



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

CAROLINE LUCAS CALHEIRANI

**ANÁLISE DE SENSIBILIDADE DO NVIDIA DIGITS AOS PARÂMETROS DO
PROCESSAMENTO DE IMAGENS**

LIMEIRA/SP
2020

CAROLINE LUCAS CALHEIRANI

**ANÁLISE DE SENSIBILIDADE DO NVIDIA DIGITS AOS PARÂMETROS DO
PROCESSAMENTO DE IMAGENS**

*Monografia apresentada à Faculdade de
Tecnologia da Universidade Estadual de
Campinas como parte dos requisitos exigidos para
a obtenção do título de Bacharel em Sistemas de
Informação.*

Orientador: Prof. Dr. André Franceschi de Angelis

LIMEIRA/SP
2020

CAROLINE LUCAS CALHEIRANI

**ANÁLISE DE SENSIBILIDADE DO NVIDIA DIGITS AOS PARÂMETROS DO
PROCESSAMENTO DE IMAGENS**

*Monografia apresentada à Faculdade de
Tecnologia da Universidade Estadual de
Campinas como parte dos requisitos exigidos para
a obtenção do título de Bacharel em Sistemas de
Informação.*

BANCA EXAMINADORA

Prof. Dr. André Franceschi de Angelis

Prof. Dra. Ieda Geriberto Hidalgo

Prof. Dr. Ulisses Martins Dias

LIMEIRA/SP
2020

AGRADECIMENTOS

Agradeço a Deus pelas oportunidades e pessoas que coloca em meu caminho. Aos meus pais, Ivanil e Ronaldo, minha gratidão por sempre acreditarem em mim, por me darem apoio, incentivo, educação e por sempre estarem ao meu lado. Ao meu orientador, Professor Dr. André Franceschi de Angelis, por todo apoio e paciência ao longo do meu TCC. Aos docentes: Dra. Ieda Geriberto Hidalgo e Dr. Ulisses Martins Dias por aceitarem o convite para compor a banca de meu trabalho de conclusão de curso. A doutoranda Thais Rocha pela colaboração no projeto e todo auxílio. A todas as pessoas que direta ou indiretamente contribuíram para a realização desse projeto. Por último, quero agradecer também à Faculdade de Tecnologia da UNICAMP e todo o seu corpo docente por toda experiência e conhecimento que adquiri durante essa etapa.

RESUMO

A classificação de imagens é uma área da tecnologia da informação que vem sendo utilizada para desenvolver pesquisas, criar novas tecnologias ou aprimorar algumas já existentes. Na classificação de imagens, pode ser usada a Inteligência Artificial (IA), que permite o reconhecimento de padrões para que, posteriormente, consigam ser mapeados. Dentre as diversas técnicas computacionais que englobam a IA, uma delas é representada pelas Redes Neurais, que podem reconhecer padrões e ter a capacidade de aprendizagem através de treinamento. Um dos programas existentes para treinar essas redes é o DIGITS. Ele facilita a tarefa de classificação de imagens, segmentação e detecção de objetos. Porém, para que seu resultado seja eficiente, é necessário que haja o melhor uso dos parâmetros de operação. Este trabalho teve por objetivo o estudo de alguns desses através da análise de sensibilidade, com o intuito de redução de treinamento e melhoria da acurácia. Para tanto, houve a definição dos parâmetros *Training epochs* (100, 200, 500 e 1.000), *Base learning rate* (0.01, 0.10 e 1.00) e *Standard networks* (AlexNet e GoogLeNet), configuradas com o *framework Caffe*. Além disso, alternou-se a quantidade de imagens (100, 500, 1000 e 2000), a fração de treinamento (25%, 50%, 75% e 100%) e quatro modelos de geração de sequência de imagens (*Moving Point*, *Rotating Point*, *Triple Cloud* e *Gaussian Cloud*). De acordo com os parâmetros escolhidos, foram realizados 1.152 testes, resultando em 10 conjuntos de dados relevantes para o objetivo descrito. Concluiu-se que, para os modelos analisados, independente da rede, é aconselhável o uso de poucas imagens e *training epochs*. Dependendo do modelo, é necessário a adequação da taxa de aprendizagem. Para os modelos *Moving Point*, *Rotating Point* e *Triple Cloud* a *standard networks* AlexNet, foi considerada a melhor para os treinamentos, assim como a fração de treinamento 75%. O modelo *Gaussian Cloud* mostrou-se contrário aos outros modelos, sendo indiferente a essas configurações. Essas conclusões permitem que haja redução no volume de dados, no tempo de processamento, e na sobrecarga de parâmetros, proporcionando a otimização de futuros trabalhos com o DIGITS.

Palavras-chaves: Sensibilidade de parâmetros. Classificação de imagens. DIGITS.

