

*Redes Neurais Recorrentes Para Previsão de Séries
Temporais de Memórias Curta e Longa*

Daniel Takata Gomes

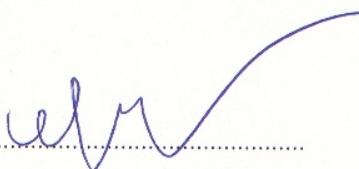
Campinas - SP

2005

MODELOS DE REDES NEURAIS RECORRENTES PARA PREVISÃO DE SÉRIES TEMPORAIS DE MEMÓRIAS CURTA E LONGA

Este exemplar corresponde à redação final da dissertação
devidamente corrigida por Daniel Takata Gomes e aprovada
pela comissão julgadora.

Campinas, 16 de dezembro de 2005



.....
Prof. Dr. Emanuel Pimentel Barbosa

Orientador

Banca Examinadora

1. Prof. Dr. Emanuel Pimentel Barbosa (Orientador) - IMECC/UNICAMP
2. Profa. Dra. Clelia Maria de Castro Toloi - IME/USP
3. Prof. Dr. Marcelo Cunha Medeiros - PUC/RJ
4. Prof. Dr. Peter Sussner - IMECC/UNICAMP

Dissertação apresentada ao Instituto de Matemática,
Estatística e Computação Científica, UNICAMP,
como requisito parcial para a obtenção do Título de
Mestre em Estatística.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecário: Maria Júlia Milani Rodrigues CRB8a / 2116

Gomes, Daniel Takata
G585m Modelos de redes neurais recorrentes para previsão de séries temporais de memórias curta e longa / Daniel Takata Gomes – Campinas, [S.P. :s.n.], 2005.
Orientador: Emanuel Pimentel Barbosa
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.
1. Redes neurais. 2. Séries temporais. 3. Previsão. 4. Kalman, Filtragem de. I. Barbosa, Emanuel Pimentel. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Recurrent neural networks for prediction of short and long memory time series

Palavras-chave em inglês (Keywords): 1. Neural networks. 2. Time series. 3. Forecasting. 4. Kalman filtering.

Área de concentração: Estatística

Titulação: Mestre em Estatística

Banca examinadora: Prof. Dr. Emanuel Pimentel Barbosa (IMECC-UNICAMP)

Profa. Dra. Clélia Maria de Castro Toloi (IME-USP)

Prof. Dr. Marcelo Cunha Medeiros (PUC-RJ)

Prof. Dr. Peter Sussner (IMECC-UNICAMP)

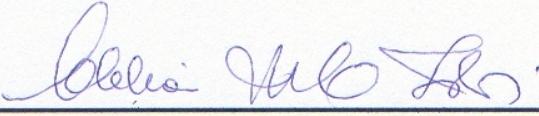
Data da defesa: 18/11/2005

**Dissertação de Mestrado defendida em 18 de novembro de 2005 e aprovada pela Banca
Examinadora composta pelos Profs. Drs.**

Prof (a). Dr (a). EMANUEL PIMENTEL BARBOSA



Prof (a). Dr (a). CLELIA MARIA DE CASTRO TOLOI



Prof (a). Dr (a). MARCELO CUNHA MEDEIROS



Prof (a). Dr (a). PETER SUSSNER



*“Run, rabbit, run.
Dig that hole, forget the sun.
And when at last the work is done,
Don’t sit down, it’s time to dig another one”*

Roger Waters

Agradecimentos

Agradeço aos meus pais, Shirley e José Carlos, e minha irmã, Luciana, pelo apoio, paciência e amor.

Ao meu orientador, Professor Emanuel Pimentel Barbosa, diretamente responsável pelas principais idéias e com quem divido os méritos deste trabalho.

Aos Professores Marcelo Cunha Medeiros, Clelia Maria de Castro Toloi e Peter Sussner, por terem aceitado fazer parte da banca examinadora e pelas contribuições para melhoria do trabalho.

Aos professores do Departamento de Estatística do IMECC pela ajuda oferecida e pelos cursos ministrados.

Aos amigos de treino da equipe de natação da UNICAMP Swimming Society Reloaded, por terem se tornado muito mais do que companheiros de piscina.

Aos meus familiares, pelo incentivo e carinho.

Aos colegas de graduação e mestrado em Estatística e aos amigos que fiz em todos esses anos de universidade.

Aos amigos de longa data, em especial Sérgio, Tode, Elias, Danilo, Gustavo e Rafael, sempre presentes nos melhores e piores momentos.

À Fapesp, Fundação de Amparo à Pesquisa do Estado de São Paulo, responsável pelo suporte financeiro.

Resumo

É de grande interesse o estudo de previsão de séries temporais, ou seja, conseguir identificar características do processo num ponto futuro. Para isso, é necessário estimar com precisão, ou pelo menos com uma boa aproximação, o processo gerador dos dados. Nos últimos anos, modelos de redes neurais artificiais vêm desempenhando um papel crescente na abordagem e solução de problemas estatísticos importantes. A estrutura mais básica de rede neural, as redes progressivas ou *feedforward*, sem retroalimentação, têm se mostrado uma alternativa vantajosa, em alguns casos, em relação aos modelos lineares tradicionais. No entanto, algumas séries apresentam características que permitem introduzir algum tipo de realimentação na rede. Tal rede é chamada de rede neural recorrente, ferramenta ainda pouco explorada pela comunidade estatística.

São dois os principais objetivos da dissertação. Primeiro, o estudo de redes neurais recorrentes para predição de séries temporais, o que compreende sua fundamentação teórica, principais arquiteturas e algoritmos de aprendizagem e sua implementação computacional. Segundo, estudo comparativo da performance preditiva dessas redes para séries temporais, tanto no caso de séries temporais de memória curta quanto de memória longa, tomando-se como referência modelos padrão tipo ARIMA e ARFIMA.

São utilizadas séries espaciais (dados de variáveis do solo) e séries temporais (séries de inflações de países desenvolvidos). Os resultados mostram que o uso das redes para predição é vantajoso em relação a modelos lineares para diversas séries. Também é desenvolvido um novo modelo (redes neurais com pesos variáveis) e seu correspondente algoritmo de estimação, baseado nas características dos dados.

Abstract

Forecasting of time series is a topic of great interest nowadays. To do so, the data generating process needs to be estimated with a good degree of accuracy. In the last years, artificial neural networks are becoming more important in the statistical community. The more basic structure of a neural network, the *feedforward* neural nets, without feedback, can be a profitable alternative, in some cases, comparing to linear traditional models. However, some time series present characteristics that allow to introduce some kind of feedback in the network. Such network is called recurrent neural network, a tool not so popular in the statistical community.

Two of the main concerns of this work are the study of recurrent neural networks for prediction of time series (theoretical fundamental, main architectures and learning algorithms) and the comparative study of the predictive performance of these networks for short and long memory time series.

Spatial series (solo science data) and time series (inflation data) are used. The results show that the networks have good prediction performance comparing to linear models in several series. A new model (neural networks with varying coefficients) and its estimation algorithm are proposed, based in the data characteristics.

Sumário

Lista de tabelas	xii
Lista de figuras	xiv
1 Introdução	1
2 Redes Neurais Artificiais	4
2.1 Introdução	4
2.2 Redes neurais como aproximadores universais de funções	6
2.3 Redes neurais <i>feedforward</i>	7
2.3.1 Treinamento de redes neurais <i>feedforward</i> : o algoritmo de retropropagação de erros	8
2.4 Redes neurais recorrentes	12
2.5 Modelos ARMA e redes neurais	13
2.5.1 Modelos ARMA(p, q) lineares	13
2.5.2 Modelos auto-regressivos não-lineares e redes neurais <i>feedforward</i>	14
2.5.3 Modelos médias móveis não-lineares e redes neurais recorrentes	14
2.5.4 Modelos ARMA não-lineares e redes neurais recorrentes	15
2.6 Treinamento de redes neurais recorrentes	15
2.6.1 Retropropagação através do tempo	15
2.6.2 Algoritmo de aprendizagem recorrente em tempo real	19
2.6.3 Filtro de Kalman e extensões	20
2.7 Intervalos de predição	26
2.7.1 Método analítico	26
2.7.2 Método numérico: reamostragem de resíduos (bootstrap)	26
3 Modelos lineares para séries temporais de memória longa	29
3.1 Modelos ARIMA	29

3.2 Modelos ARFIMA	32
3.2.1 Estimação da ordem de integração fracionária	33
4 Redes Neurais Recorrentes para Séries Espaciais de Memória Curta: predição em transects	36
4.1 Introdução	36
4.2 Análise preliminar dos dados	37
4.3 Modelos utilizados	39
4.3.1 Regressão linear	39
4.3.2 Auto-regressão vetorial - VAR	41
4.3.3 Regressão não-linear (não-paramétrica)	42
4.3.4 Comparação de modelos	44
4.4 Modelo proposto: redes neurais recorrentes com pesos variáveis ao longo do espaço	47
4.4.1 Algoritmo: filtro de Kalman estendido de 2 ^a ordem	48
4.5 Resultados	51
4.6 Intervalos de predição	52
4.7 Um estudo de simulação	55
5 Análise de séries econômicas de inflação	57
5.1 Introdução	57
5.2 Série de inflação dos Estados Unidos	59
5.2.1 Análise descritiva	60
5.2.2 Ajuste de modelos	60
5.2.3 Resultados	66
5.3 Série de inflação do Reino Unido	68
5.3.1 Análise descritiva	68
5.3.2 Ajuste de modelos	68
5.3.3 Resultados	73
5.3.4 Intervalos de predição	76
5.4 Outras séries de inflação de memória longa: Canadá, Itália e França	78
5.4.1 Análise Descritiva	78
5.4.2 Ajuste de modelos	79
5.4.3 Resultados	83
5.4.4 Versão heterocedástica do modelo de redes neurais: o caso da Itália	83
5.5 Séries de inflação de memória curta: Alemanha e Japão	88
5.5.1 Análise Descritiva	88

5.5.2	Ajuste de modelos	88
5.5.3	Resultados	95
5.5.4	Versão heterocedástica do modelo de redes neurais: o caso do Japão	95
6	Conclusões	97
	Apêndices	99
A	Algoritmo para treinamento de redes neurais com pesos variáveis e programas utilizados	99
A.1	Filtro de Kalman	99
A.2	Programa em MATLAB: treinamento de redes recorrentes com pesos variáveis ao longo do espaço/tempo	102
A.3	Programa em MATLAB: treinamento de redes <i>feedforward</i> com pesos variáveis ao longo do espaço/tempo	105
A.4	Programa em MATLAB: cálculo de intervalos de previsões via bootstrap	111
A.5	Resultados: ajuste dos modelos AR lineares	112
B	Dados utilizados	118
B.1	Dados de solo	118
B.2	Dados de inflação	119
	Referências Bibliográficas	133

Listas de Tabelas

2.1 Relação entre terminologias de rede e de estatística (Maier & Dandy, 2000)	8
4.1 Comparação de modelos de redes neurais	44
4.2 Performance preditiva (10 últimos pontos do transect)	45
4.3 Performance preditiva (10 primeiros pontos do transect)	45
4.4 Performance preditiva: 10 últimos pontos do transect	49
4.5 Performance preditiva: 10 primeiros pontos do transect	50
4.6 Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos fixos	53
4.7 Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos variáveis .	54
4.8 Resultados de simulação: série gerada com rede de pesos fixos	55
4.9 Resultados de simulação: série gerada com rede de pesos variáveis	56
5.1 Redes ajustadas para inflação dos Estados Unidos	64
5.2 Redes com pesos variáveis ajustadas para inflação dos Estados Unidos	65
5.3 Comparação de modelos: Inflação dos Estados Unidos (erros acumulados por horizonte)	67
5.4 Comparação de modelos: Inflação dos Estados Unidos (erros absolutos por mês) . . .	67
5.5 Valores de \hat{d} por diferentes métodos (inflação do Reino Unido)	69
5.6 Modelos escolhidos para redes de pesos fixos (inflação do Reino Unido)	72
5.7 Modelos escolhidos para redes de pesos variáveis (inflação do Reino Unido)	72
5.8 Comparação de modelos: Inflação do Reino Unido (erros acumulados por horizonte) .	74
5.9 Comparação de modelos: Inflação do Reino Unido (erros absolutos por mês)	74
5.10 Comparação de algoritmos: Inflação do Reino Unido (redes <i>feedforward</i> com pesos variáveis)	75
5.11 Intervalos de predição de 95% (12 últimos meses): Rede <i>feedforward</i> de pesos fixos . .	76
5.12 Intervalos de predição de 95% (12 últimos meses): Rede <i>feedforward</i> de pesos variáveis .	77
5.13 Estimação da ordem de integração fracionária	79
5.14 Modelos ARFIMA para séries de inflação com memória longa	81
5.15 Modelos escolhidos para redes de pesos fixos	81

5.16 Modelos escolhidos para redes de pesos variáveis	82
5.17 Comparação de modelos: Inflação do Canadá (erros acumulados por horizonte)	84
5.18 Comparação de modelos: Inflação do Canadá (erros absolutos por mês)	84
5.19 Comparação de modelos: Inflação da Itália (erros acumulados por horizonte)	85
5.20 Comparação de modelos: Inflação da Itália (erros absolutos por mês)	85
5.21 Comparação de modelos: Inflação da França (erros acumulados por horizonte)	86
5.22 Comparação de modelos: Inflação da França (erros absolutos por mês)	86
5.23 Modelos de redes com variância heterogênea: Inflação da Itália	88
5.24 Modelos ARFIMA para séries de inflação com memória curta	89
5.25 Modelos escolhidos para redes de pesos fixos	91
5.26 Modelos escolhidos para redes de pesos variáveis	92
5.27 Comparação de modelos: Inflação da Alemanha (erros acumulados por horizonte)	93
5.28 Comparação de modelos: Inflação da Alemanha (erros absolutos por mês)	93
5.29 Comparação de modelos: Inflação do Japão (erros acumulados por horizonte)	94
5.30 Comparação de modelos: Inflação do Japão (erros absolutos por mês)	94
5.31 Modelos de redes com variância heterogênea: Inflação do Japão	95

Listas de Figuras

2.1	Rede recorrente tipo Elman	12
2.2	Rede recorrente tipo Jordan	13
3.1	Autocorrelação simples de um processo com memória longa	33
4.1	Séries espaciais	37
4.2	Gráficos de correlações, dependência e variância	38
4.3	Evolução dos coeficientes da variável carbono ao longo dos pontos do transect	40
4.4	Comparação de predições: 10 últimos pontos do transect	49
4.5	Comparação de predições: 10 primeiros pontos do transect	50
4.6	Gráficos de resíduos: modelo de redes neurais de pesos variáveis (10 primeiros pontos) .	51
4.7	Intervalos de previsão de 95% (10 primeiros pontos): Rede Elman de pesos fixos	53
4.8	Intervalos de previsão de 95% (10 primeiros pontos): Rede Elman de pesos variáveis .	54
5.1	Gráficos da série de inflação dos Estados Unidos	59
5.2	Gráficos da série de inflação dos Estados Unidos diferenciada fracionalmente	61
5.3	Gráficos da série de inflação dos Estados Unidos diferenciada fracionalmente e sazonalmente	61
5.4	Resíduos do ajuste do modelo ARFIMA $(0, d, 0) \times (0, 1, 1)_{12}$ para a série dos Estados Unidos	62
5.5	Gráficos da série de inflação do Reino Unido	68
5.6	Inflação do Reino Unido filtrada: comparação de métodos de estimação de d	70
5.7	Inflação do Reino Unido filtrada: comparação de número de termos truncados	71
5.8	Gráficos da série de inflação do Reino Unido diferenciada fracionalmente e sazonalmente	72
5.9	Gráficos de resíduos: modelo de redes para inflação do Reino Unido	75
5.10	Intervalos de previsão: Rede <i>feedforward</i> com pesos fixos	76
5.11	Intervalos de previsão: Rede <i>feedforward</i> com pesos variáveis	77
5.12	Gráficos da série de inflação do Canadá	78

5.13 Gráficos da série de inflação da Itália	78
5.14 Gráficos da série de inflação da França	79
5.15 Gráficos da série de inflação do Canadá diferenciada fracionalmente e sazonalmente	80
5.16 Gráficos da série de inflação da Itália diferenciada fracionalmente e sazonalmente	80
5.17 Gráficos da série de inflação da França diferenciada fracionalmente e sazonalmente	80
5.18 Resíduos padronizados: modelo de redes para inflação da Itália	87
5.19 Gráficos da série de inflação da Alemanha	89
5.20 Gráficos da série de inflação do Japão	90
5.21 Gráficos da série de inflação da Alemanha diferenciada sazonalmente	90
5.22 Gráficos da série de inflação do Japão diferenciada sazonalmente	91

Capítulo 1

Introdução

É de grande interesse o estudo de métodos de previsão de séries temporais, ou seja, conseguir identificar e predizer algumas características do processo num ponto futuro. As condições presentes determinam em algum grau o futuro possivelmente envolvendo relações complexas entre as variáveis.

Uma situação ideal para a realização de previsões seria o conhecimento das equações que modelam os mecanismos responsáveis pela geração das séries temporais. No entanto, em muitos problemas reais essas informações não são disponíveis, e não se tem condições ideais para construir equações que governem o comportamento das variáveis de interesse. Quando isso acontece, o usual é utilizar uma abordagem baseada em modelos, na qual tenta-se identificar ou aproximar o processo gerador dos dados.

Redes Neurais Artificiais (“Artificial Neural Networks” - ANN) vêm desempenhando um papel crescente nos últimos anos na abordagem e solução de problemas estatísticos importantes. Sua contribuição se dá principalmente em problemas de reconhecimento de padrões ou classificação e problemas de previsão, tanto para dados transversos (regressão) quanto para dados de séries temporais ou espaciais.

A estrutura mais básica de rede neural, as chamadas redes multicamadas (ou redes *feedforward*), sem retroalimentação, é vista como uma alternativa não-linear aos modelos estatísticos lineares tradicionais, como por exemplo modelos de regressão linear ou modelos ARIMA. As redes *feedforward* têm se mostrado uma alternativa vantajosa, em alguns casos, em relação aos modelos lineares tradicionais. A característica marcante desse tipo de modelo é sua capacidade de modelar tanto estruturas lineares quanto não-lineares, podendo, inclusive, aproximar, com qualquer grau de acurácia, uma função arbitrária. Essa característica permite ao modelo lidar com mais facilidade com outliers no sentido que o modelo não é afetado de maneira global como um modelo de regressão linear, e sim apenas de maneira local (o ajuste é prejudicado apenas nas imediações do ponto aberrante).

No entanto, se a série temporal apresentar algum componente MA (média móvel) ou estrutura de memória longa, é natural que algum tipo de realimentação deva ser introduzida na rede. Uma rede com

essa característica é chamada de rede neural recorrente (“recurrent neural network”). Um modelo do tipo rede recorrente é uma ferramenta ainda pouco explorada pela comunidade estatística (ao contrário das redes *feedforward*).

São dois os principais objetivos desta tese. Primeiro, o estudo de redes neurais recorrentes para predição de séries temporais, o que compreende sua fundamentação teórica, principais arquiteturas e algoritmos de aprendizagem e sua implementação computacional. Segundo, estudo comparativo da performance preditiva dessas redes para séries temporais, tanto no caso de séries de memória curta quanto de memória longa, tomando-se como referência modelos padrão tipo VAR (“Vector Auto Regression”), espaço de estado, ARIMA e ARFIMA, com ou sem sazonalidade, ajustados a séries espaciais (variáveis de solo) e séries temporais (dados de inflação de países desenvolvidos).

Esta tese está dividida em quatro capítulos principais.

O capítulo 2 aborda um resumo da teoria básica de redes neurais, iniciando historicamente com a motivação de construir um modelo computacional para o neurônio biológico, em 1943. É apresentado o modelo mais comum de rede neural, rede *feedforward*, e seu método de estimativa (algoritmo de retropropagação). A seguir, são mostradas as arquiteturas mais comuns de redes neurais recorrentes (Elman e Jordan), além da relação de modelos ARMA não-lineares e redes recorrentes. São apresentados os algoritmos de treinamento mais conhecidos para redes recorrentes (retropropagação através do tempo, aprendizagem recorrente em tempo real e filtro de Kalman), seguido por métodos de cálculo de intervalos de confiança para redes neurais.

O capítulo 3 traz uma revisão de modelos lineares de séries temporais de memória curta (ARIMA) e também de memória longa (ARFIMA), bem como seus métodos de estimativa.

O capítulo 4 traz implementações práticas realizadas com dados de variáveis de solo (memória curta). O comportamento do conjunto de dados utilizado é particularmente interessante, pois apresenta três características principais: dependência espacial, não-homogeneidade do solo e não-linearidade da relação entre variável resposta (Nitrogênio total) e variável regressora (Carbono orgânico). Após uma análise preliminar com avaliação comparativa de performance preditiva (redes neurais e outros modelos), é observado que nenhum dos modelos consegue incorporar as três características principais do problema. Os resultados apontam para um novo modelo. O modelo proposto é redes neurais com pesos variáveis. Também é desenvolvido um algoritmo para estimativa, baseado em extensão de 2^a ordem do filtro de Kalman, com resultados superiores aos obtidos pelos algoritmos padrão.

O capítulo 5 também trata de implementações práticas, desta vez com dados econômicos de inflação dos países do G7 (memórias curta e longa): Estados Unidos, Canadá, Reino Unido, Alemanha, França, Itália e Japão. Para todas as séries, foram comparados modelos de redes neurais e modelos lineares (ARMA/ARFIMA e AR linear). Vale salientar que os modelos de redes neurais utilizados aqui podem ser considerados modelos híbridos, pois contam com uma prévia filtragem dos dados com o filtro de

memória longa antes do ajuste dos modelos. Também são utilizados modelos de redes neurais com pesos variáveis, como proposto capítulo 3, além de uma versão heterocedástica (considerando variância não-constante) destes modelos (caso da Itália e do Japão). A avaliação comparativa de performance preditiva (para diferentes horizontes de predição) mostra resultados variados, com modelos lineares tendo melhor performance para algumas séries e redes neurais obtendo melhor performance para outras.

Por fim, as discussões finais e conclusões obtidas são apresentadas, seguido pelos apêndices e pelas referências bibliográficas.

Capítulo 2

Redes Neurais Artificiais

O objetivo deste capítulo é oferecer um panorama geral sobre redes neurais, desde suas origens, na década de 1940, até os processos de treinamento atuais, passando pelas arquiteturas de redes a serem utilizadas nesse trabalho (*feedforward* e recorrentes).

2.1 Introdução

O estudo de redes neurais artificiais teve início quando apareceram as primeiras investigações sistemáticas que tentavam definir um modelo computacional para a forma de processamento de informação do cérebro. A primeira verdadeira referência para a teoria de redes neurais artificiais data de 1943, quando Warren McCulloch e Walter Pitts publicaram o artigo “A Logical Calculus of the Ideas Immanent in Nervous Activity”, onde apresentaram o que ficou conhecido como neurônio booleano de McCulloch (Kovács, 1996; Haykin, 1999). Foi o primeiro modelo computacional para o neurônio biológico. O neurônio de McCulloch tinha sua saída calculada com base em suas variáveis de entrada. A saída era determinada calculando-se a soma ponderada das entradas, com os chamados “ganhos sinápticos” como fatores de ponderação. Se o resultado fosse maior ou igual a um certo valor limiar (“threshold”), a saída do neurônio era um (pulso). Caso contrário, zero (não pulso). Matematicamente, o modelo pode ser representado como

$$y_i = \Theta \left(\sum_j w_{ij} x_j - \mu_i \right) \quad (2.1)$$

onde y_i é a saída do neurônio i , x_j é a j -ésima entrada do neurônio, w_{ij} é o ganho sináptico (peso) que relaciona a entrada j ao neurônio i , μ_i é o valor limiar para o neurônio i e Θ é uma função de ativação

definida por

$$\Theta(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.2)$$

As operações do neurônio artificial de McCulloch são justificadas no artigo citado: “A inteligência é equivalente ao cálculo de predicados que por sua vez pode ser implementado por funções booleanas. Por outro lado, o sistema nervoso é composto de redes de neurônios, que com as devidas simplificações, tem a capacidade básica de implementar estas funções booleanas. Conclusão: a ligação entre inteligência e atividade nervosa fica estabelecida de forma científica” (Kovács, 1996).

Com base nas idéias de McCulloch, as redes neurais artificiais, semelhante às que conhecemos hoje, apareceram pela primeira vez no final da década de 1950. Frank Rosenblatt criou o perceptron, uma rede de múltiplos neurônios. No perceptron, os neurônios estão dispostos em diversas camadas. Os neurônios que recebem as entradas constituem a primeira camada (camada de entrada). Os neurônios que recebem as saídas dos neurônios localizados na camada de entrada formam a segunda camada, e assim sucessivamente, até a última camada (camada de saída). Uma grande vantagem em relação ao modelo de McCulloch e Pitts é que o perceptron de Rosenblatt pode lidar com estímulos de valores contínuos, e não apenas binários.

De acordo com Rosenblatt, é simples escolher os ganhos sinápticos no caso de implementação de funções booleanas. No entanto, para a implementação de uma função arbitrária, essa escolha não é trivial. Rosenblatt propôs um método inspirado nos neurônios biológicos, que têm a propriedade de aprender uma função. A idéia seria ensinar a rede artificial, até que ela aprendesse a função desejada, no sentido de encontrar o valor ótimo dos ganhos sinápticos. O processo de aprendizagem de redes será visto logo a seguir, mas a idéia é comparar as saídas da rede com valores desejados correspondentes e criar um algoritmo de otimização. Rosenblatt demonstrou que se existe uma solução para um problema de classificação, o perceptron aprenderá a solução em um número finito de passos (Warner & Misra, 1996). No entanto, ele não conseguiu oferecer uma solução totalmente definitiva para o problema.

As maiores contribuições ao longo da década de 1970 foram referentes a topologias e utilizações de redes neurais. No entanto, o maior impulso para o estabelecimento das redes neurais viria apenas em 1986. David Rumelhart desenvolveu um método para treinar redes neurais multicamada chamado algoritmo de retropropagação e resolveu o problema pendente desde o fim da década de 1950. O algoritmo resolveu diversos problemas pendentes no tocante à estimativa de pesos de redes neurais e hoje é um dos métodos mais utilizado para treinamento de percepitrons multimacadas.

Desde então, muitos trabalhos foram feitos para melhorar o algoritmo de retropropagação, tais como evitar sobre-ajuste dos dados, acelerar a velocidade do algoritmo e evitar mínimos locais (Haykin, 1999; Fine, 1999).

Redes neurais também estão entre as ferramentas mais utilizadas no campo chamado Inteligência Artificial (ao lado de sistemas de produção e lógica nebulosa), no qual se busca prover máquinas com a capacidade de realizar algumas atividades mentais do ser humano (Nascimento Jr. & Yoneyama, 2000).

Há muitos outros tipos de redes neurais além do perceptron multicamada (também chamado de rede neural progressiva ou *feedforward*) com diferentes objetivos. Muitos trabalhos foram desenvolvidos abordando uma perspectiva estatística para redes neurais (White, 1989; Cheng & Titterington, 1994; Warner & Misra, 1996; e outros). Neste trabalho, o objetivo é predição de séries temporais (ou espaciais), assunto freqüentemente tratado não somente com o uso de redes neurais *feedforward*, mas também com redes retroalimentadas, ditas recorrentes, por exemplo redes tipo Elman, tipo Jordan e o algoritmo de aprendizagem recorrente em tempo real (Williams & Peng, 1990). Esse tipo específico de rede neural será abordado mais à frente.

Em geral, as redes neurais podem executar diferentes tarefas: classificação ou reconhecimento, associação ou memorização, codificação, predição e simulação. Há dois tipos de aprendizagem de redes neurais: aprendizagens supervisionada e não-supervisionada. Aprendizagem supervisionada é utilizada quando há um valor alvo (desejável) associado com cada entrada do conjunto de treinamento (amostra utilizada para ajuste dos pesos). A saída da rede é comparada com o valor alvo e sua diferença é utilizada para alterar os pesos. Aprendizagem não-supervisionada ocorre quando não há valores alvo para a saída da rede. Nesse caso, a rede aprende através de correlações entre os padrões das variáveis de entrada (Haykin, 1999). Como o objetivo deste trabalho é predição, apenas aprendizagem supervisionada será utilizada.

Redes neurais artificiais podem ser consideradas uma ferramenta recente no campo de predição. No entanto, seu uso tem crescido muito nos últimos anos nas mais diversas áreas para este fim, como finanças, medicina e ciência experimental (Maier & Dandi, 2000).

2.2 Redes neurais como aproximadores universais de funções

As estruturas de redes neurais visam aproximar relações não-lineares desconhecidas (entre saídas e entradas) através de uma composição (função de função) de relações lineares generalizadas (RLG's). Em 1957, o matemático Kolmogorov publicou um teorema que quantifica essa aproximação. O teorema, relativo a representação de funções contínuas, pode ser diretamente adaptado a redes neurais, sendo reformulado da seguinte maneira (Bishop, 1995; Fine, 1999):

Teorema de Kolmogorov-Nielsen: Dada uma função $f : [0, 1]^n \rightarrow \Re^m$, $f(x) = y$, existe sempre para f uma implementação exata com uma rede neural de três camadas com n entradas, sendo a primeira camada intermediária composta por $n(2n + 1)$ neurônios, a segunda camada intermediária com $2n + 1$ neurônios e a camada de saída com m neurônios representando as m componentes do vetor y .

As funções de ativação envolvidas na rede neural dependem da natureza da função f a ser implementada. Como o teorema é um teorema de existência, não oferece nenhum procedimento para a determinação dessas funções de ativação. A seguinte versão simplificada do teorema acima tem maior importância prática no contexto deste trabalho:

Cybenko (1989): Uma rede *feedforward* com duas camadas (intermediária e saída) com funções de ativação sigmoidais pode aproximar qualquer função contínua com um dado grau de acurácia pré-definido.

2.3 Redes neurais *feedforward*

Uma rede neural do tipo perceptron multicamada, com ligações unidirecionais entre os neurônios (na direção das entradas para as saídas), é comumente chamada de rede neural progressiva ou *feedforward*. Uma rede neural é um conjunto de unidades computacionais interconectadas e organizadas em camadas (Warner & Misra, 1995). Essas unidades, também chamadas de nós, representam (tem como metáfora) o neurônio biológico. As ligações entre neurônios, ou entre entradas e neurônios, têm, cada uma, um peso sináptico correspondente, denotado por w_{ij} (conexão do j -ésimo nó, ou j -ésima entrada, para o i -ésimo nó). A saída do j -ésimo neurônio, ou a j -ésima entrada, é multiplicada por w_{ij} e é uma entrada para o i -ésimo nó. A chamada excitação ou entrada resultante do i -ésimo nó é dada por $\text{excitação}_i = \sum_j w_{ij} \text{entrada}_j + \mu_i$ onde μ_i pode ser interpretado como um termo de intercepto (ou “viés”, no linguajar de redes). Na realidade, não é difícil ver que uma rede *feedforward* com apenas um neurônio, com função de ativação linear, nada mais é do que uma regressão linear padrão.

No entanto, no caso geral, cada neurônio tem uma função não-linear correspondente. Ou seja, a saída do neurônio i é dada por $\text{saída}_i = g(\text{excitação}_i)$ onde g é uma função não-linear diferenciável e limitada. Entre outras, duas escolhas comuns para g são

$$g(x) = \frac{1}{1 + e^{-x}} \quad g(x) = \tanh(x) \quad (2.3)$$

Para escrever a saída da rede como função de suas entradas, vamos considerar uma rede *feedforward* com uma camada intermediária e uma camada de saída.

A saída do j -ésimo neurônio da camada intermediária é dada por $v_{pj} = g(\sum_{i=1}^N w_{ji} x_{pi})$, onde N é o número total de entradas, w_{ji} é o peso que relaciona a entrada i ao neurônio j , x_{pi} é o valor da p -ésima observação (“pattern”) da i -ésima entrada e g é uma função não-linear diferenciável e limitada. Note que o viés não está incluído na notação, mas esse termo pode ser acrescentado ao modelo com a inclusão de uma variável de entrada com valor 1.

O k -ésimo neurônio dos O neurônios da camada de saída tem como saída $\hat{y}_{pk} = r(\sum_{j=1}^M W_{kj} v_{pj})$, onde M é o número de neurônios na camada anterior, W_{kj} é o peso que relaciona a saída do j -ésimo

Terminologia de redes	Terminologia estatística
Variável de entrada	Variável independente
Variável de saída	Valor ajustado (“fitted values”)
Valor alvo ou desejável	Variável dependente ou resposta
Treinamento ou aprendizagem	Otimização não-linear
Pesos da rede	Parâmetros do modelo
“Patterns”	Observações
Viés	Termo de intercepto

Tabela 2.1: Relação entre terminologias de rede e de estatística (Maier & Dandy, 2000)

neurônio da camada intermediária ao neurônio k da camada de saída e r é uma função de ativação. Em muitos casos, inclusive neste trabalho, onde o objetivo é predição, essa função de ativação é simplesmente a identidade, pois os valores alvo para a rede podem estar fora do intervalo $[0, 1]$ ou $[-1, 1]$. Portanto, daqui para frente, vamos considerar que $\hat{y}_{pk} = f_{pk}$.

A fórmula geral para as saídas de uma rede *feedforward* com uma camada intermediária e uma camada de saída fica, então, para $k = 1, \dots, O$:

$$\hat{y}_{pk} = \sum_{j=1}^M W_{kj} g\left(\sum_{i=1}^N w_{ji} x_{pi}\right) \quad (2.4)$$

o que pode ser visto simplesmente como uma regressão não-linear de parâmetros w_{ji} e W_{kj} desconhecidos.

2.3.1 Treinamento de redes neurais *feedforward*: o algoritmo de retropropagação de erros

O algoritmo de aprendizagem por retropropagação do erro (“back-propagation”) é um método de estimativa dos pesos sinápticos de uma rede feedforward multicamada. Foi desenvolvido por Rumelhart et al. (1986), apesar de ter sido explorado anos antes (Werbos, 1974), com resultados não reconhecidos e difundidos. O objetivo do algoritmo nada mais é do que minimizar a soma de quadrados de erros, o que representa estimativa por máxima verossimilhança dos pesos da rede quando os erros têm distribuição normal.

Para efetuar essa aprendizagem, alguma medida de performance é necessária. A mais utilizada notadamente no caso de respostas gaussianas é a soma de quadrados de erros:

$$E = \frac{1}{2} \sum_p \sum_k (y_{pk} - \hat{y}_{pk})^2 \quad (2.5)$$

onde p é referente às observações, o índice k é referente aos nós de saída, y é a resposta desejada (alvo) e \hat{y} é a saída da rede (valor ajustado). Os pesos da rede são ajustados de forma a minimizar esse erro.

A cada passo do algoritmo, ou seja, quando a observação p é processada, o objetivo é minimizar

$$E_p = \frac{1}{2} \sum_k (y_{pk} - \hat{y}_{pk})^2 \quad (2.6)$$

o que implica que $E = \sum_p E_p$.

Vamos novamente nos ater ao caso de uma rede com duas camadas (intermediária e saída), que é o tipo de rede que será utilizado nesse trabalho. O objetivo é encontrar os pesos w_{ji} e W_{kj} que minimizem a função E . Para tanto, usa-se o procedimento de minimização por gradiente descendente, percorrendo a rede da saída para a entrada, onde a correção nos pesos é definida pela “regra delta” (Haykin, 1999; Fine, 1999):

$$\Delta W_{kj} = -\eta \frac{\partial E_p}{\partial W_{kj}} \quad (2.7)$$

onde η é a taxa de aprendizagem. O sinal negativo na equação 2.7 é necessário para alterar o valor dos pesos na direção de uma redução no valor de E . Substituindo a equação 2.4 na equação 2.6, temos

$$E_p = \frac{1}{2} \sum_{k=1}^O (y_{pk} - (\sum_{j=1}^M W_{kj} g(\sum_{i=1}^M w_{ji} x_{pi}))^2 \quad (2.8)$$

Pela regra da cadeia, temos

$$\frac{\partial E_p}{\partial W_{kj}} = \frac{\partial E_p}{\partial \hat{y}_{pk}} \frac{\partial \hat{y}_{pk}}{\partial f_{pk}} \frac{\partial f_{pk}}{\partial W_{kj}} \quad (2.9)$$

Não é difícil ver que

$$\frac{\partial E_p}{\partial \hat{y}_{pk}} = -(y_{pk} - \hat{y}_{pk}) \quad (2.10)$$

$$\frac{\partial \hat{y}_{pk}}{\partial f_{pk}} = 1 \quad (2.11)$$

$$\frac{\partial f_{pk}}{\partial W_{kj}} = v_{pj} \quad (2.12)$$

Substituindo em 2.7, temos

$$\Delta W_{kj} = \eta (y_{pk} - \hat{y}_{pk}) v_{pj} \quad (2.13)$$

Portanto, a fórmula para atualização dos pesos da camada de saída fica

$$W_{kj}^{p+1} = W_{kj}^p + \Delta W_{kj} \quad (2.14)$$

Similarmente, vamos agora calcular, para os pesos da camada intermediária,

$$\Delta w_{ji} = -\eta \frac{\partial E_p}{\partial w_{ji}} \quad (2.15)$$

Pela regra da cadeia, temos

$$\frac{\partial E_p}{\partial w_{ji}} = \sum_{k=1}^O \frac{\partial E_p}{\partial \hat{y}_{pk}} \frac{\partial \hat{y}_{pk}}{\partial f_{pk}} \frac{\partial f_{pk}}{\partial v_{pj}} \frac{\partial v_{pj}}{\partial h_{pj}} \frac{\partial h_{pj}}{\partial w_{ji}} \quad (2.16)$$

Mas

$$\frac{\partial E_p}{\partial \hat{y}_{pk}} = -(y_{pk} - \hat{y}_{pk}) \quad (2.17)$$

$$\frac{\partial \hat{y}_{pk}}{\partial f_{pk}} = 1 \quad (2.18)$$

$$\frac{\partial f_{pk}}{\partial v_{pj}} = W_{kj} \quad (2.19)$$

$$\frac{\partial v_{pj}}{\partial h_{pj}} = g'(h_{pj}) \quad (2.20)$$

$$\frac{\partial h_{pj}}{\partial w_{ji}} = x_{pi} \quad (2.21)$$

O que nos dá, pelas equações 2.15 e 2.16,

$$\Delta w_{ji} = \eta \sum_{k=1}^O (y_{pk} - \hat{y}_{pk}) W_{kj} g'(h_{pj}) x_{pi} \quad (2.22)$$

onde a parte interna da expressão acima (entre η e x_{pi}), por analogia com a equação 2.13 (onde $y_{pk} - \hat{y}_{pk}$ é o erro de saída) é o erro da camada intermediária, obtido por retropropagação do erro de saída.

Algoritmo de retropropagação

Inicialização: $w_{ji} = w_{ji}^1$, $W_{kj} = W_{kj}^1$

Para $p = 1, 2, \dots$:

1. Com w_{ij}^p e W_{kj}^p , calcule as saídas dos neurônios da rede, i. e. v_{pj} e \hat{y}_{pk} ;
2. Calcule os erros da camada de saída (supondo g dada por 2.3)

$$\delta_{pk} = (y_{pk} - \hat{y}_{pk});$$

3. Calcule os erros da camada intermediária

$$\psi_{pj} = \sum_{k=1}^O \delta_{pk} W_{kj} v_{pj} (1 - v_{pj});$$

4. Atualize os pesos:

$$w_{ji}^{p+1} = w_{ji}^p + \eta \psi_{pj} x_{pi}$$

$$W_{kj}^{p+1} = W_{kj}^p + \eta \delta_{pk} v_{pj}$$

Modos de treinamento: sequencial e por batelada (*batch*)

Na aplicação prática do algoritmo de retropropagação, o modo de treinamento pode variar de acordo com a maneira com que o conjunto de treinamento (amostra utilizada para a estimativa dos pesos) é apresentado. Uma apresentação de todo o conjunto de treinamento durante o processo de aprendizagem é chamado de época e os pesos são alterados depois que o algoritmo processa uma observação ou todo o conjunto de treinamento. Assim, o algoritmo de retropropagação pode ser implementado de duas formas (Haykin, 1999; Fine, 1999):

1. Modo sequencial: também referido como modo on-line. A atualização dos pesos é feita após a apresentação de cada uma das observações do conjunto de treinamento (forma que o algoritmo de retropropagação é apresentado nesta seção). Quando o conjunto de treinamento inteiro é processado, o treinamento é recomeçado e o processo continua, até que seja interrompido através de um critério de parada.
2. Modo por batelada (*batch*): a atualização dos pesos é feita após a apresentação de todas as observações do conjunto de treinamento, que constituem uma época. A cada passo do algoritmo, a função E (equação 2.5) é minimizada, ao invés de apenas E_p (equação 2.6). O conjunto de treinamento é apresentado ao algoritmo quantas vezes forem necessárias para a estimativa dos pesos (o treinamento é interrompido através de um critério de parada).

Critério de parada

Os métodos de otimização para encontrar os pesos são iterativos. Isso significa que, em um número grande de iterações, o valor da função E pode ser tão pequeno quanto se queira. Isso normalmente representa um bom ajuste dentro da amostra, mas em geral o modelo não conseguirá ajustar valores fora do plano amostral, pois neste caso o modelo estaria incluindo o erro em sua predição, o que não interessa. Para evitar esse inconveniente, usa-se em redes neurais, entre outras, a técnica do “early-stopping”. O conjunto de dados é separado em uma amostra de treinamento, que será utilizada para estimar os pesos, e em uma amostra de validação. Quando os pesos são estimados pela amostra de treinamento, ao mesmo tempo avaliam-se os erros nas amostras, ou seja, verifica-se se o valor de E continua pequeno mesmo fora da amostra de treinamento. Desta forma, tomam-se os pesos da iteração que corresponde ao valor mínimo de E na amostra de validação (mas com os pesos calculados baseados na amostra de treinamento), evitando-se assim o que se chama de sobre-ajuste. Pode ser mostrado que este procedimento corresponde a um método de regularização (Bishop, 1995).

É comum separar o conjunto de dados em mais uma amostra, chamada de amostra de teste, para eventuais cálculos de performance preditiva e comparações entre diferentes modelos.

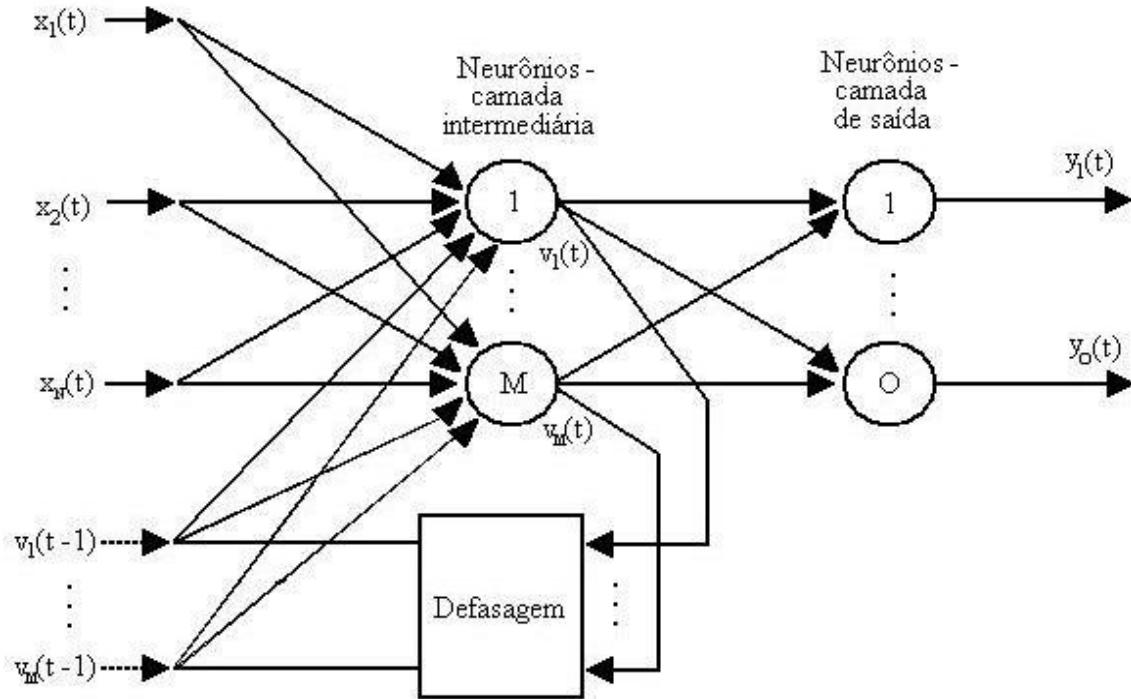


Figura 2.1: Rede recorrente tipo Elman

2.4 Redes neurais recorrentes

A estrutura de uma rede neural recorrente tem por base uma rede *feedforward* com algumas modificações, notadamente a introdução de realimentação, o que amplia sua potencialidade de modelagem de dados temporais ou espaciais. Uma rede *feedforward*, portanto, pode ser vista como um caso particular de uma rede recorrente. As realimentações consistem em saídas de neurônios de determinada camada serem reintroduzidas como entradas de neurônios de camadas anteriores ou da própria. Um neurônio pode ser realimentado por sua própria saída. Essas possibilidades fazem com que a arquitetura de uma rede recorrente possa tomar diversas formas.

No caso de tempo discreto, como utilizado aqui, as realimentações são acompanhadas de defasagens de uma ou mais unidades do tempo.

Com essa topologia de rede, ampliam-se as possibilidades de modelagem de estruturas de auto-dependência de dados em caso de séries temporais ou espaciais, não se limitando apenas às estruturas auto-regressivas padrão passíveis de representação por redes *feedforward*.

Para o caso de predição de séries temporais, duas das redes recorrentes mais utilizadas são as do tipo Elman e as do tipo Jordan (Mandic & Chambers, 2001). As saídas da camada intermediária (Elman)

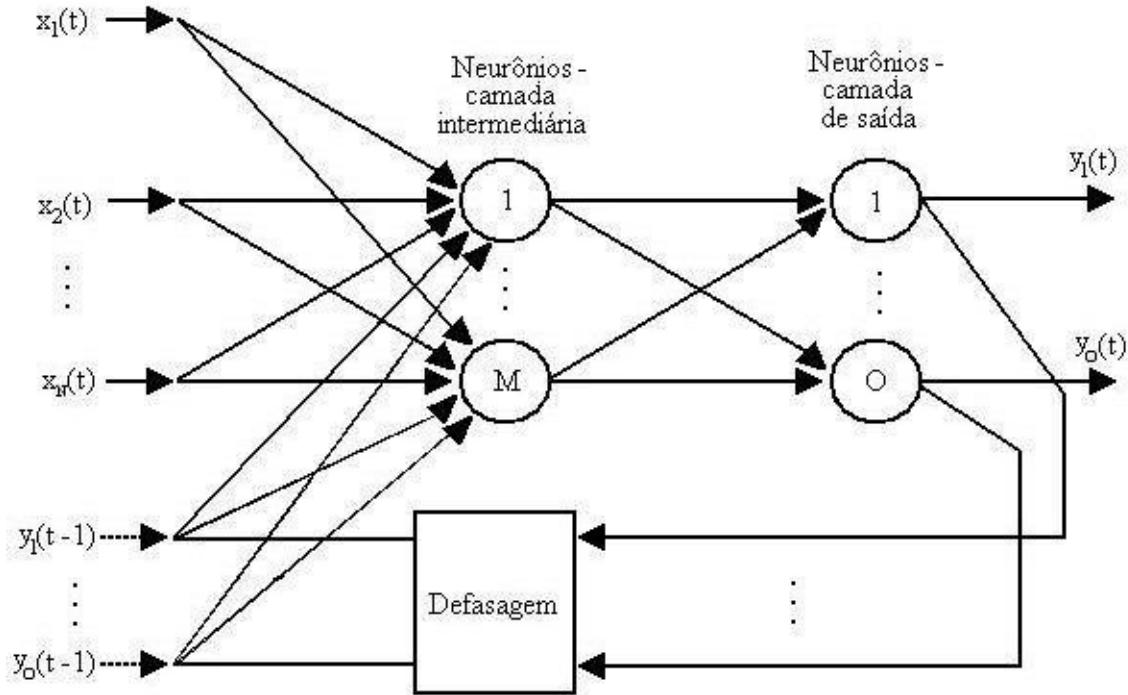


Figura 2.2: Rede recorrente tipo Jordan

ou da camada de saída (Jordan) são defasadas e reintroduzidas como entradas da rede. As figuras 2.1 e 2.2 mostram tipos particulares dessas redes, com uma camada intermediária.

2.5 Modelos ARMA e redes neurais

Nesta seção, será mostrado como específicas arquiteturas de redes neurais *feedforward* e recorrentes podem servir como extensões não-lineares de modelos lineares tradicionais de séries temporais, a saber modelos ARMA (Connor et al., 1994).

2.5.1 Modelos ARMA(p, q) lineares

Uma série temporal $x_t : t = 0, 1, \dots, n$ é dita seguir um processo ARMA(p, q) se

$$x(t) = \mu + \sum_{i=1}^p \phi_i x(t-i) + \sum_{j=1}^q \theta_j a(t-j) + a(t) \quad (2.23)$$

onde μ é a média do processo e, por suposição, $E(a(t)|x(t-1), x(t-2), \dots) = 0$, condição satisfeita quando, por exemplo, a_t é ruído branco (média zero, não-correlacionados e com variância finita σ^2). Os coeficientes ϕ_i e θ_j são, respectivamente, coeficientes auto-regressivos e média móvel do processo. No caso

de $\theta_j = 0, j = 1, 2, \dots, q$, o processo é dito auto-regressivo de ordem p (AR(p)). Se $\phi_i = 0, i = 1, 2, \dots, p$, o processo é dito média móvel de ordem q (MA(q)). Sem perda de generalidade, pode-se considerar $\mu = 0$, o que é equivalente a substituir $x(t)$ por $x(t) - E(x(t))$ (Shumway & Stoffer, 2000).

2.5.2 Modelos auto-regressivos não-lineares e redes neurais *feedforward*

Suponha que um processo $x(t) : t = 0, 1, \dots, n$ seja definido por

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-p)) + e(t) \quad (2.24)$$

onde f é uma função não-linear, porém desconhecida. Pode-se dizer, então, que $x(t)$ é um processo auto-regressivo não-linear. Neste caso, é possível usar uma rede neural *feedforward* para uma estimativa \hat{f} de f . Lapedes e Farber foram os primeiros que propuseram redes neurais *feedforward* como um modelo autoregressivo não-linear - NAR (Connor et al., 1994). Uma aproximação para f via redes com uma camada intermediária é dada por

$$\hat{x}(t) = \hat{f}(x(t-1), x(t-2), \dots, x(t-p)) = \sum_{i=1}^I W_i g\left(\sum_{j=1}^p w_{ij} x(t-j) + \beta_i\right) \quad (2.25)$$

onde I é o número de neurônios na camada intermediária, g é uma função limitada e diferenciável, tipicamente sigmoidal, W_i e w_{ij} são os pesos da rede e β_i é o viés do neurônio i . Esses pesos são obtidos através da minimização de uma função perda, por exemplo a soma dos quadrados dos resíduos $\sum_{t=1}^n (x(t) - \hat{x}(t))^2$, através de algum procedimento de otimização.

2.5.3 Modelos médias móveis não-lineares e redes neurais recorrentes

Similarmente à extensão não-linear de processos autoregressivos, uma extensão não-linear de um processo média móvel pode ser dada por

$$x(t) = f(e(t-1), e(t-2), \dots, e(t-q)) + e(t) \quad (2.26)$$

onde, novamente, f é uma função não-linear.

Suponha que este modelo MA não-linear seja inversível no sentido que existe uma função h tal que

$$x(t) = h(x(t-1), x(t-2), \dots) + e(t) \quad (2.27)$$

Com isso, em princípio, poderia-se calcular os termos $e(t-j), j = 1, \dots, q$, como função das infinitas observações do passado $x(t-1), x(t-2), \dots$. Como, na prática, isso não é possível, o cálculo do preditor $x(t)$ é feito por analogia aos cálculos recursivos utilizados para o modelo MA linear.

$$x(t) = f(\hat{e}(t-1), \hat{e}(t-2), \dots, \hat{e}(t-q)) \quad (2.28)$$

$$\hat{e}(j) = x(j) - \hat{x}(j), j = t-1, \dots, t-q \quad (2.29)$$

Em geral, é utilizada a condição inicial

$$\hat{e}(0) = \dots = \hat{e}(-q+1) = 0 \quad (2.30)$$

Desta forma, a estimativa para f dada por uma rede neural recorrente é

$$\hat{x}(t) = \sum_{i=1}^I W_i f \left(\sum_{j=1}^q w_{ij} (x(t-j) - \hat{x}(t-j)) + \beta_i \right) \quad (2.31)$$

Assim como para a rede *feedforward*, os pesos são estimados minimizando uma função perda.

2.5.4 Modelos ARMA não-lineares e redes neurais recorrentes

Uma série $x_t : t = 1, \dots, n$ segue um processo NARMA(p, q) se

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-p), e(t-1), e(t-2), \dots, e(t-q)) + e(t) \quad (2.32)$$

Através das equações dos processos AR e MA não-lineares, é fácil ver que o preditor \hat{x}_t dado por rede neural recorrente é

$$\hat{x}(t) = \sum_{i=1}^I W_i f \left(\sum_{j=1}^p w_{ij} x(t-j) + \sum_{j=1}^q w'_{ij} (x(t-j) - \hat{x}(t-j)) + \beta_i \right) \quad (2.33)$$

2.6 Treinamento de redes neurais recorrentes

Aqui serão apresentados os três algoritmos mais utilizados para o treinamento de redes neurais recorrentes: o algoritmo de retropropagação através do tempo (uma extensão do algoritmo de retropropagação padrão), o algoritmo de aprendizagem recorrente em tempo real e o filtro de Kalman estendido (que também pode ser utilizado no treinamento de redes *feedforward*).

2.6.1 Retropropagação através do tempo

Para a estimação dos pesos de uma rede recorrente, considera-se em geral um número finito pré-estabelecido de iterações. O algoritmo de retropropagação através do tempo (“back-propagation through time” - BPTT) é uma extensão do algoritmo de retropropagação padrão.

Suponha que uma rede recorrente tenha um total de n nós (neurônios) e que o peso w_{ij} represente a conexão do neurônio i com a saída do neurônio j , que pode ser também uma entrada da rede (note que vários desses pesos podem ser nulos). Neste algoritmo, a rede recorrente é desdobrada em uma rede *feedforward* que cresce uma camada a cada passo de tempo. Em cada passo de tempo $t = 0, 1, 2, \dots, T$, a rede é alimentada pelo vetor de entradas exógenas $x(t)$ ($x(0) = 0$) e as saídas $y_1(t), \dots, y_n(t)$ de todos os neurônios são calculadas. As saídas dos n neurônios no passo de tempo t servirão como entradas

para a camada de n neurônios que se formará no passo $t + 1$. O algoritmo de retropropagação através do tempo consiste em aplicar o algoritmo de retropropagação padrão nessa rede desdobrada.

Antes de partir para o algoritmo propriamente dito, é conveniente fazer algumas suposições e definições (Williams & Zipser, 1995).

Sejam $\mathbf{y}(t)$ um vetor de dimensão n que contém as saídas dos nós da rede e $\mathbf{x}^{rede}(t)$ um vetor de dimensão m que contém as entradas exógenas da rede no tempo t . Definimos

$$x_k(t) = \begin{cases} x_k^{rede}(t), & k \in I \\ y_k(t), & k \in U \end{cases} \quad (2.34)$$

onde I é o conjunto de índices tais que x_k é uma entrada exógena e U é o conjunto de índices tais que x_k é saída de um nó da rede.

O peso w_{ij} da rede representa a conexão da i -ésima unidade com a j -ésima unidade, se $j \in U$, ou a j -ésima entrada, se $j \in I$ (sem levar em consideração os vieses).

Seja s_k a excitação da k -ésima unidade da rede

$$s_k(t+1) = \sum_{l \in U} w_{kl} y_l(t) + \sum_{l \in I} w_{kl} x_l(t) \quad (2.35)$$

onde $x_l(t)$ são as entradas exógenas da rede. Assim, a saída do k -ésimo neurônio é dada por

$$y_k(t+1) = g_k(s_k(t+1)) \quad (2.36)$$

onde g_k é uma função de ativação, tipicamente sigmoidal.

O objetivo é estimar os pesos da rede que minimizem uma função do erro. Seja $y_i(t), i = 1, \dots, n$, a saída do i -ésimo neurônio no passo de tempo t . Seja $S(t)$ o conjunto de índices $k \in U$ para os quais existe um valor alvo $d_k(t)$ para a saída do neurônio k . Definimos o erro do neurônio k como

$$e_k(t) = \begin{cases} d_k(t) - y_k(t), & k \in S(t) \\ 0, & c.c. \end{cases} \quad (2.37)$$

Para a estimação dos pesos, é conveniente maximizar a função

$$E^{total}(t', t) = \sum_{\tau=t'+1}^t E(\tau) \quad (2.38)$$

onde

$$E(t) = -\frac{1}{2} \sum_{k \in U} (e_k(t))^2 \quad (2.39)$$

O algoritmo de retropropagação é um algoritmo de aprendizagem baseado em gradiente. Dessa forma, a taxa de mudança dos pesos de um passo para outro é dada por

$$\Delta w_{ij} = \eta \frac{\partial E(t', t)}{\partial w_{ij}} \quad (2.40)$$

onde η é uma constante positiva que recebe o nome de taxa de aprendizagem. Seja

$$\varepsilon_k(\tau) = \frac{\partial E}{\partial y_k(\tau)} \quad (2.41)$$

onde τ é um passo qualquer no tempo. Seja $y_k^*(\tau)$ apenas as saídas dos neurônios que têm respostas desejadas associadas. Pela definição de E , temos

$$e_k(\tau) = \frac{\partial E}{\partial y_k^*(\tau)} \quad (2.42)$$

E depende de $y_k(\tau)$ apenas através de $y_k^*(\tau)$ e através das variáveis $s_k(\tau + 1)$, quando l varia em U .

$$\frac{\partial E(\tau)}{\partial y_k^*(\tau)} = \frac{\partial E(\tau)}{\partial y_k(\tau)} \frac{\partial y_k(\tau)}{\partial y_k^*(\tau)} + \sum_{l \in U} \frac{\partial E(\tau)}{\partial s_k(\tau + 1)} \frac{\partial \tau s_k(\tau + 1)}{\partial y_k(t)} \quad (2.43)$$

O segundo termo da soma é zero se $\tau = t$, onde t é o passo atual. Logo,

$$\varepsilon_k(\tau; E) = \begin{cases} e_k(t), & \tau = t \\ e_k(t) + \sum_{l \in U} w_{lk} \delta_l(\tau + 1), & \tau < t \end{cases} \quad (2.44)$$

onde

$$\delta_k(\tau; E) = \frac{\partial E}{\partial s_k(\tau)} \quad (2.45)$$

Para $\tau < t$,

$$\frac{\partial E(\tau)}{\partial s_k(\tau)} = \frac{\partial E(\tau)}{\partial y_k(\tau)} \frac{\partial y_k(\tau)}{\partial s_k(\tau)} \Rightarrow \delta_k(\tau; E) = g'_k(s_k(\tau)) \varepsilon_k(\tau) \quad (2.46)$$

Para i e j apropriados,

$$\frac{\partial E(\tau)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t-1} \frac{\partial E(\tau)}{\partial w_{ij}(\tau)} \frac{\partial w_{ij}(\tau)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t-1} \frac{\partial J(\tau)}{\partial w_{ij}(\tau)} \quad (2.47)$$

Para qualquer τ ,

$$\frac{\partial E(\tau)}{\partial w_{ij}(\tau)} = \frac{\partial E(\tau)}{\partial s_i(\tau + 1)} \frac{\partial s_i(\tau + 1)}{\partial w_{ij}(\tau)} = \delta_i(\tau + 1; E) x_j(\tau) \quad (2.48)$$

onde x_j é a entrada do neurônio j . Das equações acima, finalmente temos:

$$\frac{\partial E(\tau)}{\partial w_{ij}} = \sum_{\tau=t_0}^{t-1} \delta_i(\tau + 1; E) x_j(\tau) \quad (2.49)$$

O algoritmo é chamado “por época” se calcula o gradiente de $E^{total}(t_0, t_1)$ em t_1 , o fim de uma época, e é chamado de algoritmo em “tempo real” se calcula $E(t)$ para cada passo de tempo t .

Dessa forma, as diferenças são:

- Algoritmo em tempo real: $E = E(t)$ e $e_k(\tau; E(t)) = 0$ para $\tau < t$

- Algoritmo por época: $t = t_1$, $E = E^{total}(t_0, t_1)$, $e_k(\tau; E^{total}(t_0, t_1)) = e_k(\tau) \forall \tau \leq t_1$

Algoritmo em tempo real

(aplicado na rede recorrente desdobrada em uma *feedforward*)

Inicialização: $w_{ij} = w_{ij}^0$, $i \in U$, $j \in U \cup I$

Para $t = 1, 2, \dots, t_1$:

1. Com w_{ij}^t , calcule as saídas dos neurônios da rede desdobrada desde a primeira camada até a camada atual t ;
2. Calcule os erros $e_k(t)$ comparando as saídas dos neurônios com seus valores desejados correspondentes (os erros $e_k(t)$ estão definidos para todos os neurônios, podendo ser nulos caso não haja valores desejados correspondentes);
3. Faça

$$\varepsilon_k(t) = e_k(t)$$

e calcule, $\forall \tau < t$:

$$\begin{aligned}\delta_k(\tau) &= g'_k(s_k(\tau))\varepsilon_k(\tau) \\ \varepsilon_k(t) &= \sum_{l \in U} w_{lk} \delta_l(\tau)\end{aligned}$$

4. Calcule o gradiente

$$\frac{\partial E(t)}{\partial w_{ij}} = \sum_{\tau=1}^t \delta_i(\tau) x_j(\tau - 1)$$

5. A variação dos pesos é dada por

$$w_{ij}^{t+1} = w_{ij}^t + \eta \frac{\partial E(t)}{\partial w_{ij}}$$

Na prática, esse algoritmo é computacionalmente muito dispendioso, pois é necessário o cálculo das saídas dos neurônios de uma rede desdobrada com um número muito grande de camadas, se t for muito grande. O que usualmente se faz é um truncamento, fazendo com que, a partir de um certo t , o número de camadas da rede desdobrada seja fixo. Ou seja, os cálculos são feitos apenas até, digamos, m passos para trás, utilizando uma rede com $m < t$ camadas (Haykin, 1999).

Algoritmo por época

(aplicado na rede recorrente desdobrada em uma *feedforward*)

Inicialização: $w_{ij} = w_{ij}^0$, $i \in U$, $j \in U \cup I$

Aqui, como o objetivo é calcular o gradiente de $E^{total}(t_0, t_1)$, é preciso rodar o algoritmo no intervalo $[t_0, t_1]$.

1. Com w_{ij}^0 , calcule as saídas dos neurônios da rede desdobrada desde a primeira camada até a camada t_1 ;
2. Calcule os erros $e_k(t_1)$ comparando as saídas dos neurônios com seus valores desejados correspondentes (os erros $e_k(t_1)$ estão definidos para todos os neurônios, podendo ser nulos caso não haja valores desejados correspondentes);
3. Faça

$$\varepsilon_k(t_1) = e_k(t_1)$$

e calcule, $\forall \tau < t_1$:

$$\begin{aligned}\delta_k(\tau) &= g'_k(s_k(\tau))\varepsilon_k(\tau) \\ \varepsilon_k(\tau) &= e_k(\tau - 1) + \sum_{l \in U} w_{lk} \delta_l(\tau)\end{aligned}$$

4. Calcule o gradiente

$$\frac{\partial E^{total}(t_0, t_1)}{\partial w_{ij}} = \sum_{\tau=t_0+1}^{t_1} \delta_i(\tau) x_j(\tau - 1)$$

5. A variação dos pesos é dada por

$$w_{ij}^1 = w_{ij}^0 + \eta \frac{\partial E^{total}(t_0, t_1)}{\partial w_{ij}}$$

Repita o processo acima quantas vezes for necessário.

2.6.2 Algoritmo de aprendizagem recorrente em tempo real

O algoritmo de aprendizagem recorrente em tempo real (“real time recurrent learning”) foi apresentado pela primeira vez por Williams & Zipser (1989). Vamos manter as mesmas definições utilizadas na seção anterior. Suponha que desejamos minimizar a mesma função do erro, ou seja, maximizar $E(t)$. Para $k \in U$, $i \in U$, $j \in U \cup I$ e $t_0 < t < t_1$, definimos

$$p_{ij}^k(t) = \frac{\partial y_k(t)}{\partial w_{ij}} \tag{2.50}$$

O objetivo é calcular a derivada da função erro em relação aos pesos, qual seja:

$$\frac{\partial E(t)}{\partial w_{ij}} = \frac{\partial E(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial w_{ij}} = \sum_{k \in U} e_k(t) p_{ij}^k(t) \tag{2.51}$$

para cada $i \in U$ e $j \in U \cup I$. Além disso, pelas fórmulas de y_k e s_k , temos

$$p_{ij}^k(t+1) = \frac{\partial y_k(t+1)}{\partial w_{ij}} = \frac{\partial y_k(t+1)}{\partial s_k(t+1)} \frac{\partial s_k(t+1)}{\partial w_{ij}} = g'_k(s_k(t+1)) \left(\sum_{l \in U} w_{kl} p_{ij}^l(t) + \delta_{ik} x_j(t) \right) \quad (2.52)$$

onde δ_{ik} é o delta de Kronecker.

Com o vetor $e(t)$ disponível no passo t , as equações acima permitem o cálculo do gradiente $\Delta_w E(t)$, pois é possível calcular $p_{ij}^k(t)$ em tempo real. Para o primeiro passo, é conveniente assumir que $p_{ij}^k(t_0) = 0$, pois é razoável assumir que o estado inicial da rede não tem dependência funcional com os pesos (Williams & Zipser, 1995).

Algoritmo de aprendizagem recorrente em tempo real

Inicialização: $w_{ij} = w_{ij}^0$, $p_{ij}^k(0) = 0$

Para $t = 1, 2, \dots, t_1$:

1. Com os pesos w_{ij}^t disponíveis, calcule as saídas dos neurônios da rede e os erros $e_k(t)$.
2. Calcule

$$p_{ij}^k(t) = g'_k(s_k(t)) \left(\sum_{l \in U} w_{kl} p_{ij}^l(t-1) + \delta_{ik} x_j(t-1) \right)$$

onde δ_{ik} é o delta de Kronecker.

3. Calcule o gradiente

$$\frac{\partial E(t)}{\partial w_{ij}} = \sum_{k \in U} e_k(t) p_{ij}^k(t)$$

4. A variação dos pesos é dada por

$$w_{ij}^{t+1} = w_{ij}^t + \eta \frac{\partial E(t)}{\partial w_{ij}}$$

2.6.3 Filtro de Kalman e extensões

O Filtro de Kalman pode ser utilizado para treinamento de redes neurais, tanto *feedforward* como recorrentes. No entanto, por se tratar de otimização de um sistema linear, o Filtro de Kalman em si não é eficiente para o treinamento de redes, sendo necessária, portanto, uma extensão do filtro. Nesta seção, são descritos tanto o Filtro de Kalman quanto o Filtro de Kalman estendido e seu uso para treinamento de redes neurais.

Modelo linear dinâmico

O modelo linear dinâmico para uma seqüência de observações $y(t), t = 1, 2, \dots$, é parametrizado por um vetor de estados $\mathbf{x}(t)$ de dimensão $p \times 1$ é dado por

$$y(t) = F'(t)\mathbf{x}(t) + v(t), \quad v(t) \sim N(0, V(t)) \quad (2.53)$$

$$\mathbf{x}(t) = G(t)\mathbf{x}(t) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N(0, W(t)) \quad (2.54)$$

com informação inicial $(\mathbf{x}(0)|D(0)) \sim N(\mathbf{m}(0), C(0))$, onde $D(t) = (D(t-1), y(t))$. Os erros observacionais e de evolução, $v(t)$ e $\mathbf{w}(t)$, são independentes entre si e no tempo, além de serem independentes de $(\mathbf{x}(0)|D(0))$ (Barbosa, E., 1991; West & Harrison, 1997).

Equivalentemente, o modelo pode ser definido por:

1. Distribuição observacional (verossimilhança): $(y(t)|\mathbf{x}(t)) \sim N(F^t(t)\mathbf{x}(t), V(t))$
2. Distribuição de evolução de estado: $(\mathbf{x}(t)|\mathbf{x}(t-1)) \sim N(G(t)\mathbf{x}(t-1), W(t))$
3. “Priori” inicial (pré-posteriori): $(\mathbf{x}(0)|D(0)) \sim N(\mathbf{m}(0), C(0))$

O filtro de Kalman

Caso padrão: $V(t)$ conhecida

Assumindo que o sistema seja linear e que o ruído é branco e gaussiano, o filtro de Kalman é ótimo no sentido que o erro médio quadrático é minimizado (Anderson & Moore, 1979). As equações básicas de estimação seqüencial para o modelo dinâmico, com a variância observacional $V(t)$ conhecida, são dadas abaixo ($t = 1, 2, \dots$):

$$\mathbf{a}(t) = G(t)\mathbf{m}(t-1) \quad (2.55)$$

$$R(t) = G(t)C(t-1)G'(t) + W(t) \quad (2.56)$$

$$\mathbf{m}(t) = \mathbf{a}(t) + A(t)e(t) \quad (2.57)$$

$$C(t) = R(t) - A(t)Q(t)A'(t) \quad (2.58)$$

onde

$$e(t) = y(t) - F'(t)\mathbf{a}(t) \quad (2.59)$$

$$A(t) = R(t)F(t)Q^{-1}(t) \quad (2.60)$$

$$Q(t) = F'(t)R(t)F(t) + V(t) \quad (2.61)$$

$\mathbf{m}(t)$ e $C(t)$ são, respectivamente, a média e a variância da posteriori $(\mathbf{x}(t)|D(t))$, e serão as estimativas para o vetor de estados $\mathbf{x}(t)$ e sua variância, respectivamente, no último passo do algoritmo.

$\mathbf{a}(t)$ e $R(t)$ são respectivamente média e variância da priori $\mathbf{x}(t)|D(t-1)$. A demonstração encontra-se no apêndice A.1.

Análise conjugada: $V(t)$ não conhecida

Em muitas situações práticas, é comum considerar $V(t)$ desconhecida e constante igual a V . Neste caso, uma análise Bayesiana com priori conjugada para V é possível (West & Harrison, 1997).

Para tanto, considera-se V uma variável aleatória, e a distribuição conjunta a priori de $(\mathbf{x}(t), V)$ é assumida ser normal-gama invertida com n graus de liberdade e média S^{-1} :

$$(\mathbf{x}(t), V|D(t-1)) \sim \text{NI}^{-1}(\mathbf{a}(t), R(t), S(t-1), n(t-1)) \quad (2.62)$$

onde $\mathbf{a}(t)$ e $R(t)$ são obtidos como anteriormente. A distribuição a posteriori conjunta é dada por

$$(\mathbf{x}(t), V|D(t)) \sim \text{NI}^{-1}(\mathbf{m}(t), C(t), S(t), n(t)) \quad (2.63)$$

onde

$$\mathbf{m}(t) = a(t) + A(t)(y(t) - F'(t)a(t)) \quad (2.64)$$

$$A(t) = R(t)F(t)Q^{-1}(t) \quad (2.65)$$

$$Q(t) = F'(t)R(t)F(t) + S(t-1) \quad (2.66)$$

$$C(t) = (R(t) - A(t)Q(t)A'(t))\frac{S(t)}{S(t-1)} \quad (2.67)$$

$$S(t) = S(t-1)\frac{\left(n(t-1) + \frac{e^2(t)}{Q(t)}\right)}{n(t)} \quad (2.68)$$

$$n(t) = n(t-1) + 1 \quad (2.69)$$

Especificação de $W(t)$

É natural considerar que a variância $W(t)$ aumente conforme t cresça, ou seja, há uma perda de informação quando o tempo aumenta.

Da fórmula da variância da priori $(\mathbf{x}(t)|D(t-1))$, temos

$$\text{Var}(\mathbf{x}(t)|D(t-1)) = R(t) = G(t) \text{Var}(\mathbf{x}(t-1)|D(t-1))G'(t) + W(t) = P(t) + W(t) \quad (2.70)$$

Dessa relação, considera-se uma taxa de perda de informação tal que $R(t) = \frac{P(t)}{\delta}$ para algum fator $0 < \delta < 1$. Com isso, $W(t) = P(t)\frac{1-\delta}{\delta}$, o que resulta em um aumento de variância sobre $x(t)$ de $100\frac{(1-\delta)}{\delta}\%$ entre os instantes $t-1$ e t . Para outros detalhes, ver Barbosa (1991) ou West & Harrison (1997).

Filtro de Kalman estendido

O filtro de Kalman supõe que o sistema é descrito por um modelo de espaço de estado linear. Quando o sistema sob consideração é não-linear, usa-se o filtro de Kalman estendido (EKF). No EKF, os termos não-lineares são aproximados pelos termos lineares de primeira (e eventualmente de segunda) ordem pela expansão em série de Taylor.

Considere o sistema não-linear descrito pelo modelo dinâmico

$$y(t) = f(\mathbf{x}(t)) + v(t), \quad v(t) \sim N(0, V(t)) \quad (2.71)$$

$$\mathbf{x}(t) = g(\mathbf{x}(t-1)) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N(0, W(t)) \quad (2.72)$$

onde f e g são funções não-lineares.

Aproximando essas funções usando expansão em série de Taylor de primeira ordem em torno das médias a priori e a posteriori $\mathbf{a}(t)$ e $\mathbf{m}(t)$, temos:

$$f(\mathbf{x}(t)) \approx f(\mathbf{a}(t)) + F^*(\mathbf{a}(t))(\mathbf{x}(t) - \mathbf{a}(t)) \quad (2.73)$$

$$g(\mathbf{x}(t)) \approx g(\mathbf{m}(t)) + G^*(\mathbf{m}(t))(\mathbf{x}(t) - \mathbf{m}(t)) \quad (2.74)$$

onde as matrizes jacobianais $F^*(\mathbf{a}(t))$ e $G^*(\mathbf{m}(t))$ são definidas como

$$F^*(\mathbf{a}(t)) = \left\{ \frac{\partial f(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right\}_{\mathbf{x}(t)=\mathbf{a}(t)} \quad (2.75)$$

$$G^*(\mathbf{m}(t)) = \left\{ \frac{\partial g(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right\}_{\mathbf{x}(t)=\mathbf{m}(t)} \quad (2.76)$$

O sistema não-linear fica, aproximadamente,

$$y(t) = F^*(\mathbf{a}(t))\mathbf{x}(t) + r(\mathbf{a}(t)) + v(t), \quad v(t) \sim N(0, V(t)) \quad (2.77)$$

$$\mathbf{x}(t) = G^*(\mathbf{m}(t))\mathbf{x}(t) + u(\mathbf{m}(t)) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N(0, W(t)) \quad (2.78)$$

onde

$$r(\mathbf{a}(t)) = f(\mathbf{a}(t)) - F^*(\mathbf{a}(t))\mathbf{a}(t) \quad (2.79)$$

$$u(\mathbf{m}(t)) = g(\mathbf{m}(t)) - G^*(\mathbf{m}(t))\mathbf{m}(t) \quad (2.80)$$

As equações do filtro de Kalman são recalculadas substituindo-se as matrizes F e G do sistema linear pelos jacobianos F^* e G^* , respectivamente, e carregando os termos constantes $r(\mathbf{a}(t))$ e $u(\mathbf{m}(t))$.

$$\mathbf{a}(t) = G^*(\mathbf{m}(t))\mathbf{m}(t-1) + u(\mathbf{m}(t)) \quad (2.81)$$

$$R(t) = G^*(\mathbf{m}(t))C(t-1)G^{*\prime}(\mathbf{m}(t)) + W(t) \quad (2.82)$$

$$\mathbf{m}(t) = \mathbf{a}(t) + A(t)e(t) \quad (2.83)$$

$$C(t) = R(t) - A(t)Q(t)A'(t) \quad (2.84)$$

onde

$$e(t) = y(t) - f(\mathbf{a}(t)) \quad (2.85)$$

$$A(t) = R(t)F^*(\mathbf{a}(t))Q^{-1}(t) \quad (2.86)$$

$$Q(t) = F^{*\prime}(\mathbf{a}(t))R(t)F^*(\mathbf{a}(t)) + V(t) \quad (2.87)$$

No caso de aproximação até o termo de segunda ordem da expansão em série de Taylor, vamos considerar que apenas a equação de observação apresente relação não-linear:

$$y(t) = f(\mathbf{x}(t)) + v(t), \quad v(t) \sim N(0, V(t)) \quad (2.88)$$

$$\mathbf{x}(t) = G(t)\mathbf{x}(t-1) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N(0, W(t)) \quad (2.89)$$

A aproximação de f fica:

$$f(\mathbf{x}(t)) \approx f(\mathbf{a}(t)) + J(\mathbf{a}(t))(\mathbf{x}(t) - \mathbf{a}(t)) + \frac{1}{2}(\mathbf{x}(t) - \mathbf{a}(t))' H(\mathbf{a}(t))(\mathbf{x}(t) - \mathbf{a}(t)) \quad (2.90)$$

onde o vetor J (jacobiana) e a matriz H (hessiana) são definidos como

$$J(\mathbf{a}(t)) = \left\{ \frac{\partial f(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right\}_{\mathbf{x}(t)=\mathbf{a}(t)} \quad (2.91)$$

$$H(\mathbf{a}(t)) = \left\{ \frac{\partial^2 f(\mathbf{x}(t))}{\partial \mathbf{x}(t) \partial \mathbf{x}(t)'} \right\}_{\mathbf{x}(t)=\mathbf{a}(t)} \quad (2.92)$$

Recalculando o filtro de Kalman, temos

$$\mathbf{a}(t) = G(t)\mathbf{m}(t-1) \quad (2.93)$$

$$R(t) = G(t)C(t-1)G'(t) + W(t) \quad (2.94)$$

$$\mathbf{m}(t) = \mathbf{a}(t) + A(t)e(t) \quad (2.95)$$

$$C(t) = R(t) - A(t)Q(t)A'(t) \quad (2.96)$$

onde

$$e(t) = y(t) - f(\mathbf{a}(t)) - \frac{1}{2}\text{Tr}\{H(\mathbf{a}(t))R(t)\} \quad (2.97)$$

$$A(t) = R(t)J(\mathbf{a}(t))Q^{-1}(t) \quad (2.98)$$

$$Q(t) = J'(\mathbf{a}(t))R(t)J(\mathbf{a}(t)) + \frac{1}{2}\text{Tr}\{H(\mathbf{a}(t))R(t)\}^2 + V(t) \quad (2.99)$$

Em ambos os casos descritos acima, é suposto que a matriz de variância observacional $V(t)$ é conhecida. Caso a matriz seja desconhecida e constante, pode-se utilizar análise conjugada da mesma maneira como descrito na seção anterior, ou seja, são acrescentadas as duas equações de estimativa de V (ver 2.68 e 2.69) e as equações de $Q(t)$ e $C(t)$ são modificadas: $V(t)$ em 2.99 é substituída por $S(t-1)$ e o lado direito da equação 2.96 é multiplicado por $\frac{S(t)}{S(t-1)}$. A matriz $W(t)$ pode ser especificada pelo método dos fatores de desconto.

Aplicação do filtro de Kalman estendido a redes neurais

O problema de estimativa de pesos em redes neurais pode ser abordado como um problema de estimativa do vetor de estado em um modelo dinâmico, que pode ser feita através do filtro de Kalman estendido (Patel,2000). Nesta situação, os pesos são organizados como um vetor $\mathbf{w}(t)$, referidos como vetor de estado no lugar de $\mathbf{x}(t)$.

Considere uma rede neural (*feedforward* ou recorrente) com um valor correspondente desejado $d(t)$ referente ao nó de saída em cada passo t . O modelo da rede no sistema de espaço de estado (caso gaussiano) é dado por

$$d(t) = f(\mathbf{w}(t), \mathbf{x}(t)) + v(t), \quad v(t) \sim \mathcal{N}(0, V(t)) \quad (2.100)$$

$$\mathbf{w}(t) = \mathbf{w}(t-1) \quad (2.101)$$

onde f é o mapeamento não-linear da rede. Neste caso, a matriz $G(t)$ é a identidade e a matriz de variância W do vetor de estado é nula. Agora, $\mathbf{x}(t)$ é o vetor de entradas exógenas da rede no passo t .

O vetor de estado \mathbf{w} pode ser estimado via filtro de Kalman estendido, utilizando expansão em série de Taylor tanto de primeira quanto de segunda ordem, através das equações já descritas acima.

A maior dificuldade do filtro de Kalman estendido aplicado a redes neurais, principalmente quando se usa aproximação de segunda ordem, é o cálculo das matrizes de derivadas. O jacobiano J e a hessiana H são definidos como

$$J(\mathbf{a}(t)) = \frac{\partial f(\mathbf{w}(t), \mathbf{x}(t))}{\partial \mathbf{w}(t)} \quad (2.102)$$

$$H(\mathbf{a}(t)) = \frac{\partial^2 f(\mathbf{w}(t), \mathbf{x}(t))}{\partial \mathbf{w}(t) \partial \mathbf{w}(t)'} \quad (2.103)$$

e esses cálculos podem ser particularmente difíceis e tediosos, devido a não-linearidade e a complexidade que as funções da rede envolvem, principalmente no caso de redes recorrentes.

Além do filtro de Kalman estendido, outros algoritmos envolvendo o filtro de Kalman vêm sendo estudados e podem servir de alternativa para o treinamento de redes neurais, como o filtro de Kalman “Sigma-Point” (Van der Merwe, 2004) e o filtro de Partícula (Ristic et al., 2004). O estudo destes algoritmos, no entanto, estão fora do escopo desta tese.

Extensões do modelo acima (equações 2.100 e 2.101) onde os pesos da rede não serão mais fixos mas variáveis suavemente longo do tempo ou do espaço serão introduzidos no capítulo 4.

2.7 Intervalos de predição

2.7.1 Método analítico

Nesta subseção serão apresentadas fórmulas para cálculo de intervalos de predição de redes neurais. Estes resultados são assintoticamente válidos e podem ser verificados em Hwang & Ding (1997).

Considere que

$$y(t) = f(\mathbf{x}(t), \mathbf{w}) + \varepsilon(t), \quad t = 1, \dots, T \quad (2.104)$$

onde $y(t)$ é a série temporal de interesse, $f(\cdot)$ é a função (não-linear) de mapeamento da rede, $\mathbf{x}(t)$ é o vetor de variáveis de entrada (que pode ser composto por defasagens da variável $y(t)$ e/ou por variáveis exógenas) de dimensão $N \times 1$, \mathbf{w} é o vetor de pesos da rede e $\varepsilon(t)$ é erro tal que $\varepsilon(t)$ e $\varepsilon(s)$ são iid se $t \neq s$.

Um intervalo de predição assintótico de $1 - \alpha$ para $y(T+k)$ é dado por

$$f(\mathbf{x}(T+k), \hat{\mathbf{w}}) \pm t_{1-\alpha/2, T-(N+2)M-1} \hat{\sigma} \sqrt{S(\hat{\mathbf{w}})} \quad (2.105)$$

onde $t_{1-\alpha/2, T-(N+2)M-1}$ é o quantil de $1 - \alpha/2$ de uma distribuição t -student com $T - (N + 2)M - 1$ graus de liberdade (sendo M o número de neurônios da camada intermediária),

$$\hat{\sigma}^2 = \frac{1}{T - (N + 2)M - 1} \sum_{i=1}^T (y(i) - f(\mathbf{x}(i), \hat{\mathbf{w}}))^2 \quad (2.106)$$

e

$$S(\hat{\mathbf{w}}) = \frac{1}{n} \left[\frac{\partial f'(\mathbf{x}(i), \mathbf{w})}{\mathbf{w}} \right]_{\mathbf{w}=\hat{\mathbf{w}}} \hat{\Sigma}^{-1}(\hat{\mathbf{w}}) \left[\frac{\partial f(\mathbf{x}(i), \mathbf{w})}{\mathbf{w}} \right]_{\mathbf{w}=\hat{\mathbf{w}}} \quad (2.107)$$

onde

$$\hat{\Sigma}(\hat{\mathbf{w}}) = \frac{1}{n} \sum_{i=1}^T \left[\frac{\partial f(\mathbf{x}(i), \mathbf{w})}{\mathbf{w}} \frac{\partial f'(\mathbf{x}(i), \mathbf{w})}{\mathbf{w}} \right]_{\mathbf{w}=\hat{\mathbf{w}}} \quad (2.108)$$

2.7.2 Método numérico: reamostragem de resíduos (bootstrap)

Devido a forte não-linearidade presente em modelos de redes neurais, muitas vezes a aplicação de métodos analíticos para cálculo de intervalos de confiança e predição (que dependem de momentos de 2^a ordem) não funcionam tão bem quanto para os momentos de 1^a ordem.

Para contornar esse problema, é possível utilizar a técnica de reamostragem (bootstrap) para calcular intervalos de predições para séries temporais. Nesses tipos de aplicações, o mais comum é fazer reamostragem dos resíduos. Johnston e Dinardo (1997) mostram como calcular intervalos de predição para um modelo linear gaussiano de série temporal. É possível fazer uma extensão para o caso não-linear de redes neurais.

Suponha que

$$y(t) = f(\mathbf{x}(t), \mathbf{w}) + \varepsilon(t), \quad t = 1, \dots, T \quad (2.109)$$

onde $y(t)$ é a série temporal de interesse, $f(\cdot)$ é a função (não-linear) de mapeamento da rede, $\mathbf{x}(t)$ é o vetor de variáveis de entrada (que pode ser composto por defasagens da variável $y(t)$ e/ou por variáveis exógenas), \mathbf{w} é o vetor de pesos da rede e $\varepsilon(t)$ é erro tal que $\varepsilon(t)$ e $\varepsilon(s)$ são iid se $t \neq s$.

O objeto de interesse é um intervalo de confiança para $y(T+k)$, $k > 0$, ou seja, um valor futuro de y . Em primeiro lugar, é preciso calcular a estimativa pontual de $y(T+k)$, ou seja, $\hat{y}(T+k)$. Para isso, basta tomar a estimativa de $\hat{\mathbf{w}}$ de \mathbf{w} (calculada através de algum método de estimação de pesos de redes neurais) e os valores de $\mathbf{x}(T+k)$ e calcular o valor de $\hat{y}(T+k) = f(\mathbf{x}(T+k), \hat{\mathbf{w}})$.

O processo de reamostragem consiste em, digamos, B repetições do seguinte processo:

- Calcule os resíduos $\hat{\varepsilon}(t) = y(t) - \hat{y}(t)$, $t = 1, \dots, T$.
- Retire uma amostra aleatória $\hat{\varepsilon}^i(t) : t = 1, \dots, T$ com reposição de tamanho T do conjunto de resíduos $\hat{\varepsilon}$.
- Reescale os resíduos:

$$\tilde{\varepsilon}^i(t) = \frac{\hat{\varepsilon}^i(t)}{\sqrt{1 - h(t)}} - \frac{1}{T} \sum_{t=1}^T \frac{\hat{\varepsilon}^i(t)}{\sqrt{1 - h(t)}} \quad (2.110)$$

onde $h(t)$ é o t -ésimo elemento diagonal da matriz chapéu H . Se estivéssemos tratando do caso linear, a matriz seria definida por $H = X(X'X)^{-1}X'$. Como estamos em um caso não-linear, é utilizada uma aproximação baseada no jacobiano da função f . A matriz é definida por

$$h = \frac{\partial f(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \left(\frac{\partial f'(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \frac{\partial f(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \right)^{-1} \frac{\partial f'(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \quad (2.111)$$

o que vamos chamar de matriz chapéu generalizada.

- A i -ésima amostra bootstrap é obtida por

$$\hat{y}^i(t) = f(\mathbf{x}(t), \hat{\mathbf{w}}) + \tilde{\varepsilon}^i(t), \quad t = 1, \dots, T \quad (2.112)$$

onde $\hat{\mathbf{w}}$ é a estimativa do vetor de pesos \mathbf{w} e $\tilde{\varepsilon}^i(t)$ é amostrado aleatoriamente com reposição de $\hat{\varepsilon}$.

Com essas i amostras, é possível calcular novamente as correspondentes estimativas de \mathbf{w} , digamos $\hat{\mathbf{w}}^i$. Com isso, obtém-se uma nova estimativa $\hat{y}^i(T+k)$. A variância bootstrap é calculada através da fórmula

$$\hat{\sigma}_{\hat{y}(T+k)}^2 = \frac{1}{B-1} \sum_{i=1}^B (\hat{y}^i(T+k) - \hat{y}^{(.)}(T+k))^2 \quad (2.113)$$

onde

$$\hat{y}^{(\cdot)}(T+k) = \frac{1}{B} \sum_{i=1}^B \hat{y}^i(T+k) \quad (2.114)$$

Utilizando a estimativa do desvio padrão, é possível calcular um intervalo de confiança para $\hat{y}(T+k)$.

Capítulo 3

Modelos lineares para séries temporais de memória longa

Séries temporais de memória longa são caracterizadas por uma forte dependência com seu passado distante. São caracterizadas por um decaimento lento (hiperbólico) das autocorrelações conforme cresce o valor dos lags. Para modelar esse tipo de processo, um dos modelos mais usados é o ARFIMA (“Auto Regressive Fractionally Integrated Moving Average”), que é uma extensão do conhecido modelo ARIMA, mas com ordem de integração fracionária. Esse tipo de modelo foi introduzido originalmente na literatura por Granger e Joyeux (1980) e Hosking (1981).

Nesta seção, é feita uma breve revisão do modelo ARMA, seguido pela definição do modelo ARFIMA e dos principais métodos de estimação da ordem fracionária do modelo.

3.1 Modelos ARIMA

Os modelos ARIMA foram introduzidos por Box & Jenkins (1970). Modelos ARFIMA são uma extensão dos modelos ARIMA (ver Box, Jenkins & Reinsel, 1994; Wei, 1990; ou Morettin e Toloi, 2004).

Considere uma série temporal $x(t) : t = 1, 2, \dots$. Sem perda de generalidade, podemos considerar $\mu = E(x(t)) = 0$, ou, equivalentemente, substituir $x(t)$ por $x(t) - E(x(t))$. Seja B o operador *backshift* definido por $B^i x(t) = x(t - i)$, $i = 1, 2, \dots$. Além disso, seja $a(t)$ ($t = 1, 2, \dots$) ruído branco (variáveis aleatórias não-correlacionadas com média zero e variância finita σ^2). Um modelo ARMA(p, q) para $x(t)$ é definido como a solução estacionária de

$$\phi(B)x(t) = \theta(B)a(t) \quad (3.1)$$

onde

$$\phi(B) = 1 - \sum_{j=1}^p \phi_j B^j \quad (3.2)$$

$$\theta(B) = 1 + \sum_{j=1}^q \theta_j B^j \quad (3.3)$$

Para que o processo seja inversível, é necessário que as raízes de $\theta(B) = 0$ estejam fora do círculo unitário. Para que o processo seja estacionário, é necessário que as raízes de $\phi(B) = 0$ estejam fora do círculo unitário. Também é necessário assumir que $\phi(B) = 0$ e $\theta(B) = 0$ não tenham raízes comuns. No processo ARMA(p, q), p representa a ordem auto-regressiva e q a ordem média móvel do processo. Se $q = 0$, o processo é chamado AR(p), e se $p = 0$, o processo é chamado MA(q).

Para um processo estacionário $x(t)$ com média $E(x(t)) = \mu$ e variância $\text{Var}(x(t)) = E[(x(t) - \mu)^2] = \sigma^2$, constantes, as covariâncias $\text{Cov}(x(t), x(s))$ são funções apenas da diferença de tempo $|t - s|$. Nesse caso, a covariância entre $x(t)$ e $x(t + k)$ é

$$\gamma(k) = \text{Cov}(x(t), x(t + k)) = E[(x(t) - \mu)(x(t + k) - \mu)] \quad (3.4)$$

e a correlação entre $x(t)$ e $x(t + k)$ é

$$\rho(k) = \frac{\text{Cov}(x(t), x(t + k))}{\sqrt{\text{Var}(x(t))}\sqrt{\text{Var}(x(t + k))}} = \frac{\gamma(k)}{\gamma(0)} \quad (3.5)$$

pois $\text{Var}(x(t)) = \text{Var}(x(t + k)) = \sigma^2 = \gamma(0)$. Como essas funções representam a covariância e a correlação entre $x(t)$ e $x(t + k)$ de um mesmo processo, separados apenas por k defasagens, elas são chamadas função de autocovariância e função de autocorrelação. Outra quantidade importante é a função de autocorrelação parcial $\phi(kk) = \text{Cov}(x(t), x(t + k)|x(t + 1), \dots, x(t + k - 1))$, que indica a correlação entre $x(t)$ e $x(t + k)$ depois que a dependência das variáveis “anteriores” $x(t + 1), x(t + 2), \dots$ e $x(t + k - 1)$ é removida. A expressão de $\phi(kk)$ é dada por

$$\phi(kk) = \begin{vmatrix} 1 & \rho(1) & \rho(2) & \dots & \rho(k-2) & \rho(1) \\ \rho(1) & 1 & \rho(1) & \dots & \rho(k-3) & \rho(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \dots & \rho(1) & \rho(k) \\ \hline 1 & \rho(1) & \rho(2) & \dots & \rho(k-2) & \rho(k-1) \\ \rho(1) & 1 & \rho(1) & \dots & \rho(k-3) & \rho(k-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \dots & \rho(1) & 1 \end{vmatrix} \quad (3.6)$$

Para maiores detalhes, ver Wei (1990).

Uma propriedade importante das funções de autocorrelação simples e parcial de processos ARMA é que, para um processo estacionário, elas decaem exponencialmente com o tempo, ou seja, a dependência de $x(t)$ com $x(t+k)$ tende a zero rapidamente quando k cresce. Os estimadores dessas funções são importantes para se determinar a ordem (p, q) de um processo ARMA.

No entanto, muitas vezes séries temporais são não-estacionárias, ou seja, algumas raízes do polinômio AR não estão fora do círculo unitário. Em muitos casos, de acordo com Box e Jenkins (1976), essas séries se comportam com uma “não-estacionaridade homogênea”, ou seja, o comportamento não-estacionário da série é constante e não depende do particular índice t . Se $\psi(B)$ é o operador auto-regressivo que descreve esse comportamento, temos

$$\psi(B)(x(t) + C) = \psi(B)x(t) \quad (3.7)$$

para qualquer constante C . Isso implica que $\psi(B)$ deve ser da forma

$$\psi(B) = \phi(B)(1 - B)^d \quad (3.8)$$

para algum $d > 0$ para o qual $\phi(B)$ seja um operador auto-regressivo estacionário. Ou seja, a série $x(t)$ não é estacionária, mas a d -ésima diferença $(1 - B)^d x(t)$, para algum inteiro $d \geq 1$, é estacionária.

Em geral, a série diferenciada $(1 - B)^d x(t)$ segue um processo estacionário ARMA(p, q), ou seja,

$$\phi(B)(1 - B)^d x(t) = \theta(B)a(t) \quad (3.9)$$

O modelo acima é chamado ARIMA de ordem (p, d, q) .

Eventualmente, o processo $x(t)$ pode ser sazonal. Muitas séries temporais econométricas contêm um fenômeno sazonal que se repete após um período de tempo regular. O menor período para esse fenômeno repetitivo é chamado de período sazonal. Por exemplo, as vendas mensais de chocolate tendem a aumentar em março-abril, época da Páscoa. O período sazonal nesse caso é 12 (meses).

Quando não sabemos que a série $x(t)$ contém sazonalidade e ajustamos um modelo ARIMA não-sazonal

$$\phi(B)(1 - B)^d x(t) = \theta(B)b(t), \quad (3.10)$$

a série $b(t)$ não será ruído branco, pois apresentará correlações entre os períodos sazonais. A relação entre esses períodos também pode ser representada por um modelo ARIMA

$$\Phi(B^s)(1 - B^s)^D b(t) = \Theta(B^s)b(t), \quad (3.11)$$

onde

$$\Phi(B^s) = 1 - \sum_{j=1}^P \Phi_j B^{js} \quad (3.12)$$

$$\Theta(B^s) = 1 + \sum_{j=1}^Q \Theta_j B^{js} \quad (3.13)$$

Os polinômios $\Phi(B)$ e $\Theta(B)$ devem ter raízes fora do círculo unitário e não podem ter raízes comuns, $a(t)$ é ruído branco, P é a ordem auto-regressiva sazonal, Q é a ordem média móvel sazonal, D é a ordem de diferenciação sazonal e s é o período de sazonalidade.

Combinando as partes não-sazonal e sazonal do modelo, temos o modelo ARIMA sazonal

$$\Phi(B^s)\phi(B)(1 - B)^d(1 - B^s)^Dx(t) = \theta(B)\Theta(B^s)a(t) \quad (3.14)$$

que é freqüentemente denotado por SARIMA($p, d, q) \times (P, D, Q)_s$.

A média μ do processo pode depender de variáveis exógenas, onde $\mu = f(Z, \beta)$, sendo Z uma matriz $T \times k$ de regressores exógenos. Neste capítulo, quando nos referirmos a tal processo, vamos considerar o caso linear, onde $f(Y, \beta) = Y\beta$, sendo β um vetor $k \times 1$.

Existem diversos métodos para estimação dos parâmetros de um modelo ARIMA, como método dos momentos ou máxima verossimilhança. Para maiores detalhes, ver Box, Jenkins & Reinsel (1994), Wei (1990) ou Morettin e Toloi (2004).

3.2 Modelos ARFIMA

Um processo ARFIMA é uma extensão natural do processo ARIMA, no qual a ordem de integração d pode assumir qualquer valor real. No caso de d inteiro ($d \geq 0$), então $(1 - B)^d$ pode ser escrito como

$$(1 - B)^d = \sum_{k=0}^d \binom{d}{k} (-1)^k B^k \quad (3.15)$$

com coeficientes binomiais

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)} \quad (3.16)$$

onde $\Gamma(\cdot)$ é a função gama. Como a função gama é definida para todos os números reais, a definição acima do coeficiente binomial pode ser estendida a todos os números reais d . Note que para inteiros negativos a função gama tem pólos tais que o coeficiente binomial é zero se $k > d$ e d é inteiro. Formalmente, podemos definir $(1 - B)^d$ para qualquer número real d por

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k \quad (3.17)$$

Para todos os inteiros positivos, apenas os primeiros $d + 1$ termos são não-nulos e com isso obtemos a definição original da d -ésima diferença com o operador $(1 - B)^d$. Para valores não-inteiros de d , a soma em 3.17 é genuína sobre um número infinito de índices. A definição 3.15 pode agora ser estendida para valores não inteiros de d .

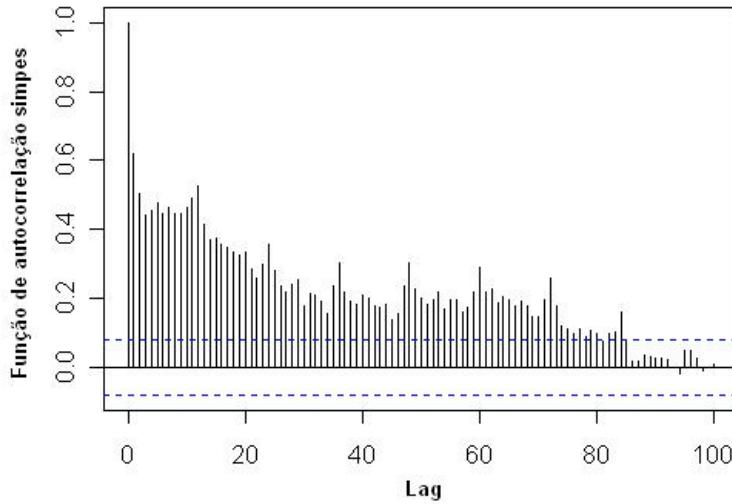


Figura 3.1: Autocorrelação simples de um processo com memória longa

Um processo ARFIMA(p, d, q) (“Auto Regressive Fractionally Integrated Moving Average”) é definido por

$$\phi(B)(1 - B)^d x(t) = \theta(B)a(t) \quad (3.18)$$

onde $|d| < \frac{1}{2}$ (Granger e Joyeux, 1980, Herencia, 2002).

No caso de memória longa, a amplitude de interesse para d é $[0, \frac{1}{2}]$. Quando $d < 0$, d não pode ser interpretada como ordem de integração no sentido que estamos acostumados. Quando $d \geq \frac{1}{2}$, o processo não é estacionário (Beran, 1994).

Para um processo ARFIMA, as autocorrelações não decaem exponencialmente, como em um processo ARMA, e sim apenas hiperbolicamente (Hosking, 1981). As autocorrelações podem ser significativas por dezenas, ou até centenas, de lags (ver figura 3.1). Chamamos esse comportamento de memória longa, e modelos do tipo ARFIMA são altamente indicados para modelar séries que apresentam essa característica.

3.2.1 Estimação da ordem de integração fracionária

Considere o modelo ARFIMA(p, d, q)

$$\phi(B)(1 - B)^d(x(t) - \mu) = \theta(B)a(t) \quad (3.19)$$

A função de autocovariância de um processo estacionário ARMA de média μ é

$$c(i) = E[(x(t) - \mu)(x(t - i) - \mu)] \quad (3.20)$$

que define a matriz de variância da distribuição conjunta de $\mathbf{x} = (x(1), \dots, x(T))'$

$$V(\mathbf{x}) = \begin{pmatrix} c(0) & c(1) & \dots & c(T-1) \\ c(1) & c(0) & \dots & c(T-2) \\ \vdots & \vdots & \ddots & \vdots \\ c(T-1) & c(T) & \dots & c(0) \end{pmatrix} = \Sigma \quad (3.21)$$

Sob normalidade, $\mathbf{x} \sim N_T(\mu, \Sigma)$.

Os métodos de estimação descritos a seguir foram implementados no programa Ox por Doornik e Ooms (2001), sendo o primeiro deles também disponível no software S-PLUS.

Estimação por máxima verossimilhança

Considere

$$\mathbf{z} = \mathbf{x} - \mu, \quad \mu = f(Y, \beta) \quad (3.22)$$

onde Y é uma matriz $T \times k$ de regressores exógenos. Usualmente, $f(Y, \beta) = Y\beta$ (caso linear), onde β é um vetor $k \times 1$, ou $f(Y, \beta) = c$ (constante), onde μ não depende de regressores exógenos.