ABSTRACT

Image classification is an area of information technology that has been used to develop research, create new technologies, or improve existing ones. In the classification of images, Artificial Intelligence (AI) can be used, which allows the recognition of patterns so that they can be mapped. Among the various computational techniques that include AI, one of them is represented by Neural Networks, which can recognize patterns and have the ability to learn through training. One of the existing programs to train these networks is DIGITS. It facilitates the task of image classification, segmentation and object detection. However, for its result to be efficient, it is necessary to make the best use of the operating parameters. This work aimed to study some of these through sensitivity analysis, with the aim of reducing training and improving accuracy. For that, the parameters Training epochs (100, 200, 500 and 1,000), Base learning rate (0.01, 0.10 and 1.00) and Standard networks (AlexNet and GoogLeNet) were defined, configured with the Caffe framework. In addition, the number of images (100, 500, 1000 and 2000), the training fraction (25%, 50%, 75% and 100%) and four image sequence generation models (Moving Point, Rotating Point, Triple Cloud and Gaussian Cloud). According to the chosen parameters, 1,152 tests were carried out, resulting in 10 data sets relevant to the described objective. It was concluded that, for the analyzed models, regardless of the network, it is advisable to use a few images and training epochs. Depending on the model, it is necessary to adjust the learning rate. For the Moving Point, Rotating Point and Triple Cloud models, the AlexNet standard networks were considered the best for training, as well as the training fraction 75%. The Gaussian Cloud model proved to be contrary to the other models, being indifferent to these configurations. These conclusions allow a reduction in the volume of data, in the processing time, and in the overload of parameters, providing the optimization of future works with DIGITS.

Keywords: *Parameter sensitivity. Image classification. DIGITS.*

LISTA DE FIGURAS

FIGURA 1 - ÁREAS RELACIONADAS A IA	13
FIGURA 2 - <i>PERCEPTRON</i>	15
FIGURA 3 - ARQUITETURA DE UMA RNA	16
FIGURA 4 - ARQUITETURA DE UMA CNN	17
FIGURA 5 - GPU TITAN XP	18
FIGURA 6 - NOVO CONJUNTO DE DADOS PARA CLASSIFICAÇÃO DE IMAGEM	20
FIGURA 7 - TELA DE CONFIGURAÇÃO PUTTY	22
FIGURA 8 - INFORMAÇÕES DO SISTEMA ACESSADO PELO PUTTY	23
FIGURA 9 - TELA <i>MODELS</i> DO DIGITS	24
FIGURA 10 - IMAGENS SINTÉTICAS GERADAS PELO <i>FISGEN</i> COM O MODELO <i>MOVING POINT</i>	26
FIGURA 11 - TELA DE CADASTRO DE GRUPO DE DADOS NO DIGITS	28
FIGURA 12 - INDICAÇÃO DOS PARÂMETROS INALTERADOS	28
FIGURA 13 - EXEMPLO DOS RESULTADOS PLANILHADOS	29
FIGURA 14 - EXEMPLO PLANILHA INTERMEDIÁRIA	30
FIGURA 15 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA ALEXNET	32
FIGURA 16 - INFLUÊNCIA DO MODELO NOS PARÂMETROS GOOGLNET	33
FIGURA 17 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA ALEXNET E FRAÇÃO 75%	34
FIGURA 18 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA GOOGLNET E FRAÇÃO 75%	35
FIGURA 19 - INFLUÊNCIA DA FRAÇÃO DE TREINAMENTO PARA A REDE ALEXNET	36
FIGURA 20 - INFLUÊNCIA DA FRAÇÃO DE TREINAMENTO PARA A REDE GOOGLNET	37
FIGURA 21 - ANÁLISE DA TAXA DE APRENDIZAGEM PARA A REDE ALEXNET	38
FIGURA 22 - ANÁLISE DA TAXA DE APRENDIZAGEM PARA A REDE GOOGLNET	39
FIGURA 23 - ANÁLISE DAS REDES PARA OS MODELOS MP E GC	40
FIGURA 24 - INFLUÊNCIA DAS REDES E FRAÇÕES DE VALIDAÇÕES PARA O MODELO GC	40

LISTA DE TABELAS

TABELA 1 - COMPARAÇÃO ENTRE REDE ARTIFICIAL E BIOLÓGICA	14
TABELA 2 - ESPECIFICAÇÕES BÁSICAS DA GPU TITAN XP	18
TABELA 3 - DESCRIÇÃO DOS PARÂMETROS ESCOLHIDOS PARA TESTE	25
TABELA 4 - PARÂMETROS DE CONFIGURAÇÃO DO <i>FISGEN</i>	26
TABELA 5 - DESCRIÇÃO DOS MODELOS DE GERAÇÃO DE SEQUÊNCIA DE IMAGENS	27
TABELA 6 - CONFIGURAÇÃO DOS CONJUNTOS DE GRÁFICOS	31
TABELA 7 - RESUMO DOS RESULTADOS	41