Considerando $a(t) \sim N(0, \sigma^2)$ e utilizando algum procedimento para cálculo das autocovariâncias em Σ , a log-verossimilhança de \mathbf{x} fica

$$\log L(d, \phi, \theta, \beta, \sigma^2) = -\frac{T}{2} \log 2\pi - \frac{1}{2} \log |\Sigma| - \frac{1}{2} \mathbf{z}' \Sigma^{-1} \mathbf{z} \quad (3.23)$$

É conveniente começar determinando σ^2 , escrevendo $\Sigma = R\sigma^2$:

$$\log L(d, \phi, \theta, \beta, \sigma^2) \propto -\frac{1}{2} \log |R| - \frac{T}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \mathbf{z}' R^{-1} \mathbf{z} \quad (3.24)$$

Derivando com relação a σ^2 , temos

$$\frac{\partial \log L(d, \phi, \theta, \beta, \sigma^2)}{\partial \sigma^2} = -\frac{T}{2\sigma^2} + \frac{\mathbf{z}' R^{-1} \mathbf{z}}{2\sigma^4} \quad (3.25)$$

Igualando a zero, obtemos o estimador de σ^2

$$\hat{\sigma}^2 = T^{-1} \mathbf{z}' R^{-1} \mathbf{z} \quad (3.26)$$

Substituindo na log-verossimilhança, obtemos

$$\ell(d, \phi, \theta, \beta) = -\frac{T}{2} \log 2\pi - \frac{1}{2} \log |R| - \frac{T}{2} \log (T^{-1} \mathbf{z}' R^{-1} \mathbf{z} - \frac{T}{2}) \quad (3.27)$$

Para eliminar β , basta fazer uma analogia ao caso de regressão linear:

$$\hat{\beta} = (X' R^{-1} X)^{-1} X' R^{-1} \mathbf{y} \quad (3.28)$$

$$\hat{\mathbf{z}} = \mathbf{y} - X \hat{\beta} \quad (3.29)$$

Substituindo na log-verossimilhança, obtemos

$$\ell(d, \phi, \theta) = -\frac{T}{2}(1 + \log 2\pi) - \frac{1}{2}\log|R| - \frac{T}{2}\log(T^{-1}\hat{\mathbf{z}}'R^{-1}\hat{\mathbf{z}}) \quad (3.30)$$

Na equação acima, o termo $-\frac{T}{2}(1 + \log 2\pi)$ é constante, e portanto não faz diferença na maximização da função. Note que $T^{-1}\hat{\mathbf{z}}'R^{-1}\hat{\mathbf{z}}$ é o estimador da variância σ^2 . Portanto, a função utilizada no procedimento de maximização é

$$-\frac{1}{2}\log|R| - \frac{T}{2}\log\sigma^2 \quad (3.31)$$

que pode ser maximizada utilizando-se de algum procedimento numérico.

Perfil de verossimilhança modificada

An e Bloomfield (1993) reduziram o viés do estimador de máxima verossimilhança devido a presença de parâmetros de perturbação desconhecidos nos regressores. A principal modificação em relação à equação 3.30 é o termo de correção $|X'R^{-1}X|$, que corrige os vieses na estimação de d , ϕ e θ . Essa correção leva a uma diminuição nos valores dos erros médios quadráticos dos estimadores, já que essa correção não aumenta a variância (Doornik e Ooms, 2003). O perfil de verossimilhança modificada é

$$\ell_m(d, \phi, \theta) = -\frac{T}{2}(1 + \log 2\pi) - \left(\frac{1}{2} - \frac{1}{T}\right)\log|R| - \frac{T-k-2}{2}\log(T^{-1}\hat{\mathbf{z}}'R^{-1}\hat{\mathbf{z}}) - \frac{1}{2}\log|X'R^{-1}X| \quad (3.32)$$

As outras correções presentes na fórmula acima são devidas aos graus de liberdade. Quando $p = q = d = 0$, o estimador da variância dos resíduos usa a conhecida correção dos graus de liberdade devida à estimação dos parâmetros da regressão:

$$\hat{\sigma}^2 = \frac{1}{T-k}\hat{\mathbf{z}}'R^{-1}\hat{\mathbf{z}} \quad (3.33)$$

Mínimos quadrados não-lineares

Definindo $\hat{a}(t)$ como os resíduos obtidos ao aplicar o filtro ARFIMA(p, d, q) em $z(t) = x(t) - \mu(t)$, a variância residual é

$$\hat{\sigma}^2 = \frac{1}{T-k} \sum_{t=1}^T \hat{a}^2(t) \quad (3.34)$$

Mínimos quadrados não-lineares simplesmente maximiza a função

$$f(d, \phi, \theta, \beta) = \frac{1}{2}\log\left(\frac{1}{T} \sum_{t=1}^T \hat{a}^2(t)\right) \quad (3.35)$$

que pode ser maximizada utilizando-se de algum procedimento numérico como por exemplo Gauss-Newton. Para maiores detalhes, ver Doornik e Ooms (2001).

Capítulo 4

Redes Neurais Recorrentes para Séries Espaciais de Memória Curta: predição em transects

A metodologia desenvolvida no capítulo anterior a respeito de redes neurais recorrentes e correspondente algoritmo de aprendizagem baseado no filtro de Kalman estendido de 2^a ordem é aqui neste capítulo não somente aplicada a dados reais (sob forma de séries espaciais) como também ampliada ou estendida de modo a permitir que os pesos da rede variem suavemente ao longo do espaço ou do tempo, com intuito de melhor expressar as características do processo gerador dos dados e obter melhor performance de predição.

4.1 Introdução

O estudo das relações entre os atributos (variáveis) do solo é assunto de grande importância na área agronômica, de modo a buscar um manejo mais racional e adequado dos recursos naturais do meio ambiente e uma melhoria da produtividade agrícola. É de grande interesse um modelo que relacione uma variável do solo de medição complexa (como Nitrogênio Total) com outras variáveis de medição simples (como Carbono Orgânico) no intuito de predizer a primeira com base na medição das demais.

Neste capítulo é proposto um modelo de redes neurais recorrentes para avaliar a relação entre as variáveis. O modelo desenvolvido é uma rede neural recorrente com pesos variáveis ao longo do espaço. Para treinamento, foi desenvolvido um algoritmo baseado em filtro de Kalman estendido de 2^a ordem. A eficiência preditiva do modelo é mostrada através da comparação de medidas de performance de

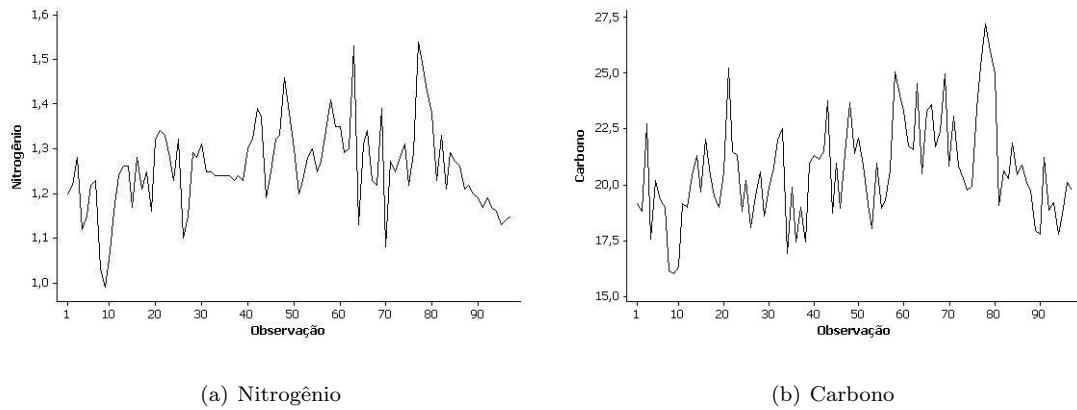


Figura 4.1: Séries espaciais

predição com outros modelos de referência. Essas relações são chamadas funções de pedotransferência (Bouma, 1989) e têm sido muito estudadas na área de ciência do solo (Wendroth et al., 1997; Timm & Barbosa et al., 2003. Reichardt & Timm, 2004). Essas relações apresentam certa dificuldade em sua modelagem devido a três fatores principais: a dependência espacial dos dados, a não linearidade da relação em si e a não homogeneidade do solo. As redes recorrentes do capítulo anterior já incorporaram os dois primeiros desses fatores, e o último será incorporado através de extensão do modelo anterior.

4.2 Análise preliminar dos dados

Em julho de 1998 foi realizado, na EMBRAPA de Jaguariúna, SP, o plantio de uma área relativa a um experimento com aveia preta (*avena strigosa*), onde foi definido um esquema amostral correspondente. Foram demarcadas no campo parcelas e subparcelas, com alocação espacial de três linhas de amostragem, ditas transects, cada uma com 192 metros de comprimento, uma em cada sistema de preparo do solo: convencional, plantio direto e cultivo mínimo. Um transect portanto representa uma linha no solo ao longo do qual a amostra é retirada (para análise em laboratório) em pontos igualmente espaçados.

Na análise em questão, são considerados apenas os dados coletados no sistema de preparo convencional; portanto, com apenas uma linha de amostragem de 192 metros, com observações a cada 2 metros, o que representa 97 dados para cada variável representando propriedade do solo a ser medida.

Com base no transect assim definido foram tomadas sub-amostras na profundidade de 0-20 cm, com pontos de amostragem a cada 2 metros e a 0,5 m ao lado. Nesses pontos foram determinados o pH em água, condutividade elétrica, carbono orgânico (C-orgânico) e nitrogênio total (N-total) no solo (EMBRAPA, 1999).

Dentre essas variáveis, são aqui consideradas apenas as 2 últimas, tendo em vista o objetivo específico de estudar o nitrogênio total do solo. Essas 2 variáveis (séries espaciais) são apresentadas graficamente

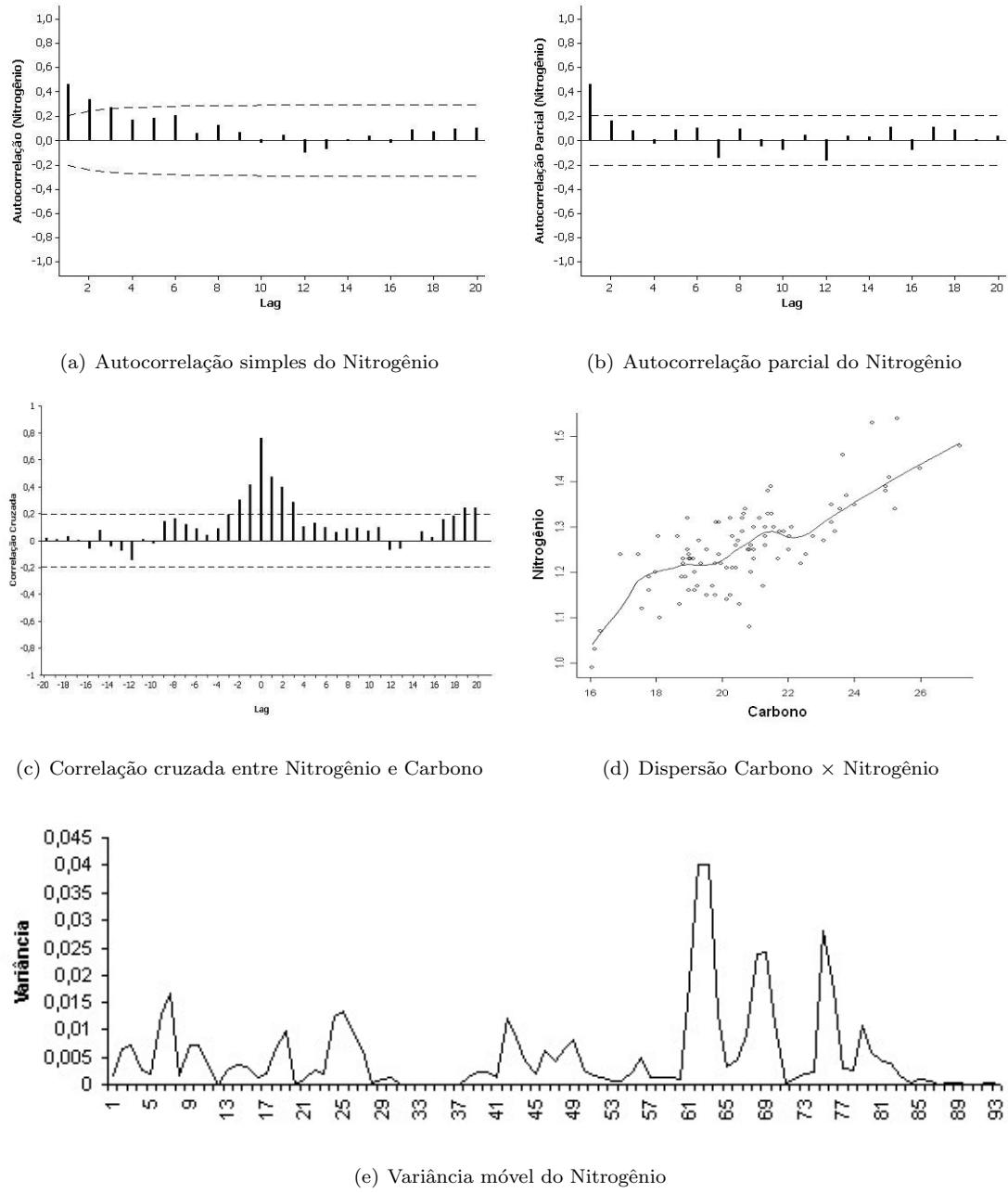


Figura 4.2: Gráficos de correlações, dependência e variância

nas figuras 4.1(a) e 4.1(b).

A dependência do nitrogênio nos valores de carbono no mesmo ponto (mostrada no diagrama de dispersão) e em pontos vizinhos é visualizada no gráfico de correlação cruzada da figura 4.2(c); já a dependência espacial do nitrogênio em relação ao nitrogênio de pontos próximos é mostrada nas figuras 4.2(a) e 4.2(b), através das autocorrelações simples e parcial, que sugerem com razoável nitidez um processo auto-regressivo de 1^a ordem para a série de nitrogênio. Portanto, esta breve análise descritiva e exploratória dessas 2 séries espaciais sugere que o nitrogênio do solo a cada ponto pode ser razoavelmente explicado através do carbono no mesmo ponto e do nitrogênio em pontos próximos.

Também é mostrado um diagrama de dispersão (figura 4.2(d)), o qual evidencia a dependência ou correlação existente entre essas variáveis do solo, assim como uma certa não-linearidade da relação entre as variáveis, expressa pelo suavizador loess (Cleveland, 1979).

A correlação expressiva mostrada no diagrama de dispersão está de acordo com os “movimentos parcialmente coincidentes” dos gráficos das séries de nitrogênio e carbono (queda seguida de subida ao início de ambas as séries, passando por uma relativa estabilidade, culminando por subida e queda ao final de ambas).

Também vale notar que a figura 4.2(e) representa a variância móvel do transect (variância calculada a cada sete pontos) e demonstra uma heterogeneidade da série, que pode ser verificada através das oscilações presentes no gráfico.

Embora não apresentada graficamente, foi verificada a hipótese de normalidade dos dados, sendo esta não rejeitada ($p\text{-valor} > 0,150$, teste de Komolgorov-Smirnov).

4.3 Modelos utilizados

4.3.1 Regressão linear

a) Regressão com erros auto-regressivos de 1^a ordem (coeficientes fixos)

Neste tipo de modelo de regressão para dados dependentes (séries espaciais y_i e x_i , $i = 1, \dots, n$), tem-se, no ponto amostral i :

$$\begin{aligned} y_i &= \mathbf{x}_i \beta + \nu_i \\ \nu_i &= \varphi_1 \nu_{i-1} + \epsilon_i \\ \epsilon_i &\sim N(0, \sigma^2) \end{aligned} \tag{4.1}$$

onde $\mathbf{x}_i = (1, x_i)$, β é um vetor de coeficientes, $|\varphi_1| < 1$ e os ϵ_i 's são não correlacionados com variância constante e distribuição gaussiana. Quando o coeficiente φ_1 é nulo, tem-se o caso de regressão linear padrão. Neste modelo, os estimadores dos β 's da regressão são corrigidos devido à autocorrelação es-

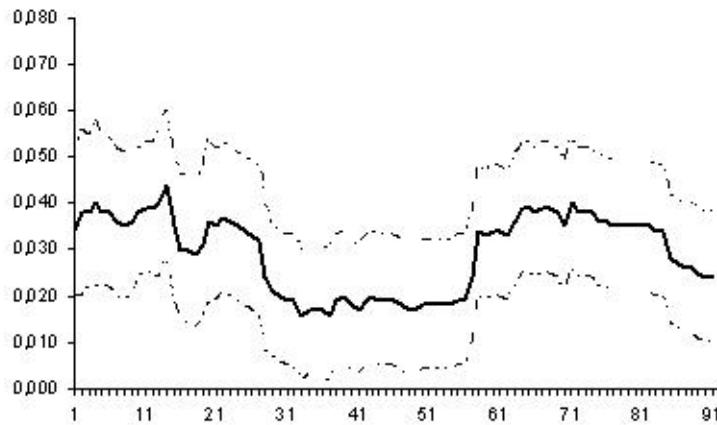


Figura 4.3: Evolução dos coeficientes da variável carbono ao longo dos pontos do transect

pacial utilizando-se o método de mínimos quadrados generalizados (Greene, 2003). Nesta aplicação, a variável resposta y_i é o nitrogênio total e o regressor x_i é o carbono orgânico.

b) Regressão com coeficientes variáveis

No modelo de regressão dinâmica, que é um caso particular do modelo linear dinâmico de West & Harrison (1997), a equação de estado, que descreve a evolução dos coeficientes de regressão β , pode ser escrita como um passeio aleatório vetorial

$$\beta_i = \beta_{i-1} + \mathbf{w}_i \quad (4.2)$$

onde $\mathbf{w}_i \sim N(0, W)$ são não correlacionados (ruído branco). Neste modelo, o vetor de estado está relacionado à variável observada através da equação de observação:

$$y_i = F_i \beta_i + v_i \quad (4.3)$$

onde y_i é a variável resposta, a matriz F_i , que se reduz a um vetor no caso de resposta unidimensional, contém os regressores e v_i são erros não correlacionados com distribuição normal de média zero e variância V , ou seja, $v_i \sim N(0, V)$. Nesse caso, y_i é a variável nitrogênio total (N_i) e o vetor F_i é composto dos regressores nitrogênio total defasado (N_{i-1}) e carbono orgânico (C_i), de modo que $F_i = (1, N_{i-1}, C_i)$. A estimativa (seqüencial) dos parâmetros é feita através de uma versão estendida do filtro de Kalman, considerando um modelo normal-gama inversa para a distribuição a priori de (β, V) e especificando uma estrutura bloco-diagonal para W através do chamado método dos fatores de desconto já descrito no capítulo anterior (West & Harrison, 1997). A evolução dos coeficientes da variável carbono é mostrada na figura 4.3, o que comprova que a relação entre nitrogênio e carbono muda ao longo do espaço, sugerindo assim a conveniência de se utilizar um modelo com coeficientes não constantes.

Para implementação computacional deste modelo foi utilizado o software BATS desenvolvido por Pole et al. (1994).

4.3.2 Auto-regressão vetorial - VAR

a) Modelo VAR padrão

O modelo VAR (Vector Auto-Regression, Greene, 2003) representa uma regressão de cada variável do sistema sobre suas defasagens e defasagens das outras variáveis. No caso de duas variáveis x_i e y_i , temos

$$x_i = \lambda + \sum_{j=1}^m \beta_j x_{i-j} + \sum_{j=1}^m \gamma_j y_{i-j} + \epsilon_{1i} \quad (4.4)$$

$$y_i = \lambda' + \sum_{j=1}^m \theta_j x_{i-j} + \sum_{j=1}^m \alpha_j y_{i-j} + \epsilon_{2i} \quad (4.5)$$

onde os λ 's, β 's, γ 's, θ 's e α 's são os coeficientes do modelo, m é a ordem auto-regressiva (no caso em questão, $m = 1$) e os ϵ 's são erros estocásticos não correlacionados, por exemplo, ruído branco, como no item anterior. É utilizada também uma versão alternativa (**VAR corrigido**), tomando y_i como nitrogênio total no ponto i e x_i como carbono orgânico no ponto $i + 1$, o que resulta no regressor carbono defasado com índice i e não $i - 1$ como no modelo VAR original.

b) Modelo VAR estrutural: espaço de estado

Na análise de espaço de estado, o estado correspondente a uma variável medida num dado ponto $p = i$ é relacionada ao estado da mesma e de outras variáveis em um ponto $i - 1$. A equação básica, chamada equação de estado, pode ser escrita como:

$$\mathbf{x}_i = \Phi x_{i-1} + \mathbf{w}_i \quad (4.6)$$

onde \mathbf{x}_i é o vetor de estado (conjunto de variáveis de estado ou latentes) em i , Φ é uma matriz de coeficientes de estado ou de transição e \mathbf{w}_i são ruídos do sistema para $i = 1, 2, 3, \dots, N$, tendo, por suposição, vetor média zero, variâncias constantes, não correlacionados e normalmente distribuídos. Esta é a estrutura de um modelo auto regressivo vetorial de 1^a ordem para o vetor de estado, o qual é “embebido” em uma equação de observação:

$$y_i = A_i \mathbf{x}_i + v_i \quad (4.7)$$

onde o vetor de observação y_i é relacionado com o vetor de estado \mathbf{x}_i por uma matriz A_i (eventualmente uma matriz identidade) e os ruídos v_i têm média zero, são não correlacionados e normalmente distribuídos, como na equação anterior. Os ruídos w_i e v_i são considerados independentes um do outro. Aqui, o vetor de estado \mathbf{x}_i representa os valores verdadeiros (não observados ou latentes, isto é, livres de

erro de medição) de nitrogênio total (N) e carbono orgânico (C), e a equação de estado pode ser escrita especificamente como:

$$\begin{pmatrix} N_i \\ C_i \end{pmatrix} = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix} \begin{pmatrix} N_{i-1} \\ C_{i-1} \end{pmatrix} + \begin{pmatrix} w_{N_i} \\ w_{C_i} \end{pmatrix} \quad (4.8)$$

onde os erros desta equação têm estrutura de ruído branco, como descrito anteriormente. Os coeficientes de estado Φ_{kj} e demais parâmetros das equações acima são aqui estimados através de procedimento recursivo baseado no Filtro de Kalman em associação com o algoritmo EM dado por Shumway e Stoffer (2000) através do software ASTSA desenvolvido por esses autores.

4.3.3 Regressão não-linear (não-paramétrica)

a) Modelos aditivos - GAM

O modelo de regressão aditivo, assim como sua versão generalizada GAM (“Generalized Additive Model”), tem seu preditor mais flexível que o caso linear, mantendo-se apenas a pressuposição de aditividade, dado por

$$y_i = \alpha + \sum_{j=1}^m f_j(x_i) + \epsilon_i \quad (4.9)$$

onde α é o intercepto do modelo, m é o número de preditores (no caso em questão, $m = 2$) e os ϵ 's são os erros aleatórios não correlacionados de média nula e variância constante como nos modelos anteriores. As f_j 's são funções suaves (i.e., que apresentam derivadas contínuas), as quais são especificadas na prática por suavizadores típicos, como, por exemplo, os splines cúbicos e os suavizadores lowess - locally weighted scatterplot smoother (Cleveland, 1979), os quais são estimados através do algoritmo backfitting (Hastie et al., 2001).

b) Redes neurais *feedforward*

Um modelo de redes neurais progressivas ou *feedforward* (Haykin, 1999; Fine, 1999), introduzido no capítulo anterior, pode ser interpretado como um caso especial de regressão não-linear de natureza mais complexa, onde os coeficientes de regressão são ditos pesos da rede, as variáveis regressoras são as entradas da rede, a variável resposta é dita saída desejável da rede e a variável resposta ajustada é a saída da rede.

A rede *feedforward* considerada aqui tem $n = 2$ entradas x_1 e x_2 (carbono e nitrogênio defasado), k neurônios na camada intermediária e uma saída y , que é a resposta nitrogênio desejável. A função de ligação utilizada na camada intermediária é a função logística $g(x) = (1 + e^{-x})^{-1}$, e na camada de

saída é utilizada a função identidade, resultando na relação entrada-saída

$$\hat{y}_p = \sum_{j=1}^k W_j g\left(\sum_{i=0}^{n-2} w_{ji} x_{pi}\right) \quad (4.10)$$

onde p (“pattern”) é o índice das observações e w_{ji} e W_j são os pesos da rede.

O processo de estimativação dos pesos w_{ji} e W_j da rede (chamado de treinamento ou aprendizado) é baseado na minimização da função perda ou erro quadrático,

$$E = \frac{1}{2} \sum_{p=1}^N (y_p - \hat{y}_p)^2 \quad (4.11)$$

considerando o método usual de retropropagação de erro (Haykin, 1999), utilizando-se otimização por gradiente através do software MATLAB (MathWorks Inc., 2003). A fórmula quadrática da função erro se deve à hipótese de normalidade para os dados y_p , $p = 1, \dots, N$, conforme justificado ao final da seção 4.2.

O processo de generalização foi dividido em duas partes. Para tal, o conjunto de dados é separado em 3 partes: uma amostra de treinamento, que é utilizada para estimar os pesos, uma amostra de verificação para implementação do “early-stopping” e uma amostra de teste (validação preditiva) para verificação da performance preditiva. O tamanho desta amostra de teste foi definido em 2 casos: os 10 últimos pontos do transect e os 10 primeiros pontos. A amostra intermediária (verificação) também foi considerada com 10 pontos, contiguamente ao longo do transect. Em resumo, as 3 sub-amostras foram definidas com tamanhos 77, 10 e 10, respectivamente, perfazendo o total de 97 observações do transect.

Na primeira etapa, para a escolha do número de neurônios na camada intermediária, foi utilizado o método de “network pruning/network growing” com validação (Sussner, 1999). Foram testados diferentes valores para o número de neurônios (de 1 a 10). As medidas de performance obtidas na amostra de validação são apresentadas na tabela 4.1, e é possível verificar que a escolha ótima é $k = 2$ neurônios na camada intermediária. Depois de escolhido o número de neurônios, a segunda etapa consiste em testar várias inicializações para os pesos, escolhidos aleatoriamente em um intervalo ao redor de zero (Rojas, 1996), e selecionar a rede que tem melhor performance na amostra de validação, com objetivo de evitar sobreajuste no conjunto de treinamento. A técnica utilizada é a do “early-stopping”, descrita na seção 2.3.1.

c) Redes Neurais Recorrentes

Para ampliar as possibilidades de modelagem de estruturas de auto-dependência dos dados (auto-correlação espacial), são utilizados aqui modelos de redes neurais recorrentes. Os modelos utilizados são os mais populares de redes recorrentes para predição de séries temporais: Elman e Jordan (Haykin, 1999).

A exemplo do item anterior, foram consideradas $n = 2$ entradas x_1 e x_2 (carbono e nitrogênio defasado), $k = 2$ neurônios na camada intermediária e uma saída y , que é a resposta nitrogênio desejável.

Nº de neurônios	MAPE	Correlação observados × ajustados
1	0,03754	0,81933
2	0,03379	0,89464
3	0,03809	0,81102
4	0,04399	0,84292
5	0,03881	0,85976
6	0,04065	0,87124
7	0,04072	0,84576
8	0,04245	0,83888
9	0,04508	0,83472
10	0,04566	0,81342

Tabela 4.1: Comparação de modelos de redes neurais

A função de ligação utilizada na camada intermediária é a função logística $g(x) = (1 + e^{-x})^{-1}$, e na camada de saída é utilizada a função identidade.

A rede Elman foi implementada via MATLAB utilizando-se de uma extensão do algoritmo de retropropagação (utilizado no caso das redes *feedforward*) conhecida como retropropagação através do tempo, que é o algoritmo mais utilizado na prática para esse tipo de rede. Esse tipo de rede é o único de rede recorrente disponível no MATLAB para os fins aqui tratados. Para a implementação do “early stopping”, foram utilizados os mesmos critérios adotados para a rede *feedforward*.

A rede Jordan foi implementada via filtro de Kalman estendido de 2^a ordem, em um algoritmo escrito em linguagem MATLAB. Em adição, a rede Elman também foi treinada por esse algoritmo, possibilitando a comparação de resultados entre dois algoritmos distintos. Também foi tentada uma versão do filtro de Kalman estendido de 1^a ordem, com resultados muito pobres (conforme esperado, de acordo com a literatura) e, por isso, não apresentados aqui.

4.3.4 Comparação de modelos

Os modelos foram ajustados em duas versões. Na primeira, os 10 últimos dados foram omitidos para que se fizesse a predição destes valores (tabela 4.2). Na segunda, os 10 primeiros foram omitidos, fazendo-se a predição da mesma maneira (tabela 4.3). As medidas de performance de predição para comparação são o MSE (“Mean Square Error” ou Erro Médio Quadrático), MAE (“Mean Absolute Error” ou Erro Médio Absoluto) e MAPE (“Mean Absolute Percentage Error” ou Erro Médio Absoluto Percentual).

Como o valor do nitrogênio no ponto depende do valor do nitrogênio nos pontos vizinhos, é natural

Modelos de predição		Medidas de performance		
		MSE	MAE	MAPE
Regressão linear escalar	Padrão	0,00388	0,04982	0,04301
	Com erros AR(1)	0,00389	0,04953	0,04279
	Com coeficientes variáveis	0,00288	0,04610	0,03960
Auto-regressão vetorial	VAR padrão	0,00713	0,07993	0,06390
	VAR corrigido	0,00350	0,04791	0,03905
	VAR estrutural	0,00096	0,02691	0,02302
Regressão não-linear (não-paramétrica)	GAM/lowess	0,00361	0,05014	0,04084
	Redes neurais <i>feedforward</i>	0,00313	0,04308	0,03727
	Redes recorrentes (Elman/gradiente)	0,00279	0,04154	0,03599
	Redes recorrentes (Elman/EKF 2ª ordem)	0,00257	0,04032	0,03311
	Redes recorrentes (Jordan/EKF 2ª ordem)	0,00330	0,04540	0,03740
Comb. melhores modelos	Modelo misto	0,00197	0,02958	0,02461

Tabela 4.2: Performance preditiva (10 últimos pontos do transect)

Modelos de predição		Medidas de performance		
		MSE	MAE	MAPE
Regressão linear escalar	Padrão	0,00483	0,05486	0,04665
	Com erros AR(1)	0,00475	0,05408	0,04601
	Com coeficientes variáveis	0,00407	0,05280	0,04589
Auto-regressão vetorial	VAR padrão	0,00713	0,07993	0,06390
	VAR corrigido	0,00423	0,05082	0,04358
	VAR estrutural	0,00314	0,04799	0,04192
Regressão não-linear (não-paramétrica)	GAM/splines	0,00793	0,06650	0,05583
	Redes neurais <i>feedforward</i>	0,00344	0,04536	0,03898
	Redes recorrentes (Elman/gradiente)	0,00213	0,03323	0,02827
	Redes recorrentes (Elman/EKF 2ª ordem)	0,00157	0,03015	0,02614
	Redes recorrentes (Jordan/EKF 2ª ordem)	0,00270	0,04060	0,03580
Comb. melhores modelos	Modelo misto	0,00147	0,03290	0,02863

Tabela 4.3: Performance preditiva (10 primeiros pontos do transect)

considerar como variável regressora não somente o valor do nitrogênio em um ponto vizinho de um dos lados, mas também o valor do outro vizinho. No entanto, o ajuste dos modelos considerando esse fator demonstrou pequena ou nenhuma vantagem em termos de predição. Por isso, os modelos consideram como variável regressora apenas o valor do nitrogênio no ponto anterior, além do carbono no mesmo ponto.

Como apenas o valor do carbono para os últimos (ou primeiros) 10 pontos do transect são observados, foi utilizado, em lugar da variável nitrogênio defasado, o valor predito do nitrogênio no ponto anterior. No caso de modelos não-lineares, isso pode representar um viés nas previsões (o valor esperado da função não-linear não é necessariamente o valor da função aplicada no valor esperado). No caso deste conjunto de dados, como a principal e mais importante variável regressora é o carbono e não o nitrogênio defasado, as previsões poderão conter apenas um pequeno viés que não prejudicará a performance preditiva dos modelos não-lineares.

Analizando-se as tabelas 4.2 e 4.3 conjuntamente, pode ser verificado que entre os modelos sem variável latente, ou seja, entre os modelos de regressão propriamente ditos, o modelo VAR padrão apresenta resultados menos satisfatórios (independentemente da medida de performance considerada), pois, ao contrário dos outros modelos, o VAR utiliza o carbono orgânico defasado como variável regressora exógena, e não o carbono orgânico no ponto correspondente, que tem uma relação linear mais forte com o nitrogênio total. Já o modelo VAR corrigido apresenta os melhores resultados entre os modelos de regressão (valores mínimos de MSE, MAE e MAPE tanto na tabela 4.2 quanto na tabela 4.3), tendo o preditor mais apropriado, ou seja, o carbono no mesmo ponto.

Em termos absolutos, a melhor predição para os últimos pontos do transect é obtida com o modelo de espaço de estado padrão (VAR estrutural), independente da medida de performance considerada, seguido pela rede Elman. Já para os 10 primeiros pontos do transect, a melhor predição é obtida através de rede neural recorrente, seguido do VAR estrutural (espaço de estado padrão). O modelo misto apresentado nas tabelas tem suas previsões calculadas através da média aritmética das previsões de cada modelo um desses dois modelos (rede Elman e VAR estrutural), conseguindo melhores resultados preditivos do que pelo menos um dos modelos concorrentes em ambos os extremos do transect.

Vale observar que na tabela 4.3 acima só foi considerada uma única opção do modelo de regressão aditiva, qual seja o GAM/splines, uma vez que o GAM/lowess tem restrição quanto ao valor do regressor para predição (deve estar dentro do intervalo dos dados utilizados para o ajuste), o que não é atendido para esses particulares dados.

A principal conclusão dessa primeira análise é que as características dos dados sugerem estruturas tipo redes recorrentes (ao invés de *feedforward*) ou modelo de espaço de estado (VAR estrutural), dependendo do experimento considerado.

4.4 Modelo proposto: redes neurais recorrentes com pesos variaveis ao longo do espaço

O objetivo do modelo aqui proposto é unir a capacidade não-linear das redes neurais com a possibilidade de captação da heterogeneidade do solo pelo modelo de espaço de estado.

A primeira motivação para a combinação dos modelos foi obtida através do que chamamos de modelo misto, que combina as previsões externas dos modelos de redes neurais recorrentes e espaço de estado padrão. As novas previsões são obtidas através do cálculo da média aritmética das previsões de cada modelo em cada ponto. Os resultados mostram que esse preditor misto tem resultados satisfatórios, conseguindo melhores resultados preditivos do que pelo menos um dos modelos concorrentes em ambos os extremos do transect. Os resultados são apresentados nas tabelas 4.2 e 4.3.

No modelo proposto nesta seção, a combinação dos modelos é feita de modo a representar a estrutura da rede em forma de espaço de estado, com os pesos da rede representando as variáveis de estado a serem estimadas,

$$y(i) = f(\mathbf{w}(i), \mathbf{x}(i)) + v(i), \quad v(i) \sim N(0, V) \quad (4.12)$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mathbf{u}(i), \quad \mathbf{u}(i) \sim N(0, \Sigma) \quad (4.13)$$

onde $y(i)$ é o valor observado (medido) do nitrogênio total no i -ésimo ponto do transect, $f(\cdot)$ é a função de mapeamento da rede, $\mathbf{x}(i)$ é o vetor composto pelas variáveis exógenas (carbono orgânico e nitrogênio total defasado) mais as realimentações e $\mathbf{w}(i)$ é o vetor composto pelos pesos da rede, o qual evolui segundo um passeio aleatório vetorial (equação 4.13). A função da rede f é dada por

$$f(\mathbf{w}(i), \mathbf{x}(i)) = \sum_{j=0}^k W_j(i) g\left(\sum_{l=0}^n w_{jl}(i)x_l(i)\right) \quad (4.14)$$

onde g é uma função de ativação sigmoidal do tipo $g(x) = (1 + e^{-x})^{-1}$, n é o numero de entradas, k é o número de neurônios na camada intermediária, $w_{jl}(i)$ e $W_j(i)$ são os pesos da rede no ponto i do espaço que compõem o vetor $\mathbf{w}(i)$ e $x_l(i)$ são as entradas da rede que compõem o vetor $x(i)$ (carbono, nitrogênio defasado e as realimentações). As perturbações aleatórias $v(i)$ e $\mathbf{u}(i)$ são por suposição independentes e normalmente distribuídas com variâncias constantes.

Além do já citado modelo misto, uma segunda motivação é que o modelo permite incorporar alguma eventual heterocedasticidade do processo gerador dos dados, uma vez que a variância do modelo depende da variância do processo $\mathbf{w}(i)$, que segue um passeio aleatório, e portanto depende do índice i . Esse fenômeno está presente em diversos conjuntos de dados, inclusive nos dados do nitrogênio analisados aqui (ver figura 4.2(e)). Além do mais, tal formulação está também motivada dentro do contexto da literatura existente presente sobre modelos não-lineares de parâmetros variáveis formulados em espaço de estado (Pole, West & Harrison, 1994; Barbosa & Sadownik, 1999; Sadownik & Barbosa, 1999).

Para a estimção dos pesos, é proposto um algoritmo tipo filtro de Kalman estendido de 2^a ordem (envolvendo Jacobiano e Hessiana da função f) em versão Bayesiana (priori normal multivariada-gama inversa), que inclui a estimção da variânci de $v(i)$, ao passo que as variâncias de $u(i)$ são especificadas através de fatores de desconto.

4.4.1 Algoritmo: filtro de Kalman estendido de 2^a ordem

Aproximação de segunda ordem da função de mapeamento da rede (expansão em série de Taylor):

$$y(t) \approx f(\mathbf{a}(t)) + J(\mathbf{a}(t))(\mathbf{w}(t) - \mathbf{a}(t)) + \frac{1}{2} (\mathbf{w}(t) - \mathbf{a}(t))' H(\mathbf{a}(t)) (\mathbf{w}(t) - \mathbf{a}(t)) \quad (4.15)$$

onde

$$J(\mathbf{a}(t)) = \left\{ \frac{\partial f(\mathbf{w}(t))}{\partial \mathbf{w}(t)} \right\}_{\mathbf{w}(t)=\mathbf{a}(t)} \quad (4.16)$$

$$H(\mathbf{a}(t)) = \left\{ \frac{\partial^2 f(\mathbf{w}(t))}{\partial \mathbf{w}(t) \partial \mathbf{w}(t)'} \right\}_{\mathbf{w}(t)=\mathbf{a}(t)} \quad (4.17)$$

No caso, a função f representa uma rede neural recorrente tipo Elman, mas o algoritmo pode ser utilizado também quando a f represente uma rede *feedforward* ou alguma outra rede recorrente, ou mesmo qualquer topologia de rede.

Algoritmo de estimção dos pesos $\mathbf{w}(t)$:

Inicialização: $\mathbf{w}(0), C(0), s(0), n(0), \delta$ (fator de desconto) dados.

Para $t = 1, \dots, N$

$$\mathbf{a}(t) = \mathbf{w}(t-1) \quad (4.18)$$

$$R(t) = \frac{C(t-1)}{\delta} \quad (4.19)$$

$$\mathbf{w}(t) = \mathbf{a}(t) + A(t) e(t) \quad (4.20)$$

$$C(t) = (R(t) - A(t) Q(t) A'(t)) \frac{s(t)}{s(t-1)} \quad (4.21)$$

$$s(t) = s(t-1) \frac{\left(n(t-1) + \frac{e^2(t)}{Q(t)} \right)}{n(t)} \quad (4.22)$$

$$n(t) = n(t-1) + 1 \quad (4.23)$$

onde

$$e(t) = y(t) - f_1(t) \quad (4.24)$$

$$A(t) = R(t) J(\mathbf{a}(t)) Q^{-1}(t) \quad (4.25)$$

$$Q(t) = J'(\mathbf{a}(t)) R(t) J(\mathbf{a}(t)) + \frac{1}{2} \text{Tr} \{ H(\mathbf{a}(t)) R(t) \}^2 + s(t-1) \quad (4.26)$$

$$f_1(t) = f(\mathbf{a}(t)) + \frac{1}{2} \text{Tr} \{ J(\mathbf{a}(t)) R(t) \} \quad (4.27)$$

Para o programa em linguagem MATLAB, ver apêndice A.2.

Modelos de predição		Medidas de performance		
		MSE	MAE	MAPE
Modelos padrão com estrutura latente	Redes neurais recorrentes (Elman/gradiente)	0,00279	0,04154	0,03599
	Espaço de estado linear (VAR estrutural)	0,00096	0,02691	0,02302
Modelos propostos e seus casos particulares	Rede Elman com pesos variáveis (EKF 2 ^a ordem, fator = 0,96)	0,00081	0,02334	0,02005
	Rede Elman (EKF 2 ^a ordem, W = 0)	0,00257	0,04032	0,03311
	Rede Elman com pesos variáveis (EKF 1 ^a ordem, fator = 0,99)	0,01202	0,11063	0,08316
	Rede Elman (EKF 1 ^a ordem, W = 0)	0,01023	0,09756	0,07684

Tabela 4.4: Performance preditiva: 10 últimos pontos do transect

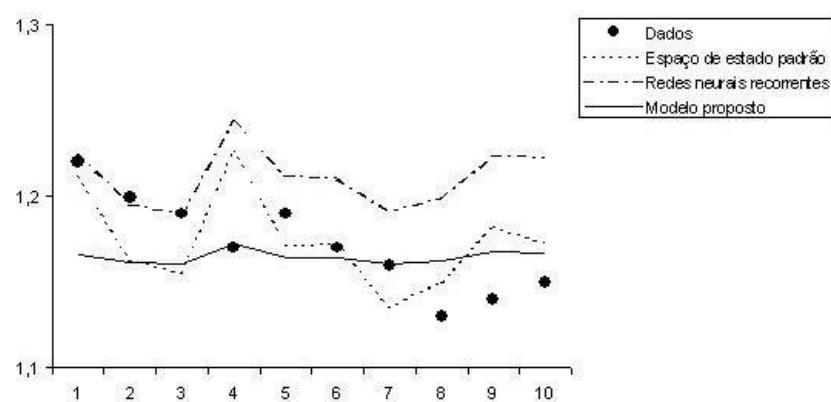


Figura 4.4: Comparação de previsões: 10 últimos pontos do transect

Modelos de predição		Medidas de performance		
		MSE	MAE	MAPE
Modelos padrão com estrutura latente	Redes neurais recorrentes (Elman/gradiente)	0,00213	0,03323	0,02827
	Espaço de estado linear (VAR estrutural)	0,00314	0,04799	0,04192
Modelos propostos e seus casos particulares	Rede Elman com pesos variáveis (EKF 2 ^a ordem, fator = 0,96)	0,00141	0,03002	0,02599
	Rede Elman (EKF 2 ^a ordem, W = 0)	0,00157	0,03015	0,02614
	Rede Elman com pesos variáveis (EKF 1 ^a ordem, fator = 0,99)	0,00311	0,04377	0,03755
	Rede Elman (EKF 1 ^a ordem, W = 0)	0,00385	0,05033	0,04279

Tabela 4.5: Performance preditiva: 10 primeiros pontos do transect

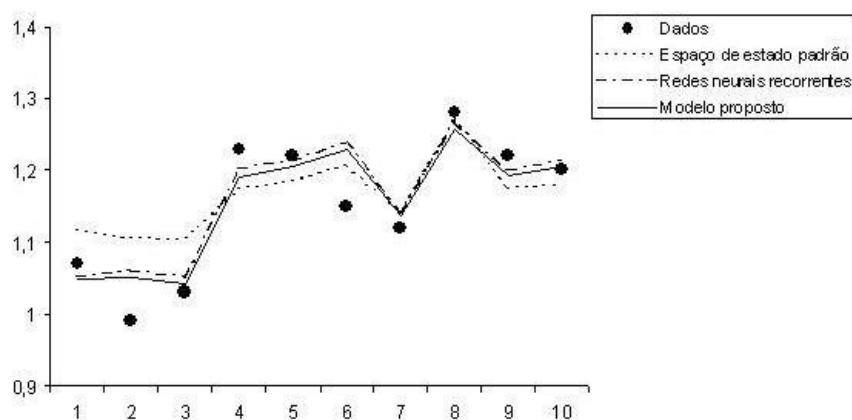


Figura 4.5: Comparação de previsões: 10 primeiros pontos do transect

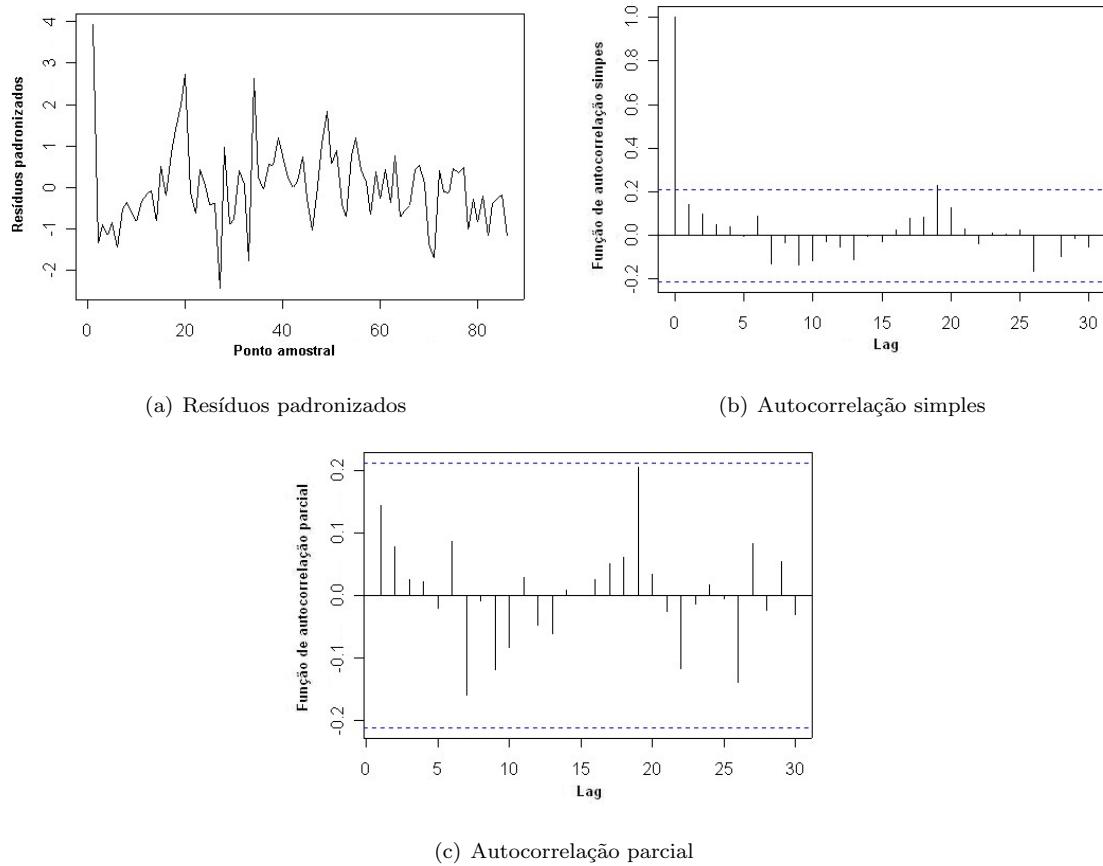


Figura 4.6: Gráficos de resíduos: modelo de redes neurais de pesos variáveis (10 primeiros pontos)

4.5 Resultados

As tabelas 4.4 e 4.5 comparam os melhores modelos já previamente analisados (redes neurais recorrentes Elman treinadas via gradiente e modelo de espaço de estado linear), além de redes neurais recorrentes Elman com pesos fixos treinadas via filtro de Kalman estendido de 2^a ordem (fator de desconto = 0) e os modelos propostos de redes neurais recorrentes Elman com pesos variáveis no espaço. Também são apresentados resultados dos modelos de redes neurais recorrentes Elman treinados por filtro de Kalman estendido de 1^a ordem.

Em termos de performance preditiva, o modelo proposto (rede neural recorrente com pesos variáveis) apresenta melhores resultados que cada um dos modelos isoladamente (rede neural recorrente padrão via métodos gradientes e via filtro de Kalman e espaço de estado linear), como mostrado nas tabelas 4.4 e 4.5.

A performance preditiva obtida pelo algoritmo baseado em filtro de Kalman de 2^a ordem é显著mente melhor do que o algoritmo de 1^a ordem, o que mostra que o algoritmo de 1^a ordem não

é capaz de lidar adequadamente com a não-linearidade presente no problema, sendo a extensão de 2^a ordem necessária para o treinamento de um modelo de rede neural.

As figuras 4.4 e 4.5 mostram os gráficos das predições dos modelos de espaço de estado padrão, redes neurais recorrentes e o novo modelo proposto.

O objetivo alcançado foi construir um novo modelo de regressão capaz de incorporar simultaneamente os três fatores de complexidade do problema (dependência espacial dos dados, não homogeneidade do solo e não-linearidade da relação), com correspondente algoritmo de estimação, superando assim as limitações dos modelos até então em uso para esse fim.

A estrutura desse modelo proposto para função de pedotransferência em transect (rede neural recorrente com pesos variáveis ao longo do espaço), além de apresentar características referentes aos três fatores desejáveis, os resultados mostraram que este modelo foi o de melhor performance preditiva (segundo três diferentes tipos de medidas) entre os diversos modelos considerados para os dados em questão, aí incluindo-se diferentes tipos de modelos lineares e não-lineares.

4.6 Intervalos de predição

As figuras 4.7 e 4.8 apresentam os intervalos de predição de 95% para os primeiros 10 pontos do transect para os modelos de redes neurais (tanto pesos fixos quanto variáveis). Os intervalos foram obtidos através de dois métodos: analítico e reamostragem dos resíduos (bootstrap), conforme os procedimentos detalhados na seção 2.7.

Foram consideradas 1000 amostras bootstrap reamostradas com reposição da série original de resíduos, cujas características (gráfico e funções de autocorrelação simples e parcial) são mostradas na figura 4.6. Esses intervalos, calculados diretamente a partir da variância bootstrap, são aproximadamente equivalente aos intervalos entre os percentis 2,5 e 97,5 da distribuição das 1000 amostras bootstrap.

Note que embora estes intervalos de predição correspondam a predição de origem fixa, suas amplitudes não crescem ao longo dos 10 pontos de predição, uma vez que o valor do carbono (que é o principal preditor) é observado ao longo desses pontos. Também é interessante notar que os intervalos analíticos são um pouco mais largos que os intervalos numéricos, com capacidade de cobrir mais pontos. Isso ocorre porque o método numérico se trata de uma aproximação, que em alguns casos pode subestimar ou superestimar os intervalos. Neste caso, a diferença observada é pequena, o que mostra que o método numérico não é muito prejudicado.

Esses gráficos, cujos correspondentes valores numéricos são apresentados nas tabelas 4.6 e 4.7, mostram que, mesmo encerrando-se as medições do nitrogênio no ponto 87 do transect e utilizando-se apenas as medições do carbono nesses 10 pontos restantes, o modelo fornece predições suficientemente precisas acerca dos valores do nitrogênio.

Ponto	Dado real	Predição	Intervalo analítico	Intervalo numérico
1	1,07	1,0527	(0,9963; 1,1091)	(1,0047; 1,1007)
2	0,99	1,0607	(1,0076; 1,1138)	(1,0117; 1,1097)
3	1,03	1,0518	(0,9964; 1,1072)	(1,0048; 1,0988)
4	1,23	1,2023	(1,1552; 1,2494)	(1,1553; 1,2493)
5	1,22	1,2141	(1,1639; 1,2643)	(1,1676; 1,2606)
6	1,15	1,2388	(1,202; 1,2756)	(1,1925; 1,2851)
7	1,12	1,1417	(1,0829; 1,2005)	(1,0937; 1,1897)
8	1,28	1,2669	(1,1942; 1,3396)	(1,2187; 1,3151)
9	1,22	1,1989	(1,1546; 1,2432)	(1,1513; 1,2465)
10	1,2	1,2135	(1,1719; 1,2551)	(1,1659; 1,2611)

Tabela 4.6: Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos fixos

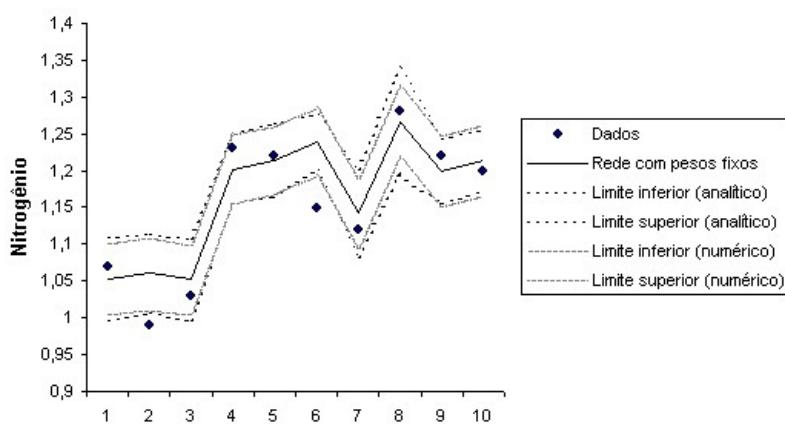


Figura 4.7: Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos fixos

Ponto	Dado real	Predição	Intervalo analítico	Intervalo numérico
1	1,07	1,0484	(0,992; 1,1048)	(0,9984; 1,0984)
2	0,99	1,0506	(0,9943; 1,1069)	(0,9977; 1,1035)
3	1,03	1,0416	(0,9836; 1,0996)	(0,9897; 1,0935)
4	1,23	1,1901	(1,1409; 1,2393)	(1,1386; 1,2416)
5	1,22	1,2054	(1,1529; 1,2579)	(1,1578; 1,253)
6	1,15	1,2305	(1,1925; 1,2685)	(1,1825; 1,2785)
7	1,12	1,1357	(1,0741; 1,1973)	(1,0863; 1,1851)
8	1,28	1,2572	(1,1807; 1,3337)	(1,2076; 1,3068)
9	1,22	1,1924	(1,1462; 1,2386)	(1,1428; 1,242)
10	1,2	1,2054	(1,1621; 1,2487)	(1,1558; 1,255)

Tabela 4.7: Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos variáveis

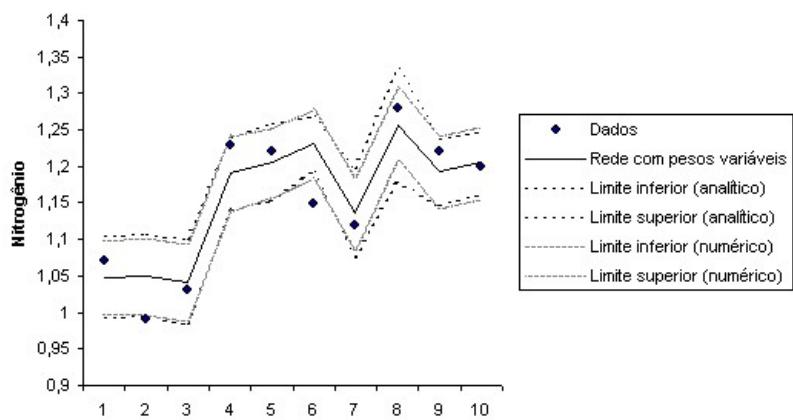


Figura 4.8: Intervalos de predição de 95% (10 primeiros pontos): Rede Elman de pesos variáveis

Rede Elman	MSE	MAE	MAPE
Pesos fixos (gradiente)	0,00185	0,03342	0,02913
Pesos fixos (EKF, fator=1)	0,00259	0,03595	0,03024
Pesos variáveis (EKF, fator=0,95)	0,00375	0,05154	0,04417
Pesos variáveis (EKF, fator=0,96)	0,00335	0,04654	0,04120
Pesos variáveis (EKF, fator=0,97)	0,00289	0,04532	0,03895
Pesos variáveis (EKF, fator=0,98)	0,00283	0,04167	0,03658
Pesos variáveis (EKF, fator=0,99)	0,00288	0,04161	0,03524

Tabela 4.8: Resultados de simulação: série gerada com rede de pesos fixos

4.7 Um estudo de simulação

Nesta seção é feito um estudo de simulação, baseado nos dados espaciais analisados neste capítulo, para avaliar a performance dos algoritmos utilizados para treinamento de redes.

Duas séries distintas são geradas através de uma rede neural recorrente tipo Elman com dois neurônios na camada intermediária. A primeira, utilizando como entradas as variáveis carbono e nitrogênio defasado, tem como modelo gerador uma rede com pesos fixos. O modelo utilizado é o seguinte:

$$y(i) = f(\mathbf{w}(i), \mathbf{x}(i)) + v(i), \quad v(i) \sim N(0, \sigma^2) \quad (4.28)$$

$$\mathbf{w}(i) = \mathbf{w}(i - 1) \quad (4.29)$$

onde $y(i)$ é o valor simulado no i -ésimo ponto, $f(\cdot)$ é a função de mapeamento da rede (tipo Elman com dois neurônios na camada intermediária), $\mathbf{x}(i)$ é o vetor composto pelas variáveis de entrada (carbono orgânico e nitrogênio total defasado) mais as realimentações e $\mathbf{w}(i)$ é o vetor composto pelos pesos da rede. O valor utilizado para σ na simulação é 0,025.

Modelos de redes neurais são ajustados utilizando-se dois algoritmos: retropropagação através do tempo (baseado em gradiente) e filtro de Kalman estendido de 2ª ordem. No caso deste último, é feita uma variação no valor do fator de desconto (0,95 a 1). É esperado que os melhores resultados sejam obtidos com o algoritmo baseado em gradiente e com o filtro de Kalman com fator de desconto 1 (ambos modelam uma rede com pesos fixos). A tabela 4.8 mostra os resultados preditivos correspondentes aos 10 primeiros pontos.

Os resultados obtidos são esperados. A estimativa de σ obtida através do algoritmo do filtro de Kalman é $\hat{\sigma} = 0,050$.

A segunda série simulada utiliza as mesmas variáveis de entrada e a mesma topologia de rede, com

Rede Elman	MSE	MAE	MAPE
Pesos fixos (gradiente)	0,00524	0,06539	0,04781
Pesos fixos (EKF, fator=1)	0,00543	0,06344	0,04731
Pesos variáveis (EKF, fator=0,95)	0,00455	0,05960	0,04368
Pesos variáveis (EKF, fator=0,96)	0,00441	0,05209	0,03883
Pesos variáveis (EKF, fator=0,97)	0,00360	0,04750	0,03422
Pesos variáveis (EKF, fator=0,98)	0,00370	0,04682	0,03339
Pesos variáveis (EKF, fator=0,99)	0,00435	0,05712	0,04126

Tabela 4.9: Resultados de simulação: série gerada com rede de pesos variáveis

a diferença que os pesos agora são variáveis. O modelo é dado por

$$y(i) = f(\mathbf{w}(i), \mathbf{x}(i)) + v(i), \quad v(i) \sim N(0, \sigma^2) \quad (4.30)$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mathbf{u}(i), \quad \mathbf{u}(i) \sim N(0, \lambda^2 I) \quad (4.31)$$

onde $y(i)$ é o valor simulado no i -ésimo ponto, $f(\cdot)$ é a função de mapeamento da rede (tipo Elman com dois neurônios na camada intermediária), $\mathbf{x}(i)$ é o vetor composto pelas variáveis de entrada (carbono orgânico e nitrogênio total defasado) mais as realimentações e $\mathbf{w}(i)$ é o vetor composto pelos pesos da rede. O valor utilizado para σ na simulação é 0,025 e para λ é 0,008.

Os mesmos modelos da tabela 4.8 são ajustados, e desta vez espera-se que os melhores resultados sejam obtidos pelo filtro de Kalman com fator de desconto diferente de 1 (rede com pesos variáveis). A tabela 4.9 mostra os resultados preditivos correspondentes aos 10 primeiros pontos.

Os resultados obtidos são esperados, com as melhores performances preditivas obtidas com fatores de desconto 0,97 e 0,98. A estimativa de σ obtida através do algoritmo do filtro de Kalman é $\hat{\sigma} = 0,0469$.

Em ambos os casos de simulação (pesos fixos e pesos variáveis), o algoritmo do filtro de Kalman superestima a variância do erro observacional. Como já observado na seção 2.7, a forte não-linearidade presente em modelos de redes neurais prejudica os cálculos que dependem de momentos de 2^a ordem, ao contrário dos cálculos que dependem de momentos de 1^a ordem. Isso explica porque as estimativas pontuais das previsões apresentam resultados satisfatórios, ao contrário das estimativas das variâncias.

Capítulo 5

Análise de séries econômicas de inflação

Neste capítulo, o objetivo é analisar dados de séries temporais econômicas tanto de memória curta quanto longa com vistas a se fazer previsões de valores futuros a diferentes horizontes. A análise é iniciada com a série de taxas de inflação dos Estados Unidos e do Reino Unido, seguida pelas séries de outros países desenvolvidos.

Os modelos ARFIMA (“Auto Regressive Fractionally Integrated Moving Average”) são utilizados como referência. Também são utilizados modelos de redes neurais *feedforward* e recorrentes (tipos Elman e Jordan), otimizadas com respeito ao número de neurônios e ao número de entradas (regressores). A partir dessas redes, são também ajustadas redes *feedforward* e recorrentes não padrão, com pesos variáveis ao longo do tempo. Todos esses modelos são comparados com respeito as suas correspondentes performances de predição a diferentes horizontes (1, 3, 6 e 12 meses).

5.1 Introdução

Dados de inflação de países industrializados já foram estudados e analisados por uma série de autores. A partir da década de 80 modelos de memória longa têm sido usados de maneira crescente, se tornando uma ferramenta muito utilizada para dados econômicos (Baillie, 1996). Bos, Franses e Ooms (2002) mostraram a utilidade de uma modelagem tipo ARFIMA para a série de inflação dos Estados Unidos. Baillie, Chung e Tieslau (1996) encontraram forte evidência de memória longa em dados de inflação de 9 dos 10 países analisados. Hassler e Wolters (1995) analisaram dados de inflação de 5 países industrializados e rejeitaram as hipóteses de raiz unitária e de estacionaridade, concluindo que um modelo com ordem de integração fracionária é um modelo plausível.

No entanto, a investigação de não-linearidades em dados de séries temporais é importante na teoria macroeconómica (Granger & Teräsvirta, 1993; McNelis & McAdam, 2004; Nakamura, 2005). Modelos do tipo ARFIMA não são capazes de lidar adequadamente com estruturas não-lineares. Se o processo gerador da inflação não for linear, tais modelos poderão apresentar resultados não muito satisfatórios em termos de predição.

Com vistas a suprir essa possível deficiência dos modelos lineares, modelos não-lineares baseados em redes neurais (redes *feedforward*) têm sido utilizados como uma ferramenta alternativa para dados econômicos. Esse crescente uso de redes neurais se deu apenas nos últimos anos, quando se percebeu que redes podem ser utilizadas em análises macro e microeconómicas, com grande potencial de melhorar a qualidade de predição econômica (Tal e Nazareth, 1995). Nakamura (2005) ajustou redes neurais para predição de dados de inflação dos Estados Unidos, assim como Binner e Gazely fizeram para dados da Itália (1999) e Reino Unido (2004). McNelis e McAdam (2004) utilizaram redes neurais para predição de inflação em países europeus, além de Estados Unidos e Japão.

Redes neurais permitem, por exemplo, lidar com mais facilidade com outliers no sentido que o modelo não é afetado de maneira global como um modelo de regressão linear, e sim apenas de maneira local (o ajuste é prejudicado apenas nas imediações do ponto aberrante), ou mais geralmente não desprezar não-linearidades por ventura presentes no processo gerador dos dados. Assim, um modelo auto-regressivo (linear) para séries temporais pode ser estendido, via redes *feedforward*, a uma estrutura auto-regressiva não-linear, com possíveis vantagens preditivas, como em Chen, Racine e Swanson (2001), Joeckes (2002) e Joeckes e Barbosa (2003), entre outros. Entretanto, se a série temporal apresentar algum componente MA (médias móveis) ou ainda, apresentar estrutura de memória longa, é natural que algum tipo de retroalimentação deva ser introduzido na rede multicamada, tornando-a uma rede recorrente. Redes neurais recorrentes não são tão difundidas na literatura econômica ou estatística como as redes *feedforward*, e podem apresentar resultados melhores em termos de predição do que as redes *feedforward*, pelas razões já citadas.

Neste capítulo, séries de inflação dos países do G7 (Estados Unidos, Canadá, Reino Unido, Alemanha, França, Itália e Japão) são analisadas, confrontando modelos ARFIMA, redes neurais *feedforward* e redes neurais recorrentes (tipos Elman e Jordan), tanto com pesos fixos como variáveis, em termos da capacidade preditiva de cada um deles para diferentes horizontes de predição. A primeira série a ser analisada é a dos Estados Unidos, a mais estudada na literatura econométrica. Na seção seguinte, o mesmo é feito para a série do Reino Unido, seguido das séries restantes de memória longa (Canadá, França e Itália) e depois as séries de memória curta (Alemanha e Japão).

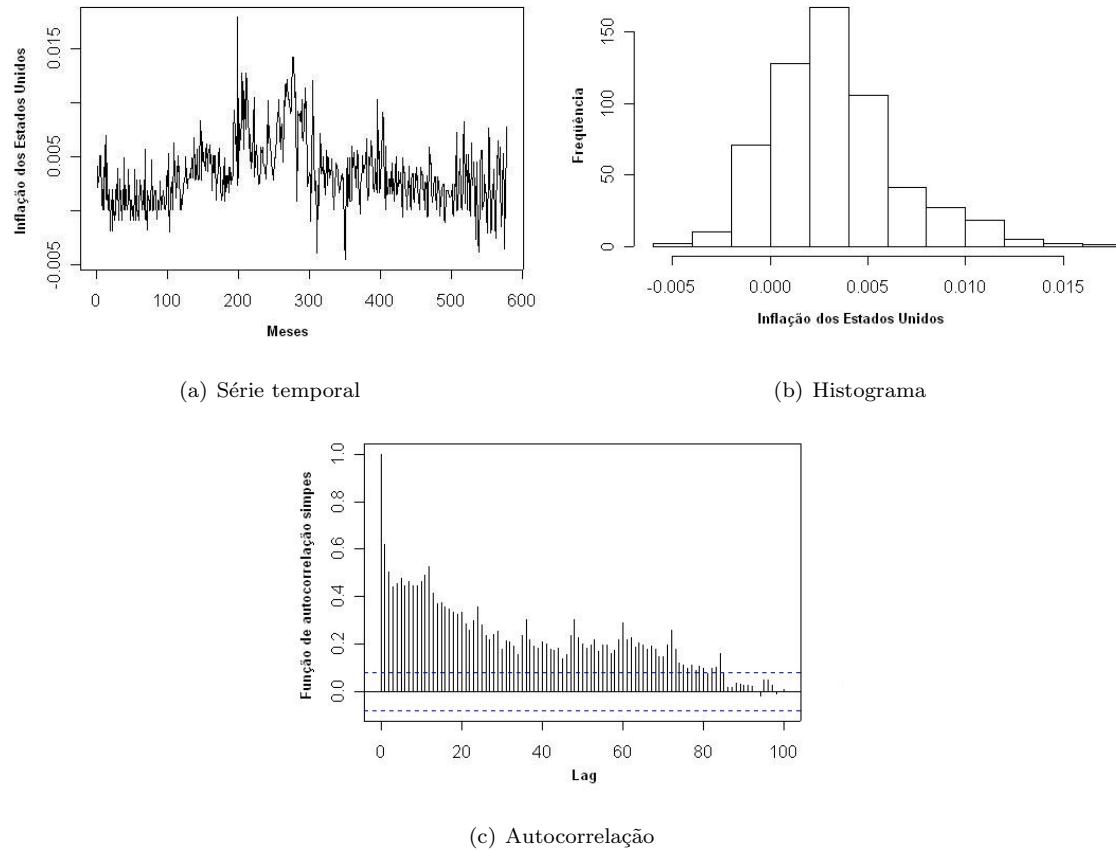


Figura 5.1: Gráficos da série de inflação dos Estados Unidos

5.2 Série de inflação dos Estados Unidos

Esta seção tem como objetivo uma análise comparativa de performance de predição entre modelos de séries temporais (ARFIMA) e modelos de redes neurais para a série de inflação dos Estados Unidos.

A inflação mensal dos Estados Unidos é definida como a primeira diferença do logaritmo da série de CPI (“Consumer Price Index”, ou Índice de Preço ao Consumidor) do país (Baillie, 1996). A série de CPI é acumulativa (índice de base fixa). A primeira diferença do logaritmo, que nada mais é que o logaritmo do quociente (índice de base móvel), tem por objetivo um índice que explique a variação de preços de uma forma aditiva e não acumulativa. Os dados foram retirados do “International Financial Statistics” (através do site <http://ifs.apdi.net/imf/logon.aspx>), tendo início em janeiro de 1957 e fim em março de 2005.

5.2.1 Análise descritiva

A figura 5.1(a) mostra a série de inflação dos Estados Unidos e sugere uma não-estacionaridade em sua média. A figura 5.1(c) mostra autocorrelações significativas que persistem por dezenas de lags, o que dá o caráter de memória longa à série. A figura 5.1(b) mostra que a série tem distribuição pouco assimétrica, ou seja, não muito afastada da normal.

5.2.2 Ajuste de modelos

Nesta seção serão ajustados os modelos. As últimas 12 observações da série (abril de 2004 a março de 2005) não foram consideradas para o ajuste, pois serão utilizadas para comparação preditiva.

Modelo ARFIMA (ordem de integração fracionária)

Nesta seção, será ajustado um modelo ARFIMA para os dados de inflação dos Estados Unidos. O primeiro passo é encontrar o estimador \hat{d} para a ordem de integração fracionária d . Utilizando o método de máxima verossimilhança através do software S-PLUS, o valor encontrado é de $\hat{d} = 0,4031212$. Com este valor, a série é filtrada através do filtro $(1 - B)^{\hat{d}}$ para que se obtenha um processo com memória curta e, assim, se possa efetuar uma análise padrão. Na verdade, foi ajustado o modelo ARFIMA(0, d , 0) com o intuito de se obter um valor numérico para o parâmetro fracionário. Para mais detalhes sobre o método de estimação, ver por exemplo Shumway & Stoffer (2000), p. 166-171, ou Morettin & Toloi (2004), p. 468-469.

Os gráficos de autocorrelação não apresentaram valores significativos nos lags de baixa ordem (figura 5.2). Entretanto, houve indícios de sazonalidade, com autocorrelações significantes em lags múltiplos de 12. As correlações sazonais não decaem exponencialmente, o que sugere uma raiz unitária sazonal. Através do software SAS (macro “DFTEST”), o teste Dickey-Fuller (Wooldridge, 2003) apresenta p-valor = 0,3424, não havendo evidência portanto para rejeitar a hipótese nula de que há raiz unitária sazonal.

Uma diferenciação sazonal de ordem 12 foi tomada. Após isso, como pode ser visto pela figura 5.3, a estrutura observada (uma autocorrelação sazonal significante e autocorrelações parciais sazonais decaendo rapidamente) sugeriam uma componente MA(1) sazonal de período 12. Entretanto, podia-se observar também autocorrelações significativas nos primeiros lags, o que também sugeriria componentes AR ou MA. Como, a princípio, a estrutura de autocorrelação não apresentava nenhum padrão, uma série de modelos ARMA foram ajustados para a série diferenciada fracionalmente, variando o número de parâmetros AR e MA. No entanto, o único modelo que apresentou resultados satisfatórios em relação aos resíduos do ajuste não continha componentes MA e AR simples, ou seja, o modelo ARIMA(0,0,0) \times (0,1,1)₁₂, como mostram os gráficos.

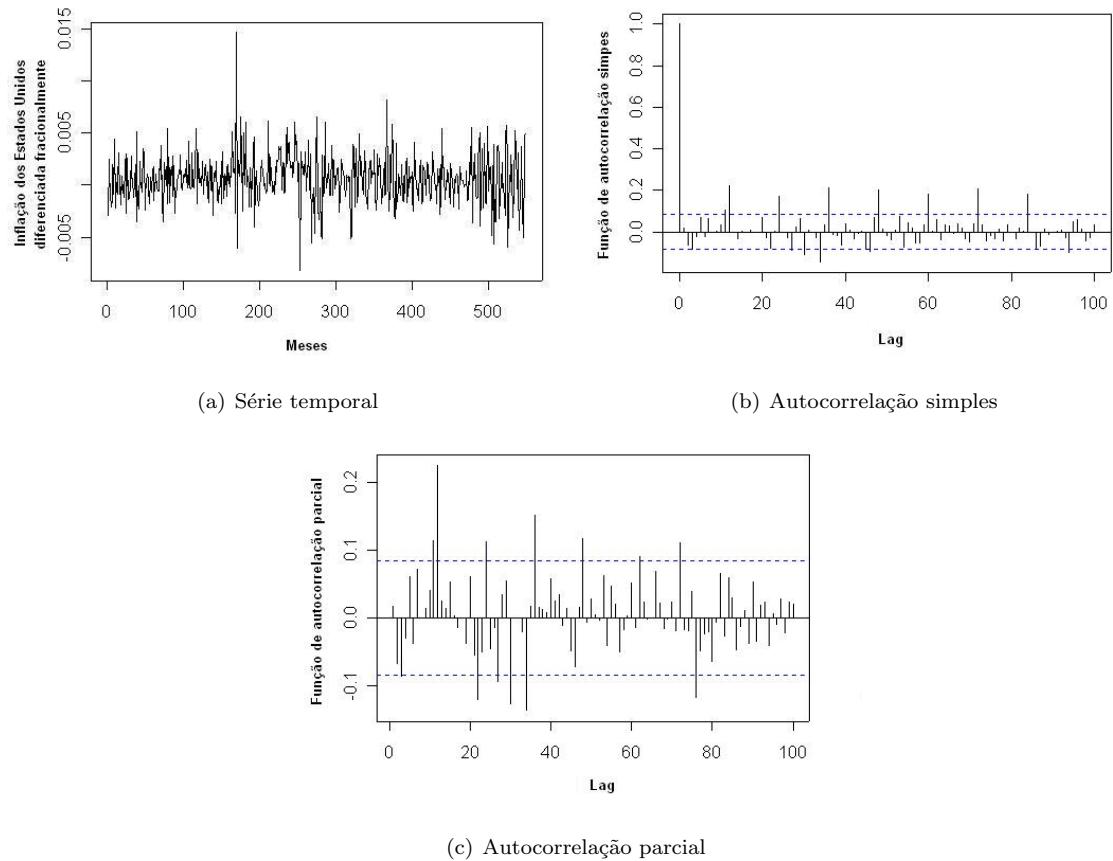


Figura 5.2: Gráficos da série de inflação dos Estados Unidos diferenciada fracionalmente

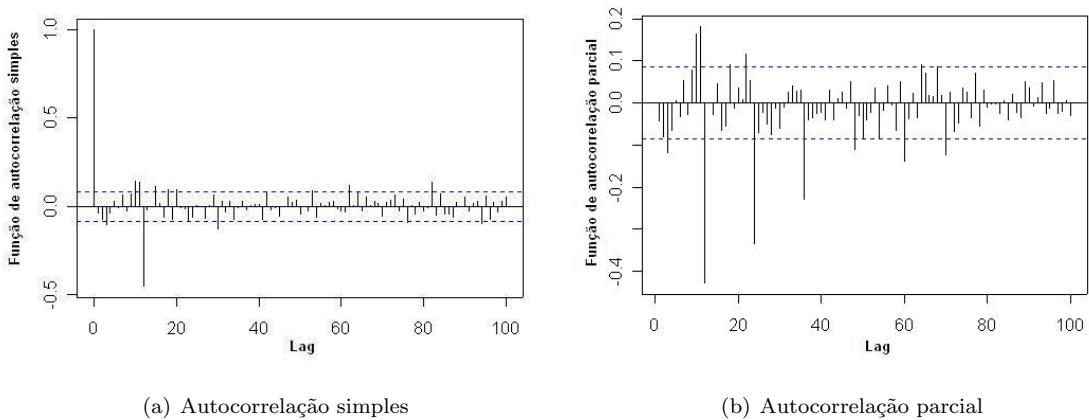


Figura 5.3: Gráficos da série de inflação dos Estados Unidos diferenciada fracionalmente e sazonalmente

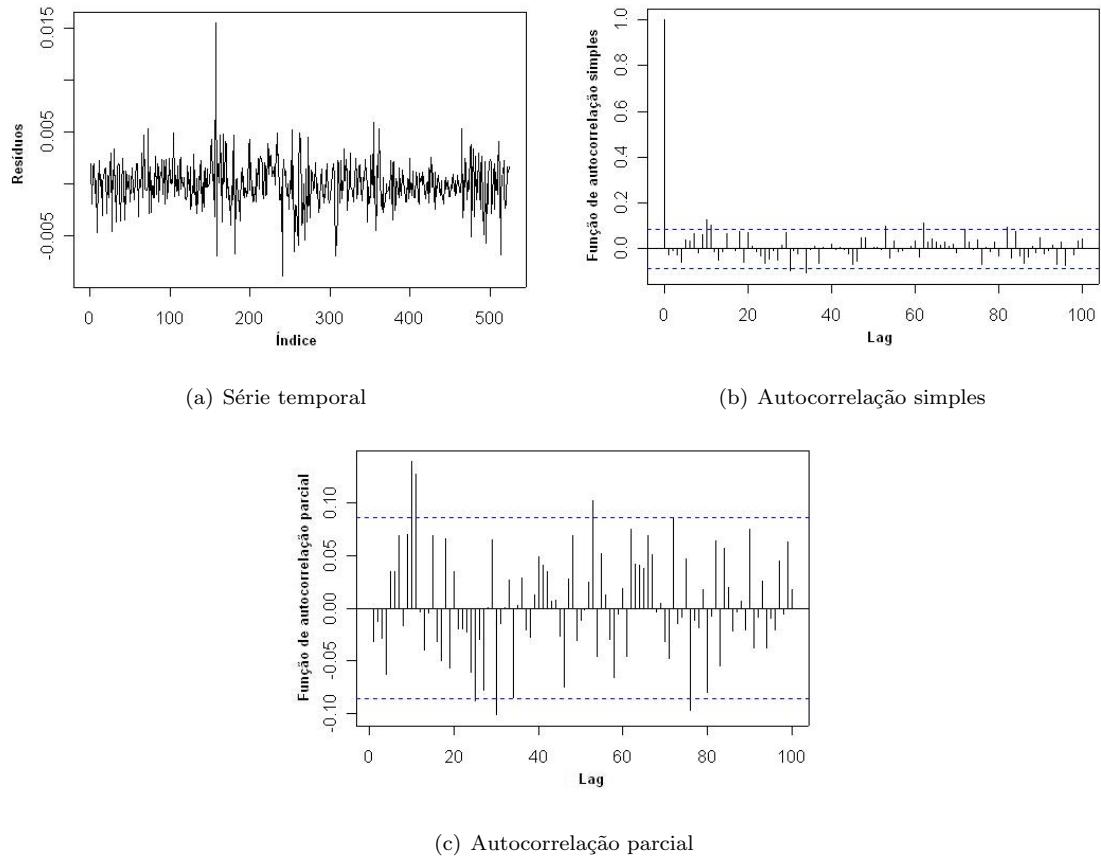


Figura 5.4: Resíduos do ajuste do modelo ARFIMA $(0, d, 0) \times (0, 1, 1)_{12}$ para a série dos Estados Unidos

Portanto, o modelo escolhido foi um ARFIMA $(0; 0, 4031212; 0) \times (0; 1; 1)_{12}$:

$$\hat{\Theta}_1 = 0,8169 \pm 0,0259 \quad (5.1)$$

$$\hat{\sigma}^2 = 4,86 \times 10^{-6} \quad (5.2)$$

Esses valores numéricos encontrados foram obtidos pelo método de máxima verossimilhança via softwares S-PLUS e Ox, com resultados praticamente coincidentes.

Modelo final com parâmetros estimados:

$$(1 - B^{12})(1 - B)^{0,4031212}x(t) = (1 + 0,8169B^{12})a(t) \quad (5.3)$$

$$a(t) \sim N(0; 4,86 \times 10^{-6}) \quad (5.4)$$

Um modelo do tipo ARFIMAX (modelo ARFIMA com variáveis explanatórias exógenas) também foi considerado nesta análise. Bos, Franses e Ooms (2001) utilizaram variáveis como desemprego e taxa de juros como *leading indicators*, ou seja, valores presentes dessas variáveis poderiam ajudar a modelar valores futuros da inflação. Eles se utilizaram da série de inflação “core”, ou seja, uma série de inflação

sem considerar alguns itens de consumo, no caso alimentação e energia. No nosso caso, a adição de tais variáveis explanatórias no modelo representou uma melhora muito pequena em termos de predição, observando-se alguma melhora em apenas previsões de curto prazo (1 mês). Por isso, tais variáveis não foram incorporadas ao modelo ARFIMA e, consequentemente, também não serão consideradas nos modelos não-lineares de redes neurais.

Modelos de redes neurais

Nesta seção, o objetivo é explorar o potencial de modelagem não-linear oferecido pelos modelos de redes neurais. Um modelo do tipo ARFIMA não é capaz de modelar um processo que em sua origem é não-linear e, caso uma série se comporte dessa maneira, é possível que modelos de redes neurais ofereçam melhores resultados preditivos do que modelos lineares.

Antes do ajuste das redes, é feito um pré-processamento dos dados, uma técnica muito utilizada em redes neurais (Haykin, 1999). Neste caso, como se trata de uma série temporal com memória longa, é natural pensar em filtrar a série com o filtro de memória longa utilizado no ajuste do modelo ARFIMA, ou seja, com o filtro/operador $(1 - B)^d = \sum_{j=1}^{\infty} \pi_j B^j$, com o mesmo valor de \hat{d} estimado por máxima verossimilhança. Como o filtro de memória longa envolve uma somatória infinita, essa somatória é truncada em 30 termos. Quando as redes estiverem ajustadas, é preciso fazer a inversão do filtro para que as previsões sejam recuperadas em na escala original, e com isso mais 30 parcelas são utilizadas.

Da mesma forma, como foi necessária uma defasagem sazonal para ajuste do ARFIMA (a ordem de integração sazonal do modelo ARFIMA é igual a um), aqui essa defasagem também é tomada e a inversão feita após o ajuste das redes.

Portanto, na prática, a série utilizada para ajuste dos modelos de redes neurais é a mesma utilizada para o modelo ARFIMA, que nada mais é que um modelo ARMA sazonal ajustado na série diferenciada fracionalmente e sazonalmente.

Como já explicado, as últimas 12 observações da série não foram consideradas para o ajuste. No caso de redes neurais, essa é a chamada “amostra de teste”. As 24 observações anteriores são tomadas como “amostra de validação”, para “early stopping” (ver seção 2.3.1). As observações restantes compõem a “amostra de treinamento”, para estimação dos pesos das redes.

Três tipos de redes são ajustadas: progressivas (*feedforward*) e duas variantes de redes recorrentes (Elman e Jordan). As redes têm duas camadas, a camada intermediária e a camada de saída. Para encontrar a “rede ótima”, alguns fatores foram considerados: a inicialização dos pesos (necessária para o algoritmo de Retropropagação e Retropropagação Através do Tempo), o número de neurônios na camada intermediária e o número de regressores (aqui, os regressores são defasagens da própria inflação dos Estados Unidos).

Para cada tipo de topologia, foram consideradas redes com 2 a 5 neurônios na camada intermediária.

Tipo de rede	Número de regressores	Melhor rede	MSE	MAE
<i>Feedforward</i>	2	5 neurônios, 6 ^a inicialização	$7,17 \times 10^{-6}$	$2,30 \times 10^{-3}$
<i>Feedforward</i>	4	4 neurônios, 6 ^a inicialização	$6,66 \times 10^{-6}$	$2,21 \times 10^{-3}$
<i>Feedforward</i>	6	5 neurônios, 7 ^a inicialização	$5,55 \times 10^{-6}$	$1,89 \times 10^{-3}$
Elman	2	2 neurônios, 4 ^a inicialização	$1,89 \times 10^{-5}$	$3,73 \times 10^{-3}$
Elman	4	5 neurônios, 10 ^a inicialização	$7,22 \times 10^{-6}$	$2,01 \times 10^{-3}$
Elman	6	2 neurônios, 3 ^a inicialização	$7,53 \times 10^{-6}$	$2,13 \times 10^{-3}$
Jordan	2	2 neurônios, 2 ^a inicialização	$8,23 \times 10^{-6}$	$2,38 \times 10^{-3}$
Jordan	4	4 neurônios, 5 ^a inicialização	$5,69 \times 10^{-6}$	$1,91 \times 10^{-3}$
Jordan	6	4 neurônios, 10 ^a inicialização	$5,62 \times 10^{-6}$	$1,92 \times 10^{-3}$

Tabela 5.1: Redes ajustadas para inflação dos Estados Unidos

Para cada número de neurônio, 10 inicializações diferentes foram atribuídas aos pesos, e o número de regressores também foi variado (2 regressores, sendo uma defasagem simples e uma defasagem sazonal; 4 regressores, sendo duas defasagens simples e duas sazonais; e 6 regressores, sendo três defasagens simples e três sazonais). Portanto, para cada tipo de rede, 120 modelos foram estimados em busca da melhor combinação. Os pesos são estimados através dos algoritmos de Retropropagação (para redes *feedforward*) e Retropropagação Através do Tempo (para redes recorrentes), utilizando-se do software MATLAB (Neural Networks Toolbox).

A comparação entre os 120 modelos diferentes estimados é feita através da amostra de validação via medidas de performance MSE e MAE. Dessa forma, apenas um modelo é selecionado para que seja feita a predição na amostra de teste. A tabela 5.1 mostra as melhores performances obtidas para os casos 2, 4 e 6 regressores, acompanhados do número de neurônios na camada intermediária e do número da inicialização dos pesos utilizada.

Ao final, as redes escolhidas foram as seguintes:

Feedforward: Rede com 5 neurônios na camada intermediária e 6 regressores.

Elman: Rede com 5 neurônios na camada intermediária e 4 regressores.

Jordan: Rede com 4 neurônios na camada intermediária e 4 regressores.

Redes Neurais com pesos variáveis no tempo

Nesta seção, é considerado o mesmo algoritmo da seção 4.4.1, onde foi utilizada uma rede Elman com pesos variáveis no espaço para dados de séries espaciais. A diferença é que, para séries de inflação, o algoritmo também foi adaptado para redes *feedforward* e Jordan.

O objetivo da utilização do modelo de redes neurais com pesos variando suavemente no tempo é captar uma possível heterogeneidade (mudança suave de regime) porventura presente no processo.

Para o ajuste destas redes, o número de neurônios e o número de regressores foram fixados, sendo

Tipo de rede	Fator de desconto	Melhor inicialização	MSE	MAE
<i>Feedforward</i>	0,95	4 ^a	$8,33 \times 10^{-6}$	$2,33 \times 10^{-3}$
<i>Feedforward</i>	0,96	5 ^a	$7,78 \times 10^{-6}$	$2,32 \times 10^{-3}$
<i>Feedforward</i>	0,97	3 ^a	$8,32 \times 10^{-6}$	$2,06 \times 10^{-3}$
<i>Feedforward</i>	0,98	7 ^a	$6,26 \times 10^{-6}$	$2,14 \times 10^{-3}$
<i>Feedforward</i>	0,99	9 ^a	$6,91 \times 10^{-6}$	$2,19 \times 10^{-3}$
Elman	0,95	8 ^a	$1,58 \times 10^{-4}$	$9,07 \times 10^{-3}$
Elman	0,96	6 ^a	$4,67 \times 10^{-5}$	$4,46 \times 10^{-3}$
Elman	0,97	9 ^a	$3,12 \times 10^{-4}$	$1,23 \times 10^{-2}$
Elman	0,98	3 ^a	$1,59 \times 10^{-4}$	$6,38 \times 10^{-3}$
Elman	0,99	9 ^a	$2,19 \times 10^{-5}$	$3,73 \times 10^{-3}$
Jordan	0,95	2 ^a	$8,08 \times 10^{-6}$	$2,30 \times 10^{-3}$
Jordan	0,96	2 ^a	$5,70 \times 10^{-6}$	$1,77 \times 10^{-3}$
Jordan	0,97	4 ^a	$7,01 \times 10^{-6}$	$2,17 \times 10^{-3}$
Jordan	0,98	1 ^a	$6,88 \times 10^{-6}$	$2,13 \times 10^{-3}$
Jordan	0,99	9 ^a	$8,17 \times 10^{-6}$	$2,36 \times 10^{-3}$

Tabela 5.2: Redes com pesos variáveis ajustadas para inflação dos Estados Unidos

os mesmos encontrados na seção anterior, para redes de pesos fixos. Para cada um dos tipos de rede, o fator de desconto para especificação da matriz de variância dos erros do sistema foi variado entre 0,95 e 0,99, com mudanças de 0,01. Para cada um dos casos, 10 inicializações dos pesos foram feitas, resultando num total de 50 modelos para cada tipo de rede. Os resultados se encontram na tabela 5.2. O melhor modelo foi escolhido com base nas medidas de performance MSE e MAE na amostra de validação.

As redes com pesos variáveis escolhidas foram as seguintes:

Feedforward: Rede com 5 neurônios na camada intermediária, 6 regressores e fator de desconto 0,98.

Elman: Rede com 5 neurônios na camada intermediária, 4 regressores e fator de desconto 0,99.

Jordan: Rede com 4 neurônios na camada intermediária, 4 regressores e fator de desconto 0,96.

Modelo autoregressivo linear

Com o intuito de verificar se de fato há não-linearidade presente no processo da série de inflação dos Estados Unidos, é ajustado nessa seção um modelo autoregressivo (AR) linear. O objetivo é simplesmente construir uma versão linear dos modelos de redes ajustados, com as mesmas variáveis regressoras (defasagens da série), com o objetivo de mensurar a possível melhora oferecida por um modelo não-linear (redes neurais) em comparação a uma versão linear correspondente do modelo.

Assim como feito para os outros modelos, a série foi previamente filtrada com o filtro de memória longa utilizando o valor \hat{d} estimado. Como os modelos de redes se utilizaram de 4 e 6 regressores, serão ajustados modelos AR lineares com essas mesmas variáveis. Os modelos finais ajustados, com 4 e 6

regressores, na ordem, são:

$$(1 - B)^{0,403121^2}x(t) = 0,000011 + 0,0059x(t-1) + 0,0104x(t-2) - \\ - 0,639x(t-12) - 0,369x(t-24) \quad (5.5)$$

$$(1 - B)^{0,403121^2}x(t) = 0,000003 + 0,0006x(t-1) + 0,0006x(t-2) - 0,0290x(t-3) - \\ - 0,741x(t-12) - 0,553x(t-24) - 0,287x(t-36) \quad (5.6)$$

Para outros detalhes acerca dos ajustes destes modelos, ver apêndice A.5.

5.2.3 Resultados

O objetivo é encontrar o melhor modelo para predição dessas séries temporais de memória longa, em particular a série de Inflação dos Estados Unidos. Para realizar tal comparação, vamos considerar que os últimos 12 dados da série não estivessem disponíveis. No caso do modelo ARFIMA, a predição foi feita normalmente. No caso das redes neurais com pesos fixos, os pesos estimados foram utilizados para a predição dos valores correspondentes à amostra de teste. No caso das redes neurais com pesos variáveis, a melhor inicialização para os pesos foi tomada e a estimativa dos pesos foi feita considerando a amostra de treinamento e a amostra de validação. Com isso, os pesos considerados para predição são tomados da última estimativa obtida.

Uma comparação envolvendo as medidas MSE e MAE para predições de 1, 3, 6 e 12 meses a frente, calculadas de forma acumulada (por exemplo, as medidas para 3 meses são calculadas se utilizando também das predições de 1 e 2 meses), se encontra na tabela 5.3, com os valores correspondentes às melhores predições em cada mês em negrito. Outra comparação, desta vez considerando apenas o erro absoluto em cada horizonte (1° , 3° , 6° e 12° meses a frente) em relação ao valor verdadeiro, é apresentada na tabela 5.4. Neste caso, para uma comparação mais justa, o ideal seria tomar o cálculo das predições de base móvel a cada horizonte. Isso exigiria um enorme esforço computacional, pois a cada horizonte seria preciso ajustar os modelos novamente (pois novos dados são incorporados), o que está fora de questão neste trabalho. Na prática, os casos de 1 e 3 meses são considerados predições de curto prazo, e as predições de 6 e 12 meses, de médio prazo.

Considerando as medidas acumuladas, entre os modelos de redes neurais, o modelo que mostrou maior capacidade preditiva foi a rede neural *feedforward*. No entanto, as redes só conseguem ter melhor performance preditiva do que o modelo ARFIMA para o médio prazo (6 ou 12 meses). Já no caso de comparação de erros absolutos, as redes conseguem uma performance melhor, superando os modelos lineares em 3 dos 4 horizontes considerados, com as redes recorrentes obtendo destaque.

Esse resultado está de acordo com Nakamura (2004), que detectou que os modelos lineares são mais úteis em termos de predição para a inflação dos Estados Unidos no curtíssimo ou curto prazo.

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARIMA(0; d; 0) \times (0; 1; 1) ₁₂	2, 22 \times 10⁻⁷	7, 21 \times 10⁻⁶	6, 03 \times 10 ⁻⁶	6, 67 \times 10⁻⁶
AR linear (4 regressores)	8, 75 \times 10 ⁻⁷	8, 20 \times 10 ⁻⁶	5, 23 \times 10⁻⁶	7, 72 \times 10 ⁻⁶
AR linear (6 regressores)	3, 35 \times 10 ⁻⁶	9, 02 \times 10 ⁻⁶	5, 57 \times 10⁻⁶	7, 83 \times 10 ⁻⁶
Rede <i>feedforward</i> padrão	2, 43 \times 10 ⁻⁶	1, 42 \times 10 ⁻⁵	9, 65 \times 10 ⁻⁶	8, 76 \times 10 ⁻⁶
Rede <i>feedforward</i> com pesos variáveis	2, 21 \times 10 ⁻⁶	9, 69 \times 10 ⁻⁶	5, 61 \times 10⁻⁶	9, 28 \times 10 ⁻⁶
Rede Elman padrão	4, 81 \times 10 ⁻⁵	2, 36 \times 10 ⁻⁵	1, 90 \times 10 ⁻⁵	1, 76 \times 10 ⁻⁵
Rede Elman com pesos variáveis	9, 25 \times 10 ⁻⁶	9, 51 \times 10 ⁻⁵	4, 89 \times 10 ⁻⁵	3, 27 \times 10 ⁻⁵
Rede Jordan padrão	7, 82 \times 10⁻⁶	1, 31 \times 10 ⁻⁵	1, 11 \times 10 ⁻⁵	9, 96 \times 10 ⁻⁶
Rede Jordan com pesos variáveis	3, 28 \times 10 ⁻⁵	3, 61 \times 10 ⁻⁵	2, 26 \times 10 ⁻⁵	1, 66 \times 10 ⁻⁵

Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA(0; d; 0) \times (0; 1; 1) ₁₂	4, 71 \times 10⁻⁴	2, 02 \times 10⁻³	2, 04 \times 10 ⁻³	7, 46 \times 10 ⁻³
AR linear (4 regressores)	9, 35 \times 10 ⁻⁴	2, 41 \times 10 ⁻³	1, 94 \times 10 ⁻³	2, 18 \times 10⁻³
AR linear (6 regressores)	1, 83 \times 10 ⁻³	2, 51 \times 10 ⁻³	1, 95 \times 10 ⁻³	2, 31 \times 10⁻³
Rede <i>feedforward</i> padrão	1, 56 \times 10 ⁻³	3, 48 \times 10 ⁻³	2, 68 \times 10 ⁻³	2, 38 \times 10⁻³
Rede <i>feedforward</i> com pesos variáveis	4, 70 \times 10 ⁻³	2, 80 \times 10 ⁻³	1, 82 \times 10⁻³	2, 41 \times 10 ⁻³
Rede Elman padrão	6, 94 \times 10 ⁻⁴	3, 96 \times 10 ⁻³	3, 77 \times 10 ⁻³	3, 58 \times 10 ⁻³
Rede Elman com pesos variáveis	3, 04 \times 10 ⁻³	7, 50 \times 10 ⁻³	4, 54 \times 10 ⁻³	3, 83 \times 10 ⁻³
Rede Jordan padrão	3, 04 \times 10 ⁻³	7, 50 \times 10 ⁻³	4, 54 \times 10 ⁻³	3, 83 \times 10 ⁻³
Rede Jordan com pesos variáveis	5, 73 \times 10 ⁻³	5, 47 \times 10 ⁻³	4, 06 \times 10 ⁻³	3, 41 \times 10 ⁻³

Tabela 5.3: Comparação de modelos: Inflação dos Estados Unidos (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARFIMA(0; d; 0) \times (0; 1; 1) ₁₂	4, 71 \times 10⁻⁴	1, 08 \times 10 ⁻³	1, 15 \times 10 ⁻³	3, 30 \times 10 ⁻³
AR linear (4 regressores)	9, 35 \times 10 ⁻⁴	1, 77 \times 10 ⁻³	9, 91 \times 10 ⁻⁴	2, 16 \times 10 ⁻³
AR linear (6 regressores)	1, 83 \times 10 ⁻³	9, 15 \times 10 ⁻⁴	1, 34 \times 10 ⁻³	3, 35 \times 10 ⁻³
Rede <i>feedforward</i> padrão	1, 56 \times 10 ⁻³	3, 83 \times 10 ⁻³	2, 34 \times 10 ⁻³	1, 59 \times 10⁻⁴
Rede <i>feedforward</i> com pesos variáveis	4, 70 \times 10 ⁻³	1, 61 \times 10 ⁻³	2, 36 \times 10 ⁻⁵	1, 05 \times 10 ⁻³
Rede Elman padrão	6, 94 \times 10 ⁻³	1, 72 \times 10⁻⁴	2, 50 \times 10 ⁻³	1, 12 \times 10 ⁻³
Rede Elman com pesos variáveis	3, 04 \times 10 ⁻³	1, 63 \times 10 ⁻²	1, 84 \times 10 ⁻³	2, 61 \times 10 ⁻³
Rede Jordan padrão	2, 80 \times 10 ⁻³	1, 23 \times 10 ⁻³	1, 14 \times 10 ⁻³	6, 91 \times 10 ⁻⁴
Rede Jordan com pesos variáveis	5, 73 \times 10 ⁻³	2, 33 \times 10 ⁻³	6, 01 \times 10⁻⁴	1, 35 \times 10 ⁻³

Tabela 5.4: Comparação de modelos: Inflação dos Estados Unidos (erros absolutos por mês)

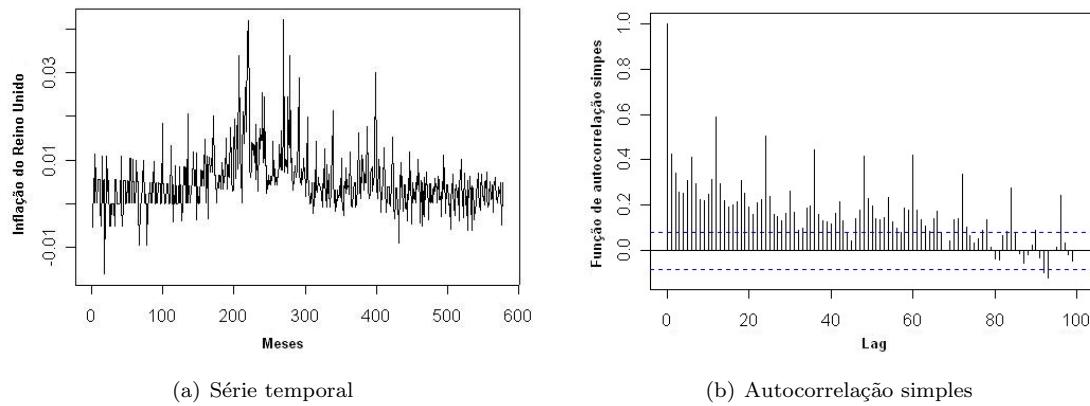


Figura 5.5: Gráficos da série de inflação do Reino Unido

5.3 Série de inflação do Reino Unido

Esta seção aborda a análise da série de inflação do Reino Unido. Os modelos a serem comparados são os mesmos da seção anterior, e os dados também foram obtidos pelo “International Financial Statistics”, tendo início em janeiro de 1957 e fim em março de 2005.

5.3.1 Análise descritiva

O gráfico da série de inflação do Reino Unido sugere não-estacionaridade em média, o que é confirmado com o gráfico da função de autocorrelação, que persiste por dezenas de lags, também confirmando a presença de memória longa (ver figura 5.5).

5.3.2 Ajuste de modelos

Modelo ARFIMA (ordem de integração fracionária)

A ordem de integração fracionária \hat{d} foi estimada através do software OX por três métodos: máxima verossimilhança, perfil de verossimilhança modificada e mínimos quadrados não-lineares. Os resultados estão apresentados na tabela 5.5. Após encontrada essa estimativa, a série foi filtrada com o filtro de memória longa (polinômio infinito nas potências do operador *backshift*).

Para avaliar a diferença dos métodos de estimação, a série foi filtrada com os valores de d estimados pelos métodos de máxima verossimilhança e mínimos quadrados não-lineares (polinômio truncado em 30 termos). Uma parte da série filtrada pelos dois métodos é apresentada na figura 5.6, o que mostra que praticamente não existe diferença significante entre os resultados obtidos pelos diferentes métodos. Por isso, o valor de \hat{d} obtido por máxima verossimilhança será utilizado na análise desta seção.

Para avaliar o impacto do número de termos necessário para truncagem para o cálculo do polinômio infinito, a figura 5.7 mostra a série filtrada com truncagem de 30 e 90 termos (utilizando \hat{d} obtido por

Método de estimação	Valor de \hat{d}
Máxima verossimilhança	0,287240
Perfil de veross. modificada	0,287237
Mínimos quadr. não-lineares	0,289238

Tabela 5.5: Valores de \hat{d} por diferentes métodos (inflação do Reino Unido)

máxima verossimilhança), e novamente nota-se pouca diferença. A análise considerá a filtragem feita com o polinômio truncado em 30 termos, como é o caso de um outro exemplo amostrado em Shumway & Stoffer (2000).