SUMÁRIO

1. INTRODUÇÃO	10
2. REVISÃO BIBLIOGRÁFICA.....	12
2.1 Inteligência Artificial.....	12
2.2 Redes Neurais Artificiais.....	13
2.3 Deep Learning	16
2.4 GPU	18
2.5 DIGITS	19
3. METODOLOGIA	21
3.1 Recursos.....	21
3.2 Métodos	21
3.2.1 Configuração do Ambiente de Trabalho	22
3.2.2 Determinação dos Parâmetros e Valores.....	24
3.2.3 Configuração de Treinamento.....	27
3.2.4 Análise dos Resultados.....	29
4. RESULTADOS E DISCUSSÃO	31
5. CONCLUSÃO	42
6. REFERÊNCIAS	43

1. INTRODUÇÃO

O avanço da tecnologia tem acontecido numa velocidade muito rápida. Passou-se, em menos de uma década, de TV's digitais a robôs que fazem cirurgias sem intervenção médica, e esse crescimento tende a ficar mais rápido.

Graças ao desenvolvimento da tecnologia, o ramo da Inteligência Artificial (IA) ganhou grande atenção de pesquisadores e vem trazendo com ela uma revolução tecnológica. A IA abrange diversas áreas de estudo, como Matemática, Biologia, Engenharia, e também se conecta a várias linhas de pesquisas.

Uma dessas áreas é a das Redes Neurais Artificiais (RNAs), que tem o objetivo de proporcionar ao computador a capacidade de emular o comportamento das redes neurais biológicas. Essas redes demandam um grande número de cálculos por possuírem diversas camadas de neurônios. Por esse motivo, as Redes Neurais são usadas em computadores com *Graphics Processing Units* (GPU). Eles são processadores com alto poder computacional, que inicialmente foram criados para processar gráficos 2D e 3D e, posteriormente, usados para o processamento de dados e classificação de imagens.

As RNAs possuem habilidade de aprender e de melhorar seu desempenho através do seu treinamento. A rede é treinada a achar padrões em imagens, por exemplo, para posteriormente classificá-los. Normalmente esse aprendizado é supervisionado, ou seja, é dada a rede à resposta desejada para um certo padrão apresentado.

Existem diversos tipos de programas que fazem treinamento de redes utilizando GPU. Um deles é o *Deep Learning GPU Training System* (DIGITS), que é um sistema de treinamento de aprendizagem profunda desenvolvido pela empresa NVIDIA¹. O DIGITS é um *software* que utiliza muitos parâmetros de treinamento para a classificação de imagens e esses parâmetros têm grande influência no seu resultado.

O número elevado de parâmetros do DIGITS e as características de cada projeto podem tornar complexa a escolha de valores ótimos para treinamento, influenciando significativamente na acurácia. Durante o treinamento, é inviável testá-los, pois isso

¹ Nvidia - <https://www.nvidia.com/>

demandaria muito tempo. Entretanto, conhecer a influência dos parâmetros no processamento pode ajudar na seleção apropriada de valores.

Tendo em vista essa importância dos parâmetros para o aprendizado de uma rede, objetivou-se com o presente trabalho o teste da sensibilidade de alguns dos parâmetros de operação do DIGITS e seus respectivos valores: *Training epochs* (Épocas) - 100, 200, 500 e 1.000, *Base learning rate* (Taxa de aprendizagem) - 0.01, 0.10 e 1.00 e *Standard networks* (Redes) - AlexNet e GoogLeNet, configuradas com o *framework Caffe*. Além disso, alternou-se a quantidade de imagens (100, 500, 1000 e 2000), a fração de treinamento (25%, 50%, 75% e 100%) e quatro modelos de geração de sequência de imagem (*Moving Point*, *Rotating Point*, *Triple Cloud* e *Gaussian Cloud*).

Em geral, as análises de sensibilidades são feitas para verificar o comportamento de um modelo no intervalo de seus parâmetros. Ou seja, são usados diferentes valores de parâmetros para determinar o efeito que eles produzem no resultado final. Optou-se pela escolha desses parâmetros, pois são os que mais influenciam no resultado do treinamento.

Este trabalho contou com o apoio de um projeto de Doutorado desenvolvido pela discente Thais Rocha² (Título: Análise de Viabilidade e Desempenho de Redes Neurais *Deep Learning* aplicadas à Previsão de Vazão em Bacias Hidrográficas a partir de Imagens de Satélite), cujo objetivo geral é avaliar o desempenho de redes neurais artificiais profundas, *Deep Neural Network* (DNN), na classificação de imagens para previsão de vazões de rios.