Foi detectado que a série possuia autocorrelações sazonais que não decaiam exponencialmente como mostrado na figura 5.8(a) (teste Dickey-Fuller para raiz unitária sazonal: p-valor = 0,4502). Dessa forma, uma diferenciação sazonal também foi tomada. Os gráficos de autocorrelação e autocorrelação parcial encontram-se na figura 5.8. Os gráficos sugerem uma componente MA(1) sazonal de período 12 (uma autocorrelação sazonal significante e autocorrelações parciais sazonais decaendo rapidamente). Assim como feito para a série americana, uma série de modelos ARMA foram ajustados. O modelo encontrado para a série diferenciada é o mesmo da série americana: ARIMA(0,0,0)×(0,1,1)₁₂.

Portanto, o modelo escolhido é um ARFIMA (0;0,2982073;0)×(0;1;1)₁₂ com parâmetros estimados via software S-PLUS:

$$\hat{\Theta}_1 = 0,7382 \pm 0,0295 \quad (5.7)$$

$$\hat{\sigma}^2 = 1,96 \times 10^{-5} \quad (5.8)$$

Modelo final com parâmetros estimados

$$(1 - B^{12})(1 - B)^{0,2982073}x(t) = (1 + 0,7382B^{12})a(t) \quad (5.9)$$

$$a(t) \sim N(0; 1,96 \times 10^{-5}) \quad (5.10)$$

Embora não apresentados aqui, os gráficos das autocorrelações dos resíduos do ajuste apresentaram resultados satisfatórios.

Modelos de redes neurais

O mesmo processo de ajuste de modelos de redes neurais feito com a série de inflação americana é feita aqui, com o intuito de verificar se alguma dessas séries segue um processo não-linear.

Antes do ajuste das redes, os dados foram diferenciados fracionalmente e sazonalmente. As últimas 12 observações da série não foram consideradas para o ajuste (“amostra de teste”). As 24 observações anteriores são tomadas como “amostra de validação”, para “early stopping”. As observações restantes compõem a “amostra de treinamento”, para estimação dos pesos das redes.

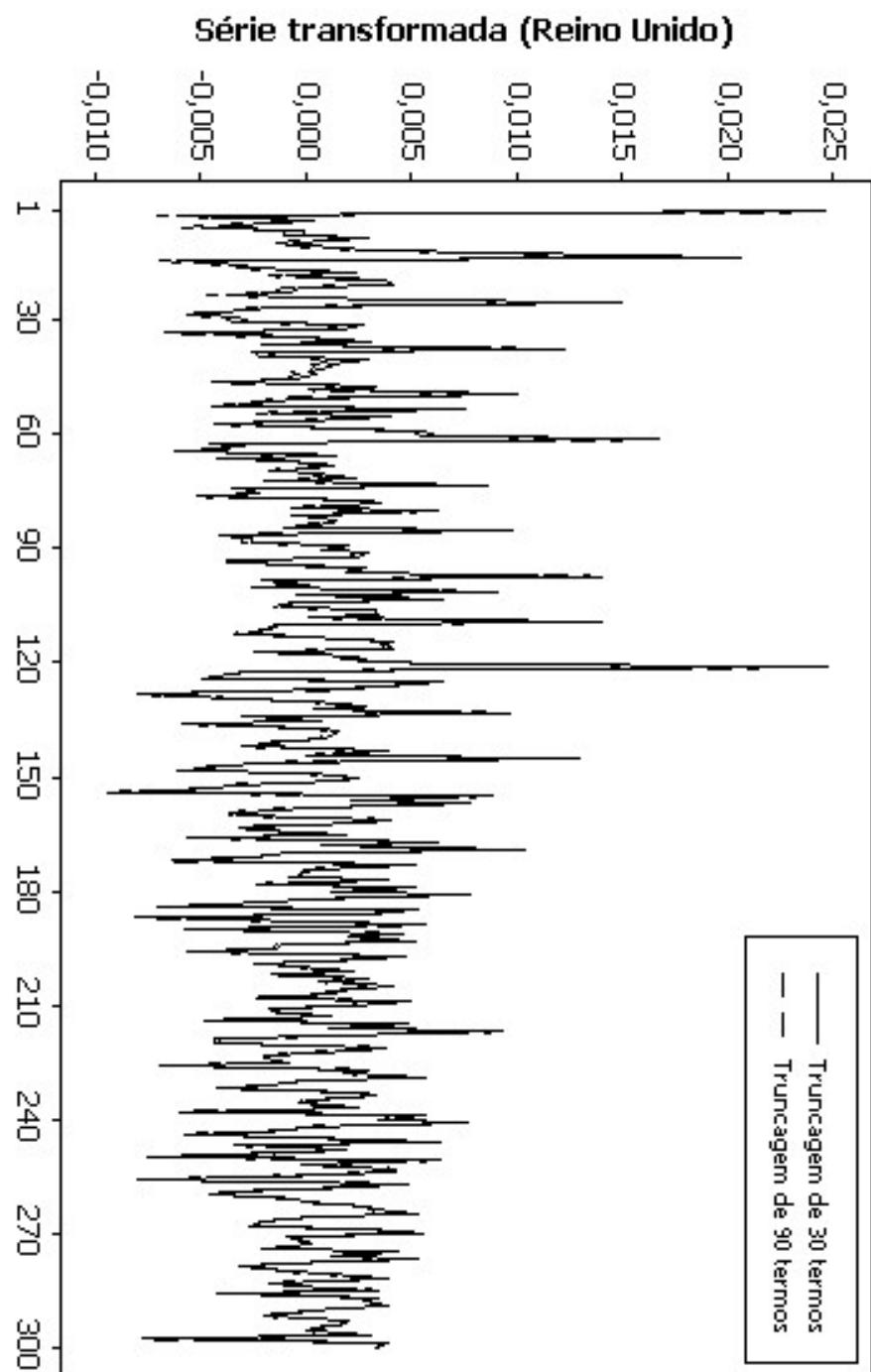


Figura 5.6: Inflação do Reino Unido filtrada: comparação de métodos de estimação de d

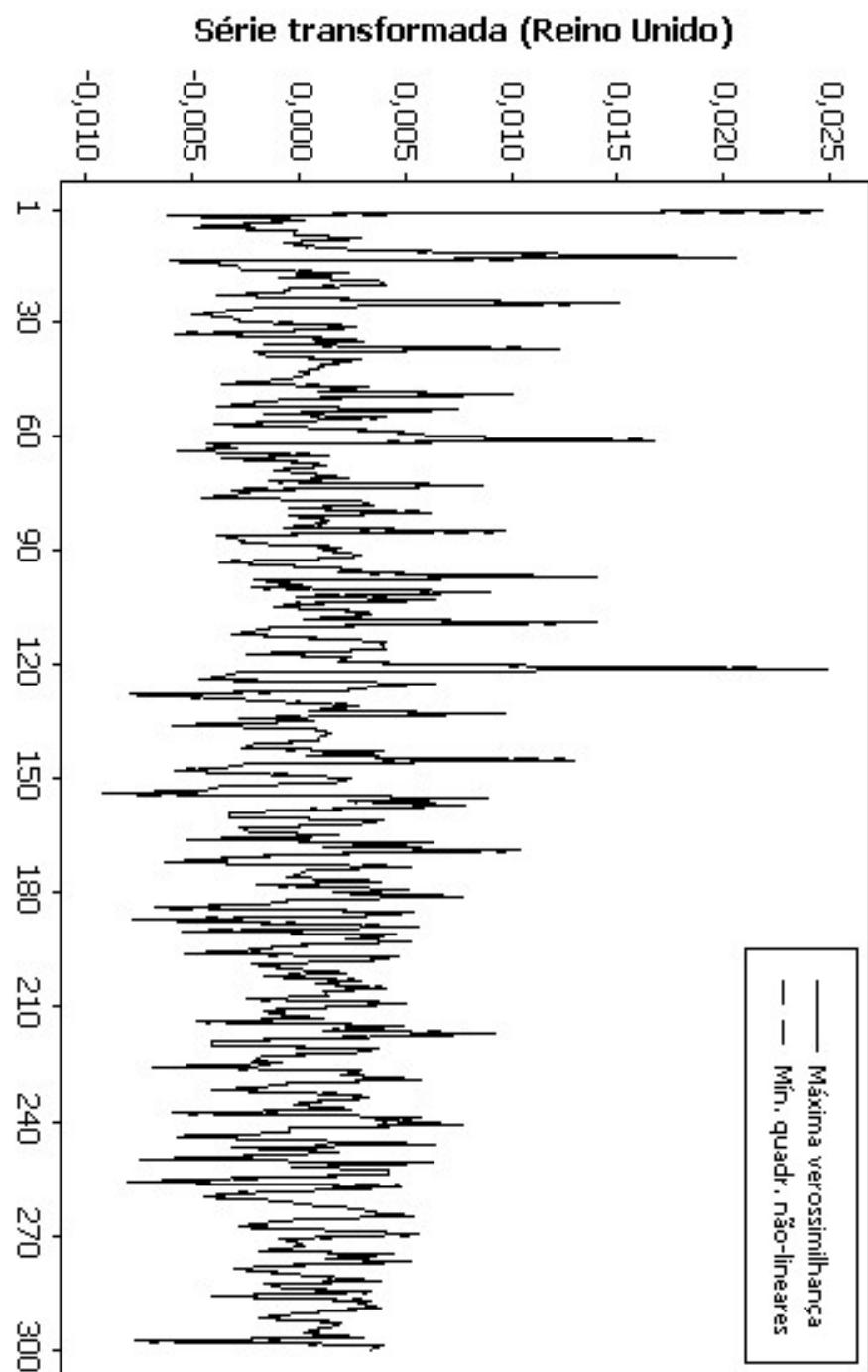


Figura 5.7: Inflação do Reino Unido filtrada: comparação de número de termos truncados

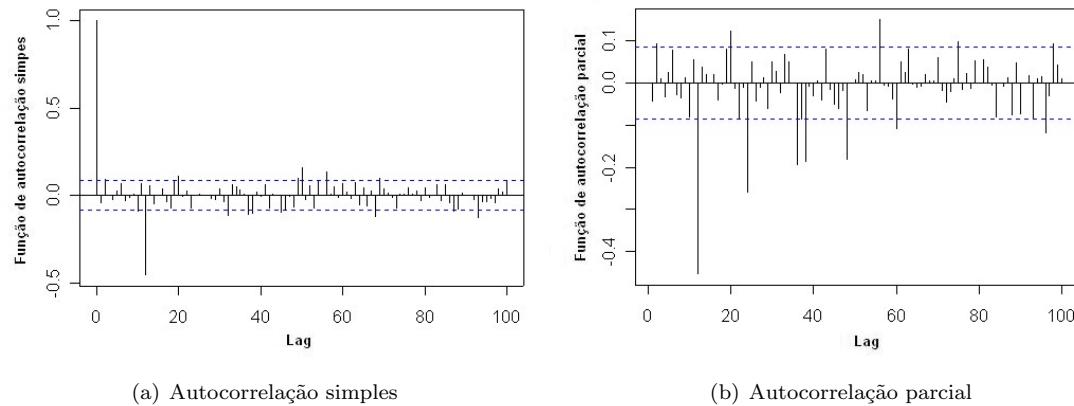


Figura 5.8: Gráficos da série de inflação do Reino Unido diferenciada fracionalmente e sazonalmente

Tipo de rede	Neurônios na camada intermediária	Regressores
<i>Feedforward</i>	5	4
Elman	3	4
Jordan	3	6

Tabela 5.6: Modelos escolhidos para redes de pesos fixos (inflação do Reino Unido)

Redes *feedforward* e recorrentes (tipos Elman e Jordan) são ajustadas, variando-se o número de neurônios na camada intermediária (2 a 5), o número de regressores (2, 4 e 6) e a inicialização dos pesos (10 inicializações diferentes). A comparação entre as 120 redes de cada um dos três tipos é feita através das medidas MSE e MAE na amostra de validação. Ao final, as redes “ótimas” obtidas são as apresentadas na tabela 5.6.

Redes Neurais com pesos variáveis no tempo

Para o ajuste de redes neurais com pesos variáveis no tempo, as redes foram fixadas com o mesmo número de neurônios e de regressores encontrados na seção anterior. Os fatores de desconto (0,95 a 0,99) e as inicializações (10) foram variadas, com a performance avaliada na amostra de validação. As melhores redes estão apresentadas na tabela 5.7.

Tipo de rede	Neurônios na camada intermediária	Regressores	Fator de desconto
<i>Feedforward</i>	5	4	0,97
Elman	3	4	0,95
Jordan	3	6	0,97

Tabela 5.7: Modelos escolhidos para redes de pesos variáveis (inflação do Reino Unido)

Modelo autoregressivo linear

Nesta seção, são ajustados modelos autoregressivos lineares para a série de inflação do Reino Unido em questão com o intuito de construir uma versão linear dos modelos de redes ajustados, com as mesmas variáveis regressoras (defasagens da série), com o objetivo de mensurar a melhora oferecida por um modelo não-linear (redes neurais) em comparação uma versão linear correspondente do modelo.

As variáveis regressoras foram escolhidas com base nos modelos de redes neurais ajustados. Assim como feito para os outros modelos, a série foi previamente filtrada com o filtro de memória longa utilizando o valor \hat{d} estimado.

$$(1 - B^{12})(1 - B)^{0,2982073}x(t) = -0,000021 - 0,0142x(t-1) + 0,0612x(t-2) - \\ - 0,573x(t-12) - 0,270x(t-24) \quad (5.11)$$

$$(1 - B^{12})(1 - B)^{0,2982073}x(t) = 0,000023 + 0,0039x(t-1) + 0,0544x(t-2) + \\ + 0,0054x(t-3) - 0,605x(t-12) - \\ - 0,369x(t-24) - 0,168x(t-36) \quad (5.12)$$

Para maiores detalhes acerca dos ajustes dos modelos, ver apêndice A.5.

5.3.3 Resultados

Considerando que os últimos 12 dados da série não estivessem disponíveis, uma comparação preditiva é feita envolvendo as medidas MSE e MAE acumuladas calculadas para previsões de 1, 3, 6 e 12 meses a frente. Os valores se encontram na tabela 5.8, com os valores correspondentes às melhores previsões em cada mês em negrito. Uma comparação considerando apenas o erro absoluto em cada horizonte (1º, 3º, 6º e 12º meses a frente) em relação ao valor verdadeiro é apresentada na tabela 5.9.

Ambas as tabelas mostram que as redes *feedforward* são os melhores modelos, alternando entre pesos fixos e variáveis dependendo do mês. O modelo ARFIMA só é competitivo em previsão a médio ou longo prazo (12 meses).

A figura 5.9 mostra o gráfico dos resíduos padronizados do modelo de redes de pesos variáveis, além dos gráficos de autocorrelação simples e parcial.

Para mostrar realmente que a aproximação de 2ª ordem é necessária, é ajustado aqui um modelo de rede neural *feedforward* com pesos variáveis para a série de inflação do Reino Unido treinada por um algoritmo baseado em extensão do filtro de Kalman de 1ª ordem. O processo de treinamento é o mesmo descrito na seção 5.4, e a arquitetura da rede é a mesma utilizada (5 neurônios com 4 regressores). Os resultados preditivos são piores que o modelo obtido com o algoritmo de 2ª ordem, como mostra a tabela 5.10, conforme esperado, o que fortalece a hipótese de que a aproximação de 1ª ordem é insuficiente para o caso de redes neurais.

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA(0; d ; 0) \times (0; 1; 1) ₁₂	$3,79 \times 10^{-6}$	$1,87 \times 10^{-6}$	$2,09 \times 10^{-6}$	$3,22 \times 10^{-6}$
AR linear (4 regressores)	$8,00 \times 10^{-7}$	$1,21 \times 10^{-6}$	$1,48 \times 10^{-6}$	$3,71 \times 10^{-5}$
AR linear (6 regressores)	$1,98 \times 10^{-6}$	$1,34 \times 10^{-6}$	$1,86 \times 10^{-6}$	$3,28 \times 10^{-5}$
Rede feedforward padrão	$2,61 \times 10^{-6}$	$8,90 \times 10^{-7}$	$2,12 \times 10^{-6}$	$3,23 \times 10^{-6}$
Rede feedforward com pesos variáveis	$7,74 \times 10^{-7}$	$1,93 \times 10^{-6}$	$1,52 \times 10^{-6}$	$3,44 \times 10^{-6}$
Rede Elman padrão	$3,06 \times 10^{-6}$	$3,00 \times 10^{-6}$	$2,77 \times 10^{-6}$	$3,36 \times 10^{-6}$
Rede Elman com pesos variáveis	$1,05 \times 10^{-6}$	$5,50 \times 10^{-6}$	$3,90 \times 10^{-6}$	$3,36 \times 10^{-6}$
Rede Jordan padrão	$4,87 \times 10^{-6}$	$7,21 \times 10^{-6}$	$2,22 \times 10^{-5}$	$3,49 \times 10^{-5}$
Rede Jordan com pesos variáveis	$1,07 \times 10^{-6}$	$2,54 \times 10^{-6}$	$7,36 \times 10^{-6}$	$1,43 \times 10^{-5}$

Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA(0; d ; 0) \times (0; 1; 1) ₁₂	$1,95 \times 10^{-3}$	$1,28 \times 10^{-3}$	$1,29 \times 10^{-3}$	$1,60 \times 10^{-3}$
AR linear (4 regressores)	$8,94 \times 10^{-4}$	$1,05 \times 10^{-3}$	$1,09 \times 10^{-3}$	$1,58 \times 10^{-3}$
AR linear (6 regressores)	$1,41 \times 10^{-3}$	$1,14 \times 10^{-3}$	$1,29 \times 10^{-3}$	$1,63 \times 10^{-3}$
Rede feedforward padrão	$1,62 \times 10^{-3}$	$6,31 \times 10^{-4}$	$9,14 \times 10^{-4}$	$1,23 \times 10^{-3}$
Rede feedforward com pesos variáveis	$8,80 \times 10^{-4}$	$1,33 \times 10^{-3}$	$1,17 \times 10^{-3}$	$1,58 \times 10^{-3}$
Rede Elman padrão	$1,75 \times 10^{-3}$	$1,73 \times 10^{-3}$	$1,51 \times 10^{-3}$	$1,44 \times 10^{-3}$
Rede Elman com pesos variáveis	$1,03 \times 10^{-3}$	$2,19 \times 10^{-3}$	$1,80 \times 10^{-3}$	$1,51 \times 10^{-3}$
Rede Jordan padrão	$2,21 \times 10^{-3}$	$2,41 \times 10^{-3}$	$3,43 \times 10^{-3}$	$4,80 \times 10^{-3}$
Rede Jordan com pesos variáveis	$1,04 \times 10^{-3}$	$1,50 \times 10^{-3}$	$2,01 \times 10^{-3}$	$2,94 \times 10^{-3}$

Tabela 5.8: Comparação de modelos: Inflação do Reino Unido (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARFIMA(0; d ; 0) \times (0; 1; 1) ₁₂	$1,95 \times 10^{-3}$	$1,04 \times 10^{-3}$	$1,44 \times 10^{-3}$	$1,22 \times 10^{-3}$
AR linear (4 regressores)	$8,95 \times 10^{-4}$	$1,49 \times 10^{-3}$	$1,38 \times 10^{-3}$	$7,39 \times 10^{-4}$
AR linear (6 regressores)	$1,41 \times 10^{-3}$	$1,11 \times 10^{-3}$	$1,77 \times 10^{-3}$	$1,18 \times 10^{-3}$
Rede feedforward padrão	$1,62 \times 10^{-3}$	$4,10 \times 10^{-5}$	$3,14 \times 10^{-3}$	$3,71 \times 10^{-4}$
Rede feedforward com pesos variáveis	$8,80 \times 10^{-4}$	$1,85 \times 10^{-3}$	$8,64 \times 10^{-4}$	$6,11 \times 10^{-4}$
Rede Elman padrão	$1,75 \times 10^{-3}$	$1,62 \times 10^{-3}$	$2,58 \times 10^{-3}$	$6,83 \times 10^{-4}$
Rede Elman com pesos variáveis	$1,03 \times 10^{-3}$	$2,80 \times 10^{-3}$	$2,04 \times 10^{-3}$	$5,60 \times 10^{-4}$
Rede Jordan padrão	$2,21 \times 10^{-3}$	$1,08 \times 10^{-3}$	$3,04 \times 10^{-4}$	$1,56 \times 10^{-3}$
Rede Jordan com pesos variáveis	$1,04 \times 10^{-3}$	$1,07 \times 10^{-3}$	$8,26 \times 10^{-4}$	$1,13 \times 10^{-3}$

Tabela 5.9: Comparação de modelos: Inflação do Reino Unido (erros absolutos por mês)

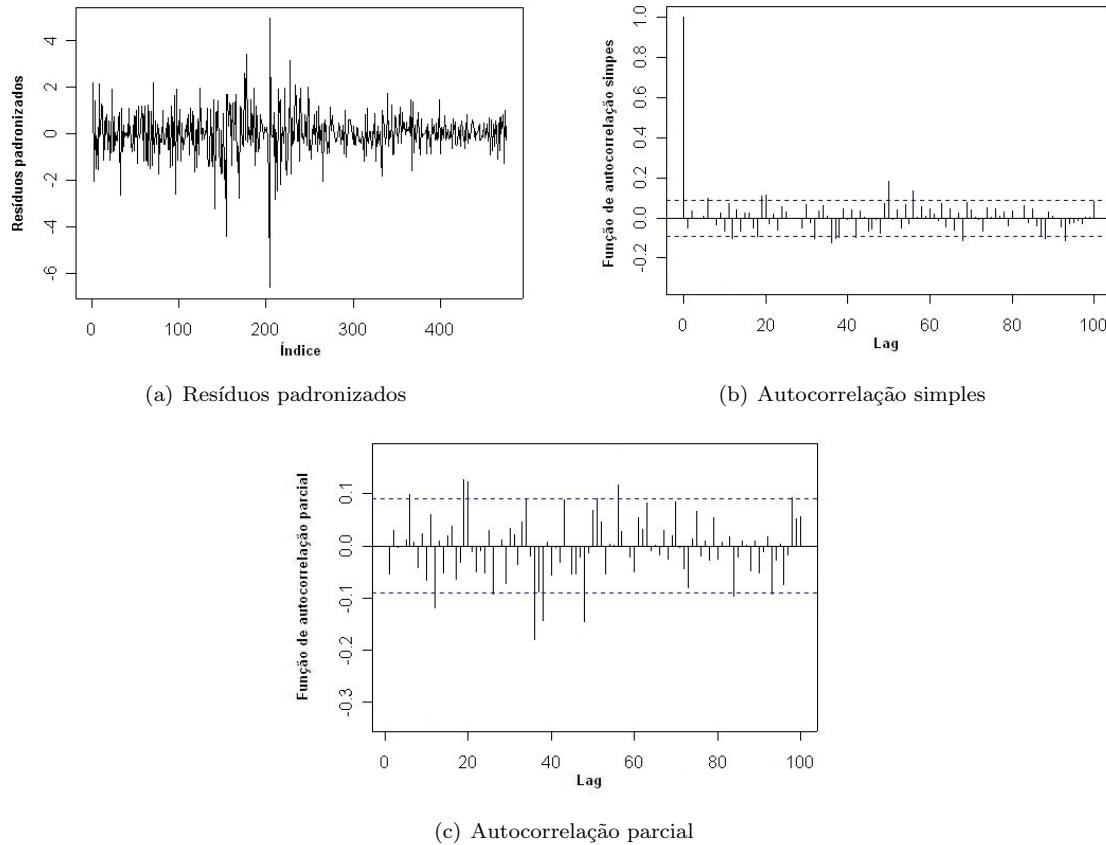
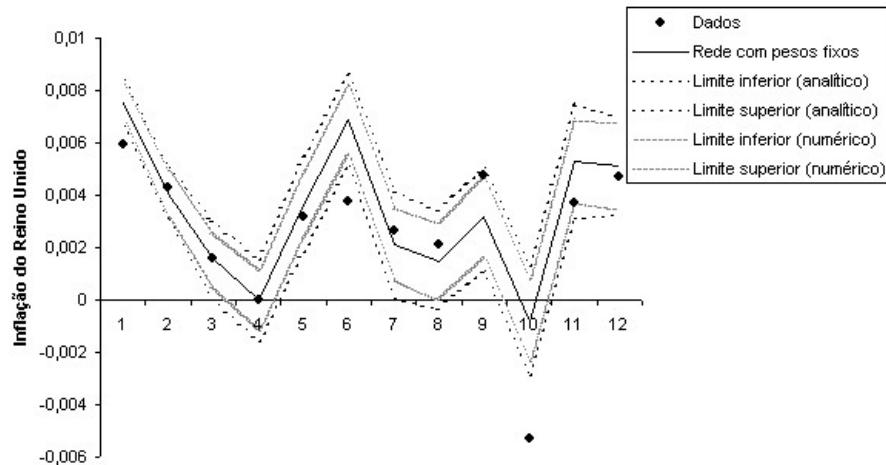


Figura 5.9: Gráficos de resíduos: modelo de redes para inflação do Reino Unido

Algoritmo	MSE			
	1 mês	3 meses	6 meses	12 meses
EKF 1 ^a ordem	$8,45 \times 10^{-7}$	$1,03 \times 10^{-6}$	$2,03 \times 10^{-6}$	$4,55 \times 10^{-6}$
EKF 2 ^a ordem	$7,74 \times 10^{-7}$	$1,93 \times 10^{-6}$	$1,52 \times 10^{-6}$	$3,44 \times 10^{-6}$
MAE				
Algoritmo	1 mês	3 meses	6 meses	12 meses
	$9,19 \times 10^{-4}$	$9,63 \times 10^{-4}$	$1,19 \times 10^{-3}$	$1,76 \times 10^{-3}$
EKF 2 ^a ordem	$8,80 \times 10^{-4}$	$1,33 \times 10^{-3}$	$1,17 \times 10^{-3}$	$1,58 \times 10^{-3}$

Tabela 5.10: Comparação de algoritmos: Inflação do Reino Unido (redes *feedforward* com pesos variáveis)

Mês	Dado real	Predição	Intervalo analítico	Intervalo numérico
1	0,00594	0,007556208	(0,00667; 0,00844)	(0,0069; 0,00821)
2	0,0043	0,004065567	(0,00306; 0,00507)	(0,00319; 0,00494)
3	0,001605	0,001563956	(0,00006; 0,00307)	(0,00053; 0,0026)
4	0	-1,16356E-05	(-0,00156; 0,00154)	(-0,00116; 0,00114)
5	0,003203	0,003646483	(0,00182; 0,00548)	(0,00241; 0,00489)
6	0,003736	0,006872363	(0,00509; 0,00865)	(0,00555; 0,00819)
7	0,002648	0,002126949	(0,00005; 0,00421)	(0,00074; 0,00351)
8	0,002119	0,001472419	(-0,00034; 0,00342)	(0,00003; 0,00292)
9	0,004754	0,003158512	(0,00113; 0,00503)	(0,00166; 0,00466)
10	-0,005286	-0,000800499	(-0,0029; 0,00129)	(-0,00235; 0,00075)
11	0,003706	0,005304297	(0,00315; 0,00746)	(0,00371; 0,0069)
12	0,00473	0,005101095	(0,00321; 0,00699)	(0,00346; 0,00674)

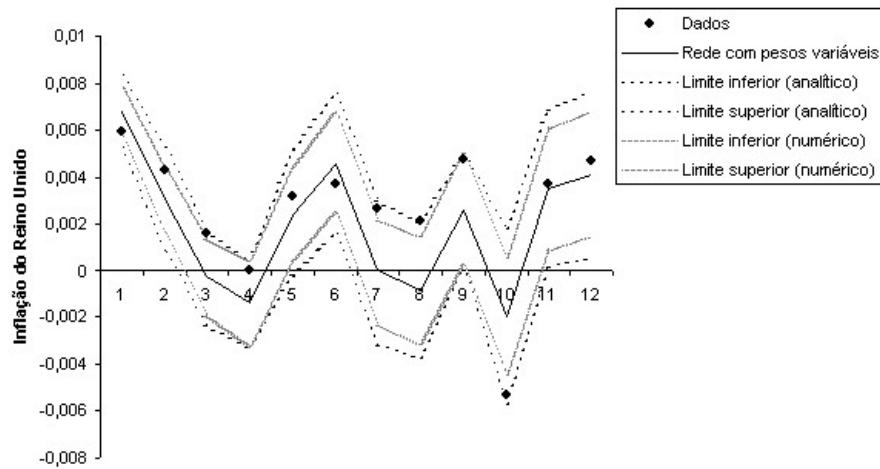
Tabela 5.11: Intervalos de predição de 95% (12 últimos meses): Rede *feedforward* de pesos fixosFigura 5.10: Intervalos de predição: Rede *feedforward* com pesos fixos

5.3.4 Intervalos de predição

Nas figuras 5.10 e 5.11 são apresentados os intervalos de predição para os modelos de redes neurais *feedforward* de pesos fixos e variáveis, respectivamente, para os últimos 12 pontos da série. Os intervalos foram obtidos por dois métodos: analítico e via reamostragem dos resíduos (bootstrap), através do procedimento descrito na seção 2.7.

É possível notar que o intervalo para a rede com pesos variáveis cobre um maior número de pontos (apenas dois estão notadamente fora do intervalo, enquanto que no caso da rede com pesos fixos são três pontos fora dos limites). Os intervalos analíticos também cobrem um número maior de pontos do que os intervalos numéricos, pois estes são calculados baseados em aproximações.

Mês	Dado real	Predição	Intervalo analítico	Intervalo numérico
1	0,00594	0,006820012	(0,0069; 0,00821)	(0,00578; 0,00786)
2	0,0043	0,003048482	(0,00319; 0,00494)	(0,00165; 0,00445)
3	0,001605	-0,000249614	(0,00053; 0,0026)	(-0,00191; 0,00141)
4	0	-0,001415703	(-0,00116; 0,00114)	(-0,00328; 0,00045)
5	0,003203	0,00242879	(0,00241; 0,00489)	(0,00043; 0,00443)
6	0,003736	0,00459976	(0,00555; 0,00819)	(0,00247; 0,00673)
7	0,002648	-4,58024E-05	(0,00074; 0,00351)	(-0,00229; 0,0022)
8	0,002119	-0,00084247	(0,00003; 0,00292)	(-0,00319; 0,0015)
9	0,004754	0,002613381	(0,00166; 0,00466)	(0,00018; 0,00505)
10	-0,005286	-0,001956071	(-0,00235; 0,00075)	(-0,00446; 0,00055)
11	0,003706	0,003499679	(0,00371; 0,0069)	(0,00092; 0,00608)
12	0,00473	0,004119424	(0,00346; 0,00674)	(0,00146; 0,00677)

Tabela 5.12: Intervalos de predição de 95% (12 últimos meses): Rede *feedforward* de pesos variáveisFigura 5.11: Intervalos de predição: Rede *feedforward* com pesos variáveis

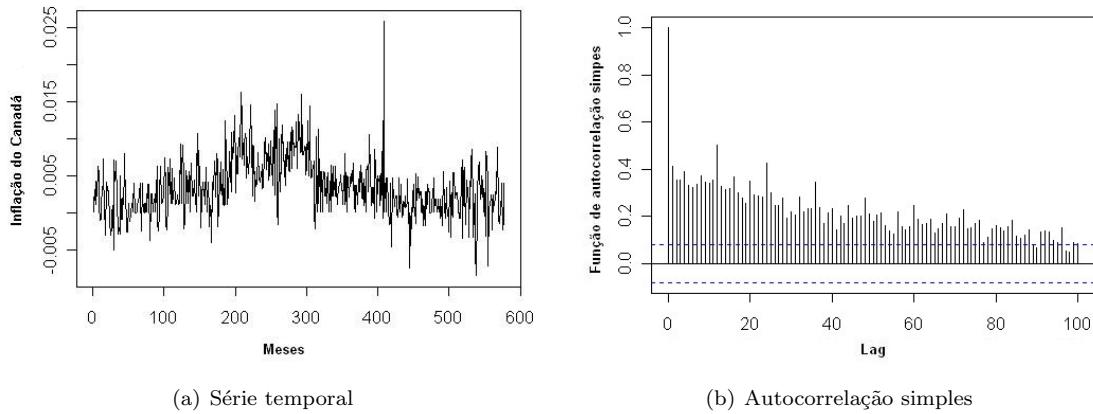


Figura 5.12: Gráficos da série de inflação do Canadá

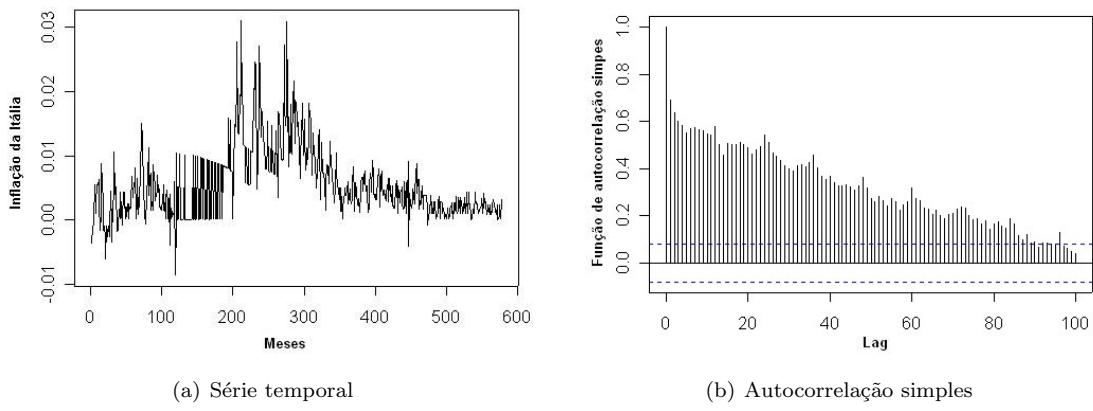


Figura 5.13: Gráficos da série de inflação da Itália

5.4 Outras séries de inflação de memória longa: Canadá, Itália e França

Esta seção aborda séries de inflação de outros três países que possuem memória longa: Canadá, Itália e França. Os modelos a serem comparados são os mesmos da seção anterior, e os dados também foram obtidos pelo “International Financial Statistics”, tendo início em janeiro de 1957 e fim em março de 2005.

5.4.1 Análise Descritiva

Os gráficos das séries sugerem não-estacionaridade em média, o que é confirmado com os gráficos das funções de autocorrelação, que persistem por dezenas de lags, também confirmando a presença de memória longa (ver figuras 5.12, 5.13 e 5.14).

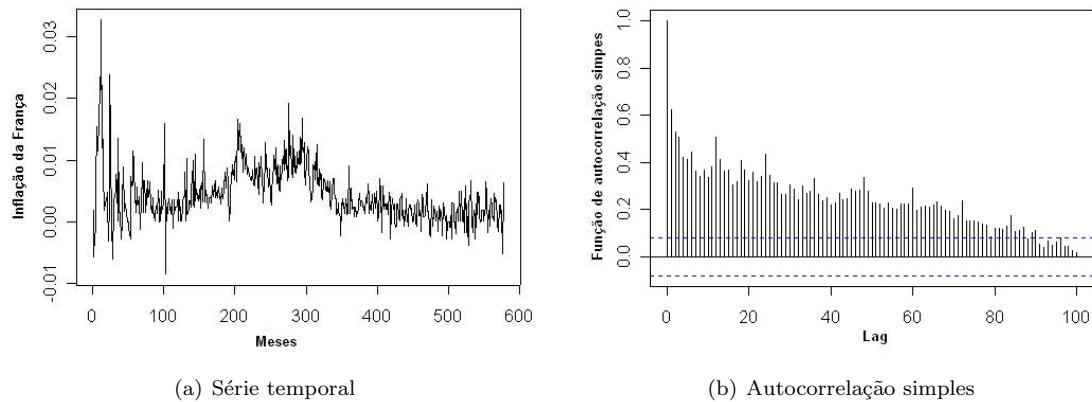


Figura 5.14: Gráficos da série de inflação da França

País	\hat{d}
Canadá	0,2844788
Itália	0,4271725
França	0,4109501

Tabela 5.13: Estimação da ordem de integração fracionária

5.4.2 Ajuste de modelos

Modelo ARFIMA (ordem de integração fracionária)

A ordem de integração fracionária \hat{d} foi estimada através do software S-PLUS (método de máxima verossimilhança), que ofereceu os resultados da tabela 5.13.

Após a filtragem da série com o filtro de memória longa, foi detectado que todas as séries possuíam autocorrelações sazonais que não decaiam exponencialmente (teste Dickey-Fuller para raiz unitária sazonal: valor-p > 0,3 para os três países). Dessa forma, uma diferenciação sazonal também foi tomada.

Os gráficos de autocorrelação e autocorrelação parcial das séries (figuras 5.15, 5.16 e 5.17) sugerem uma componente MA(1) sazonal de período 12 (uma autocorrelação sazonal significante e autocorrelações parciais sazonais decaendo rapidamente). Assim como feito para a série americana, uma série de modelos ARMA foram ajustados às séries. Os modelos com resíduos satisfatórios estão apresentados na tabela 5.14.

Modelos de Redes Neurais

O mesmo processo de ajuste de modelos de redes neurais feito com a série de inflação americana e do Reino Unido é feita aqui também para Canadá, Itália e França, com o intuito de verificar se alguma dessas séries segue um processo não-linear.

Antes do ajuste das redes, os dados dos quatro países foram diferenciados fracionalmente e sazonal-

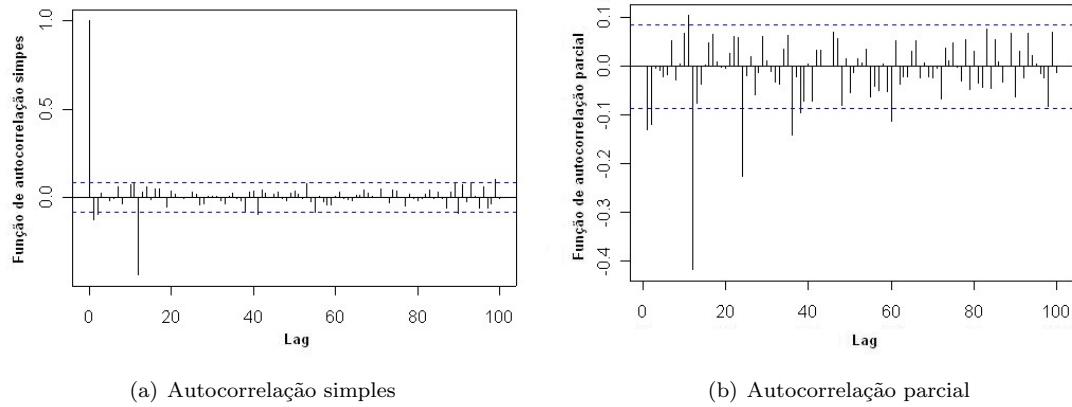


Figura 5.15: Gráficos da série de inflação do Canadá diferenciada fracionalmente e sazonalmente

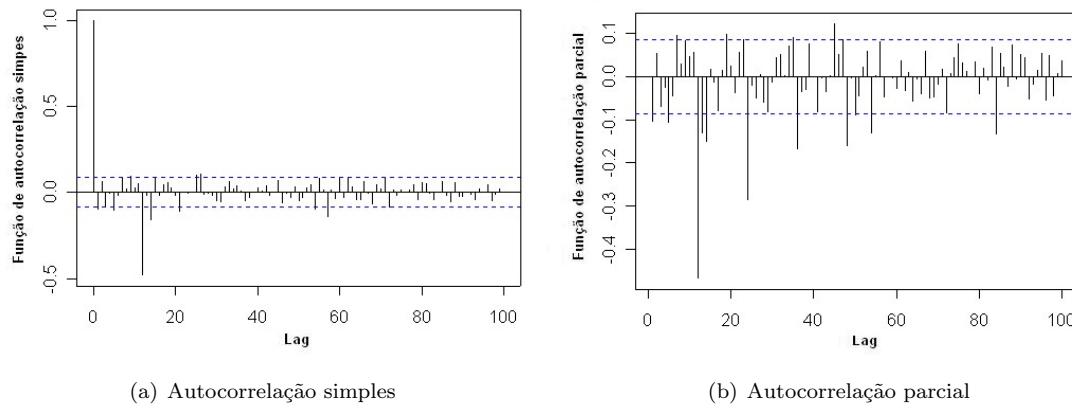


Figura 5.16: Gráficos da série de inflação da Itália diferenciada fracionalmente e sazonalmente

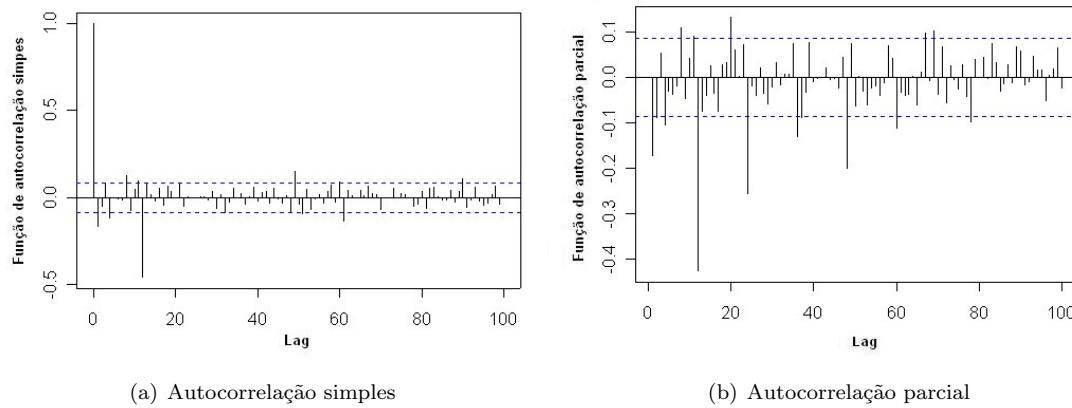


Figura 5.17: Gráficos da série de inflação da França diferenciada fracionalmente e sazonalmente

Canadá
$\text{ARFIMA}(0;0,2844788;1) \times (0;1;1)$
Modelo: $(1 - B^{12})(1 - B)^{0,2844788}x(t) = (1 + 0,7126B^{12})(1 + 0,1415B)a(t)$
Itália
$\text{ARFIMA}(0;0,4271725;1) \times (0;1;1)$
Modelo: $(1 - B^{12})(1 - B)^{0,4271725}x(t) = (1 + 0,8155B^{12})(1 + 0,1147B)a(t)$
França
$\text{ARFIMA}(0;0,4109501;1) \times (0;1;1)$
Modelo: $(1 - B^{12})(1 - B)^{0,4109501}x(t) = (1 + 0,7346B^{12})(1 + 0,1836B)a(t)$

Tabela 5.14: Modelos ARFIMA para séries de inflação com memória longa

Tipo de rede	Neurônios na camada intermediária	Regressores
Canadá		
<i>Feedforward</i>	5	4
Elman	2	2
Jordan	5	2
Itália		
<i>Feedforward</i>	2	4
Elman	3	4
Jordan	3	4
França		
<i>Feedforward</i>	3	2
Elman	2	4
Jordan	2	2

Tabela 5.15: Modelos escolhidos para redes de pesos fixos

mente. As últimas 12 observações da série não foram consideradas para o ajuste (“amostra de teste”). As 24 observações anteriores são tomadas como “amostra de validação”, para “early stopping”. As observações restantes compõem a “amostra de treinamento”, para estimação dos pesos das redes.

Redes *feedforward* e recorrentes (tipos Elman e Jordan) são ajustadas, variando-se o número de neurônios na camada intermediária (2 a 5), o número de regressores (2, 4 e 6) e a inicialização dos pesos (10 inicializações diferentes). A comparação entre as 120 redes de cada um dos três tipos é feita através das medidas MSE e MAE na amostra de validação. Ao final, as redes “ótimas” obtidas são as apresentadas na tabela 5.15.

Redes Neurais com pesos variáveis no tempo

Para o ajuste de redes neurais com pesos variáveis no tempo, as redes foram fixadas com o mesmo número de neurônios e de regressores encontrados na seção anterior. Os fatores de desconto (0,95 a 0,99) e as inicializações (10) foram variadas, com a performance avaliada na amostra de validação. As

Tipo de rede	Neurônios na camada intermediária	Regressores	Fator de desconto
Canadá			
<i>Feedforward</i>	5	4	0,95
Elman	2	2	0,95
Jordan	5	2	0,99
Itália			
<i>Feedforward</i>	2	4	0,95
Elman	3	4	0,98
Jordan	3	4	0,99
França			
<i>Feedforward</i>	3	2	0,99
Elman	2	4	0,96
Jordan	2	2	0,97

Tabela 5.16: Modelos escolhidos para redes de pesos variáveis

melhores redes estão apresentadas na tabela 5.16.

Modelo autoregressivo linear

Nesta seção, são ajustados modelos autoregressivos lineares para as séries de inflação de memória longa em questão com o intuito de construir uma versão linear dos modelos de redes ajustados, com as mesmas variáveis regressoras (defasagens da série), com o objetivo de mensurar a melhora oferecida por um modelo não-linear (redes neurais) em comparação uma versão linear correspondente do modelo.

As variáveis regressoras foram escolhidas com base nos modelos de redes neurais ajustados. Assim como feito para os outros modelos, as séries foram previamente filtradas com o filtro de memória longa utilizando os valores \hat{d} estimados.

Canadá:

$$(1 - B^{12})(1 - B)^{0,2844788}x(t) = 0,000009 - 0,0956x(t-1) - 0,456x(t-12) \quad (5.13)$$

$$\begin{aligned} (1 - B^{12})(1 - B)^{0,2844788}x(t) &= 0,000010 + 0,102x(t-1) + 0,0831x(t-2) - \\ &\quad - 0,570x(t-12) - 0,266x(t-24) \end{aligned} \quad (5.14)$$

Itália:

$$\begin{aligned} (1 - B^{12})(1 - B)^{0,4271725}x(t) &= -0,000030 - 0,0697x(t-1) + 0,0723x(t-2) - \\ &\quad - 0,647x(t-12) - 0,320x(t-24) \end{aligned} \quad (5.15)$$

França:

$$(1 - B^{12})(1 - B)^{0,4109501}x(t) = 0,000003 - 0,116x(t-1) - 0,458x(t-12) \quad (5.16)$$

$$\begin{aligned} (1 - B^{12})(1 - B)^{0,4109501}x(t) &= -0,000031 - 0,124x(t-1) - 0,0734x(t-2) - \\ &\quad - 0,599x(t-12) - 0,298x(t-24) \end{aligned} \quad (5.17)$$

Para maiores detalhes acerca dos ajustes dos modelos, ver apêndice A.5.

5.4.3 Resultados

Considerando que os últimos 12 dados da série não estivessem disponíveis, uma comparação preditiva é feita envolvendo as medidas MSE e MAE acumuladas calculadas para previsões de 1, 3, 6 e 12 meses a frente. Os valores se encontram na tabela 5.17, 5.19 e 5.21, com os valores correspondentes às melhores previsões em cada mês em negrito. Uma comparação considerando apenas o erro absoluto em cada horizonte (1º, 3º, 6º e 12º meses a frente) em relação ao valor verdadeiro é apresentada nas tabelas 5.18, 5.20 e 5.22.

Resultados muito distintos são encontrados aqui. No caso do Canadá, ARFIMA e redes *feedforward* são os melhores modelos. Interessante notar que não há predominância de nenhum desses modelos em relação a previsão de curto e longo prazo. O modelo ARFIMA é melhor para previsão de 1 e 12 meses a frente, enquanto redes *feedforward* (de pesos variáveis) têm melhor performance nos intervalos intermediários (3 e 6 meses).

Os casos de Itália e França são os que há maior predominância de um único modelo. A série italiana apresenta melhores previsões quando modelada pela rede Elman de pesos fixos, independente do horizonte considerado, no caso das medidas acumuladas. Considerando apenas os erros absolutos por horizonte, a rede Elman também é a que mais se destaca. Já a série francesa é melhor predita pelo modelo ARFIMA, também independente do horizonte, considerando as medidas acumuladas. Observando os erros absolutos por horizonte, redes recorrentes tipo Jordan conseguem ter uma boa performance a médio prazo (6 e 12 meses).

A figura 5.18 mostra os resíduos do modelo de rede Elman de pesos fixos para a série italiana. O gráfico correspondente à série italiana sugere uma heterocedasticidade dos resíduos. Na seção seguinte, será sugerido um novo modelo, com a finalidade de modelar essa heterocedasticidade.

5.4.4 Versão heterocédastica do modelo de redes neurais: o caso da Itália

O gráfico da série de inflação italiana sugere um comportamento atípico em relação às outras séries. Visualmente, é possível notar uma mudança na variabilidade dos dados. Por exemplo, no começo e no meio da série a variabilidade é muito maior do que no final. O gráfico dos resíduos para o modelo de rede Elman padrão (5.18) confirma essa suspeita.

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA (0; d ; 1) \times (0; 1; 1) ₁₂	3,04 $\times 10^{-7}$	$1,63 \times 10^{-5}$	$1,06 \times 10^{-5}$	8,23 $\times 10^{-6}$
AR linear (2 regressores)	$2,56 \times 10^{-7}$	$3,64 \times 10^{-5}$	$2,25 \times 10^{-5}$	$1,43 \times 10^{-5}$
AR linear (4 regressores)	$1,59 \times 10^{-7}$	$2,21 \times 10^{-5}$	$1,32 \times 10^{-5}$	$1,22 \times 10^{-5}$
Rede <i>feedforward</i> padrão	$4,09 \times 10^{-7}$	$6,58 \times 10^{-5}$	$3,73 \times 10^{-5}$	$4,28 \times 10^{-5}$
Rede <i>feedforward</i> com pesos variáveis	$1,24 \times 10^{-6}$	1,18 $\times 10^{-6}$	4,10 $\times 10^{-6}$	$2,64 \times 10^{-5}$
Rede Elman padrão	$2,04 \times 10^{-6}$	$7,22 \times 10^{-5}$	$3,83 \times 10^{-5}$	$2,32 \times 10^{-5}$
Rede Elman com pesos variáveis	$8,83 \times 10^{-4}$	$2,97 \times 10^{-4}$	$1,66 \times 10^{-4}$	$8,75 \times 10^{-5}$
Rede Jordan padrão	$1,68 \times 10^{-6}$	$1,90 \times 10^{-5}$	$1,07 \times 10^{-5}$	$2,57 \times 10^{-5}$
Rede Jordan com pesos variáveis	$3,71 \times 10^{-6}$	$3,48 \times 10^{-5}$	$1,87 \times 10^{-5}$	$2,98 \times 10^{-5}$
MAE				
Modelo	1 mês	3 meses	6 meses	12 meses
	5,51 $\times 10^{-4}$	$3,27 \times 10^{-3}$	$2,67 \times 10^{-3}$	2,33 $\times 10^{-3}$
AR linear (2 regressores)	$5,97 \times 10^{-4}$	$4,94 \times 10^{-3}$	$3,75 \times 10^{-3}$	$2,83 \times 10^{-3}$
AR linear (4 regressores)	$3,99 \times 10^{-4}$	$3,70 \times 10^{-3}$	$2,68 \times 10^{-3}$	$2,63 \times 10^{-3}$
Rede <i>feedforward</i> padrão	$6,39 \times 10^{-4}$	$6,19 \times 10^{-3}$	$4,36 \times 10^{-3}$	$4,95 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$1,11 \times 10^{-3}$	1,08 $\times 10^{-3}$	1,58 $\times 10^{-3}$	$3,21 \times 10^{-3}$
Rede Elman padrão	$1,43 \times 10^{-3}$	$7,34 \times 10^{-3}$	$4,52 \times 10^{-3}$	$3,41 \times 10^{-3}$
Rede Elman com pesos variáveis	$2,97 \times 10^{-2}$	$1,13 \times 10^{-2}$	$8,54 \times 10^{-3}$	$5,53 \times 10^{-3}$
Rede Jordan padrão	$1,30 \times 10^{-3}$	$3,03 \times 10^{-3}$	$2,20 \times 10^{-3}$	$3,56 \times 10^{-3}$
Rede Jordan com pesos variáveis	$1,93 \times 10^{-3}$	$4,02 \times 10^{-3}$	$2,75 \times 10^{-3}$	$3,91 \times 10^{-3}$

Tabela 5.17: Comparação de modelos: Inflação do Canadá (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARFIMA(0; d ; 1) \times (0; 1; 1) ₁₂	5,51 $\times 10^{-4}$	$6,35 \times 10^{-3}$	$2,98 \times 10^{-3}$	$4,36 \times 10^{-4}$
AR linear (2 regressores)	$5,97 \times 10^{-4}$	$9,08 \times 10^{-3}$	$4,12 \times 10^{-3}$	$2,97 \times 10^{-4}$
AR linear (4 regressores)	$3,99 \times 10^{-4}$	$7,45 \times 10^{-3}$	$3,37 \times 10^{-3}$	2,47 $\times 10^{-5}$
Rede <i>feedforward</i> padrão	$6,39 \times 10^{-4}$	$4,71 \times 10^{-3}$	$3,35 \times 10^{-3}$	$5,34 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$1,11 \times 10^{-3}$	$9,21 \times 10^{-4}$	4,69 $\times 10^{-4}$	$3,28 \times 10^{-4}$
Rede Elman padrão	$1,43 \times 10^{-3}$	$9,17 \times 10^{-3}$	$3,43 \times 10^{-3}$	$2,10 \times 10^{-3}$
Rede Elman com pesos variáveis	$2,97 \times 10^{-2}$	$2,64 \times 10^{-3}$	$7,21 \times 10^{-3}$	$1,34 \times 10^{-3}$
Rede Jordan padrão	$1,30 \times 10^{-3}$	3,60 $\times 10^{-4}$	$1,72 \times 10^{-3}$	$2,47 \times 10^{-3}$
Rede Jordan com pesos variáveis	$1,93 \times 10^{-3}$	$9,63 \times 10^{-5}$	$2,32 \times 10^{-3}$	$3,60 \times 10^{-3}$

Tabela 5.18: Comparação de modelos: Inflação do Canadá (erros absolutos por mês)

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA($0; d; 1 \times (0; 1; 1)_{12}$)	$6,05 \times 10^{-7}$	$3,86 \times 10^{-7}$	$8,21 \times 10^{-7}$	$2,02 \times 10^{-6}$
AR linear (4 regressores)	$4,18 \times 10^{-7}$	$3,40 \times 10^{-7}$	$6,31 \times 10^{-7}$	$1,66 \times 10^{-6}$
Rede <i>feedforward</i> padrão	$2,80 \times 10^{-7}$	$9,63 \times 10^{-7}$	$6,73 \times 10^{-7}$	$1,18 \times 10^{-6}$
Rede <i>feedforward</i> com pesos variáveis	$2,58 \times 10^{-7}$	$4,09 \times 10^{-7}$	$7,19 \times 10^{-7}$	$1,16 \times 10^{-6}$
Rede Elman padrão	$5,16 \times 10^{-8}$	$2,32 \times 10^{-7}$	$5,20 \times 10^{-7}$	$1,14 \times 10^{-6}$
Rede Elman com pesos variáveis	$5,22 \times 10^{-8}$	$6,64 \times 10^{-7}$	$6,00 \times 10^{-7}$	$1,20 \times 10^{-6}$
Rede Jordan padrão	$9,40 \times 10^{-6}$	$4,17 \times 10^{-6}$	$3,08 \times 10^{-6}$	$4,03 \times 10^{-6}$
Rede Jordan com pesos variáveis	$4,70 \times 10^{-6}$	$1,80 \times 10^{-6}$	$2,03 \times 10^{-6}$	$3,71 \times 10^{-6}$

Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA($0; d; 1 \times (0; 1; 1)_{12}$)	$7,78 \times 10^{-4}$	$5,78 \times 10^{-4}$	$7,37 \times 10^{-4}$	$1,19 \times 10^{-3}$
AR linear (4 regressores)	$6,46 \times 10^{-4}$	$5,76 \times 10^{-4}$	$6,41 \times 10^{-4}$	$1,05 \times 10^{-3}$
Rede <i>feedforward</i> padrão	$5,29 \times 10^{-4}$	$8,86 \times 10^{-4}$	$7,15 \times 10^{-4}$	$8,85 \times 10^{-4}$
Rede <i>feedforward</i> com pesos variáveis	$5,08 \times 10^{-4}$	$5,31 \times 10^{-4}$	$6,91 \times 10^{-4}$	$9,26 \times 10^{-4}$
Rede Elman padrão	$2,27 \times 10^{-4}$	$3,89 \times 10^{-4}$	$5,98 \times 10^{-4}$	$8,84 \times 10^{-4}$
Rede Elman com pesos variáveis	$2,29 \times 10^{-4}$	$6,66 \times 10^{-4}$	$6,76 \times 10^{-4}$	$9,33 \times 10^{-4}$
Rede Jordan padrão	$3,07 \times 10^{-3}$	$1,79 \times 10^{-3}$	$1,39 \times 10^{-3}$	$1,54 \times 10^{-3}$
Rede Jordan com pesos variáveis	$2,17 \times 10^{-3}$	$1,06 \times 10^{-3}$	$9,92 \times 10^{-4}$	$1,40 \times 10^{-3}$

Tabela 5.19: Comparação de modelos: Inflação da Itália (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARFIMA($0; d; 1 \times (0; 1; 1)_{12}$)	$7,78 \times 10^{-4}$	$6,98 \times 10^{-4}$	$7,23 \times 10^{-4}$	$7,75 \times 10^{-4}$
AR linear (4 regressores)	$6,46 \times 10^{-4}$	$6,38 \times 10^{-4}$	$4,31 \times 10^{-4}$	$6,62 \times 10^{-4}$
Rede <i>feedforward</i> padrão	$5,29 \times 10^{-4}$	$1,48 \times 10^{-3}$	$6,63 \times 10^{-4}$	$7,73 \times 10^{-4}$
Rede <i>feedforward</i> com pesos variáveis	$5,08 \times 10^{-4}$	$9,78 \times 10^{-4}$	$1,51 \times 10^{-3}$	$5,40 \times 10^{-4}$
Rede Elman padrão	$2,27 \times 10^{-4}$	$7,89 \times 10^{-4}$	$1,31 \times 10^{-3}$	$2,15 \times 10^{-4}$
Rede Elman com pesos variáveis	$2,29 \times 10^{-4}$	$1,32 \times 10^{-3}$	$1,01 \times 10^{-3}$	$4,20 \times 10^{-4}$
Rede Jordan padrão	$3,07 \times 10^{-3}$	$1,63 \times 10^{-3}$	$1,38 \times 10^{-4}$	$4,97 \times 10^{-4}$
Rede Jordan com pesos variáveis	$2,17 \times 10^{-3}$	$8,09 \times 10^{-4}$	$6,28 \times 10^{-5}$	$6,49 \times 10^{-4}$

Tabela 5.20: Comparação de modelos: Inflação da Itália (erros absolutos por mês)

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARFIMA (0; d ; 1) \times (0; 1; 1) ₁₂	$1,86 \times 10^{-6}$	$2,74 \times 10^{-6}$	$2,42 \times 10^{-6}$	$5,27 \times 10^{-6}$
AR linear (2 regressores)	$7,11 \times 10^{-6}$	$7,80 \times 10^{-6}$	$4,74 \times 10^{-6}$	$5,96 \times 10^{-6}$
AR linear (4 regressores)	$2,23 \times 10^{-6}$	$3,35 \times 10^{-6}$	$2,59 \times 10^{-6}$	$6,49 \times 10^{-6}$
Rede <i>feedforward</i> padrão	$1,23 \times 10^{-5}$	$1,22 \times 10^{-5}$	$7,00 \times 10^{-6}$	$6,95 \times 10^{-6}$
Rede <i>feedforward</i> com pesos variáveis	$1,47 \times 10^{-5}$	$1,21 \times 10^{-5}$	$7,07 \times 10^{-6}$	$6,91 \times 10^{-6}$
Rede Elman padrão	$3,30 \times 10^{-5}$	$2,18 \times 10^{-5}$	$1,21 \times 10^{-5}$	$9,38 \times 10^{-6}$
Rede Elman com pesos variáveis	$7,82 \times 10^{-3}$	$3,25 \times 10^{-3}$	$1,77 \times 10^{-3}$	$9,56 \times 10^{-4}$
Rede Jordan padrão	$5,18 \times 10^{-6}$	$4,85 \times 10^{-6}$	$4,25 \times 10^{-6}$	$1,40 \times 10^{-5}$
Rede Jordan com pesos variáveis	$1,15 \times 10^{-5}$	$9,12 \times 10^{-6}$	$5,72 \times 10^{-6}$	$1,45 \times 10^{-5}$
MAE				
Modelo	1 mês	3 meses	6 meses	12 meses
ARFIMA (0; d ; 1) \times (0; 1; 1) ₁₂	$1,36 \times 10^{-3}$	$1,53 \times 10^{-3}$	$1,44 \times 10^{-3}$	$1,83 \times 10^{-3}$
AR linear (2 regressores)	$2,67 \times 10^{-3}$	$2,50 \times 10^{-3}$	$1,88 \times 10^{-3}$	$1,84 \times 10^{-3}$
AR linear (4 regressores)	$1,49 \times 10^{-3}$	$1,47 \times 10^{-3}$	$1,35 \times 10^{-3}$	$1,91 \times 10^{-3}$
Rede <i>feedforward</i> padrão	$3,51 \times 10^{-3}$	$3,02 \times 10^{-3}$	$2,15 \times 10^{-3}$	$2,06 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$3,83 \times 10^{-3}$	$3,15 \times 10^{-3}$	$2,25 \times 10^{-3}$	$2,01 \times 10^{-3}$
Rede Elman padrão	$5,74 \times 10^{-3}$	$4,15 \times 10^{-3}$	$2,72 \times 10^{-3}$	$2,43 \times 10^{-3}$
Rede Elman com pesos variáveis	$8,85 \times 10^{-2}$	$4,98 \times 10^{-2}$	$3,34 \times 10^{-2}$	$2,25 \times 10^{-2}$
Rede Jordan padrão	$2,28 \times 10^{-3}$	$1,83 \times 10^{-3}$	$1,63 \times 10^{-3}$	$2,90 \times 10^{-3}$
Rede Jordan com pesos variáveis	$3,40 \times 10^{-3}$	$2,55 \times 10^{-3}$	$1,88 \times 10^{-3}$	$2,70 \times 10^{-3}$

Tabela 5.21: Comparação de modelos: Inflação da França (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARFIMA (0; d ; 1) \times (0; 1; 1) ₁₂	$1,36 \times 10^{-3}$	$8,38 \times 10^{-4}$	$1,76 \times 10^{-3}$	$2,56 \times 10^{-3}$
AR linear (2 regressores)	$2,67 \times 10^{-3}$	$8,93 \times 10^{-4}$	$1,53 \times 10^{-3}$	$2,47 \times 10^{-3}$
AR linear (4 regressores)	$1,49 \times 10^{-3}$	$1,14 \times 10^{-4}$	$1,29 \times 10^{-3}$	$2,53 \times 10^{-3}$
Rede <i>feedforward</i> padrão	$3,51 \times 10^{-3}$	$6,65 \times 10^{-4}$	$1,48 \times 10^{-3}$	$2,99 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$3,83 \times 10^{-3}$	$1,10 \times 10^{-3}$	$1,89 \times 10^{-3}$	$2,57 \times 10^{-3}$
Rede Elman padrão	$5,74 \times 10^{-3}$	$1,13 \times 10^{-3}$	$1,88 \times 10^{-3}$	$3,50 \times 10^{-3}$
Rede Elman com pesos variáveis	$8,85 \times 10^{-2}$	$2,51 \times 10^{-2}$	$1,78 \times 10^{-2}$	$7,40 \times 10^{-3}$
Rede Jordan padrão	$2,28 \times 10^{-3}$	$1,62 \times 10^{-4}$	$4,31 \times 10^{-4}$	$2,48 \times 10^{-3}$
Rede Jordan com pesos variáveis	$3,40 \times 10^{-3}$	$2,96 \times 10^{-4}$	$1,02 \times 10^{-3}$	$1,26 \times 10^{-3}$

Tabela 5.22: Comparação de modelos: Inflação da França (erros absolutos por mês)

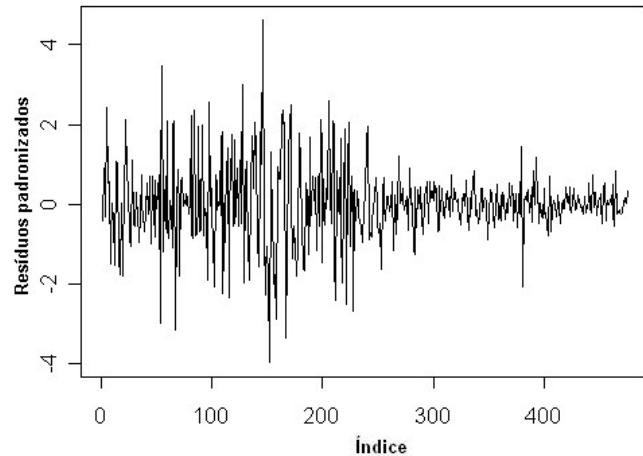


Figura 5.18: Resíduos padronizados: modelo de redes para inflação da Itália

No modelo escrito na forma de espaço de estado (como na seção 2.6.3), a saber

$$d(t) = f(\mathbf{w}(t), \mathbf{x}(t)) + v(t), \quad v(t) \sim N(0, V(t)) \quad (5.18)$$

$$\mathbf{w}(t) = \mathbf{w}(t-1) \quad (5.19)$$

essa variabilidade é representada pela variância do erro observacional $v(t)$. Isto nos dá uma outra maneira de se introduzir heterocedasticidade no modelo, que não através dos pesos variáveis, como já considerado. O usual é considerar essa variância constante, mas no caso da série italiana vamos considerar um modelo onde essa variância possa variar suavemente ao longo do tempo, assim como fizemos no caso de pesos variáveis. A idéia é a mesma: especificar a variância em cada instante do tempo através de um fator de desconto, que na prática é um valor entre 0,95 e 1.

Como visto na tabela 5.19, o modelo que tem melhor performance preditiva para a série italiana é a rede Elman com pesos fixos, seguida pela rede Elman com pesos variáveis. A tabela 5.23 compara estes modelos com os modelos sugeridos nesta seção: rede de pesos fixos e de pesos variáveis com variância heterogênea (com fator de desconto 0,99, escolhido na amostra de validação após uma avaliação de valores dentro do intervalo 0,95 a 0,99).

Os resultados mostram que a rede Elman padrão com variância heterogênea apresenta resultados muito ruins em comparação com os outros modelos. A rede Elman com pesos variáveis e variância heterogênea tem melhor performance a curtíssimo prazo (um mês), mas perde para a rede Elman padrão e tem performance muito semelhante à rede Elman com pesos variáveis nos outros períodos.

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
Rede Elman padrão	$5,16 \times 10^{-8}$	$2,32 \times 10^{-7}$	$5,20 \times 10^{-7}$	$1,14 \times 10^{-6}$
Rede Elman com pesos variáveis	$5,22 \times 10^{-8}$	$6,64 \times 10^{-7}$	$6,00 \times 10^{-7}$	$1,20 \times 10^{-6}$
Rede Elman padrão com variância heterogênea	$1,16 \times 10^{-5}$	$2,41 \times 10^{-5}$	$5,99 \times 10^{-5}$	$1,11 \times 10^{-6}$
Rede Elman com pesos variáveis com variância heterogênea	$1,38 \times 10^{-8}$	$4,27 \times 10^{-7}$	$6,04 \times 10^{-7}$	$1,20 \times 10^{-6}$
Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
Rede Elman padrão	$2,27 \times 10^{-4}$	$3,89 \times 10^{-4}$	$5,98 \times 10^{-4}$	$8,84 \times 10^{-4}$
Rede Elman com pesos variáveis	$2,29 \times 10^{-4}$	$6,66 \times 10^{-4}$	$6,76 \times 10^{-4}$	$9,33 \times 10^{-4}$
Rede Elman padrão com variância heterogênea	$3,40 \times 10^{-3}$	$4,80 \times 10^{-3}$	$7,25 \times 10^{-3}$	$1,00 \times 10^{-2}$
Rede Elman com pesos variáveis com variância heterogênea	$1,18 \times 10^{-4}$	$4,96 \times 10^{-4}$	$6,39 \times 10^{-4}$	$9,07 \times 10^{-4}$

Tabela 5.23: Modelos de redes com variância heterogênea: Inflação da Itália

5.5 Séries de inflação de memória curta: Alemanha e Japão

Esta seção abordará séries de inflação de dois países que possuem memória curta: Alemanha e Japão. Os dados foram obtidos pelo International Financial Statistics, tendo início em janeiro de 1957 e fim em março de 2005. No caso da Alemanha, o período que cobre de 1957 a 1990 corresponde à antiga Alemanha Ocidental.

5.5.1 Análise Descritiva

Os gráficos de autocorrelação simples de Alemanha e Japão (figuras 5.19(b) e 5.20(b)) não sugerem memória longa ao contrário das outras séries já estudadas. Parece não haver estacionaridade em média, mas devido às autocorrelações sazonais apenas.

5.5.2 Ajuste de modelos

Modelo ARMA

Não é necessária nenhuma diferenciação finita de baixa ordem, como visto através dos gráficos de autocorrelação simples e parcial. No entanto, as autocorrelações sazonais não decaem exponencialmente (teste Dickey-Fuller para raiz unitária sazonal: valor-p > 0,4 para os dois países). Portanto, uma diferenciação sazonal é tomada em ambas as séries.

As autocorrelações sazonais apresentam comportamento MA(1) (figuras 5.21 e 5.22). Algumas autocorrelações de baixa ordem são significantes, e em ambos os casos o modelo que melhor apresentou

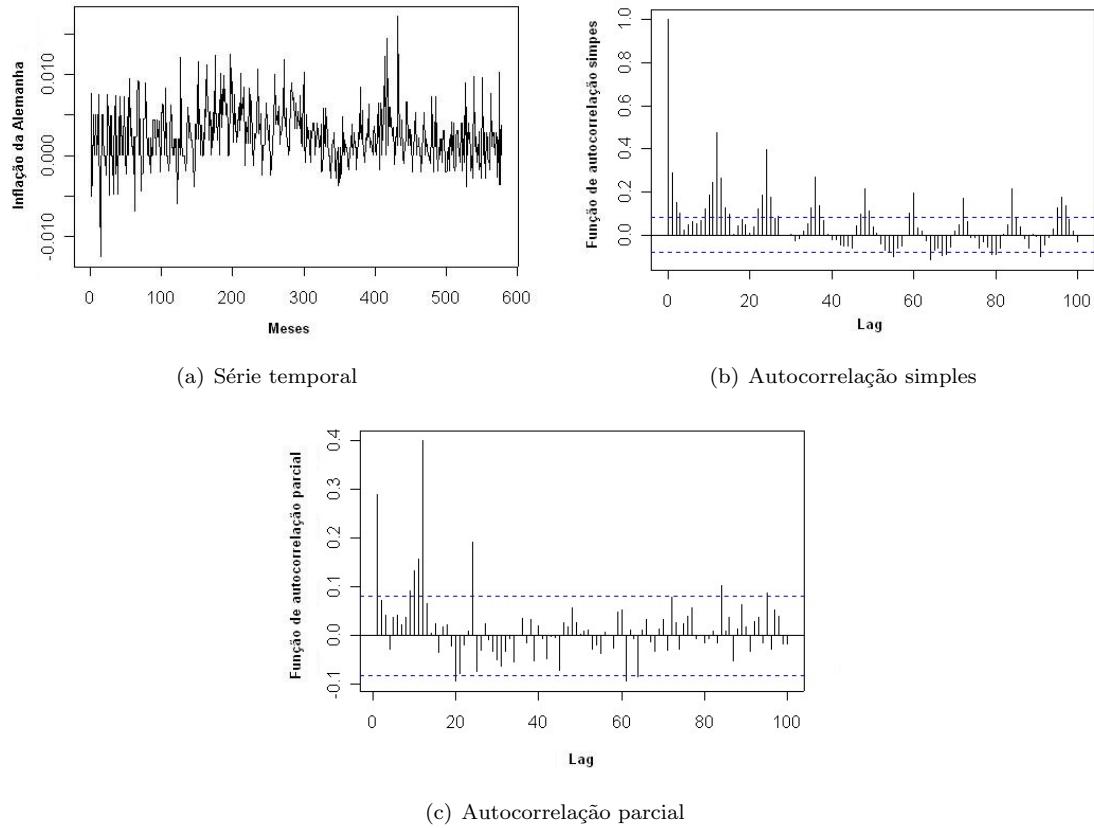


Figura 5.19: Gráficos da série de inflação da Alemanha

Alemanha
ARMA(1;1) \times (0;1;1)
Modelo: $(1 - B^{12})(1 - 0,9572B)x(t) = (1 + 0,7242B^{12})(1 + 0,8737B)a(t)$
Japão
ARMA(1;1) \times (0;1;1)
Modelo: $(1 - B^{12})(1 - 0,9216B)x(t) = (1 + 0,7988B^{12})(1 + 0,8144B)a(t)$

Tabela 5.24: Modelos ARFIMA para séries de inflação com memória curta

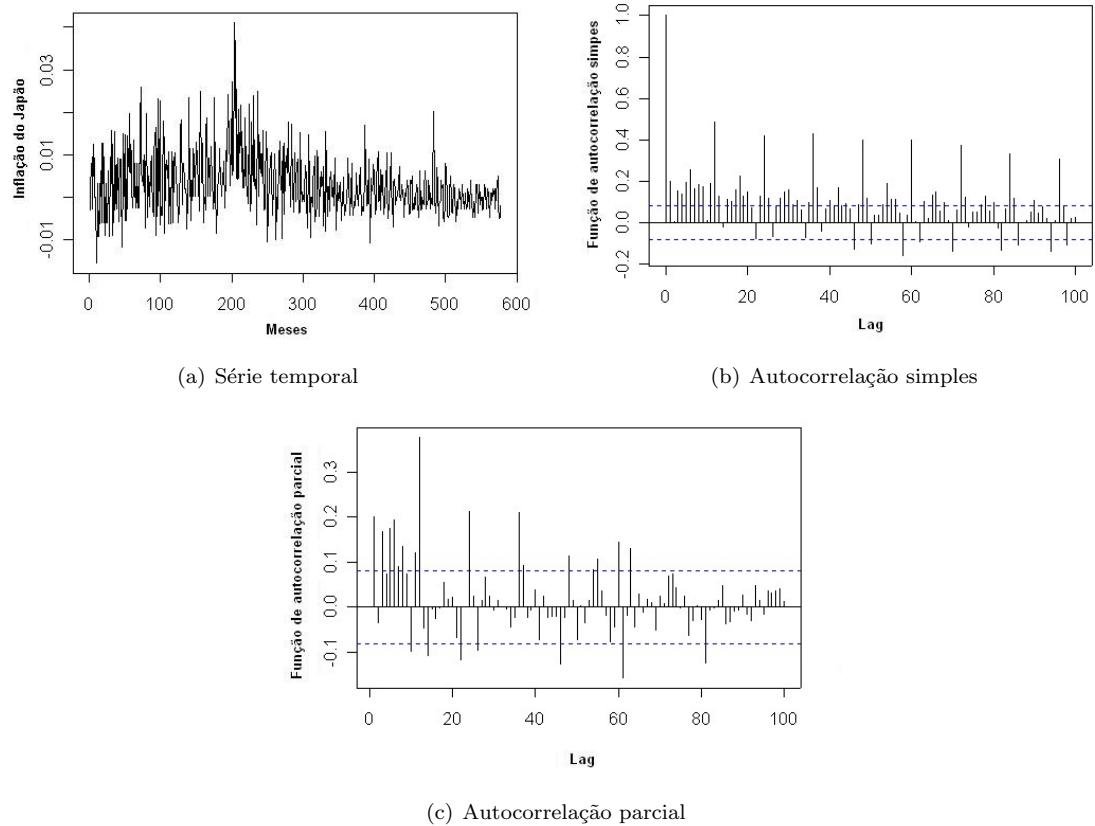


Figura 5.20: Gráficos da série de inflação do Japão

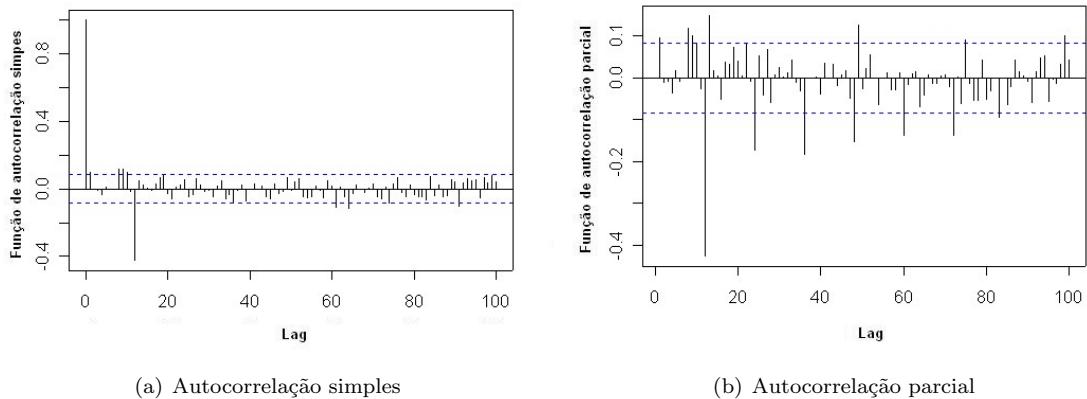


Figura 5.21: Gráficos da série de inflação da Alemanha diferenciada sazonalmente

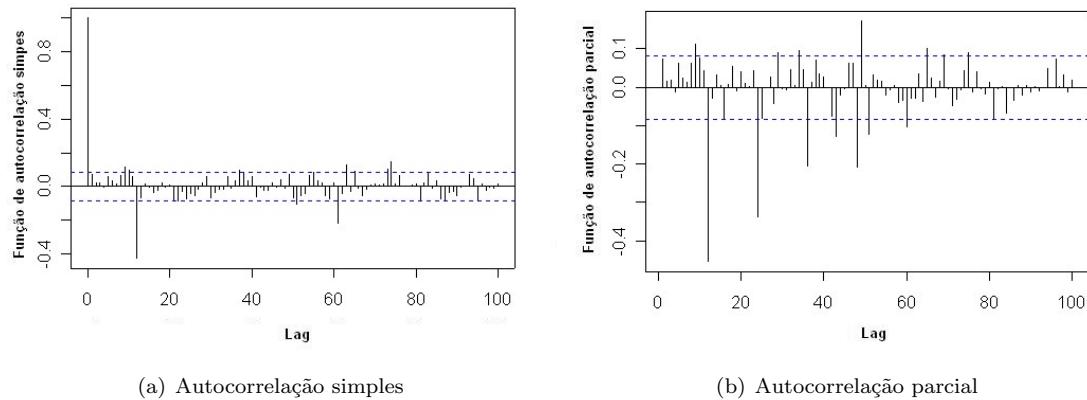


Figura 5.22: Gráficos da série de inflação do Japão diferenciada sazonalmente

Tipo de rede	Neurônios na camada intermediária	Regressores	
		Alemanha	Japão
<i>Feedforward</i>	3	2	2
Elman	3	4	4
Jordan	4	2	2
		Japão	
<i>Feedforward</i>	2	2	2
Elman	2	6	6
Jordan	5	6	6

Tabela 5.25: Modelos escolhidos para redes de pesos fixos

resultados em termos de resíduos aleatórios e não-correlacionados foram modelos ARMA(1;1). Os modelos estimados encontram-se na tabela 5.24.

Modelos de Redes Neurais

A única diferença desta análise para a análise das séries anteriores é que não foi feita uma diferenciação fracionária nas séries antes do ajuste das redes. Os dados dos dois países foram diferenciados apenas sazonalmente. As últimas 12 observações da série não foram consideradas para o ajuste (“amostra de teste”). As 24 observações anteriores são tomadas como “amostra de validação”, para “early stopping”. As observações restantes compõem a “amostra de treinamento”, para estimação dos pesos das redes.

Redes *feedforward* e recorrentes (tipos Elman e Jordan) são ajustadas, variando-se o número de neurônios na camada intermediária (2 a 5), o número de regressores (2, 4 e 6) e a inicialização dos pesos (10 inicializações diferentes). A comparação entre as 120 redes de cada um dos três tipos é feita através das medidas MSE e MAE na amostra de validação. As redes “ótimas” obtidas estão apresentadas na tabela 5.25.

Tipo de rede	Neurônios na camada intermediária	Regressores	Fator de desconto
Alemanha			
<i>Feedforward</i>	3	2	0,97
Elman	3	4	0,95
Jordan	4	2	0,96
Japão			
<i>Feedforward</i>	2	2	0,96
Elman	2	6	0,96
Jordan	5	6	0,96

Tabela 5.26: Modelos escolhidos para redes de pesos variáveis

Redes Neurais com pesos variáveis no tempo

Para o ajuste de redes neurais com pesos variáveis no tempo, as redes foram fixadas com o mesmo número de neurônios e de regressores encontrados na seção anterior. Os fatores de desconto (0,95 a 0,99) e as inicializações (10) foram variadas, com a performance avaliada na amostra de validação. As melhores redes encontram-se na tabela 5.26.

Modelo autoregressivo linear

Nesta seção, são ajustados modelos autoregressivos lineares para as séries de inflação de memória longa em questão com o intuito de construir uma versão linear dos modelos de redes ajustados, com as mesmas variáveis regressoras (defasagens da série), com o objetivo de mensurar a melhora oferecida por um modelo não-linear (redes neurais) em comparação uma versão linear correspondente do modelo.

As variáveis regressoras foram escolhidas com base nos modelos de redes neurais ajustados.

Alemanha:

$$(1 - B^{12})x(t) = -0,000008 - 0,0097x(t-1) - 0,459x(t-12) \quad (5.20)$$

$$\begin{aligned} (1 - B^{12})x(t) = & -0,000021 - 0,0142x(t-1) + 0,0612x(t-2) - \\ & - 0,573x(t-12) - 0,270x(t-24) \end{aligned} \quad (5.21)$$

Japão:

$$(1 - B^{12})x(t) = -0,000014 + 0,119x(t-1) - 0,437x(t-12) \quad (5.22)$$

$$\begin{aligned} (1 - B^{12})x(t) = & -0,000116 + 0,126x(t-1) + 0,0371x(t-2) + 0,0637x(t-3) - \\ & - 0,667x(t-12) - 0,432x(t-24) - 0,202x(t-36) \end{aligned} \quad (5.23)$$

Para maiores detalhes acerca dos ajustes dos modelos, ver apêndice A.5.

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARMA(0; 1) × (0; 1; 1)₁₂	$8,03 \times 10^{-6}$	$4,30 \times 10^{-6}$	$4,27 \times 10^{-6}$	$6,12 \times 10^{-6}$
AR linear (2 regressores)	$2,37 \times 10^{-5}$	$1,10 \times 10^{-5}$	$7,29 \times 10^{-6}$	$6,16 \times 10^{-6}$
AR linear (4 regressores)	$1,66 \times 10^{-5}$	$8,04 \times 10^{-6}$	$6,07 \times 10^{-6}$	$5,78 \times 10^{-6}$
Rede <i>feedforward</i> padrão	$2,98 \times 10^{-5}$	$1,26 \times 10^{-5}$	$9,10 \times 10^{-6}$	$7,04 \times 10^{-6}$
Rede <i>feedforward</i> com pesos variáveis	$2,80 \times 10^{-5}$	$1,58 \times 10^{-5}$	$9,27 \times 10^{-6}$	$7,74 \times 10^{-6}$
Rede Elman padrão	$3,72 \times 10^{-5}$	$2,03 \times 10^{-5}$	$1,13 \times 10^{-5}$	$9,07 \times 10^{-6}$
Rede Elman com pesos variáveis	$1,36 \times 10^{-5}$	$1,29 \times 10^{-5}$	$9,44 \times 10^{-6}$	$1,04 \times 10^{-5}$
Rede Jordan padrão	$8,12 \times 10^{-7}$	$2,17 \times 10^{-6}$	$5,96 \times 10^{-6}$	$7,22 \times 10^{-6}$
Rede Jordan com pesos variáveis	$4,17 \times 10^{-6}$	$7,01 \times 10^{-6}$	$7,50 \times 10^{-6}$	$8,24 \times 10^{-6}$

Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
ARMA(0; 1) × (0; 1; 1)₁₂	$2,83 \times 10^{-3}$	$1,97 \times 10^{-3}$	$1,84 \times 10^{-3}$	$2,18 \times 10^{-3}$
AR linear (2 regressores)	$4,87 \times 10^{-3}$	$3,04 \times 10^{-3}$	$2,36 \times 10^{-3}$	$2,11 \times 10^{-3}$
AR linear (4 regressores)	$4,07 \times 10^{-3}$	$2,65 \times 10^{-3}$	$2,24 \times 10^{-3}$	$2,09 \times 10^{-3}$
Rede <i>feedforward</i> padrão	$5,46 \times 10^{-3}$	$2,85 \times 10^{-3}$	$2,41 \times 10^{-3}$	$2,19 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$5,29 \times 10^{-3}$	$3,83 \times 10^{-3}$	$2,69 \times 10^{-3}$	$2,38 \times 10^{-3}$
Rede Elman padrão	$6,10 \times 10^{-3}$	$4,29 \times 10^{-3}$	$2,90 \times 10^{-3}$	$2,57 \times 10^{-3}$
Rede Elman com pesos variáveis	$3,69 \times 10^{-3}$	$3,35 \times 10^{-3}$	$2,62 \times 10^{-3}$	$2,51 \times 10^{-3}$
Rede Jordan padrão	$9,01 \times 10^{-4}$	$1,41 \times 10^{-3}$	$2,19 \times 10^{-3}$	$2,25 \times 10^{-3}$
Rede Jordan com pesos variáveis	$2,04 \times 10^{-3}$	$2,61 \times 10^{-3}$	$2,58 \times 10^{-3}$	$2,37 \times 10^{-3}$

Tabela 5.27: Comparação de modelos: Inflação da Alemanha (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARMA(0; 1) × (0; 1; 1) ₁₂	$2,83 \times 10^{-3}$	$1,82 \times 10^{-3}$	$2,19 \times 10^{-3}$	$1,50 \times 10^{-3}$
AR linear (2 regressores)	$4,87 \times 10^{-3}$	$1,74 \times 10^{-3}$	$1,81 \times 10^{-3}$	$6,63 \times 10^{-4}$
AR linear (4 regressores)	$4,07 \times 10^{-3}$	$2,01 \times 10^{-3}$	$1,96 \times 10^{-3}$	$6,44 \times 10^{-4}$
Rede <i>feedforward</i> padrão	$5,46 \times 10^{-3}$	$2,88 \times 10^{-4}$	$1,27 \times 10^{-3}$	$1,95 \times 10^{-3}$
Rede <i>feedforward</i> com pesos variáveis	$5,29 \times 10^{-3}$	$2,75 \times 10^{-3}$	$1,92 \times 10^{-3}$	$5,76 \times 10^{-6}$
Rede Elman padrão	$6,10 \times 10^{-3}$	$2,86 \times 10^{-3}$	$1,10 \times 10^{-3}$	$1,93 \times 10^{-4}$
Rede Elman com pesos variáveis	$3,69 \times 10^{-3}$	$4,72 \times 10^{-3}$	$4,00 \times 10^{-3}$	$2,20 \times 10^{-3}$
Rede Jordan padrão	$9,01 \times 10^{-4}$	$1,40 \times 10^{-3}$	$2,46 \times 10^{-3}$	$1,33 \times 10^{-3}$
Rede Jordan com pesos variáveis	$2,04 \times 10^{-3}$	$2,68 \times 10^{-3}$	$3,74 \times 10^{-3}$	$7,93 \times 10^{-5}$

Tabela 5.28: Comparação de modelos: Inflação da Alemanha (erros absolutos por mês)

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
ARMA(1; 1) × (0; 1; 1)₁₂	2,04 × 10⁻¹⁰	7,41 × 10 ⁻⁶	7,03 × 10 ⁻⁶	7,96 × 10⁻⁶
AR linear (2 regressores)	2,18 × 10 ⁻⁷	3,70 × 10⁻⁶	5,06 × 10⁻⁶	9,62 × 10 ⁻⁶
AR linear (6 regressores)	7,10 × 10 ⁻⁷	2,09 × 10⁻⁶	5,20 × 10⁻⁶	9,86 × 10 ⁻⁶
Rede <i>feedforward</i> padrão	1,02 × 10 ⁻⁶	3,97 × 10 ⁻⁶	6,48 × 10 ⁻⁶	1,07 × 10 ⁻⁵
Rede <i>feedforward</i> com pesos variáveis	2,02 × 10 ⁻⁶	5,42 × 10 ⁻⁶	6,99 × 10 ⁻⁶	1,11 × 10 ⁻⁵
Rede Elman padrão	6,20 × 10 ⁻⁶	3,10 × 10⁻⁶	5,69 × 10⁻⁶	1,11 × 10 ⁻⁵
Rede Elman com pesos variáveis	1,67 × 10 ⁻⁵	1,23 × 10 ⁻⁵	1,07 × 10 ⁻⁵	1,42 × 10 ⁻⁵
Rede Jordan padrão	2,82 × 10 ⁻⁶	5,44 × 10 ⁻⁶	6,12 × 10 ⁻⁶	1,05 × 10 ⁻⁵
Rede Jordan com pesos variáveis	1,67 × 10 ⁻⁵	1,71 × 10 ⁻³	2,35 × 10 ⁻³	2,62 × 10 ⁻³

Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
ARMA(1; 1) × (0; 1; 1)₁₂	1,43 × 10⁻⁵	1,89 × 10 ⁻³	1,88 × 10⁻³	2,17 × 10⁻³
AR linear (2 regressores)	4,67 × 10 ⁻⁴	1,62 × 10⁻³	1,77 × 10⁻³	2,43 × 10 ⁻³
AR linear (6 regressores)	8,42 × 10 ⁻⁴	1,21 × 10⁻³	1,67 × 10⁻³	2,49 × 10 ⁻³
Rede <i>feedforward</i> padrão	1,01 × 10 ⁻³	1,74 × 10 ⁻³	2,05 × 10 ⁻³	2,79 × 10 ⁻³
Rede <i>feedforward</i> com pesos variáveis	1,42 × 10 ⁻³	2,12 × 10 ⁻³	2,31 × 10 ⁻³	2,93 × 10 ⁻³
Rede Elman padrão	2,49 × 10 ⁻³	1,47 × 10⁻³	2,01 × 10 ⁻³	2,79 × 10 ⁻³
Rede Elman com pesos variáveis	4,09 × 10 ⁻³	3,33 × 10 ⁻³	2,99 × 10 ⁻³	3,39 × 10 ⁻³
Rede Jordan padrão	1,68 × 10 ⁻³	2,22 × 10 ⁻³	2,19 × 10 ⁻³	2,78 × 10 ⁻³
Rede Jordan com pesos variáveis	4,09 × 10 ⁻³	3,51 × 10 ⁻²	4,48 × 10 ⁻²	4,92 × 10 ⁻²

Tabela 5.29: Comparação de modelos: Inflação do Japão (erros acumulados por horizonte)

Modelo	Erro absoluto			
	1º mês	3º mês	6º mês	12º mês
ARMA(1; 1) × (0; 1; 1) ₁₂	1,43 × 10⁻⁵	1,07 × 10 ⁻³	1,31 × 10 ⁻³	1,90 × 10⁻⁴
AR linear (2 regressores)	4,67 × 10 ⁻⁴	1,43 × 10 ⁻³	1,35 × 10 ⁻³	6,41 × 10 ⁻⁴
AR linear (4 regressores)	8,42 × 10 ⁻⁴	4,78 × 10⁻⁴	1,64 × 10 ⁻³	5,11 × 10 ⁻⁴
Rede <i>feedforward</i> padrão	1,01 × 10 ⁻³	1,10 × 10 ⁻³	1,10 × 10⁻³	2,02 × 10 ⁻³
Rede <i>feedforward</i> com pesos variáveis	1,42 × 10 ⁻³	1,45 × 10 ⁻³	1,46 × 10 ⁻³	2,44 × 10 ⁻³
Rede Elman padrão	2,49 × 10 ⁻³	1,75 × 10 ⁻³	1,37 × 10 ⁻³	3,18 × 10 ⁻³
Rede Elman com pesos variáveis	4,09 × 10 ⁻³	1,73 × 10 ⁻³	1,47 × 10 ⁻³	2,79 × 10 ⁻³
Rede Jordan padrão	1,68 × 10 ⁻³	1,70 × 10 ⁻³	1,57 × 10 ⁻³	1,32 × 10 ⁻³
Rede Jordan com pesos variáveis	4,09 × 10 ⁻³	2,08 × 10 ⁻³	1,43 × 10 ⁻³	1,54 × 10 ⁻⁵

Tabela 5.30: Comparação de modelos: Inflação do Japão (erros absolutos por mês)

Modelo	MSE			
	1 mês	3 meses	6 meses	12 meses
Rede Elman padrão	$6,20 \times 10^{-6}$	$3,10 \times 10^{-6}$	$5,69 \times 10^{-6}$	$1,11 \times 10^{-5}$
Rede Elman com pesos variáveis	$1,67 \times 10^{-5}$	$1,23 \times 10^{-5}$	$1,07 \times 10^{-5}$	$1,42 \times 10^{-5}$
Rede Elman padrão com variância heterogênea	$3,62 \times 10^{-5}$	$4,04 \times 10^{-5}$	$2,77 \times 10^{-5}$	$3,06 \times 10^{-6}$
Rede Elman com pesos variáveis com variância heterogênea	$5,37 \times 10^{-6}$	$7,78 \times 10^{-6}$	$7,61 \times 10^{-6}$	$1,21 \times 10^{-5}$
Modelo	MAE			
	1 mês	3 meses	6 meses	12 meses
Rede Elman padrão	$2,49 \times 10^{-3}$	$1,47 \times 10^{-3}$	$2,01 \times 10^{-3}$	$2,79 \times 10^{-3}$
Rede Elman com pesos variáveis	$4,09 \times 10^{-3}$	$3,33 \times 10^{-3}$	$2,99 \times 10^{-3}$	$3,39 \times 10^{-3}$
Rede Elman padrão com variância heterogênea	$6,02 \times 10^{-3}$	$6,29 \times 10^{-3}$	$4,87 \times 10^{-3}$	$4,80 \times 10^{-2}$
Rede Elman com pesos variáveis com variância heterogênea	$2,31 \times 10^{-3}$	$2,61 \times 10^{-3}$	$2,53 \times 10^{-3}$	$3,04 \times 10^{-3}$

Tabela 5.31: Modelos de redes com variância heterogênea: Inflação do Japão

5.5.3 Resultados

Considerando que os últimos 12 dados da série não estivessem disponíveis, uma comparação preditiva é feita envolvendo as medidas MSE e MAE acumuladas calculadas para previsões de 1, 3, 6 e 12 meses a frente. Os valores se encontram na tabela 5.27 e 5.29, com os valores correspondentes às melhores previsões em cada mês em negrito. Uma comparação considerando apenas o erro absoluto em cada horizonte (1º, 3º, 6º e 12º meses a frente) em relação ao valor verdadeiro é apresentada nas tabelas 5.28 e 5.30.

Nenhum modelo é predominantemente melhor para essas séries de memória curta em todos os horizontes de previsão. No caso da Alemanha, a rede Jordan com pesos fixos consegue melhor previsão a curto prazo (1 e 3 meses). No médio prazo (6 e 12 meses), o modelo ARMA (medidas acumuladas) e redes recorrentes Elman e Jordan (erros absolutos por horizonte) têm melhores previsões. Na série japonesa, o modelo ARMA é melhor nos extremos (1 e 12 meses), e nos intervalos intermediários (3 e 6 meses) se destacam os modelos auto-regressivos lineares.

5.5.4 Versão heterocedástica do modelo de redes neurais: o caso do Japão

Similarmente à série de inflação italiana, o gráfico da série de inflação japonesa sugere uma mudança na variabilidade dos dados. Da mesma maneira aplicada na seção 5.4.4, é utilizado aqui um fator de desconto para a especificação da variância observacional, agora por suposição heterogênea.

Assim como para a série italiana, o modelo de redes que tem melhor performance preditiva para a série japonesa é a rede Elman com pesos fixos. A tabela 5.31 compara estes modelos com redes de pesos

fixos e de pesos variáveis com variância heterogênea (aqui também com fator de desconto 0,99, escolhido na amostra de validação após uma avaliação de valores dentro do intervalo 0,95 a 0,99).

A exemplo do que acontece com a série italiana, o modelo com pesos variáveis e variância heterogênea tem melhor performance a curtíssimo prazo (1 mês), mas perde para a rede Elman padrão nos outros períodos. Ao menos o modelo obtém melhor performance que a rede Elman de pesos variáveis em todos os períodos.

Capítulo 6

Conclusões

Os principais objetivos do trabalho foram alcançados. O estudo de redes neurais recorrentes para predição de séries temporais foi realizado, compreendendo sua fundamentação teórica, principais arquiteturas e algoritmos de aprendizagem e sua implementação computacional. A implementação prática não se limitou somente às arquiteturas e algoritmos presentes no software MATLAB, tendo sido realizada programação de outras arquiteturas e algoritmos de aprendizagem. A formulação unificada de redes neurais através da abordagem de espaço de estado e correspondente algoritmo baseado em filtro de Kalman trouxe as redes para o universo estatístico. Além disso, alguns modelos de redes baseados em arquiteturas conhecidas (*feedforward*, Elman e Jordan) foram aqui estendidas para o caso de pesos variáveis no tempo ou espaço, mostrando-se úteis tais extensões em termos de melhoria de performance preditiva.

O estudo comparativo da performance preditiva das redes neurais foi bem sucedido. Séries de memórias curta e longa foram utilizadas, tomando por base de comparação outros modelos tradicionais da literatura. Redes neurais recorrentes apresentaram bons resultados preditivos para as séries espaciais analisadas no capítulo 3, devido a não-linearidade presente na relação estudada (função de pedotransferência). No estudo de séries econômicas de inflação (capítulo 4), as redes recorrentes apresentaram bons resultados. Para comparação considerando os erros absolutos de previsões por horizonte, em quase todos os países as redes recorrentes apresentam melhores resultados que outros modelos em pelo menos um horizonte considerado. Em termos gerais, redes recorrentes, redes *feedforward* e modelos lineares se alternam como melhores modelos dependendo do país analisado e do horizonte de previsão, sugerindo que as séries de inflação de diferentes países apresentem diferenças quanto a sua estrutura intrínseca, com algumas necessitando de modelagem não-linear (com ou sem realimentação) para sua representação e outras não necessitando mais do que um modelo linear.

A avaliação da performance preditiva em séries espaciais (capítulo 3) sugeriu a construção de um novo modelo, redes neurais recorrentes com pesos variáveis no espaço, baseado na combinação da estrutura

de outros dois modelos (espaço de estado linear e redes neurais recorrentes). Esse novo modelo (e seu algoritmo de estimação proposto) foi novamente utilizado no capítulo 4, em séries econômicas, com alguns resultados promissores. Vale notar que no caso das séries de memória longa, as redes neurais propostas/consideradas atuam em seguida a um transformação prévia dos dados (pré-processamento baseado em estrutura ARFIMA($0, d, 0$)), sendo portanto um modelo híbrido.

O estudo também mostrou que o algoritmo de estimação proposto para redes neurais de pesos variáveis, baseado em extensão de segunda ordem do Filtro de Kalman, é bastante útil também para o caso particular de redes com pesos constantes, como uma alternativa ao algoritmo de Retropropagação Através do Tempo.

Os resultados dos capítulos 4 e 5 mostraram a real necessidade de se utilizar o algoritmo baseado em filtro de Kalman de segunda ordem. O filtro de Kalman estendido de 1^a ordem, mais difundido na literatura, não consegue lidar adequadamente com a não-linearidade presente em modelos de redes neurais. Foram também calculados intervalos de predição para algumas séries através de métodos analíticos e numéricos (bootstrap), complementando assim o algoritmo baseado no filtro de Kalman. Inclusive o software utilizado neste projeto, o MATLAB, não conta com nenhum método para cálculo de intervalos de predição para redes neurais.

Apêndice A

Algoritmo para treinamento de redes neurais com pesos variáveis e programas utilizados

A.1 Filtro de Kalman

O modelo linear dinâmico para uma seqüência de observações $y(t)$, $t = 1, 2, \dots$, é parametrizado por um vetor de estados $\mathbf{x}(t)$ de dimensão $p \times 1$ é definido por

$$y(t) = F'(t)\mathbf{x}(t) + v(t), \quad v(t) \sim N(0, V(t)) \quad (\text{A.1})$$

$$\mathbf{x}(t) = G(t)\mathbf{x}(t) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N(0, W(t)) \quad (\text{A.2})$$

com informação inicial $(\mathbf{x}(0)|D(0)) \sim N(\mathbf{m}(0), C(0))$, onde $D(t) = (D(t-1), y(t))$. Os erros observacionais e de evolução, $v(t)$ e $\mathbf{w}(t)$, são independentes entre si e no tempo, além de serem independentes de $(\mathbf{x}(0)|D(0))$ (Barbosa, E., 1991; West & Harrison, 1997).

Equivalentemente, o modelo pode ser definido por:

1. Verossimilhança: $(y(t)|\mathbf{x}(t)) \sim N(F^t(t)\mathbf{x}(t), V(t))$
2. Distribuição de evolução: $(\mathbf{x}(t)|\mathbf{x}(t-1)) \sim N(G(t)\mathbf{x}(t-1), W(t))$
3. Priori inicial: $(\mathbf{x}(0)|D(0)) \sim N(\mathbf{m}(0), C(0))$

O filtro de Kalman consiste no cálculo recursivo das expressões das distribuições a priori e a posteriori para o vetor de estados $\mathbf{x}(t)$, assim como a distribuição preditiva $(y(t)|D(t-1))$.

Considere $(\mathbf{x}(t-1)|D(t-1)) \sim N(\mathbf{m}(t-1), C(t-1))$.