Através desta pesquisa, espera-se que futuros trabalhos que utilizarão o DIGITS para classificar imagens sejam otimizados.

Os próximos capítulos apresentam os seguintes conteúdos: no Capítulo 2, a revisão bibliográfica; no Capítulo 3, a metodologia proposta para este trabalho; no Capítulo 4, os resultados obtidos e discussão; no Capítulo 5, a conclusão que se obteve com esse estudo; e por último, são mostradas as referências bibliográficas.

² <http://lattes.cnpq.br/1348914219247337>

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados cinco tópicos com conceitos importantes para o desenvolvimento deste trabalho: Inteligência Artificial, Rede Neural Artificial, *Deep Learning*, GPU e DIGITS.

2.1 Inteligência Artificial

A Inteligência Artificial (IA) é um dos campos mais estudados da ciência e tecnologia. Seu desenvolvimento se iniciou após a Segunda Guerra Mundial, mas só em 1956, este termo foi usado pela primeira vez (RUSSELL; NORVIG, 2013). Agora, com o avanço da tecnologia, o estudo da IA vem crescendo e se desenvolvendo mais e mais a cada ano. A IA é uma área de pesquisa da Ciência da Computação que busca simular a capacidade humana de resolver problemas através de métodos ou dispositivos computacionais (MACHADO, 2019).

Segundo Copeland (2017), esse termo é usado quando há a criação de projetos com capacidade de raciocinar, descobrir significados, generalizar ou aprender com a experiência passada.

Russel e Norving (2013) baseados no estudo de Alan Turing (1950), apontaram seis disciplinas que compõem a maior parte da IA e que estão em sua maioria presentes em computadores “inteligentes”:

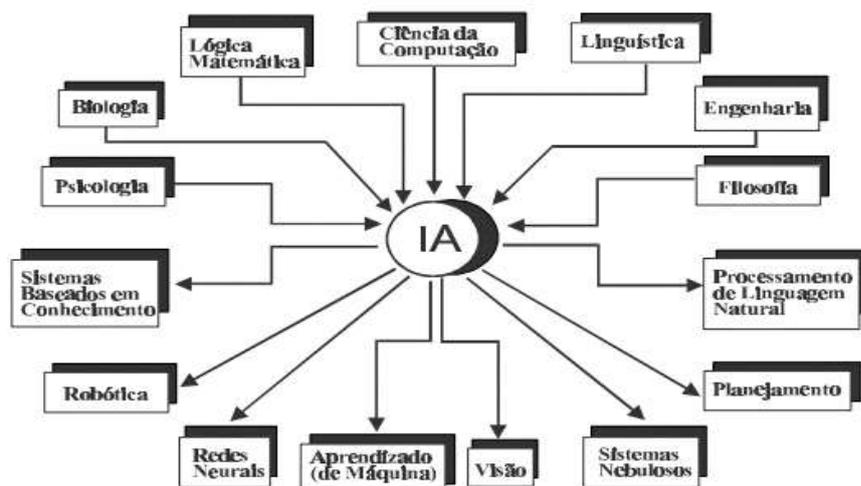
- Processamento de linguagem natural: para permitir que o computador se comunique em determinado idioma de maneira natural;
- Representação do conhecimento: para armazenar o que ele sabe e o que recebe de novo;
- Raciocínio automatizado: para poder responder a perguntas ou tirar conclusões a partir das informações nele armazenadas;
- Aprendizado de máquina: para se adaptar a novas circunstâncias;
- Visão computacional: para detectar objetos;
- Robótica: para movimentos e manipulação de objetos.

Nas definições de Kurzweil (1990, *apud* RUSSELL; NORVIG, 2013), pode-se entender a IA como a arte de criar máquinas que façam suas funções da mesma forma que uma pessoa a executa usando sua inteligência. Um exemplo é uma partida de xadrez. Aplica-se ao

computador conhecimento (como boas jogadas ou procedimentos de avaliação “inteligentes”) de tal maneira que ele consiga encontrar a melhor jogada em relação a partida em que ele está (ESCOLANO *et al.*, 2003).

Atualmente, a IA abrange uma variedade de subcampos, que vão desde o geral (aprendizagem e percepção), até tarefas específicas como por exemplo a partida de xadrez já citada, demonstrações de teoremas matemáticos, criação de poesias e diagnóstico de doenças. A IA é considerada relevante para qualquer tarefa intelectual humana e, por este motivo, ela é um campo universal (RUSSELL; NORVIG, 2013).

FIGURA 1 - ÁREAS RELACIONADAS A IA



Fonte: MONARD; BARANAUKAS, 2000.

A IA também está se desenvolvendo em várias linhas de pesquisas a fim de fornecer ao computador “habilidades humanas”, conforme mostra Figura 1. Uma dessas linhas é a Rede Neural Artificial, que é detalhada na próxima seção.

2.2 Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é desenvolvida a partir de modelos matemáticos e tem esse nome pois possui características de desempenho em comum com as redes neurais biológicas. A RNA é um sistema de processamento de dados que, por ser inspirado nas redes

neurais biológicas, é estudado por várias áreas, como neurofisiologia, psicologia, engenharia, computação e física (SILVA, 1998).

O cérebro humano possui cerca de 86 bilhões de neurônios (VARELLA, 2019). Esses neurônios têm papel essencial para o comportamento, raciocínio e funcionamento do corpo humano. Os neurônios são formados por dendritos, que são um conjunto de terminais de entrada, e pelos axônios que são os terminais de saída. Eles se comunicam através de sinapses, que são regiões onde dois neurônios entram em contato e impulsos nervosos são transmitidos entre eles (CARVALHO, 2019).

A ideia da RNA é processar informações tendo como princípio a organização dos neurônios. O cérebro é capaz de tomar decisões baseado no que ele aprendeu, então uma RNA deve saber fazer o mesmo. Seguindo essa linha de pensamento, podemos interpretar uma rede neural como um esquema de processamento capaz de armazenar conhecimento baseado em aprendizagem e aplicar esse conhecimento em acontecimentos futuros (GOLDSCHMIDT, 2010). A Tabela 1 resume a relação de uma RNA com uma rede biológica.

TABELA 1 - COMPARAÇÃO ENTRE REDE ARTIFICIAL E BIOLÓGICA

Modelo Natural	Modelo Artificial
Cérebro	RNA
Neurônio Biológico	Neurônio artificial/ elementos processadores
Rede de Neurônio	Estrutura em camadas
10 bilhões de Neurônios ³	Centenas/ milhares de neurônios
Aprendizado	Aprendizado
Generalização	Generalização
Associação	Associação
Reconhecimento de Padrões	Reconhecimento de Padrões

Fonte: GOLDSCHMIDT, 2010.

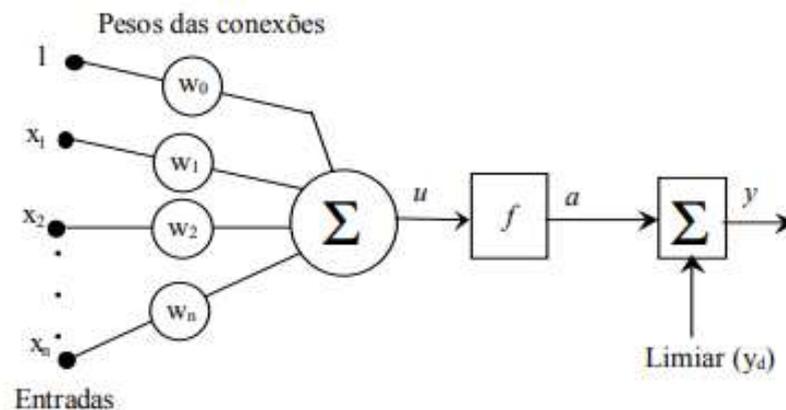
O primeiro modelo de neurônio em RNAs foi proposto por McCulloch e Pitts em 1943 e a primeira rede neural capaz de aprender padrões, denominada *perceptron*, foi criada

³ Atualmente se considera que são 86 bilhões de neurônios, (VARELLA, 2019).

por Frank Rosenblatt em 1958, que segundo Fernandes (2015) pode ser resumida da seguinte maneira (Figura 2):

1. Sinais são apresentados à entrada;
2. Cada um deles é multiplicado por um peso que indica a influência que terá na saída da unidade;
3. É feita a soma ponderada dos sinais que produz um nível de atividade;
4. Se esse nível exceder um certo limite a unidade produz uma determinada resposta de saída.