1. Cálculo da distribuição a priori de $\mathbf{x}(t)$:

$$\begin{aligned} \mathbb{E}(\mathbf{x}(t)|D(t-1)) &= \mathbb{E}(G(t)\mathbf{x}(t-1) + \mathbf{w}(t)|D(t-1)) = G(t)\mathbb{E}(\mathbf{x}(t-1)|D(t-1)) = \\ &= G(t)\mathbf{m}(t-1) \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} \text{Var}(\mathbf{x}(t)|D(t-1)) &= \text{Var}(G(t)\mathbf{x}(t-1) + \mathbf{w}(t)|D(t-1)) = \\ &= \text{Var}(G(t)\mathbf{x}(t-1)|D(t-1)) + \text{Var}(\mathbf{w}(t)|D(t-1)) = \\ &= G(t)\text{Var}(\mathbf{x}(t-1)|D(t-1))G'(t) + \text{Var}(\mathbf{w}(t)|D(t-1)) = \\ &= G(t)C(t-1)G'(t) + W(t) \end{aligned} \quad (\text{A.4})$$

$$\Rightarrow (\mathbf{x}(t)|D(t-1)) \sim N(\mathbf{a}(t), R(t)) \quad (\text{A.5})$$

onde

$$\begin{aligned} \mathbf{a}(t) &= G(t)\mathbf{m}(t-1) \\ R(t) &= G(t)C(t-1)G'(t) + W(t) \end{aligned}$$

2. Cálculo da distribuição da previsão a um passo $y(t)|D(t-1)$:

$$\begin{aligned} \mathbb{E}(y(t)|D(t-1)) &= \mathbb{E}(F'(t)\mathbf{x}(t) + v(t)|D(t-1)) = \\ &= F'(t)\mathbb{E}(\mathbf{x}(t)|D(t-1)) = F'(t)G(t)\mathbf{m}(t-1) = F'(t)\mathbf{a}(t) \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} \text{Var}(y(t)|D(t-1)) &= \text{Var}(F'(t)\mathbf{x}(t) + v(t)|D(t-1)) = F'(t)\text{Var}(\mathbf{x}(t)|D(t-1))F(t) + \\ &+ \text{Var}(v(t)|D(t-1)) = F'(t)(G(t)C(t-1)G'(t) + W(t))F(t) + V(t) = \\ &= F'(t)R(t)F(t) + V(t) \end{aligned} \quad (\text{A.7})$$

$$\Rightarrow (y(t)|D(t-1)) \sim N(f(t), Q(t)) \quad (\text{A.8})$$

onde

$$\begin{aligned} f(t) &= F'(t)\mathbf{a}(t) \\ Q(t) &= F'(t)R(t)F(t) + V(t) \end{aligned}$$

3. Cálculo da distribuição a posteriori de $\mathbf{x}(t)$:

A distribuição conjunta de $(y(t), \mathbf{x}(t)|D(t-1))$ é dada por

$$\begin{pmatrix} y(t) \\ \mathbf{x}(t) \end{pmatrix} \sim N\left(\begin{pmatrix} f(t) \\ \mathbf{a}(t) \end{pmatrix}, \begin{pmatrix} \text{Var}(y(t)|D(t-1)) & \text{Cov}(y(t), \mathbf{x}(t)|D(t-1)) \\ \text{Cov}(\mathbf{x}(t), y(t)|D(t-1)) & \text{Var}(\mathbf{x}(t)|D(t-1)) \end{pmatrix}\right) \quad (\text{A.9})$$

onde

$$\begin{aligned} \text{Cov}(y(t), \mathbf{x}(t)|D(t-1)) &= \text{Cov}(F'(t)\mathbf{x}(t) + v(t), \mathbf{x}(t)|D(t-1)) = \\ &= F'(t)\text{Cov}(\mathbf{x}(t), \mathbf{x}(t)|D(t-1)) + \text{Cov}(v(t), \mathbf{x}(t)|D(t-1)) = \\ &= F'(t)\text{Var}(\mathbf{x}(t)|D(t-1)) + 0 = F'(t)R(t) \end{aligned} \quad (\text{A.10})$$

Como a matriz $R(t)$ é simétrica, é simples verificar que $\text{Cov}(\mathbf{x}(t), y(t)|D(t-1)) = R(t)F(t)$.

Lema 1 (*West e Harrison, 1997*) Por teoria normal multivariada, se $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)'$ e

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right) \quad (\text{A.11})$$

então

$$(\mathbf{x}_2|\mathbf{x}_1) \sim N(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \quad (\text{A.12})$$

O lema é válido para $(\mathbf{x}(t), y(t)|D(t-1))$, com $\mu_1 = f(t)$, $\mu_2 = a(t)$, $\Sigma_{11} = Q(t)$, $\Sigma_{12} = F'(t)R(t)$, $\Sigma_{21} = R(t)F(t)$ e $\Sigma_{22} = R(t)$. Portanto,

$$\begin{aligned} E(\mathbf{x}(t)|y(t), D(t-1)) &= E(\mathbf{x}(t)|D(t)) = \\ &= \mathbf{a}(t) + R(t)F(t)Q^{-1}(t)(y(t) - f(t)) = \mathbf{a}(t) + A(t)e(t) \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} \text{Var}(\mathbf{x}(t)|y(t), D(t-1)) &= \text{Var}(\mathbf{x}(t)|D(t)) = \\ &= R(t) - R(t)F(t)Q^{-1}(t)F'(t)R(t) = \\ &= R(t) - A(t)Q(t)A'(t) \end{aligned} \quad (\text{A.14})$$

onde $A(t) = R(t)F(t)Q^{-1}(t)$; $e(t) = y(t) - f(t)$.

$$\Rightarrow (\mathbf{x}(t)|D(t)) \sim N(\mathbf{m}(t), C(t)) \quad (\text{A.15})$$

onde

$$\begin{aligned} \mathbf{m}(t) &= \mathbf{a}(t) + A(t)e(t) \\ C(t) &= R(t) - A(t)Q(t)A'(t) \end{aligned}$$

As equações acima formam o filtro de Kalman.

A.2 Programa em MATLAB: treinamento de redes recorrentes com pesos variáveis ao longo do espaço/tempo

Programa utilizado para treinamento de redes recorrentes para dados de solo. Arquitetura de rede utilizada: rede Elman com dois neurônios na camada intermediária e uma saída. Algoritmo de aprendizagem baseado em filtro de Kalman estendido de segunda ordem (o algoritmo encontra-se na seção 4.4.1).

Variáveis de entrada do programa:

- peso: valor inicial para o vetor de pesos da rede
- x1: primeira variável de entrada (carbono)
- x2: segunda variável de entrada (nitrogênio defasado)
- xpredicao: valores do carbono para o conjunto de predição
- ypredicao: valores desejados no nitrogênio no conjunto de predicao
- C: valor inicial para a variância a priori C
- s e n: valores iniciais de s e n para análise conjugada quando a variância do erro é desconhecida
- delta: fator de desconto (número entre 0 e 1, na prática acima de 0,95)

```
function kalman(peso,x1,x2,y,xpredicao,ypredicao,C,s,ene,delta);
n=length(x1);
x1p(1)=0;
x2p(1)=0;
yp(1)=0;
for i=2:n+1
x1p(i)=x1(i-1);
x2p(i)=x2(i-1);
yp(i)=y(i-1);
end
for i=1:13
hes(i,i)=0;
end
neuronio1saida(1)=0;
neuronio2saida(1)=0;
for i=2:n+1
for j=2:i
excitacao1(j)=peso(1)+peso(2)*x1p(j)+peso(3)*x2p(j)+peso(4)*neuronio1saida(j-1)+peso(5)*neuronio2saida(j-1);
excitacao2(j)=peso(6)+peso(7)*x1p(j)+peso(8)*x2p(j)+peso(9)*neuronio1saida(j-1)+peso(10)*neuronio2saida(j-1);
neuronio1saida(j)=1/(1+exp(-excitacao1(j)));
neuronio2saida(j)=1/(1+exp(-excitacao2(j)));
saidarede(j)=peso(11)+peso(12)*neuronio1saida(j)+peso(13)*neuronio2saida(j);
end
```

```
%calculo do jacobiano
jacob(1)=peso(12)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2);
jacob(2)=peso(12)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*x1p(i);
jacob(3)=peso(12)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*x2p(i);
jacob(4)=peso(12)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*neuronio1saida(i-1);
jacob(5)=peso(12)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*neuronio2saida(i-1);
jacob(6)=peso(13)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2);
jacob(7)=peso(13)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*x1p(i);
jacob(8)=peso(13)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*x2p(i);
jacob(9)=peso(13)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*neuronio1saida(i-1);
jacob(10)=peso(13)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*neuronio2saida(i-1);
jacob(11)=1;
jacob(12)=neuronio1saida(i);
jacob(13)=neuronio2saida(i);
for eme=1:13
hes(eme,eme)=0;
end
%calculo da hessiana
base1=peso(12)*(((exp(-excitacao1(i)))*((exp(-excitacao1(i))-1)/((1+exp(-excitacao1(i)))^3)));
hes(1,1)=base1;
hes(1,2)=base1*x1p(i);
hes(1,3)=base1*x2p(i);
hes(1,4)=base1*neuronio1saida(i-1);
hes(1,5)=base1*neuronio2saida(i-1);
hes(1,6)=0;
hes(1,7)=0;
hes(1,8)=0;
hes(1,9)=0;
hes(1,10)=0;
hes(1,11)=0;
hes(1,12)=(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2);
hes(1,13)=0;
for ele=2:13
hes(2,ele)=hes(1,ele)*x1p(i);
end
for ele=3:13
hes(3,ele)=hes(1,ele)*x2p(i);
end
for ele=4:13
hes(4,ele)=hes(1,ele)*neuronio1saida(i-1);
end
for ele=5:13
hes(5,ele)=hes(1,ele)*neuronio2saida(i-1);
end
base2=peso(13)*(((exp(-excitacao2(i)))*((exp(-excitacao2(i))-1)/((1+exp(-excitacao2(i)))^3)));
hes(6,6)=base2;
hes(6,7)=base2*x1p(i);
hes(6,8)=base2*x2p(i);
hes(6,9)=base2*neuronio1saida(i-1);
hes(6,10)=base2*neuronio2saida(i-1);
hes(6,11)=0;
hes(6,12)=0;
hes(6,13)=(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2);
for ele=7:13
hes(7,ele)=hes(6,ele)*x1p(i);
```

```

end
for ele=8:13
hes(8,ele)=hes(6,ele)*x2p(i);
end
for ele=9:13
hes(9,ele)=hes(6,ele)*neuronio1saida(i-1);
end
for ele=7:13
hes(10,ele)=hes(6,ele)*neuronio2saida(i-1);
end
hes(11,11)=0;
hes(11,12)=0;
hes(11,13)=0;
hes(12,12)=0;
hes(12,13)=0;
hes(13,13)=0;
for m=1:13
for g=1:13
if m ==g
hes(g,m)=hes(m,g);
end
end
end
%filtro de kalman
R=C/delta;
eneanterior=ene;
ene=ene+1;
santerior=s;
Q=jacob*R*jacob'+((trace((hes*R)^2))/2)+santerior;
erro(i)=yp(i)-saidarede(i)-(trace(hes*R))/2;
s=santerior*((eneanterior+((erro(i))^2)/Q))/ene;
ganho=R*jacob'*inv(Q);
peso=peso+ganho*(erro(i));
C=(R-ganho*Q*ganho')*(s/santerior);
end
for i=2:n+1
excitacao1(i)=peso(1)+peso(2)*x1p(i)+peso(3)*x2p(i)+peso(4)*neuronio1saida(i-1)+peso(5)*neuronio2saida(i-1);
excitacao2(i)=peso(6)+peso(7)*x1p(i)+peso(8)*x2p(i)+peso(9)*neuronio1saida(i-1)+peso(10)*neuronio2saida(i-1);
neuronio1saida(i)=1/(1+exp(-excitacao1(i)));
neuronio2saida(i)=1/(1+exp(-excitacao2(i)));
saidarede(i)=peso(11)+peso(12)*neuronio1saida(i)+peso(13)*neuronio2saida(i);
end
%calculo das predicoes
previsao(1)=saidarede(n+1);
neuronio1s(1)=neuronio1saida(n+1);
neuronio2s(1)=neuronio2saida(n+1);
for i=2:11
xpredicao1(i)=xpredicao(i-1);
end
for i=2:11
exc1(i)=peso(1)+peso(2)*xpredicao1(i)+peso(3)*previsao(i-1)+peso(4)*neuronio1s(i-1)+peso(5)*neuronio2s(i-1);
exc2(i)=peso(6)+peso(7)*xpredicao1(i)+peso(8)*previsao(i-1)+peso(9)*neuronio1s(i-1)+peso(10)*neuronio2s(i-1);
neuronio1s(i)=1/(1+exp(-exc1(i)));
neuronio2s(i)=1/(1+exp(-exc2(i)));
previsao(i)=peso(11)+peso(12)*neuronio1s(i)+peso(13)*neuronio2s(i);

```

```

end
for i=2:11
    previsaocerto(i-1)=previsao(i);
end
%calculo das medidas MSE, MAE e MAPE
erros2=(ypredicao-previsaocerto').^2
MSE=sum(erros2)/10
erros1=abs(ypredicao-previsaocerto')
MAE=sum(erros1)/10
for i=1:10
    erroscorr(i)=erros1(i)/previsaocerto(i);
end
MAPE=sum(erroscorr)/10

```

A.3 Programa em MATLAB: treinamento de redes *feedforward* com pesos variáveis ao longo do espaço/tempo

Programa utilizado para treinamento de redes *feedforward* para dados de inflação, em particular para a série do Reino Unido. Arquitetura de rede utilizada: rede *feedforward* com cinco neurônios na camada intermediária e uma saída, com quatro entradas (duas defasagens simples e duas defasagens sazonais da série). Algoritmo de aprendizagem baseado em filtro de Kalman estendido de segunda ordem (o algoritmo encontra-se na seção 4.4.1).

```

function (KALMANfator095,KALMANfator096,KALMANfator097,KALMANfator098,KALMANfator099,
KALMANfator100,KALMANinicializacao095,KALMANinicializacao096,KALMANinicializacao097,
KALMANinicializacao098,KALMANinicializacao099,KALMANinicializacao100)=
kalmanfeed4regr5neur(x,y,xpredicao,ypredicao);
tic
deltacraft=6
delta=1-deltacraft/100+0.01
for contador=1:10
    contador
    if deltacraft==1
        for i=1:31
            KALMANinicializacao100(i,contador)=unifrnd(-1,1);
        end
        for i=1:31
            peso(i,1)=KALMANinicializacao100(i,contador);
        end
    end
    if deltacraft==2
        for i=1:31
            KALMANinicializacao099(i,contador)=unifrnd(-1,1);
        end
        for i=1:31
            peso(i,1)=KALMANinicializacao099(i,contador);
        end
    end
    if deltacraft==3

```

```
for i=1:31
KALMANinicializacao098(i,contador)=unifrnd(-1,1);
end
for i=1:31
peso(i,1)=KALMANinicializacao098(i,contador);
end
end
if deltacraft==4
for i=1:31
KALMANinicializacao097(i,contador)=unifrnd(-1,1);
end
for i=1:31
peso(i,1)=KALMANinicializacao097(i,contador);
end
end
if deltacraft==5
for i=1:31
KALMANinicializacao096(i,contador)=unifrnd(-1,1);
end
for i=1:31
peso(i,1)=KALMANinicializacao096(i,contador);
end
end
if deltacraft==6
for i=1:31
KALMANinicializacao095(i,contador)=unifrnd(-1,1);
end
for i=1:31
peso(i,1)=KALMANinicializacao095(i,contador);
end
end
for i=1:31
C(i,i)=1000;
end
s=1000;
ene=0;
n=length(x);
Q=0;
for i=1:31
ganho(i,1)=0;
end
for i=1:n
xp(i,1)=x(1,i);
xp(i,2)=x(2,i);
xp(i,3)=x(3,i);
xp(i,4)=x(4,i);
yp(i)=y(i);
end
for i=1:31
hes(i,i)=0;
end
for i=1:31
jacob(i)=0;
end
for i=1:n
```

```

excitacao1(i)=peso(1)+peso(2)*xp(i,1)+peso(3)*xp(i,2)+peso(4)*xp(i,3)+peso(5)*xp(i,4);
excitacao2(i)=peso(6)+peso(7)*xp(i,1)+peso(8)*xp(i,2)+peso(9)*xp(i,3)+peso(10)*xp(i,4);
excitacao3(i)=peso(11)+peso(12)*xp(i,1)+peso(13)*xp(i,2)+peso(14)*xp(i,3)+peso(15)*xp(i,4);
excitacao4(i)=peso(16)+peso(17)*xp(i,1)+peso(18)*xp(i,2)+peso(19)*xp(i,3)+peso(20)*xp(i,4);
excitacao5(i)=peso(21)+peso(22)*xp(i,1)+peso(23)*xp(i,2)+peso(24)*xp(i,3)+peso(25)*xp(i,4);
neuronio1saida(i)=1/(1+exp(-excitacao1(i)));
neuronio2saida(i)=1/(1+exp(-excitacao2(i)));
neuronio3saida(i)=1/(1+exp(-excitacao3(i)));
neuronio4saida(i)=1/(1+exp(-excitacao4(i)));
neuronio5saida(i)=1/(1+exp(-excitacao5(i)));
saidarede(i)=peso(26)+peso(27)*neuronio1saida(i)+peso(28)*neuronio2saida(i)+peso(29)*neuronio3saida(i)+peso(30)*neuronio4saida(i)+peso(31)*neuronio5saida(i);
jacob(1)=peso(27)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2);
jacob(2)=peso(27)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*xp(i,1);
jacob(3)=peso(27)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*xp(i,2);
jacob(4)=peso(27)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*xp(i,3);
jacob(5)=peso(27)*(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2)*xp(i,4);
jacob(6)=peso(28)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2);
jacob(7)=peso(28)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*xp(i,1);
jacob(8)=peso(28)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*xp(i,2);
jacob(9)=peso(28)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*xp(i,3);
jacob(10)=peso(28)*(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2)*xp(i,4);
jacob(11)=peso(29)*(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2);
jacob(12)=peso(29)*(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2)*xp(i,1);
jacob(13)=peso(29)*(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2)*xp(i,2);
jacob(14)=peso(29)*(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2)*xp(i,3);
jacob(15)=peso(29)*(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2)*xp(i,4);
jacob(16)=peso(30)*(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2);
jacob(17)=peso(30)*(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2)*xp(i,1);
jacob(18)=peso(30)*(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2)*xp(i,2);
jacob(19)=peso(30)*(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2)*xp(i,3);
jacob(20)=peso(30)*(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2)*xp(i,4);
jacob(21)=peso(31)*(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2);
jacob(22)=peso(31)*(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2)*xp(i,1);
jacob(23)=peso(31)*(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2)*xp(i,2);
jacob(24)=peso(31)*(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2)*xp(i,3);
jacob(25)=peso(31)*(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2)*xp(i,4);
jacob(26)=1;
jacob(27)=neuronio1saida(i);
jacob(28)=neuronio2saida(i);
jacob(29)=neuronio3saida(i);
jacob(30)=neuronio4saida(i);
jacob(31)=neuronio5saida(i);
%hessiana
for eme=1:31
hes(eme,eme)=0;
end
%primeira linha
base1=peso(27)*(((exp(-excitacao1(i)))*((exp(-excitacao1(i))-1))/((1+exp(-excitacao1(i)))^3)));
hes(1,1)=base1;
hes(1,2)=base1*xp(i,1);
hes(1,3)=base1*xp(i,2);
hes(1,4)=base1*xp(i,3);
hes(1,5)=base1*xp(i,4);
for hesin=6:26

```

```

hes(1,hesin)=0;
end
hes(1,27)=(exp(-excitacao1(i))/(1+exp(-excitacao1(i)))^2);
for hesin=28:31
hes(1,hesin)=0;
end
%segunda linha
for ele=2:31
hes(2,ele)=hes(1,ele)*xp(i,1);
end for ele=3:31 hes(3,ele)=hes(1,ele)*xp(i,2);
end
for ele=4:31
hes(4,ele)=hes(1,ele)*xp(i,3);
end
for ele=5:31
hes(5,ele)=hes(1,ele)*xp(i,4);
end
base2=peso(28)*(((exp(-excitacao2(i)))*((exp(-excitacao2(i))-1))/((1+exp(-excitacao2(i)))^3)));
hes(6,6)=base2;
hes(6,7)=base2*xp(i,1);
hes(6,8)=base2*xp(i,2);
hes(6,9)=base2*xp(i,3);
hes(6,10)=base2*xp(i,4);
for hesin=11:27
hes(6,hesin)=0;
end
hes(6,28)=(exp(-excitacao2(i))/(1+exp(-excitacao2(i)))^2);
for hesin=29:31
hes(6,hesin)=0;
end
%terceira linha
for ele=7:31
hes(7,ele)=hes(6,ele)*xp(i,1);
end
for ele=8:31
hes(8,ele)=hes(6,ele)*xp(i,2);
end
for ele=9:31
hes(9,ele)=hes(6,ele)*xp(i,3);
end
for ele=10:31
hes(10,ele)=hes(6,ele)*xp(i,4);
end
base3=peso(29)*(((exp(-excitacao3(i)))*((exp(-excitacao3(i))-1))/((1+exp(-excitacao3(i)))^3)));
hes(11,11)=base3;
hes(11,12)=base3*xp(i,1);
hes(11,13)=base3*xp(i,2);
hes(11,14)=base3*xp(i,3);
hes(11,15)=base3*xp(i,4);
for hesin=16:28
hes(11,hesin)=0;
end
hes(11,29)=(exp(-excitacao3(i))/(1+exp(-excitacao3(i)))^2);
for hesin=30:31
hes(11,hesin)=0;

```

```

end
%quarta linha
for ele=12:31
hes(12,ele)=hes(11,ele)*xp(i,1);
end
for ele=13:31 hes(13,ele)=hes(11,ele)*xp(i,2);
end
for ele=14:31
hes(14,ele)=hes(11,ele)*xp(i,3);
end
for ele=15:31
hes(15,ele)=hes(11,ele)*xp(i,4);
end
base4=peso(30)*(((exp(-excitacao4(i)))*((exp(-excitacao4(i))-1))/((1+exp(-excitacao4(i)))^3)));
hes(16,16)=base4;
hes(16,17)=base4*xp(i,1);
hes(16,18)=base4*xp(i,2);
hes(16,19)=base4*xp(i,3);
hes(16,20)=base4*xp(i,4);
for hesin=20:29
hes(16,hesin)=0;
end
hes(16,30)=(exp(-excitacao4(i))/(1+exp(-excitacao4(i)))^2);
hes(16,31)=0;
%quinta linha
for ele=17:31
hes(17,ele)=hes(16,ele)*xp(i,1);
end
for ele=18:31 hes(18,ele)=hes(16,ele)*xp(i,2);
end
for ele=19:31
hes(19,ele)=hes(16,ele)*xp(i,3);
end
for ele=20:31
hes(20,ele)=hes(16,ele)*xp(i,4);
end
base5=peso(31)*(((exp(-excitacao5(i)))*((exp(-excitacao5(i))-1))/((1+exp(-excitacao5(i)))^3)));
hes(21,21)=base5;
hes(21,22)=base5*xp(i,1);
hes(21,23)=base5*xp(i,2);
hes(21,24)=base5*xp(i,3);
hes(21,25)=base5*xp(i,4);
for hesin=26:30
hes(21,hesin)=0;
end
hes(21,31)=(exp(-excitacao5(i))/(1+exp(-excitacao5(i)))^2);
for ele=22:31
hes(22,ele)=hes(21,ele)*xp(i,1);
end
for ele=23:31
hes(23,ele)=hes(21,ele)*xp(i,2);
end
for ele=24:31
hes(24,ele)=hes(21,ele)*xp(i,3);
end

```

```

for ele=25:31
hes(25,ele)=hes(21,ele)*xp(i,4);
end
for hesin=26:31
hes(26,hesin)=0;
end
for hesin=27:31
hes(27,hesin)=0;
end
for hesin=28:31
hes(28,hesin)=0;
end
for hesin=29:31
hes(29,hesin)=0;
end
for hesin=30:31
hes(30,hesin)=0;
end
hes(31,31)=0;
for m=1:31
for g=1:31
if m ==g
hes(g,m)=hes(m,g);
end
end
end
hes;
R=C/delta;
eneanterior=ene;
ene=ene+1;
santerior=s;
Q=jacob*R*(jacob')+((trace((hes*R)^ 2))/2)+santerior;
erro(i)=yp(i)-saidarede(i)-(trace(hes*R))/2;
;
s=santerior*((eneanterior+((erro(i))^ 2)/Q))/ene;
ganho=R*(jacob')*inv(Q);
peso=peso+ganho*(erro(i));
C=(R-ganho*Q*ganho')*(s/santerior);
end
peso;
saidarede;
previsao(1)=y(n-1);
previsao(2)=y(n);
for i=2:13
exc1(i)=peso(1)+peso(2)*previsao(i)+peso(3)*previsao(i-1)+peso(4)*y(n+i-13)+peso(5)*y(n+i-25);
exc2(i)=peso(6)+peso(7)*previsao(i)+peso(8)*previsao(i-1)+peso(9)*y(n+i-13)+peso(10)*y(n+i-25);
exc3(i)=peso(11)+peso(12)*previsao(i)+peso(13)*previsao(i-1)+peso(14)*y(n+i-13)+peso(15)*y(n+i-25);
exc4(i)=peso(16)+peso(17)*previsao(i)+peso(18)*previsao(i-1)+peso(19)*y(n+i-13)+peso(20)*y(n+i-25);
exc5(i)=peso(21)+peso(22)*previsao(i)+peso(23)*previsao(i-1)+peso(24)*y(n+i-13)+peso(25)*y(n+i-25);
neuronio1s(i)=1/(1+exp(-exc1(i)));
neuronio2s(i)=1/(1+exp(-exc2(i)));
neuronio3s(i)=1/(1+exp(-exc3(i)));
neuronio4s(i)=1/(1+exp(-exc4(i)));
neuronio5s(i)=1/(1+exp(-exc5(i)));
previsao(i+1)=peso(26)+peso(27)*neuronio1s(i)+peso(28)*neuronio2s(i)+peso(29)*neuronio3s(i) +

```

```

peso(30)*neuronio4s(i)+peso(31)*neuronio5s(i);
end
for i=14:25
exc1(i)=peso(1)+peso(2)*previsao(i)+peso(3)*previsao(i-1)+peso(4)*previsao(i-12)+peso(5)*y(n+i-25);
exc2(i)=peso(6)+peso(7)*previsao(i)+peso(8)*previsao(i-1)+peso(9)*previsao(i-12)+peso(10)*y(n+i-25);
exc3(i)=peso(11)+peso(12)*previsao(i)+peso(13)*previsao(i-1)+peso(14)*previsao(i-12)+peso(15)*y(n+i-25);
exc4(i)=peso(16)+peso(17)*previsao(i)+peso(18)*previsao(i-1)+peso(19)*previsao(i-12)+peso(20)*y(n+i-25);
exc5(i)=peso(21)+peso(22)*previsao(i)+peso(23)*previsao(i-1)+peso(24)*previsao(i-12)+peso(25)*y(n+i-25);
neuronio1s(i)=1/(1+exp(-exc1(i)));
neuronio2s(i)=1/(1+exp(-exc2(i)));
neuronio3s(i)=1/(1+exp(-exc3(i)));
neuronio4s(i)=1/(1+exp(-exc4(i)));
neuronio5s(i)=1/(1+exp(-exc5(i)));
previsao(i+1)=peso(26)+peso(27)*neuronio1s(i)+peso(28)*neuronio2s(i)+peso(29)*neuronio3s(i)+
peso(30)*neuronio4s(i)+peso(31)*neuronio5s(i);
end
for i=3:26
previsaoocerto(i-2,contador)=previsao(i);
end
end
if deltacraft==1
KALMANfator100=previsaoocerto;
end
if deltacraft==2
KALMANfator099=previsaoocerto;
end
if deltacraft==3
KALMANfator098=previsaoocerto;
end
if deltacraft==4
KALMANfator097=previsaoocerto;
end
if deltacraft==5
KALMANfator096=previsaoocerto;
end
if deltacraft==6
KALMANfator095=previsaoocerto;
end
end
tempo=toc

```

A.4 Programa em MATLAB: cálculo de intervalos de predições via bootstrap

```

function (previsao)=bootstrap(peso,x1,x2,y,xpredicao,ypredicao,C,s,ene,delta,B);
n=length(ytrain);
(saidarede,jacob)=kalman(peso,x1,x2,y,xpredicao,ypredicao,C,s,ene,delta)
for j=1:n
residuo(j)=ytrain(j)-ajustado(j);
end
for i=1:B
indice=unidrnd(n,1,n);
for j=1:n

```

```

residuoaleatorio(j)=residuo(indice(j));
end
h=jacob*inv(jacob'*jacob)*jacob';
diagonal=diag(h);
correcao=0;
for j=1:n
correcao=correcao+(1/n)*(residuoaleatorio(j)/sqrt(1-diagonal(j)));
end
for j=1:n
residuo1(j)=residuoaleatorio(j)/sqrt(1-diagonal(j))+correcao;
end
for j=1:n
bootstrap(j)=ytrain(j)+residuo1(j);
end
(pesofinal,jacob)=kalman(peso,x1,x2,bootstrap,xpredicao,ypredicao,C,s,ene,delta);
(previsaocerto)=predicao(pesofinal,x1,x2,xpredicao);
for k=1:10
previsao(k,i)=previsaocerto(k);
end
end

```

A.5 Resultados: ajuste dos modelos AR lineares

Programa utilizado: MINITAB.

EUA 4

```

The regression equation is
ALVO = 0,000011 + 0,0059 DEF 1 + 0,0104 DEF 2 - 0,639 DEF 12 - 0,369 DEF 24

```

```

Predictor Coef SE Coef T P
Constant 0,0000105 0,0001072 0,10 0,922
DEF 1 0,00594 0,03767 0,16 0,875
DEF 2 0,01038 0,03770 0,28 0,783
DEF 12 -0,63868 0,04319 -14,79 0,000
DEF 24 -0,36892 0,04276 -8,63 0,000

```

```
S = 0,00239684 R-Sq = 31,6% R-Sq(adj) = 31,1%
```

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,00131484	0,00032871	57,22	0,000
Residual Error	495	0,00284369	0,00000574		
Total	499	0,00415853			

EUA 6

```

The regression equation is
ALVO = 0,000003 + 0,0006 DEF 1 + 0,0006 DEF 2 - 0,0290 DEF 3 - 0,741 DEF 12
- 0,553 DEF 24 - 0,287 DEF 36

```

```
Predictor Coef SE Coef T P
Constant 0,0000033 0,0001043 0,03 0,975
DEF 1 0,00063 0,03659 0,02 0,986
DEF 2 0,00059 0,03654 0,02 0,987
DEF 3 -0,02904 0,03625 -0,80 0,423
DEF 12 -0,74057 0,04458 -16,61 0,000
DEF 24 -0,55335 0,04988 -11,09 0,000
DEF 36 -0,28714 0,04480 -6,41 0,000
```

S = 0,00230474 R-Sq = 37,9% R-Sq(adj) = 37,1%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	6	0,00156074	0,00026012	48,97	0,000
Residual Error	481	0,00255500	0,00000531		
Total	487	0,00411573			

UK 4

The regression equation is

ORIGINAL = - 0,000021 - 0,0142 DEF 1 + 0,0612 DEF2 - 0,573 DEF 12 - 0,270 DEF 24

```
Predictor Coef SE Coef T P
Constant -0,0000214 0,0002067 -0,10 0,917
DEF 1 -0,01417 0,03839 -0,37 0,712
DEF2 0,06121 0,03834 1,60 0,111
DEF 12 -0,57326 0,04266 -13,44 0,000
DEF 24 -0,27024 0,04247 -6,36 0,000
```

S = 0,00462133 R-Sq = 27,6% R-Sq(adj) = 27,1%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,0040400	0,0010100	47,29	0,000
Residual Error	495	0,0105716	0,0000214		
Total	499	0,0146116			

UK 6

The regression equation is

ORIGINAL = 0,000023 - 0,0039 DEF 1 + 0,0544 DEF 2 + 0,0054 DEF 3 - 0,605 DEF 12
- 0,369 DEF 24 - 0,168 DEF 36

```
Predictor Coef SE Coef T P
Constant 0,0000230 0,0002038 0,11 0,910
DEF 1 -0,00395 0,03865 -0,10 0,919
DEF 2 0,05443 0,03857 1,41 0,159
DEF 3 0,00535 0,03861 0,14 0,890
DEF 12 -0,60549 0,04397 -13,77 0,000
DEF 24 -0,36911 0,04871 -7,58 0,000
DEF 36 -0,16799 0,04328 -3,88 0,000
```

$S = 0,00450225$ R-Sq = 29,1% R-Sq(adj) = 28,2%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	6	0,00399277	0,00066546	32,83	0,000
Residual Error	481	0,00974999	0,00002027		
Total	487	0,01374277			

CAN 2

The regression equation is
 $\text{ORIGINAL} = 0,000009 - 0,0956 \text{ DEF 1} - 0,456 \text{ DEF 12}$

Predictor	Coef	SE	Coef	T	P
Constant	0,0000089	0,0001522	0,06	0,953	
DEF 1	-0,09564	0,03958	-2,42	0,016	
DEF 12	-0,45638	0,04012	-11,38	0,000	

$S = 0,00343624$ R-Sq = 21,7% R-Sq(adj) = 21,4%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	2	0,00165784	0,00082892	70,20	0,000
Residual Error	507	0,00598652	0,00001181		
Total	509	0,00764435			

CAN 4

The regression equation is
 $\text{ORIGINAL} = - 0,000010 - 0,102 \text{ DEF 1} - 0,0831 \text{ DEF 2} - 0,570 \text{ DEF 12} - 0,266 \text{ DEF 24}$

Predictor	Coef	SE	Coef	T	P
Constant	-0,0000096	0,0001487	-0,06	0,949	
DEF 1	-0,10168	0,03872	-2,63	0,009	
DEF 2	-0,08306	0,03889	-2,14	0,033	
DEF 12	-0,57019	0,04402	-12,95	0,000	
DEF 24	-0,26560	0,04478	-5,93	0,000	

$S = 0,00332045$ R-Sq = 27,8% R-Sq(adj) = 27,2%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,00209621	0,00052405	47,53	0,000
Residual Error	494	0,00544655	0,00001103		
Total	498	0,00754277			

ITA 4

The regression equation is
 $\text{ORIGINAL} = - 0,000030 - 0,0697 \text{ DEF 1} + 0,0723 \text{ DEF 2} - 0,647 \text{ DEF 12}$
 $- 0,320 \text{ DEF 24}$

```
Predictor Coef SE Coef T P
Constant -0,0000299 0,0001765 -0,17 0,866
DEF 1 -0,06968 0,03694 -1,89 0,060
DEF 2 0,07229 0,03690 1,96 0,051
DEF 12 -0,64742 0,04195 -15,43 0,000
DEF 24 -0,31993 0,04160 -7,69 0,000
```

S = 0,00394655 R-Sq = 33,5% R-Sq(adj) = 32,9%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,00387800	0,00096950	62,25	0,000
Residual Error	495	0,00770974	0,00001558		
Total	499	0,01158774			

FRA 2

The regression equation is
 ORIGINAL = 0,000003 - 0,116 DEF 1 - 0,458 DEF 12

```
Predictor Coef SE Coef T P
Constant 0,0000034 0,0001228 0,03 0,978
DEF 1 -0,11648 0,03862 -3,02 0,003
DEF 12 -0,45764 0,03796 -12,06 0,000
```

S = 0,00277842 R-Sq = 24,3% R-Sq(adj) = 24,0%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	2	0,00125851	0,00062926	81,51	0,000
Residual Error	509	0,00392928	0,00000772		
Total	511	0,00518780			

FRA 4

The regression equation is
 ORIGINAL = - 0,000031 - 0,124 DEF 1 - 0,0734 DEF 2 - 0,599 DEF 12 - 0,298 DEF 24

```
Predictor Coef SE Coef T P
Constant -0,0000309 0,0001173 -0,26 0,792
DEF 1 -0,12412 0,03776 -3,29 0,001
DEF 2 -0,07338 0,03760 -1,95 0,052
DEF 12 -0,59888 0,04225 -14,17 0,000
DEF 24 -0,29757 0,04141 -7,19 0,000
```

S = 0,00262321 R-Sq = 31,5% R-Sq(adj) = 30,9%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,00156433	0,00039108	56,83	0,000
Residual Error	495	0,00340621	0,00000688		

Total 499 0,00497054

GER 2

The regression equation is

ORIGINAL = 0,000007 + 0,0708 DEF 1 - 0,428 DEF 12

Predictor Coef SE Coef T P
 Constant 0,0000071 0,0001284 0,06 0,956
 DEF 1 0,07077 0,03849 1,84 0,066
 DEF 12 -0,42814 0,03759 -11,39 0,000

S = 0,00298830 R-Sq = 19,9% R-Sq(adj) = 19,6%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	2	0,00119625	0,00059812	66,98	0,000
Residual Error	539	0,00481324	0,00000893		
Total	541	0,00600949			

GER 4

The regression equation is

ORIGINAL = - 0,000000 + 0,0593 DEF 1 + 0,0528 DEF 2 - 0,475 DEF 12
 - 0,154 DEF 24

Predictor Coef SE Coef T P
 Constant -0,0000001 0,0001255 -0,00 0,999
 DEF 1 0,05932 0,03874 1,53 0,126
 DEF 2 0,05284 0,03889 1,36 0,175
 DEF 12 -0,47550 0,04200 -11,32 0,000
 DEF 24 -0,15375 0,04080 -3,77 0,000

S = 0,00288787 R-Sq = 20,1% R-Sq(adj) = 19,5%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	0,00110169	0,00027542	33,03	0,000
Residual Error	525	0,00437840	0,00000834		
Total	529	0,00548009			

JAP 2

The regression equation is

ORIGINAL = - 0,000014 + 0,119 DEF 1 - 0,437 DEF 12

Predictor Coef SE Coef T P
 Constant -0,0000138 0,0002817 -0,05 0,961
 DEF 1 0,11893 0,03854 3,09 0,002
 DEF 12 -0,43659 0,03795 -11,50 0,000

S = 0,00655261 R-Sq = 20,4% R-Sq(adj) = 20,1%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	2	0,0059261	0,0029630	69,01	0,000
Residual Error	538	0,0230999	0,0000429		
Total	540	0,0290260			

JAP 6

The regression equation is

$$\text{ORIGINAL} = -0,000116 + 0,126 \text{ DEF 1} + 0,0371 \text{ DEF 2} + 0,0637 \text{ DEF 3} - 0,667 \text{ DEF 12}$$
$$- 0,432 \text{ DEF 24} - 0,202 \text{ DEF 36}$$

Predictor	Coef	SE Coef	T	P
Constant	-0,0001160	0,0002616	-0,44	0,658
DEF 1	0,12558	0,03625	3,46	0,001
DEF 2	0,03712	0,03654	1,02	0,310
DEF 3	0,06374	0,03640	1,75	0,081
DEF 12	-0,66747	0,04203	-15,88	0,000
DEF 24	-0,43220	0,04610	-9,38	0,000
DEF 36	-0,20190	0,04087	-4,94	0,000

 $S = 0,00594737$ R-Sq = 34,2% R-Sq(adj) = 33,5%**Analysis of Variance**

Source	DF	SS	MS	F	P
Regression	6	0,0093921	0,0015654	44,26	0,000
Residual Error	510	0,0180393	0,0000354		
Total	516	0,0274315			

Apêndice B

Dados utilizados

B.1 Dados de solo

Nitrogênio total	Carbono orgânico	Nitrogênio total	Carbono orgânico	Nitrogênio total	Carbono orgânico	Nitrogênio total	Carbono orgânico
1,20	19,17	1,10	18,10	1,30	22,09	1,31	19,80
1,22	18,82	1,15	19,53	1,20	20,87	1,22	19,97
1,28	22,74	1,29	20,60	1,23	19,12	1,29	23,41
1,12	17,57	1,28	18,64	1,28	18,07	1,54	25,30
1,15	20,24	1,31	19,89	1,30	20,96	1,48	27,20
1,22	19,35	1,25	20,78	1,25	18,95	1,43	25,99
1,23	18,99	1,25	22,01	1,27	19,30	1,38	24,96
1,03	16,14	1,24	22,53	1,34	20,69	1,23	19,11
0,99	16,05	1,24	16,92	1,41	25,06	1,33	20,66
1,07	16,32	1,24	19,90	1,35	24,01	1,21	20,31
1,16	19,17	1,24	17,45	1,35	23,31	1,29	21,86
1,24	18,99	1,23	19,03	1,29	21,74	1,27	20,48
1,26	20,42	1,24	17,45	1,30	21,57	1,26	20,86
1,26	21,31	1,23	20,96	1,53	24,54	1,21	20,14
1,17	19,71	1,30	21,31	1,13	20,52	1,22	19,78
1,28	22,03	1,32	21,13	1,31	23,31	1,20	17,97
1,21	20,42	1,39	21,48	1,34	23,58	1,19	17,79
1,25	19,53	1,37	23,76	1,23	21,69	1,17	21,23
1,16	18,99	1,19	18,77	1,22	22,38	1,19	18,88
1,32	20,60	1,25	20,96	1,39	24,96	1,17	19,24
1,34	25,24	1,32	18,95	1,08	20,83	1,16	17,79
1,33	21,49	1,33	21,39	1,27	23,07	1,13	18,70
1,28	21,31	1,46	23,66	1,25	20,83	1,14	20,14
1,23	18,82	1,38	21,39	1,28	20,31	1,15	19,78
1,32	20,24						

B.2 Dados de inflação

ESTADOS UNIDOS	0,00191	0,00301	0,003968	0,003771	0,000681
	0,0019	0,005006	0,003163	0,005892	0,007133
0,004146	0	0,002986	0,002368	0,006391	0,0054
0,002063	0,000952	0,001989	0,003149	0,007403	0,003359
0,00309	0,001895	0,002977	0,003135	0,008917	0,004018
0,003087	0	0,002963	0,007024	0,009354	0,003336
0,005111	0,005675	0,002959	0,009286	0,010295	0,004982
0,005091	-0,000945	0,003927	0,00691	0,007142	0,002647
0,001016	0	0,002939	0,006102	0,005578	0,00165
0,001009	-0,001886	0,004874	0,006821	0,007555	0,00132
0	0,001886	0,002916	0,002265	0,007996	0,005584
0,004045	0,000945	0,002907	0,017924	0,005459	0,004578
0	0,000938	0,006746	0,002955	0,004446	0,00228
0,006034	0	0,003833	0,008087	0,008833	0,004869
0,001004	0	0,002869	0,007295	0,011656	0,00291
0,006989	0,00376	0,00286	0,006516	0,009612	0,003224
0,001991	0,004682	0,005691	0,008629	0,011412	0,003212
0	0	0,003775	0,012803	0,012216	0,004163
0,000988	0	0,002825	0,011244	0,01161	0,004781
0,001985	0,000935	0,002817	0,005575	0,010562	0,002541
-0,001985	0,001861	0,003739	0,011058	0,01	0
0	0,001864	0,008369	0,009575	0,010351	0,000633
0	0,000925	0,00646	0,007462	0,008912	0,001901
0,001985	-0,000925	0,002758	0,012754	0,009273	0,004104
-0,001985	0,000925	0,006401	0,011937	0,010494	0,004401
0,000993	0,00093	0,004546	0,008532	0,014252	0,004071
-0,000993	0	0,00453	0,008463	0,013629	0,003741
0	0,001851	0,004505	0,007102	0,014281	0,003108
0,001985	0,002779	0,003588	0,004494	0,011195	0,00155
0,000991	0	0,005365	0,007023	0,009848	0,002166
0,003943	0,000919	0,006219	0,003809	0,010965	0,003087
0,002956	0,000924	0,003534	0,005057	0,000808	0,003077
-0,000984	0,001845	0,005281	0,004404	0,006436	0,003374
0,002945	0,000916	0,005257	0,00813	0,00918	0,002445
0,002936	0,000397	0,006094	0,01053	0,008702	0,00305
0,000979	0	0,004329	0,003075	0,009018	-0,002745
-0,000979	0,001068	0,005171	0,004902	0,00855	-0,004591
-0,000974	0,003199	0,003436	0,006095	0,008095	-0,002149
0,001952	0,002121	0,001713	0,006055	0,01031	0,00307
0	0,005297	0,005118	0,004218	0,007194	0,004892
0,004873	0,001056	0,005092	0,002404	0,006392	0,000305
0	-0,002113	0,003383	0,002395	0,008212	0,001827
0,001939	0,002113	0,005049	0,002392	0,008514	0,004857
0,000971	0,00105	0,00084	0,00417	0,011362	0,000909
0	0,002107	0,001678	0,005928	0,007625	0,000906
0,000964	0,003152	0,003342	0,005305	0,010075	0,000907
0,003865	0	0,003335	0,005859	0,002145	0,006022
0,000965	0,006268	0,004978	0,004665	0,002854	0,003896
0,000965	0,003122	0,005777	0,004067	0,002846	0,004475
-0,000965	0,004142	0,002468	0,004047	0,003547	0,005345
0,000965	0,001034	0,002457	0,00288	0,003181	0,002956
0	0,003096	0,000819	0,002872	-0,00106	0,004125
0	0,003082	0,001636	0,005722	0,004231	0,002057
-0,000965	0,005124	0,001634	0,010217	0,0098	0,005559
0,001928	0,002037	0,004068	0,00619	0,012118	0,004949
0,003834	0,004072	0,000813	0,007828	0,005491	0,00261
-0,000954	0	0,004857	0,00555	0,002051	0,001446
0,002871	0,001015	0,001615	0,006624	0,001706	-0,000395
0	0	0,002414	0,004391	0,002724	0,002596
0	0,001014	0,003215	0,003826	-0,001701	0,002589
-0,000958	0,00202	0,002404	0,003811	-0,004095	0,004302
0	0,002022	0,00399	0,002715	0,002391	0,005137
0,002866	0,003025	0,001593	0,004864	0,000341	0,00341

0,004247	0,000686	0,007247	0,002115
0,004227	0	0	0,005246
0,004211	0,00274	0	0,000532
0,006701	0,003413	0,003003	-0,003676
0,003334	0,003403	0,002397	0,002097
0,000831	0,001357	0,004776	0,005754
0,001662	0,000679	0,001785	0,00779
0,004966	0,003385	0,000595	
0,004121	0,002699	0	
0,00574	0,004035	0,002967	
0,006519	0,002681	0,005906	
0,00567	0,000669	0,008211	
0,002421	0,001336	0,000584	
0,002414	0	0,001166	
0,001606	0,004	0,005233	
0,003205	0,003985	0,002315	
0,004788	0,003307	0	
0,002387	0,003298	0,005198	
0,001587	0,001973	0,001723	
0,010256	0,001969	0,000574	
0,004699	0	-0,000574	
0,005454	0,00262	0,006304	
0,001552	0,00196	0,003995	
0,002324	0,003259	0,00227	
0,005403	-0,000651	0,00396	
0,003842	-0,000652	0,004516	
0,009161	0,005847	0,001685	
0,008324	0,003232	-0,002809	
0,00601	0,005152	0	
0,002244	0,003847	0,004491	
0	0,001917	-0,003367	
0,005962	0,000638	-0,001688	
0,001485	0,001914	-0,003959	
0,001482	0,001909	0,002268	
0,001481	0,003173	0,00394	
0,002953	0,003164	0,005611	
0,002946	0,001893	0,00557	
0,00147	0	0	
0,002933	0,003148	0,000565	
0,004382	0,003137	0,00111	
0,001456	0,002503	0,003322	
0,002907	0,00125	0,001657	
0,000726	-0,000625	0,001664	
0,001449	0,001248	0	
0,003615	0,001247	-0,002215	
0,005038	0,001867	0,004417	
0,001435	0,002485	0,007675	
0,001432	0,002478	0,005992	
0,003573	-0,000619	-0,002181	
0,002137	-0,001239	-0,001632	
0,002843	0,001858	0,001088	
0,002834	0,001855	0,001087	
0,003533	0,001851	0,003804	
0,00141	0,001848	0,003241	
-0,000705	0,001844	-0,001079	
0,004921	0,001228	-0,002703	
0,003499	0,001226	-0,001083	
0,003488	0,001225	0,004865	
0,002782	0,001223	0,005388	
0,001387	0,002442	0,006425	
0,001387	0	0,003193	
0	-0,00061	0,005836	
0,002766	0,002438	0,003173	
0,002069	0,001217	-0,00159	
0,004126	0,000304	0,000527	

CANADÁ	0,0013304	-0,0011492	0,0027489	0,0005622	0,005521
	0,0026435	0,0034331	0,0109331	-0,0016874	0,0054923
0,0021484	0,0039521	0,0056923	0,0072188	0,0100897	0,0021882
0	0,0026321	0,0045317	0,0089539	0,0083297	0,0032725
0,0032043	-0,0026321	0,0011284	0,0088704	0,0027621	0,0010895
0,0010723	0,0039426	0,0033777	0,0131657	0,0076939	0,0043431
0,0042583	0,0026158	0,0056093	0,0060836	0,0092622	0,0054017
0,0021256	0	0	0,0025943	0,0118582	0
0,0063371	0,0013053	0,0033526	0,0077505	0,0069421	0,0010778
0,0052578	0	0,0066666	0,0051326	0,0095339	0,0021505
0,0010457	0	0,0022107	0,0076517	0,0047343	0,006424
-0,0010457	0,0026115	0,0033069	0,0101095	0,0078402	0
-0,0010468	0	0,0022035	0,0100083	0,0036369	0,0042583
0,0020925	0,0038956	0,0043826	0,0074392	0,0087791	0,006356
0,0020945	0,0038864	0,0032794	0,0163388	0,0071713	0,0021097
0,0052076	0,0038655	0,002177	0,0128837	0,0096516	0,0042061
0,0072555	-0,0038655	0	0,0071759	0,0060483	0,0020964
-0,0010369	0,0012902	0,0054267	0,0094861	0,0055127	0,0062631
0	0,0025753	0,0107659	0,0062733	0,0084602	0,0031177
-0,0031045	0,0025746	0,0021374	0,0093385	0,0108435	0,0010361
0,0041413	0	0,008509	0,0107847	0,0058651	0,0020704
0,0030917	0,0012819	0,0042299	0,0099152	0,0116279	0,0030983
0,0030759	0,0012802	0,0031619	0,004542	0,0110183	0,0041153
0,0030727	0,0025556	-0,0021084	0,0075221	0,0075922	0,0040984
-0,0010211	0	0,0021084	0,0052364	0,0094132	0,0050987
-0,0010221	0,0025549	0,004194	0,0052058	0,0088595	0,003048
-0,003079	0,0063443	0,0041813	0,0081276	0,0087817	0,0030373
-0,00206	-0,0012657	0,0020817	0,0146082	0,0123496	0,0010114
0	-0,0025361	0,0041552	0,0136834	0,0058913	0,0050371
0,0010337	0	0,0010349	0,0099666	0,0133918	0,0010053
-0,0051602	0,0025361	0,0061919	0,0021208	0,0097437	0,0080001
0,0072118	0,006318	0	0,0091447	0,0123874	0,0029829
0,0040969	0,0012577	0,0020586	0,0090618	0,0081745	0
0,0061172	0,0012561	0,0040952	0,0013891	0,0081082	0,0049537
0,0070877	0,0012603	0	0,0055276	0,0160218	0,0049279
0,0020185	0,0025043	-0,0020455	0,0034405	0,0079156	0,0019646
-0,0030262	0,0024981	0,0010233	0,0041124	0,0078535	0,0019608
-0,0030354	0,0074627	0	0,0040955	0,0077923	0,0048859
-0,0020246	0,0037134	-0,0041041	0,0081411	0,0090158	0,0038911
-0,0030508	-0,0012344	0,0041041	0,004718	0,0089334	0,0048419
0,0050754	-0,002479	0,0040873	0,0040268	0,0050697	0,0057804
-0,0010118	0,0012374	0,0030511	0,004677	0,0063003	0,0028784
0,0020225	0,0061699	0,0070878	0,0046583	0,0124846	0,0066819
-0,0010108	0,0049098	0,0040297	0,0066117	0,0123307	0,0009517
0,0030354	0,0024431	0,0020065	0,003291	0,00489	0
0,0040308	0,0072992	0,007995	0,0032802	0,0145282	0,0037951
0,0080193	0,001216	0,0069402	0,0084768	0,0095695	0,0037807
0,0009956	0,0060304	-0,0019767	0,0090517	0,0047506	0,0009423
0	0,0012072	0,0009888	0,0102413	0,0047282	0,0018832
-0,0025761	0,0023991	0,0039456	0,0057175	0,0058798	0,0037559
-0,0026743	0,0035879	0,0068765	0,0081988	0,0058437	0,0056075
0,0013411	0,0047694	0,0029289	0,0068862	0,0069687	0,0027927
0	0,0011874	0,0038963	0,0093124	0	0,0064839
-0,0013411	0,001186	0,0009706	0,0043193	-0,0023175	0,001845
0	0,00119	0,0058083	0,0055217	0,0046297	0,0055147
0	0,002365	0,000964	0,0097451	0,0103399	0,0027467
0,0013411	0,0011804	0,0009631	0,0066449	0	0,000913
0	0,001179	0,0124498	0,0065983	0,0022831	0,0045552
0,0013332	0,0023593	0,0075814	0,0035821	0,011338	0,0027242
0,0039889	0,0093659	0,0037714	0,0065344	0,0033757	0
0	0,0011639	0,0009396	0,0111886	0,0056031	0,005425
0	0,0058047	0,0028134	0,0023374	0	0,0062915
0	0,0092148	0,0065383	0,0139212	0,0055703	0,005362
0	0,0045784	0,008347	0,0086047	0	0,0026709
0,0013261	-0,002284	0,0055248	0,0147385	0,0033286	0,0106102