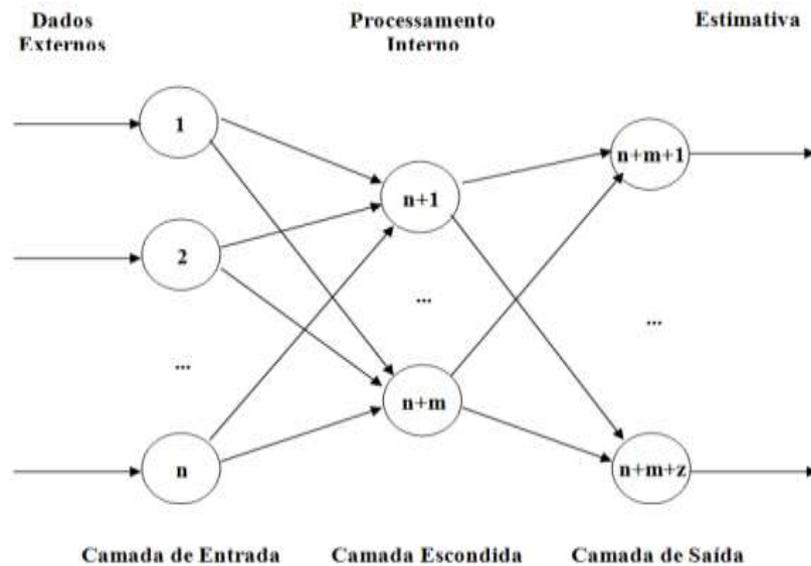
FIGURA 2 - PERCEPTRON



Fonte: SILVA, 1998.

Com semelhança ao sistema biológico, uma RNA possui um sistema de neurônios e conexões dadas por valores reais chamados de pesos. Na Figura 3, pode-se ver uma ilustração da arquitetura de uma rede neural simples. Os neurônios são representados pelos círculos e as linhas representam os pesos das conexões. Eles são divididos em múltiplas camadas, as que recebem os dados são as camadas de entrada, as que mostram os resultados são as camadas de saída. Existe ainda a camada interna, também conhecida como camada escondida, que tem a capacidade de classificação de dados em classes e de solucionar problemas de regressão não-linear. Uma RNA pode ter mais de uma camada escondida (*Multilayer Perceptron*), mas isso depende da complexidade do problema (GOLDSCHMIDT, 2010).

FIGURA 3 - ARQUITETURA DE UMA RNA



Fonte: GOLDSCHMIDT, 2010.

Existem dois tipos de aprendizado para estas redes: o aprendizado supervisionado e o não-supervisionado. No aprendizado supervisionado, uma pessoa indica para a rede a resposta desejada para determinado padrão de entrada, a fim de minimizar erros. No aprendizado não-supervisionado, o computador não recebe as respostas desejadas para os padrões de entrada, ele as acha sozinho.

As RNAs têm um processo de aprendizagem iterativo, pois possuem a capacidade de aprimorar seu próprio desempenho à medida em que vão interagindo com o mundo externo (REZENDE, 2003). Elas são usadas algumas vezes para descoberta de valores, mas estão mais associadas ao processo de classificação, seja de imagens, conjuntos de sintomas, animais entre outros. Uma das técnicas usadas para melhorar esse aprendizado é o *Deep Learning*, que é visto a seguir.

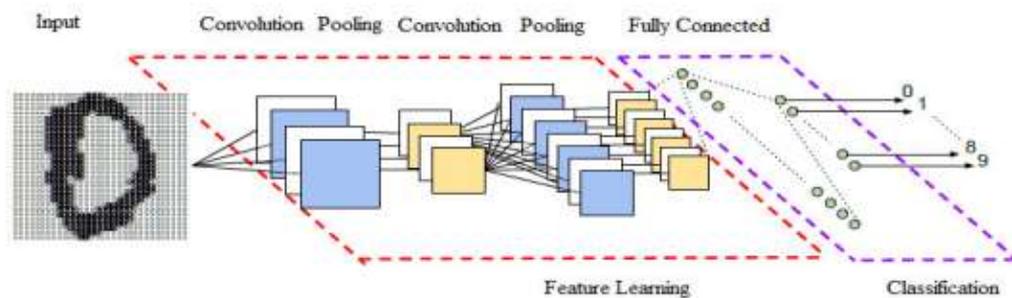
2.3 Deep Learning

Deep Learning (DL) é um subconjunto de IA e Aprendizado de Máquina que usa RNA com múltiplas camadas e que, segundo LeCun *et al.* (2015), permite com que modelos computacionais aprendam representação de dados com os mais variados níveis de abstração, gerando uma grande melhora nos resultados de reconhecimento de fala, reconhecimento visual de objetos e muitos outros domínios.

DL se difere de outras técnicas de aprendizado de máquina, pois ela tem a capacidade de aprendizado automático a partir de representação de uma grande quantidade de dados sem a necessidade de introduzir regras condicionais a mão ou conhecimento do domínio humano. Sua estrutura permite que possam aprender com dados brutos (NVIDIA, 2019).

A Rede Neural Convolutiva (*Convolutional Neural Network* - CNN) é uma rede DL utilizada para o processamento e análise de imagens. A CNN possui camadas de convolução e camadas de *pool*, que se alternam entre si, conforme Figura 4.

FIGURA 4 - ARQUITETURA DE UMA CNN



Fonte: VARGAS *et al.*, 2016.

Inicialmente, aplica-se o dado de entrada à camada de convolução. Essa camada é composta por diversos neurônios e cada um aplica um filtro em um pedaço específico da imagem e calcula sua matriz de peso. Em seguida, essas informações passam pela camada de agrupamento (*Pooling*), que simplifica as informações da camada anterior reduzindo a dimensionalidade dos dados na rede. Posteriormente, passa-se para a camada totalmente conectada. Seu papel é, para cada classe de resposta, traçar um caminho de decisão a partir das informações vindas das camadas anteriores. Essa camada influencia o aprendizado dos filtros e o resultado da rede (VARGAS *et al.*, 2016).

Hoje, existem diversas ferramentas de treinamento de uma CNN. Uma delas é o DIGITS, que é um sistema de treinamento que utiliza GPU e foi desenvolvido pela NVIDIA.

2.4 GPU

A *Graphics Processing Unit* (GPU) está presente nas placas de vídeos ou então em placas-mãe. Ela é um processador especializado que faz operações ligadas a objetos gráficos e mostra uma eficiência muito grande em relação ao paralelismo. As GPU's foram criadas originalmente para dar um efeito visual 3D em superfícies 2D, normalmente encontrados em jogos e animação. Mas, nos dias de hoje, a GPU passou a ser usada em qualquer problema com altas necessidades computacionais e processamento paralelo.

FIGURA 5 - GPU TITAN XP



Fonte: (NVIDIA, 2019).

Neste trabalho, foi utilizada a TITAN Xp, que é uma GPU da NVIDIA, ilustrada na Figura 5 e suas especificações encontram-se na Tabela 2.

TABELA 2 - ESPECIFICAÇÕES BÁSICAS DA GPU TITAN XP

Especificação dos Mecanismos	
Núcleos NVIDIA CUDA	3.840
<i>Boost Clock</i> (MHz)	1.582
Especificação de Memória	
Velocidade da Memória	11.4 Gbps
Configuração de Memória Padrão	12 GB GDDR5X
Largura da Interface da Memória	384-bit
Largura da Banda da Memória (GB/s)	547.7 GB/s

Fonte: (NVIDIA, 2019).

As GPUs são bastante utilizadas em sistemas embarcados, jogos, estações de trabalho, etc. Para o aprendizado de máquina, é usada uma grande quantidade de dados e isso exige alto grau de processamento paralelo. Devido ao seu alto poder computacional, a GPU tem sido a plataforma escolhida para esse fim (PARKER, 2017), tornando fundamental para o funcionamento do DIGITS que é detalhado a seguir.

2.5 DIGITS

O *Deep Learning GPU Training System* (DIGITS) foi lançado em 2015 com intuito de desenvolvimento, treinamento e visualização de Redes Neurais Profundas. Seu objetivo é simplificar a construção de um modelo de aprendizado profundo através dos *frameworks* populares: *Caffe*, *Torch* e *Tensorflow* e das CNN's: LeNet, AlexNet e GoogLeNet. Seu trabalho é paralelizado com as GPU's e, por esse motivo, ele não funciona em uma máquina sem a presença de uma (JANAKIRAM, 2018).

O DIGITS pode ser usado para classificação de imagens, segmentação, tarefas de detecção de objetos entre outros. Sua interface inclui desde gerenciamento de dados até treinamento de modelos. É um sistema interativo com uma interface intuitiva baseada em navegador, permitindo que seus usuários tenham visualização do comportamento de sua rede em tempo real, além de possuir código aberto (NVIDIA, 2019).

Na Figura 6, pode-se observar a tela de configuração para um novo modelo de classificação de imagens. Percebe-se que há um conjunto de parâmetros a serem preenchidos. Por exemplo, à esquerda, em *Select dataset*, é possível escolher um conjunto de dados criado anteriormente. Esses dados são referentes ao tipo da imagem, tamanho e quantidade de imagens que serão utilizadas. Há também as opções do *Solver option* onde são definidos os parâmetros de rede para o treinamento, são eles: *Training epochs*, *Blob format*, *Base learning rate* entre outros (NVIDIA, 2019). Alguns destes parâmetros têm valores reais e outros não possuem limitante, sendo o usuário quem os definem. Por último, em *Standard Networks*, é escolhido o modelo de rede que será utilizado, dentre os disponíveis no software: LeNet, AlexNet ou GoogLeNet, e um dos *frameworks*: *Caffe*, *Torch* ou *Tensorflow*.