0,0052632	0,0053421	-0,0035522
0,006105	0,0015209	0,0053231
0,0008698	0,0037036	0,006178
0,0017361	0,0048379	0,0035087
0,0034632	0,0019291	-0,0017528
0,0034513	0,0028856	0,0043762
-0,0008623	0,0028784	0,0017551
0,0085838	0	0,0034812
0,0059657	0,0019132	0,0008684
0,0033927	-0,0019132	-0,0034781
0	0,0009576	0,0043458
0,0050676	-0,0009576	0,0034631
0,0042029	0,002869	0,0069001
0,005021	-0,0019114	0,0085563
0,000835	0,0028663	0,0008507
0,0024984	0,000954	-0,003407
0,0082851	0,0038057	0
0,0057596	0,0028445	0,0025563
-0,0008213	0,0028375	-0,0051289
0,0259335	0	-0,0086005
0	0	0,0008722
0,0039915	0	0,0025827
0	0,0018862	0,0060006
0,0047695	0,0018837	0,0068237
0,0047469	0,0046926	0,0059339
0,0007895	0	0,0016862
0,0007878	0,0018703	0,003364
-0,0015773	0,0018678	0,0050343
-0,0015798	0,0018644	0,0041681
0,0039454	0	0
-0,0047356	0,0009298	0,003322
0,0047356	0,0018592	0,0024843
0,0007865	0	-0,0033138
0,0031422	0,0018557	0,0082732
0,0007846	-0,0009274	0,0073888
0,0015662	0,0009274	0,0040763
0,0023441	-0,0018557	-0,0073586
0,0023398	-0,0009291	0,0008193
0	0,0055609	0,0008279
-0,0007797	0,000924	0,0008179
0,0015576	0,0009231	0,0024498
0,0046584	-0,0009231	0,0016298
0	0,0034107	-0,0024458
0,0038661	0,0017463	0,0024458
0,0030817	-0,0005508	0,0008139
-0,0007701	0	0,0008133
0	-0,0027618	0,0024358
0,0015385	0,0043227	0,0032478
0,0007689	0,0008259	0,002422
0,002301	-0,0034904	0,00884
0,0007666	0,0027562	0,000799
0,0007649	-0,0007337	-0,000799
0,0015291	0,0054937	-0,0015998
0,0045732	0,0054647	0,0008002
-0,0015221	0,0018153	0,0023969
0	0,001811	0,0039821
-0,0076453	0,0027119	-0,0023873
-0,0015361	0,0027035	-0,000797
0,0007689	0,0026972	0,0039789
-0,0023074	0,0008965	
0,0023074	-0,0008965	
0,0038324	0,0008965	
0,0007654	-0,0008965	
0,0007637	0,0053707	
-0,0015291	0,0071181	

REINO UNIDO	0,004963	0,008475	0,005848	0,005961	-0,001153
	0,004939	0,00421	0,017335	0,00395	0,004599
0	0	0,008369	0,00856	0,005899	0,003434
-0,005698	-0,009902	0	0,005666	0,005868	0,012493
0,005698	0	0,008298	0,002821	0,009697	0,004505
0	0	0	0,002813	0,013424	0,002243
0,011299	0	0,020451	0,011174	0,009479	-0,001121
0,005603	0,004963	0	0,019259	0,007517	0,008939
0	0,004939	0,00404	0,005435	0,016716	0,002223
0	0,009804	0	0,008098	0,007338	0,006637
0,005571	0	0	0,016419	0,018117	0,003302
0,005539	0,004867	0,004024	0,019571	0,042186	-0,001099
0,00551	0	0,004008	0,007724	0,00686	0,003295
0	0	0,003991	0,034027	0,010206	0,007645
-0,00551	-0,009756	0,011881	0,014757	0,0101	0,009747
0,00551	0	0,003927	0,010935	0,009999	0,021322
0,01093	0	0,007813	0,007216	0,006613	0,004211
0	0,004889	0,003884	0	0,02441	0,002097
0,005419	0,004867	0,011562	0,010733	0,014365	-0,002097
-0,016349	0	-0,003839	0,021132	0,012599	0,003146
0	0,004843	0,003839	0,017267	0,033852	-0,001049
0	0	0	0,013609	0,009036	0,002095
0,01093	0,004818	0	0,026668	0,010439	0,003133
0,005419	0,00957	0	0,016314	0,007391	0,001043
0	0,009478	0,007634	0,01923	0,002942	0,00208
0,005391	0	0,003795	0,037385	0,005855	0,00415
-0,005391	0	0,007548	0,041926	0,005824	0,001034
0	0,004706	0,007491	0,020317	0,00867	0,010289
-0,005419	0	0,003725	0,008583	0,00574	0,001022
0	0,004684	0,007408	0,005683	0,005705	0
0	0,004661	0,014653	0,008462	0,008498	-0,003072
0	0,004641	0,00363	0,013948	0,015395	0,003072
0	0,004619	0	0,013756	0,02875	0,0051
-0,00545	0	0,010812	0,010867	0,006725	0,002033
0,00545	0	-0,003591	0,013424	0,005349	0,008089
0,005419	0,018264	0,007169	0,013246	0,003992	0,003016
0	0,004515	0,010649	0,005251	0,007936	0,004009
0	0,004495	0,007043	0,01816	0,005255	0,003991
0	0	0,006994	0,010228	0,009133	0,00199
0	0	0,0104	0,005077	0,010336	0,011857
0	0	0,006873	0,002527	0,006406	0,000983
0	0,004475	0,010222	0,015037	0,005094	0
0,010754	0,004454	0,020136	0,012362	0,001271	-0,000983
0	0,004435	0,006623	0,017054	0,007585	0,002942
-0,005362	0	0,006579	0,014388	0,019951	0,002934
0	0	0,006536	0,014184	0,00738	0,004871
0,005362	0	0	0,025494	0,003669	0,004848
0,005332	0,013187	0,003252	0,009114	0	-0,000967
0,005305	0,008695	0,003242	0,011275	0	0
0	0,00432	0,006452	0,024364	0	0,003864
0	-0,00432	0,003211	0,008713	0,004873	0,003851
0,005277	0,00432	0,00639	0,008641	0,004847	0,016198
0,005248	0	0,006342	0,002147	-0,002422	0,003774
0	0	0,00316	0,004284	0,001213	0,003759
0,010418	0,008584	0,00942	0,006389	0,00483	0,000937
0	0	0,006231	0,004238	0,001205	0,011185
0,010309	0	0,003101	0,004217	0,014336	0,004623
0	0	0,006173	0,006299	0,003552	0,010096
0	0	0,006135	0,004173	0,002362	0,004556
0,010205	0,00851	0,006098	0,00623	0,005879	0,002723
0	0	0,015084	0,006195	0,004677	0,006325
0,005062	0,004229	0,00299	0,014298	0,004658	0,007181
0	-0,004229	0,002981	0,006066	0,003478	0,004463
0,005038	-0,004247	0,00889	0,008034	0,003466	0,017654
0,01	0	0,005883	0,003989	0,002305	0,006105

0,003473	0,000689	0,010044
0,000866	0,004805	0,003518
0,002594	0	0,002341
0,006885	0,006146	-0,003509
0,00769	0,004076	0
0,008474	0,010119	0,007015
0,002528	0,004018	-0,000585
0,005876	0,001336	0,002913
0,005841	-0,004684	0,000574
0,009933	0,005351	-0,006407
0,030023	0,004658	0,00525
0,008754	-0,005325	0,001157
0,003954	0	0,005216
0,000788	0,00599	0,006334
0,0102	-0,003324	0,001153
0,009323	0,00465	-0,006336
0,007705	0,003968	0,004039
-0,002305	0,007234	0,003438
-0,000769	0,001964	-0,001718
0,002307	0,000653	-0,004023
0,005361	-0,003929	-0,001158
0,003813	0,004583	-0,000579
0,012854	0,004561	0,002884
0,003001	0	0,004018
0,004484	0,000651	0,006855
-0,002239	0,003244	0,002845
0,002239	0	0
0,003722	0,003878	-0,001702
0,003708	0,002577	0,002832
0,003694	0,005776	0,006781
0,000737	0,003831	0,001686
-0,000737	0,003817	0,001692
0,005148	0	0,00168
0,002931	0,006329	-0,000563
0,015246	0,005035	0,005036
0,003596	0,001255	0,003337
0	0,000626	0,007204
-0,003596	0,002503	0,001652
0,00072	-0,003129	-0,001107
0,003593	0,005003	0
0,003581	0,003113	0,001661
-0,001431	0,011132	0,004938
-0,003586	0,005519	0,00055
-0,009383	-0,000611	0,00055
0,006506	-0,00245	0,00437
0,003596	0,004285	-0,002183
0,009288	0,004267	0,003814
0,00355	0,000608	0,004344
-0,000709	-0,000608	0,00594
-0,00213	0	0,0043
0,004255	-0,006102	0,001605
0,004237	0,001834	0
-0,000705	0,002441	0,003203
-0,001412	0,006681	0,003736
0,002116	0,002418	0,002648
-0,004237	0	0,002119
0,005646	-0,003024	0,004754
0,00281	0,00242	-0,005286
0,01186	0,00422	0,003706
0,003462	0,001804	0,00473
0	0,0012	
-0,00485	0,003594	
0,00485	-0,004193	
0,002071	0,005387	
0,001377	0,005359	

ALEMANHA	0,0023229	0	0	0,0100001	0,0038948
	0,0023175	0	0	0,0049628	-0,0019455
0,0076239	0,0069204	0	0,0078927	0,003706	-0,0019493
-0,0050762	0,0091534	0	0,0125002	0,0049201	0,0009751
-0,0025478	0,0090704	0	0,0092736	0,0036742	0,0058309
0,0050891	0	-0,001998	0,006135	0,0048781	0,0019361
0,0025349	0,0045045	0	0,0091325	0,0060643	0,0009667
0	-0,0045045	0,001998	0,001514	0,0012085	0,0057804
0,0050505	-0,0022599	0,0059702	0,0060332	0,0036166	0,0038351
0,0050251	-0,002265	0,0039604	0,0074907	0,0024038	0,0028667
0	-0,0022701	0,0019743	0,0029806	0,003595	0,0019066
0,0025031	0,0045352	0,003937	0,0029718	0,0047733	0
0,007472	0,0045147	0,0019627	0,0014826	0,0047506	0
0,0024783	0,0044944	0,0019589	0,0029586	0,0117787	-0,0009528
-0,0049628	0,0089286	0,001955	0,005891	0,0058377	-0,002864
-0,0125158	0,0044346	0	0,0058565	0,0034863	0,0009556
-0,0050505	0	0	0,0029155	0,0057837	0,0009547
0,0050505	0,0022099	-0,0039139	0,0101377	0,0034542	0,0019066
0,0050251	0	0	0,0043135	0,0011488	0,0009519
0,0025031	0	0,0039139	0,0057225	0,0011474	0,0028504
0	-0,0022099	0,0019512	0,0071074	0	-0,0009492
0	0,0022099	0,0058309	0,0056497	0,0011461	-0,0028531
0	0	0,0115608	0,0084152	0,0079864	-0,0009528
-0,0025031	0	0,0019139	0	0,006795	0
0,0025031	0,0044053	0,0057197	-0,0013976	0,0089888	0,0009528
0,007472	0,004386	0,0018993	0,004187	0,006689	-0,0038168
0,0049505	0,0043668	0	0,0027816	0,0077477	-0,0028722
0,0024661	0,0043478	0,0037879	0,004158	0,0065934	0,0009583
-0,0049383	0	0	0,0013822	0,0043716	-0,0028777
0,0049383	0,004329	0	0,0082531	0,0032662	-0,0019231
0,004914	0,0043104	0	0,006826	0,0065006	0,0019231
0	0,0021482	0,0056551	0,0013596	0,0032345	0,0047916
0	0,0042827	0,0018779	0,0067705	0,0053677	0,0009556
-0,004914	0,0042644	0,0074767	0,0013486	0,0042735	0
0	-0,0021299	0,0111112	0,0013468	0,0042553	0,0019084
0,004914	0,0021299	0,0073395	-0,0013468	0,0031797	0,0009528
0,007326	0,0042463	0,007286	0,0026918	0,0073801	0,0009519
-0,0024361	0,0063358	0,0036232	0,0013432	0,0020986	0
0,0024361	0,0042017	0,0018067	0,002681	0,0010477	-0,0009519
-0,0048781	0,0062696	0,0054006	0,0026738	0,0031364	-0,0019066
-0,0024448	0	0,0035842	0,0026667	0,0072803	0,0009537
0,0024448	0,004158	0	0,0105961	0,0103094	0,0009528
0,0073082	0,0082645	0,0017873	0,0039448	0,0020492	0,0009519
0	0	0,0053428	0,0026212	-0,0010241	0,0037987
0,0048426	0	0,0017746	0,003919	0,0030691	0,0018939
0	0	0,0035398	0,0026042	0,0050943	0
0,0024125	-0,0020597	0,0122916	0,0051881	0,0020305	0,0028342
0	0	0,0052219	-0,0025907	0,003038	0,0028262
0,0072029	0,0020597	0,0051948	0,0012962	0,0030288	0,0009403
0,0071514	0,0061539	0,0017256	0,0012945	0,0010076	0,0009394
0	0,0040816	0,0034423	0,0012928	-0,0010076	-0,0018797
-0,0023781	0,0020346	0,0034305	0,0012912	0,0030196	0,0009403
-0,0023838	0	0,006826	0,0025773	0,002008	0,0009394
0,0023838	0,0020305	0	0,0064144	0,004004	0,0028129
0,0071175	0	0,0101524	0,003829	0,0029925	0,0018709
0	0,0020263	0,0050378	0,0038144	0,0039761	0,008376
0,0094118	0	0,0033445	0,0025349	0,0019822	0,0037003
0,0023392	0,0020222	0,0066556	0,001265	0	0,001845
0,004662	-0,006079	0,0099011	0,0037855	0,0019782	0,0055147
0,0023229	0	0,0065467	-0,0025221	0,00296	0,0018315
0	0,0020305	0,0081235	-0,0012634	0,0039331	0,0009145
0	0,0020263	0,0048426	-0,001265	0,0029397	-0,0009145
0,0023175	0	0,0064206	0,001265	0,000978	-0,0009153
-0,0069687	0,0120726	0,0063796	0,0037855	0,000977	0,0018298
0,0023283	0,001998	0,0031746	0,0025157	0,0009761	0,0027385

0,0018215	0,0010793	0
0,002726	0,0021552	-0,0010045
0,006332	0,0032241	0,004012
0,0036004	0,0053505	0,003996
0,0008981	0	-0,001996
0,0017937	0,0010667	0,001996
0,0017905	0,0010655	-0,0009975
0,0008941	0,0010644	0,0009975
0	0,0021254	0,0089331
0,0026774	0	-0,0039604
0,0035587	0	0,0059347
0,0018762	-0,0021254	0
0,0009368	0	0,003937
0	0,0031864	0,0039216
0,0028051	0,0021186	0,001955
0,0037279	0,0052771	0,0009761
0,0064905	0,0010521	-0,0019531
-0,0018501	0	0
0,0009255	0,0021008	-0,0029369
0,0064546	0,0010488	-0,0019627
0,0049875	0,0010477	0,0097752
0	0	0,0009723
0,0037244	0	0,0029112
0,0037106	0	0,0019361
0,0049261	-0,0010477	-0,0009676
0,0122102	0,0031397	0,0009676
0	0,0072879	0
0	0,0031072	0,0028972
0,0144581	-0,0010347	-0,0019305
0,0047733	-0,0020725	-0,0009667
0,0011898	0,0041408	-0,0009676
0,0059277	0,0010325	-0,0029084
0,0058928	0,007198	0,0096619
0,0035191	0,0010241	0
0,0046729	-0,0020492	0,0047962
0,0023283	-0,0010262	0,0009565
0,0023229	0	-0,0028722
0,0034742	0,0020513	-0,0019194
0	0	0,0028777
-0,0011567	0,0020471	0,0019139
0,0011567	-0,001023	0
0,0034622	0,001023	-0,0009565
0,0011514	0,0030628	0
0,0171139	0,0010188	-0,0019157
0,0078874	0,0030503	0,0076409
0,0022422	-0,0020325	0,000951
0,0033538	-0,0020367	0,0018993
0,0011154	-0,0020408	0,0028423
0,0033389	0	0,0028342
0,0044346	0,0010209	0,001885
0	-0,0020429	0
-0,0011068	0,0020429	0,0028209
0	0,0010199	0,0018762
0,0011068	0,0040692	-0,0028156
0,0022099	0	0,0018779
0,0066007	0,0010147	-0,0037594
0,0065574	0,0040486	0,0103045
0,0010887	-0,0010106	-0,0037348
0,0010875	-0,0020243	0,0037348
0,0032556	-0,0010137	0,002792
0,0010828	0,0020263	
0,0032415	0,0030318	
0,0021552	0,0030227	
-0,0021552	0,0020101	
-0,0010793	0	

FRANÇA	0,002773	0,004665	0,00517	0,005943	0,006945
	0,004148	0,008115	0,006846	0,006397	0,006208
0,001932	0,004131	0	0,009344	0,009282	0,006854
-0,005809	-0,001378	0,010337	0,007578	0,005332	0,006129
-0,001944	0,00412	0,001145	0,008355	0,004824	0,005417
-0,001946	0,001372	0,001143	0,006635	0,009101	0,004714
0,00583	0,009529	0,002277	0,009048	0,006653	0,006698
0,015385	0,00135	0,003402	0,010592	0,008962	0,005327
0,011386	0,006744	0,003397	0,008872	0,00981	0,004638
0,013121	0,006698	0,002256	0,006406	0,010635	0,006588
0,014788	0,002665	0,005622	0,016619	0,008247	0,003276
0,023577	0,001334	0,006707	0,013256	0,013143	0,001962
0,021277	0,003983	0,009971	0,011553	0,010304	0,005212
0,032788	0,006601	0,003304	0,015954	0,008431	0,005185
0,011813	0,003934	0,002194	0,011983	0,011861	0,007086
0,013336	0,003926	0,010906	0,011108	0,006529	0,007035
0,008245	0,006507	0,00325	0,012441	0,007777	0,005086
0,001635	0,002589	0,005391	0,007966	0,019148	0,003798
0,003272	0,002583	0,00429	0,010765	0,010558	0,003783
0,00327	0,001292	0,004271	0,012061	0,011489	0,001258
0,00325	0,00386	0,003195	0,009127	0,012385	0,001256
0,00486	0,001279	0,0053	0,008351	0,008172	0,003134
0,003224	0,001284	0,002111	0,011025	0,007094	0,001876
-0,003224	0	0,005261	0,007513	0,014043	0,001247
0,003224	0,001282	0,006279	0,00813	0,009911	0,000624
0,023869	0,002553	0,00416	0,008735	0,007859	-0,002496
0,014049	0,002547	0,003112	0,007329	0,010704	0,002496
-0,003104	0,001274	0,01337	0,007276	0,007715	0,003734
0	0,005067	0,003055	0,007227	0,009561	0,002482
-0,006246	0,002524	0,003052	0,006524	0,011352	0,003093
0,003132	0	0,005064	0,008418	0,009364	0,001235
0,003114	0	0,00403	0,007707	0,009277	0,001233
0,004648	0,005029	0,00402	0,006374	0,013756	0,00369
0,004635	0,001257	0,002001	0,005708	0,009065	0,002453
0,007675	0,002502	0,00399	0,010689	0,009879	0,001224
0,006103	0,002496	0,00398	0,006854	0,016838	0,001223
0	0,002496	0,003958	0,008661	0,012226	0,009121
0,013583	0,016035	0,003943	0,008583	0,011222	0,001814
0,002993	-0,008604	0,00295	0,006691	0,012792	0,001208
0	-0,002471	0,005873	0,004236	0,008439	0,005417
0,001497	0,002471	0,004866	0,009617	0,005865	0,001799
-0,002996	0,001237	0,003874	0,007151	0,010803	0,001796
0	0,002462	0,005788	0,01063	0,009867	0,00239
0,002996	0,003685	0,006708	0,009359	0,012201	0,002383
0,008911	0,002447	0,003811	0,008115	0,011255	0,000595
0,001482	0,002441	0,004746	0,00346	0,007961	0,002377
0,002948	0,001222	0,003779	0,002873	0,007112	0,001186
0,00294	0,003642	0,004706	0,006861	0,003145	0,000593
0,001471	0,003636	0,00562	0,009074	0,003135	0,001776
0,00146	-0,001206	0,003726	0,012905	0,003905	0,001773
0	0,00362	0,004641	0,009431	0,005441	0,002947
0	0,001208	0,002775	0,0077	0,009258	0,004698
-0,00146	0,002406	0,005522	0,008728	0,008414	0,002341
-0,001471	0,0024	0,004581	0,005417	0,009853	0,002918
-0,00294	0,002394	0,003653	0,008609	0,006763	0,003489
0,007337	0,002395	0,005447	0,008	0,009691	0,0029
0,005828	0,00476	0,005422	0,003715	0,012533	0,002314
0,005802	0	0,008074	0,002644	0,006571	0,002308
0,011489	0,003558	0,00535	0,00474	0,005806	0,001152
0,009949	0	0,006202	0,007329	0,008646	0,001726
0,002831	0,002363	0,008796	0,008829	0,005723	0,004015
0,005623	0	0,00611	0,010799	0,007814	0,002858
0,001405	0,002357	0,005206	0,009674	0,007755	0,00285
0,005584	0,003529	0	0,007568	0,004205	0,00624
0,00278	0,003516	0,002595	0,011997	0,003491	0,003952

0,001126	0	0
0,00281	-0,000906	0,001961
0,001682	0,002718	0,001958
0,002239	0,003613	-0,001958
0,004461	0,002702	0,001958
0,001668	0,000899	0,005847
0,001111	0,001795	-0,001949
0,00457	0	0,002921
0,002028	-0,001795	-0,000973
0,003035	0,004482	-0,003891
0,005037	0,003571	0,002917
0,001005	0,000891	0,00388
0	0,00089	0,004841
0,001003	0,00089	0,006728
0,006	0,001776	0
0,006955	0,003543	-0,001921
0,004938	0,00617	0
-0,001972	0,001756	0,001921
-0,000987	0,001753	0,000949
0,005911	-0,000876	-0,00287
0,002942	-0,001754	0,000961
0,001957	-0,002638	0,00478
0,002929	0,003515	0,000955
0,00292	0,002629	0,004753
0,001942	-0,000876	0,003794
0,002905	0,00175	0,000937
0,000967	0,002619	0
0,00193	0,001742	0
0,004809	0,000869	0,00189
0,003831	0	0,001886
-0,002872	0,001738	0,001873
0,001916	0	0
0,00382	-0,001738	0,001879
0,003806	0,002606	0,001876
0,001898	0,001733	0,006527
0,002839	0	0,004634
0	0,00173	-0,001855
-0,000945	0	-0,000929
-0,000947	-0,002423	0,001856
0,002837	0,003011	-0,000928
0,002828	0,002002	0,001855
0,000942	0,001998	0,00369
0	0	0,002765
0,003756	0,000997	0,00092
0,003742	-0,002995	0,000919
0,004658	0	0
0,000928	0	0,004574
0,001855	0	0,003643
-0,000927	-0,001001	0,00273
0,000927	0,001001	0,00362
0	-0,003005	0
0,0037	0,003005	-0,001813
0,001845	0,002995	0,002718
0,000922	0,002986	0,000904
-0,000922	0	0,002699
0,001842	0	0
0,002757	-0,00199	0,001801
0,001833	0,000996	-0,005405
0,002743	0,001988	0,005405
0,001825	0,000992	0,00627
0	0	
0	0,004949	
0	0	
0,002732	0,000986	
0,002723	0,004919	

ITÁLIA	0	0	0,015874	0,007041	0,012067
	0,00326	0	0,007843	0,013936	0,011921
-0,003635	0,006489	0	0,015503	0,010331	0,007381
-0,002737	0,000808	0,010153	0,007663	0,01022	0,007325
-0,001829	0,003226	0	0,007605	0,003384	0,007274
0	0,004018	0	0,007547	0,016749	0,00578
0	0,006394	0	0	0,016478	0,00144
0,005476	0,015028	0	0,007491	0,012986	0,00574
0,000909	0,010929	0	0,014815	0,012824	0,008546
0,002724	0,013879	0	0,014598	0,009508	0,008476
0,005424	0,003057	0	0,014389	0,009418	0,008402
0,005395	0,006089	0	0,021203	0,00933	0,005563
0,006259	0	0	0,027588	0,009248	0,011035
0,002669	0,001517	0	0,013512	0,02723	0,012271
-0,001778	-0,000759	0	0,013332	0,017754	0,008096
0,00089	0,003783	0,010051	0,013156	0,014554	0,008033
0,008857	0,006775	0	0,019425	0,017194	0,006644
0,006155	0,011191	0	0,019046	0,030772	0,005285
0,002625	0,003702	0,009951	0,030958	0,019102	0,002631
-0,001749	0,00663	0	0,018134	0,008074	0,002624
-0,000876	0,007315	0	0,017802	0,015956	0,005228
-0,002633	0,001456	0,009852	0,011695	0,010501	0,010377
-0,006169	0,002907	0	0,011559	0,010388	0,007713
-0,000886	0,001451	0	0,011427	0,017928	0,005108
-0,002657	0,008657	0	0,005673	0,0101	0,005082
-0,001777	0,006445	0,009756	0,011235	0,019902	0,003796
-0,000889	0,005694	0	0,00557	0,017092	0,005037
-0,002673	0,002835	0,009662	0,011048	0,021558	0,001255
-0,001785	0,004237	0,009568	0,005479	0,011781	0,005006
0,004458	0,004219	0	0,005449	0,01856	0,002495
-0,00089	0,006993	0,009479	0,005419	0,018225	0
-0,003569	0,004174	0	0,010751	0,015679	0,002487
0,002678	0,004845	0	0,010645	0,013244	0,002481
0,003559	0,00207	0	0,010525	0,015235	0,004944
0,0106	0,002066	0,00939	0,010415	0,01074	0,004921
0,004384	0,001374	0	0,020511	0,006389	0,003674
0,003494	0,002059	0,009303	0,020106	0,008459	0,004877
0,005217	0,002737	0,009217	0,024568	0,014627	0,006065
-0,001735	0,005454	0	0,023986	0,016462	0,002415
-0,001738	0,00136	0	0,004728	0,018201	0,003612
-0,000872	0,002711	0,009133	0,004705	0,009969	0,004797
0,001742	0	0	0,009345	0,015749	0,003582
0	0,003381	0,00905	0,018431	0,01358	0,002381
0,004338	0,004711	0	0,027032	0,007677	0,00356
0,001729	0,002683	0	0,021976	0,009514	0,005908
-0,000864	0	0,008969	0,012964	0,011301	0,00821
0	0,003343	0	0,012792	0,009319	0,0035
0,004316	-0,004013	0,008888	0,016805	0,012904	0,001163
0,001722	0,004013	0	0,012421	0,01815	0,005796
0,00429	0	0,008811	0,012274	0,016058	0,004613
0	0,000667	0	0,01212	0,015804	0,004594
0,000855	0,001334	0,008734	0,007999	0,013839	0,002288
0,001709	0,000666	0	0,007936	0,006851	0,003422
0,002559	0,00133	0	0,007873	0,013558	0,003412
0	0,005966	0,008658	0,0117	0,013379	0,001134
0,002553	-0,00863	0,008584	0,015383	0,009916	0,004524
0,001697	0	0	0,007604	0,009821	0,005627
0,001694	0,010363	0,008511	0,007546	0,009726	0,006712
0,002538	0	0,008439	0,007495	0,004826	0,006666
0,006735	0	0,008368	0,007434	0,007993	0,004421
0,00335	0	0,008298	0,014704	0,006351	0,006593
0,004172	0,010257	0,008231	0,007272	0,012577	0,007638
0,005811	0	0,008163	0,010815	0,013965	0,0065
0,005779	0	0,008097	0,007142	0,012252	0,004311
0,008196	0	0,008032	0,007091	0,003039	0,005361

0,00427	0,004066	0,000892
0,001064	0,002433	0,003562
0,002125	0,004329	0,002664
0,005293	0,007209	0,001768
0,007365	0,008741	0,000888
0,006269	0,003948	0,001764
0,004159	0,006284	0,003528
0,009293	0,006246	0,002635
0,007168	0,001555	0,00087
0,004074	0,003103	0,00438
0,004056	0,002321	0,003478
0,003031	0,003857	0,000873
0,004029	0,005374	0,004332
0,00301	0,00153	0,002592
0,006989	0,003835	0,002586
0,005953	0,003895	0,000864
0,007882	0,00388	0
0,00587	0,003865	0
0,001949	0,00385	0,001715
0,006793	0,001919	0,001712
0,007708	-0,000959	0,00086
0,005741	0,000959	0,004269
0,00381	0,000959	0,003408
0,003795	0,000957	0,002546
0,004723	0,003821	0,00254
0,003764	0,000952	0,002533
0,002813	0,002854	0,000837
0,00467	0,000949	0,00169
0,00465	0,002843	0,001678
0,006472	0,000945	0,001675
0,002762	0,002832	0,002512
0,006412	0	0,003343
0,004557	0	0,000836
0,004535	0,000942	0,004152
0,002711	0,000942	0,001654
0,005401	0,002818	0,003309
0,002689	0,003746	0,002473
0,001789	0	0,002466
0,002678	0,002799	0,000815
0,001781	0,002792	0,001646
0,005323	0,002784	0,002454
0,006176	0,000927	0,00163
0,000879	0,00185	0,000819
0,005259	0,000923	0,002442
0,004361	0	0,000807
0,002607	0,000923	0,00163
0,004332	0	0,002421
0,004312	0,001843	0,003235
0,005151	0,00184	0,002417
0,002565	0	0,002411
0,001706	0,000919	0,001601
0,001703	0,001835	0,000804
0,00594	0,001831	0,002399
0,003378	0,00274	0
0,000843	0,001823	0
0,00588	0	0,000802
0,003344	0,002729	0,002392
0,00914	0,000907	0
-0,004144	0,001813	0,003174
0,004144	0,003618	0,003173
0,002479	0,001803	
0,002472	0,0009	
0,001645	0,001799	
0,003281	0,003587	
0,004903	0,002683	

JAPÃO	0,002682	0,018236	0,024034	0,002963	0,003608
	-0,002682	-0,002008	0,0183	0,010302	0,005985
-0,003186	0,008028	0,008014	0,017973	0,002924	0,002384
0,003186	0	0,007954	0,002541	-0,010271	0,002378
0,006348	-0,002667	0,003952	0,005063	-0,001477	0,007101
0,00945	0,00799	0,001968	0,01005	0,001477	-0,007101
-0,003136	-0,002658	0,001968	0,027129	-0,002955	0,001187
0,012498	0,018445	0	0,004854	0,008837	-0,008338
0,006191	0,025774	-0,007888	0,007239	0,013111	0,01543
-0,006191	0,00508	0	0,0331	0,008645	0,007042
0	0,00505	0,007888	0,041007	0,001434	-0,004689
-0,01565	0,002513	0,023301	0,032935	0,00856	0,002347
-0,003162	0,00751	0	0,00431	-0,009994	0,002342
0,003162	0,009928	0,003831	0,025479	0,011413	-0,003515
-0,00951	-0,007436	-0,00575	0,004184	0,012686	0,003515
-0,003186	-0,002492	0,003838	0,004166	-0,004212	0,005832
0,003186	0,019754	0	0,020577	0,007009	0,002323
0	-0,004901	0,01143	0,010131	0,009728	0
0,012662	-0,004925	0,009425	0,014015	0,008265	0,002317
-0,006314	0,004925	0	0,021633	0,005472	-0,002317
0,012582	0,002451	0,001872	0,00968	0,01758	0,001159
-0,006268	-0,002451	0,013023	0,001925	0,008011	0,009228
0,009385	0,007345	0,0092	0,011473	0,003981	-0,00807
-0,009385	0,009707	0,007301	0,001899	0,001324	0,001157
-0,003152	0,004818	0,003629	0,007561	-0,002649	0,003462
0,003152	-0,002404	-0,003629	0,018658	0,017094	-0,001153
-0,006314	-0,004834	0,00544	0,00553	0,001303	0
0,003162	0,009641	0,024887	0	0,002601	0,002304
0,006288	0,014287	0,00321	0,001836	-0,003903	0,003447
-0,00945	0,016412	0,006389	-0,001836	0,010377	-0,005751
0,00945	-0,009348	0,01266	0,021819	-0,001291	-0,002309
0	0	-0,006312	0,014287	0,003868	-0,002315
0,015552	0,023203	0	-0,003552	0,007692	0,004624
-0,009303	-0,002294	0	0	0,008901	0,001153
0,012387	0,004586	-0,003169	0,017637	0,002528	-0,004619
-0,006172	0,022625	0,01575	0,006968	-0,003795	-0,002317
0	-0,006735	0,018575	0,003466	-0,003809	-0,003486
0,015361	-0,002252	0	0,023932	0,015152	0
0	-0,004524	0,006116	0,003374	0,002503	0,003486
-0,003056	0,002262	0,009107	0,001682	-0,001251	0,009238
0,006096	0,017938	0	0,0067	0,001251	0,002296
0,009078	0,01105	0,003016	-0,0067	0	-0,002296
-0,006041	-0,008828	0,011975	0,024897	-0,001251	-0,004609
0	0,006627	0,002972	0,004906	0,001251	0,001154
0,009051	0,008774	0,005917	0	0,007471	0,008042
0,00897	0,008694	-0,002954	0,012967	0,004951	0
0,002968	0,004323	0	0,008019	0	-0,004587
-0,011938	0,010718	0,023394	0,006371	-0,007436	-0,001151
0,005987	-0,006418	0,002886	0,00475	0,008669	-0,002304
0,014815	0,002141	-0,008682	0,015675	0,014688	-0,002309
0,005864	0,002141	0	0,009287	0,002427	0,003462
-0,005864	-0,006432	0,002903	-0,003086	-0,009745	0,005745
0,014599	0,010695	0,002894	-0,003096	0	0
-0,005813	0,006363	0,008632	0,001549	0	-0,001146
0	-0,006363	0,011395	0,015362	-0,002451	-0,002296
0,014472	0,008477	0,002831	0,007592	0,004896	0,003442
0,008585	0,010493	0,002821	-0,010646	0,004873	0,006849
0,005681	0,004166	0	-0,003064	0,009673	0,005673
0,019635	0,004149	0,005618	0,004592	-0,006035	-0,004536
0,005539	0,00207	0,005586	0,004569	-0,003638	-0,002275
0,010992	-0,006218	0,008321	0,009076	-0,002434	-0,00228
0,002725	-0,006257	-0,002766	0,010488	0,010903	-0,00343
-0,002725	-0,002096	0,005524	0,005943	0,008399	0,005711
0,002725	0,008353	0,010961	-0,004454	-0,004791	0,01694
0,013537	0,016496	0,00814	0,002972	-0,003608	0,005584

0	-0,001044	0,005049	0	-0,002998	-0,002032
-0,002229	0,004171	-0,003025	-0,003972	0,001999	0
-0,001117	0,010352	-0,003036	0,000994	0,001998	0
0,008898	0,001029	0	0,006934	0,000997	-0,003056
0,007723	-0,001029	-0,003044	0,002957	-0,001996	-0,003067
-0,01105	-0,007236	-0,001017	-0,006913	-0,002998	0,003067
0,00111	0,003108	0,003047	-0,001984	0,002998	0,003056
0,004429	0,004128	0,002026	-0,000993	-0,000999	0,002032
0,001105	0,001029	-0,001013	-0,000994	0	-0,00305
0,004405	-0,00206	-0,005078	0,003972	-0,003001	-0,002039
0,007662	0	0,001018	0,00198	0,001002	0,002039
0,005438	0	0,006085	0,002963	0	0,001018
-0,004347	0	-0,003038	-0,003952	-0,00301	0
-0,002182	0,003088	-0,003047	-0,005958	-0,002012	-0,005099
0,004358	0,007169	0	-0,000996	0,002012	0,001022
0,007579	0,00102	-0,001018	0,007944	0,001004	-0,002046
0,01073	-0,00102	-0,002039	0,006899	-0,003016	0
-0,001068	0,002039	0,002039	-0,000983	-0,003025	0,002046
-0,001069	0,00305	0,006091	-0,003941	0,004032	0
0,006396	0,001015	0,002023	-0,004948	-0,002014	0,001021
-0,003193	-0,001015	-0,003036	-0,003976	0	0,002039
0,005317	-0,005089	-0,001014	0,000995	-0,005054	-0,003059
0,005288	0,00102	-0,001015	0,004962	-0,001014	0,001021
0,00526	0,001019	0,004052	0	-0,00203	0,003056
-0,004206	0	0,002021	-0,002974	-0,005094	0,005073
-0,001054	0,004065	-0,003033	-0,00398	0,002041	-0,002026
0,002107	0,002026	0,001012	0,002986	0,003053	-0,005083
0,002103	0,001012	-0,001012	0,002977	0,003044	-0,00511
0,01045	-0,004052	-0,002026	0,00198	-0,001014	-0,002051
0,002077	-0,004069	0,001014	-0,005952	-0,004065	
-0,0052	0,004069	0,020057	-0,002989	0,00305	
-0,001043	0,003041	0,001984	-0,000998	0	

Referências Bibliográficas

- [1] Anderson, B. D. O, Moore. J (1979). *Optimal Filtering*. Prentice Hall.
- [2] Baillie, R. T. (1996), *Long Memory Processes and Fractional Integration in Econometrics*, Journal of Econometrics 73, p. 5-59
- [3] Baillie, R. T., Chung, C. F., Tieslau, M. A. (1996). *Analysing Inflation by the Fractionally Integrated ARFIMA-GARCH Model*. Journal of Applied Econometrics, 11, p. 23-40.
- [4] Barbosa, E. P. (1990). *Dynamic Bayesian Models for Vector Time Series Analysis and Forecasting*. Ph.D. Thesis, University of Warwick, UK, Inglaterra.
- [5] Barbosa, E. P. (1991). *Séries Temporais Múltiplas: Uma Análise via Modelos Lineares Dinâmicos*. Mini-curso apresentado na 4^a Escola de Séries Temporais e Econometrica, IMPA, Rio de Janeiro, jul/1991.
- [6] Barbosa, E. P., Sadownik, R. (1999). *A Multiplicative Seasonal Growth Model for Multivariate Time Series Analysis and Forecasting*. Communication in Statistics, 28, n. 2, p. 291-308.
- [7] Beran, J. (1994). *Statistics for Long Memory Processes*. Chapman and Hall, New York.
- [8] Binner, J. M., Gazely, A. M. (1999). *A Neural Network Approach to Inflation Forecasting: The Case of Italy*. Business & Economics Review, 1, n. 1, p. 76-92.
- [9] Binner, J. M., Gazely, A. M., Elger, T. (2004). *Dynamic Neural Network Based Inflation Forecasts for the UK*. Global Business and Economics Review Anthology, p. 546-564
- [10] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- [11] Bos, C. S., Franses, P. H, Ooms, M. (2002). *Inflation, Forecast Intervals and Long Memory Regression Models*. International Journal of Forecasting, 18, n. 2, p. 243-264.
- [12] Box, G. E P., Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco.

- [13] Box, G. P., Jenkins, G. M. , Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control.* 3rd. edition. Prentice Hall, Upper Saddle River, New Jersey.
- [14] Bouma, J. (1989). *Using Soil Data for Quantitative Land Evaluation.* Advances in Soil Science, 9, p. 177-213.
- [15] Breiman, L. (2001). *Statistical modeling: the two cultures.* Statistical Science, 9, n. 3, 199-231.
- [16] Chen, X., Racine, J., Swanson, N. R. (2001). *ARX Neural Network Models with an Application to Forecasting Inflation.* IEEE Transactions in Neural Networks, 12, p. 674-683.
- [17] Cheng, B., Titterington, D. M. (1994). *Neural networks: a review from a statistical perspective.* Statistical Science, 9, n. 1, p. 2-54.
- [18] Cleveland, W. S. (1979). *Robust Locally Weighted Regression and Smoothing Scatterplots.* Journal of the American Statistical Association, 74, p. 829-836.
- [19] Connor, J. T., Martin, R. D., Atlas, L. E. (1994). *Recurrent Neural Networks and Robust Time Series Prediction.* IEEE Transactions on Neural Networks, 5, n. 2, p. 240-254.
- [20] Coppi, R. (2002). *A theoretical framework for Data Mining: the “Informational Paradigm”.* Computational Statistics & Data Analysis, 38, n. 4, p. 501-515.
- [21] Cybenko, G. (1989). *Approximation by Superpositions of a Sigmoidal Function.* Math. Control Signals Systems, 2, p. 303-314.
- [22] Diebolt, C., Guiraud, V. (2002). *A Note on Long Memory Time Series.* Technical Report, Université Montpellier, Faculté des Sciences Economiques, Montpellier, France.
- [23] Doornik, J. A., Ooms, M. (2001). *A Package for Estimating, Forecasting and Simulating ARFIMA Models: Arfima package 1.01 for Ox.*
- [24] Doornik, J. A., Ooms, M. (2003). *Computational Aspects of Maximum Likelihood Estimation of Autoregressive Fractionally Integrated Moving Average Models.* Computational Statistics & Data Analysis, 42, n. 3, p. 333-348.
- [25] Empresa Brasileira de Pesquisa Agropecuária - EMBRAPA (1999). *Avaliação de Tipos de Manejo do Solo com Sistemas Alternativos de Adubação e Seus Impactos no Ambiente.* (Relatório Técnico).
- [26] Fine, T. L. (1999). *Feedforward Neural Network Methodology.* Springer.
- [27] Gomes, D. T., Barbosa, E. P., Timm, L. C. (2005). *Recurrent Neural Net Regression Models with space-varying coefficients for pedotransfer function estimation and prediction of soil properties.*

- RP 16/05, IMECC/Unicamp. Trabalho apresentado (comunicação oral) na 11^a Escola de Séries Temporais e Econometria, Vila Velha, ES, agosto/2005 (submetido para publicação).
- [28] Gomes, D. T., Barbosa, E. P., Timm, L. C. (2005). *Estimação de funções de pedotransferência: redes neurais recorrentes de pesos variáveis no espaço e outros modelos de predição de propriedades do solo*. Trabalho apresentado (comunicação oral) na 9^a Escola de Modelos de Regressão, São Pedro, SP, fevereiro/2005, e no 7º Congresso Brasileiro de Redes Neurais, Natal, RN, outubro/2005.
- [29] Granger, C. W. J., Joyeux, R. (1980). *An introduction to long-memory time series models and fraction differencing*. Journal of Time Series Analysis, 1, p. 15-29.
- [30] Granger, C. W. J., Teräsvirta, T. (1993). *Modelling Nonlinear Economic Relationships: Advanced Texts in Econometrics*. Oxford University Press, UK.
- [31] Greene, W. H. (2003). *Econometrics Analysis*. 5th. edition. Prentice Hall, New Jersey.
- [32] Hassler, U., Wolters, J. (1995). *Long Memory in Inflation Rates: International Evidence*. Journal of Business & Economic Statistics, 13, n. 1, p. 37-45.
- [33] Hastie, T. J., Tibshirani, R. J., Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Verlag, New York.
- [34] Haykin, S. (1999). *Neural Networks - A Comprehensive Foundation*. 2nd edition. Prentice Hall, New Jersey.
- [35] Herencia, M. Z. (2002). *Modelling Long-Range Dependence*. PhD Thesis, Pontificia Universidad Católica, Santiago, Chile.
- [36] Hosking, J. R. M. (1981). *Fractional differencing*. Biometrika, 68, p. 165-176.
- [37] Hwang, J. T. G., Ding, A. A. (1997). *Prediction intervals for artificial neural networks*. Journal of the American Statistical Association, 92, p. 748-757.
- [38] Joekes, S. (2002). *Aplicación de Redes Neuronales Robustas en Series de Tiempo: un estudio comparativo*. Tese de Mestrado, Univ. Córdoba, Argentina, jul/2002.
- [39] Joekes, S., Barbosa, E. P. (2003). *Time Series Modeling and Forecasting With Feedforward Neural Networks: A Comparative Study Using the Resex Data*. RP 13/03, IMECC/Unicamp.
- [40] Johnston, J., Dinardo, J. (1997). *Econometric Methods*. Fourth Edition, McGraw Hill, New York.
- [41] Kovács, Z. L. (1996). *Redes Neurais Artificiais: Fundamentos e Aplicações*. 2^a ed. Edição Acadêmica/USP.

- [42] Lin, T., Horne, B. G., Giles, C. L. (1996). *How Embedded Memory in Recurrent Neural Network Architectures Helps Learning Long-term Temporal Dependencies*. Computer Science Technical Report CS-TR-3626 and UMIACS-TR-96-28, University of Maryland, College Park, USA.
- [43] Maier, H. R., Dandy, G. C. (2000). *Neural Networks for the Prediction and Forecasting of Water Resources Variables: a Review of Modelling Issues and Applications*. Environmental Modelling & Software, 15, p. 101-124.
- [44] Mandic, D. P., Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction: Architectures, Learning Algorithms and Stability*. Wiley.
- [45] MATLAB (2003). *The Language of Technical Computing. v. 6.1*. MathWorks Inc.
- [46] McCulloch, C. E., Searle, S. R. (2001). *Generalized, Linear and Mixed Models*. Wiley.
- [47] McNelis, P. D., McAdam, P. (2004). *Forecasting Inflation with Thick Models and Neural Networks*. ECB Working Paper n. 352.
- [48] Morettin, P. A., Toloi, M. C. (2004). *Análise de Séries Temporais*. Edgard Blücher, São Paulo.
- [49] Nakamura, E. (2005). *Inflation Forecasting using a Neural Network*. Economics Letters, 86, n. 3, p. 373-378.
- [50] Nascimento Jr, C. L., Yoneyama, T. (2000). *Inteligência Artificial em controle e automação*. Editora Edgard Blücher Ltda, São Paulo.
- [51] Patel G.S. (2000). *Modeling nonlinear dynamics with extended Kalman filter trained recurrent multilayer perceptrons*. M.Sc. Thesis, McMaster University, Canadá.
- [52] Pole, A., West, M., Harrison, P.J. (1994). *Applied Bayesian Forecasting and Time Series Analysis*. Chapman & Hall, New York.
- [53] Reichardt, K., Timm, L. C. (2004). *Solo, Planta e Atmosfera: conceitos, processos e aplicações*. Manole, Barueri, São Paulo.
- [54] Ristic, B., Arulampalam, S., Gordon, N. (2004). *Beyond the Kalman Filter*. Artech House, London.
- [55] Rojas, R. (1996). *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin-N.Y.
- [56] Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*. In: Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations, ed. Rumelhart, D. E., McClelland, J. L. MIT Press, Cambridge.
- [57] S-PLUS User's Guide, Insightful Corporation, Seattle, WA

- [58] Sadownik, R., Barbosa, E. P. (1999). *Short-term forecasting of industrial electricity consumption in Brasil*. J. Forecast, 18, p. 215-224.
- [59] Searle, S. R. (1987). *Linear Models for Unbalanced Data*. Wiley.
- [60] Shumway, R. H., Stoffer, D. S. (2000). *Time Series Analysis and Its Applications*. Springer Verlag, New York.
- [61] Sussner, P. (1999). *Perceptrons*. In: Encyclopedia for Electrical and Electronics Engineering. John Wiley & Sons, Vol. 16, p. 44-59.
- [62] Tal, B., Nazareth, L. (1995). *Artificial Intelligence and Economic Forecasting*. Canadian Business Economics, 3, n. 3, Spring, 69-74
- [63] Tang, Z. e Fishwick, P. A. (1993), *Feed-Forward Neural Nets as Models for Time Series Forecasting*. Journal of Computing, 5, n. 4, p. 374-386
- [64] Timm, L. C., Barbosa, E. P., Souza, M. D., Dynia, J. F., Reichardt, K. (2003), *State-Space Analysis of Soil Data: An Approach Based on Space-Varying Regression Models*. Scientia Agricola, 60, 371-376.
- [65] Timm, L. C., Gomes, D. T., Barbosa, E. P., Reichardt, K., Souza, M. D., Dynia, J. F.(2005). *Neural networks and state space models for studying the soil attributes relationships: a comparative study*. Artigo submetido para publicação em periódico da área agronômica.
- [66] Van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph.D. Thesis, OGI School of Science & Engineering, OHSU, Estados Unidos.
- [67] Warner, B., Misra, M. (1996). *Understanding Neural Networks as Statistical Tools*. The American Statistician, 50, n. 4, p. 284-293.
- [68] Wei, W. S. (1990). *Time Series Analysis, Univariate and Multivariate Methods*. Addison Wesley.
- [69] Wendroth, O., Jürschik, P., Kersebaum, K.C., Reuter, H., Van Kessel, C., Nielsen, D.R. (1997). *Statistical approaches to the analysis of soil quality data*. In: *Soil Quality for Crop Production and Ecosystem Health*, ed. Gregorich, E. G., Carter, M. R.), Elsevier.
- [70] Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University.
- [71] West, M., Harrison, P. J. (1997). *Bayesian Forecasting and Dynamic Model*. Springer, London.

- [72] White, H. (1989). *Learning in artificial neural networks: a statistical perspective*. Neural Computation, 1, n. 4, p. 425-464.
- [73] Williams, R. J., Peng, J. (1990). *An efficient Gradient-Based Algorithm for On-line Training of Recurrent Network Trajectories*. Neural Computation, 2, p. 490-501.
- [74] Williams, R. J., Zipser, D. (1995). *Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity*. In: Backpropagation: theory, architectures, and applications, ed. Chauvin, Y., Rumelhart, D. E. Erlbaum, Hillsdale, New Jersey.
- [75] Wooldridge, J. M. (2003). *Introductory Econometrics: A Modern Approach*. South Western College Publishing.