FIGURA 6 - NOVO CONJUNTO DE DADOS PARA CLASSIFICAÇÃO DE IMAGEM

New Image Classification Model

Select Dataset

mnist

Done 04:43:48 PM

Image Size
28x28

Image Type
GRAYSCALE

DB backend
Imdb

Create DB (train)
45002 images

Create DB (val)
14998 images

Python Layers

Server-side file

Use client-side file

Solver Options

Training epochs: 30

Snapshot Interval (in epochs): 1

Validation Interval (in epochs): 1

Random seed: [none]

Batch size: [network defaults] multiples allowed

Batch Accumulation

Solver type: Stochastic gradient descent (SGD)

Base Learning Rate: 0.01 multiples allowed

show advanced learning rate options

Data Transformations

Subtract Mean: image

Crop Size: none

Standard Networks | Previous Networks | Pretrained Networks | Custom Network

Caffe | Torch | Tensorflow

Network	Details	Intended image size
<input checked="" type="radio"/> LeNet	Original paper [1998]	28x28 (gray) Customize
<input type="radio"/> AlexNet	Original paper [2012]	256x256
<input type="radio"/> GoogLeNet	Original paper [2014]	256x256

Fonte: autor.

A análise de sensibilidade foi importante devido ao grande número de combinações de parâmetros. A metodologia utilizada para a sua realização e os resultados obtidos são apresentados nos capítulos seguintes.

3. METODOLOGIA

Este capítulo apresenta os recursos e métodos utilizados neste projeto.

3.1 Recursos

Foram utilizados os seguintes recursos:

- Imagens sinteticamente geradas pelo programa *Fisgen (Fast Image Sequence)*;
- Software NVIDIA *Deep Learning GPU Training System (DIGITS)*, versão 6;
- GPU NVIDIA Titan XP;
- Microcomputadores, com processador Core i5;
- Navegador Firefox;
- Sistemas Operacionais Linux e Windows;
- GNUPlot 5.4.0.

3.2 Métodos

Em linhas gerais, o trabalho se dividiu em 5 etapas listadas a seguir.

1. Configuração do ambiente de trabalho: instalação e configuração de *hardware* e software, especificamente a ferramenta DIGITS;
2. Determinação dos parâmetros dentre os existentes no DIGITS: foram escolhidos para estudo os parâmetros *Training epochs*, *Base Learning rate* e *Standard networks*, quantidade de imagens, modelo e fração de treinamento;
3. Determinação dos valores para teste: dentre os parâmetros escolhidos, foi selecionada uma faixa discreta entre 3 e 4 valores para cada um deles. Durante os testes, enquanto um parâmetro estava sendo alternado, os demais ficaram fixos;
4. Configuração de treinamento: criação do grupo de dados, criação dos modelos;
5. Análise dos resultados: foram analisados o TOP-1 *Accuracy* de cada teste realizado e feita a análise para determinar a influência do parâmetro na rede. Pode-se ver com mais detalhes os resultados obtidos no capítulo 4.

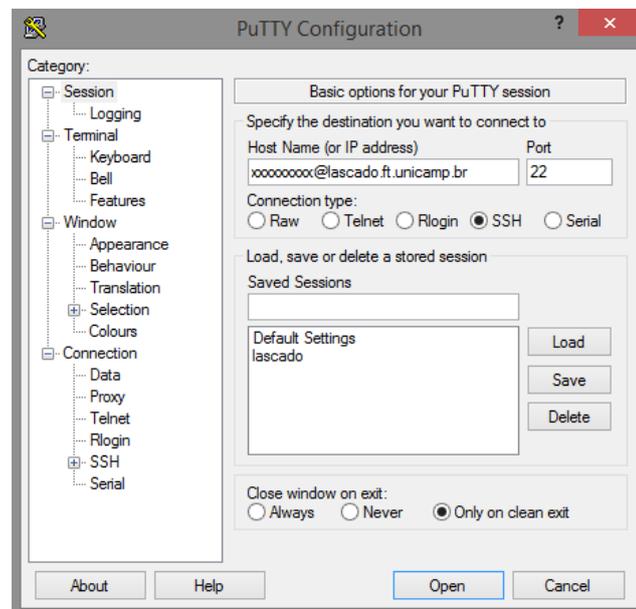
Essas cinco etapas são detalhadas a seguir.

3.2.1 Configuração do Ambiente de Trabalho

O software DIGITS já havia sido configurado anteriormente, para a realização de outros projetos. Nessa configuração feita, houve instalação e configuração da GPU Titan XP NVIDIA e, posteriormente, do DIGITS.

Para ter acesso ao DIGITS remotamente, foi necessária a criação de uma conta para acesso ao Laboratório de Simulação e de Computação de Alto Desempenho (LaSCADo), local onde está a máquina de trabalho que o DIGITS foi instalado. O acesso ao LaSCADo é permitido ao setor de TI da Faculdade de Tecnologia (FT) e pesquisadores.

FIGURA 7 - TELA DE CONFIGURAÇÃO PUTTY



Fonte: autor.

No terminal de trabalho foi necessária fazer a instalação do software de emulação de terminal PuTTY⁴, e configurá-lo conforme Figura 7. Para acessá-lo é preciso abrir o *Prompt de Comando* e digitar:

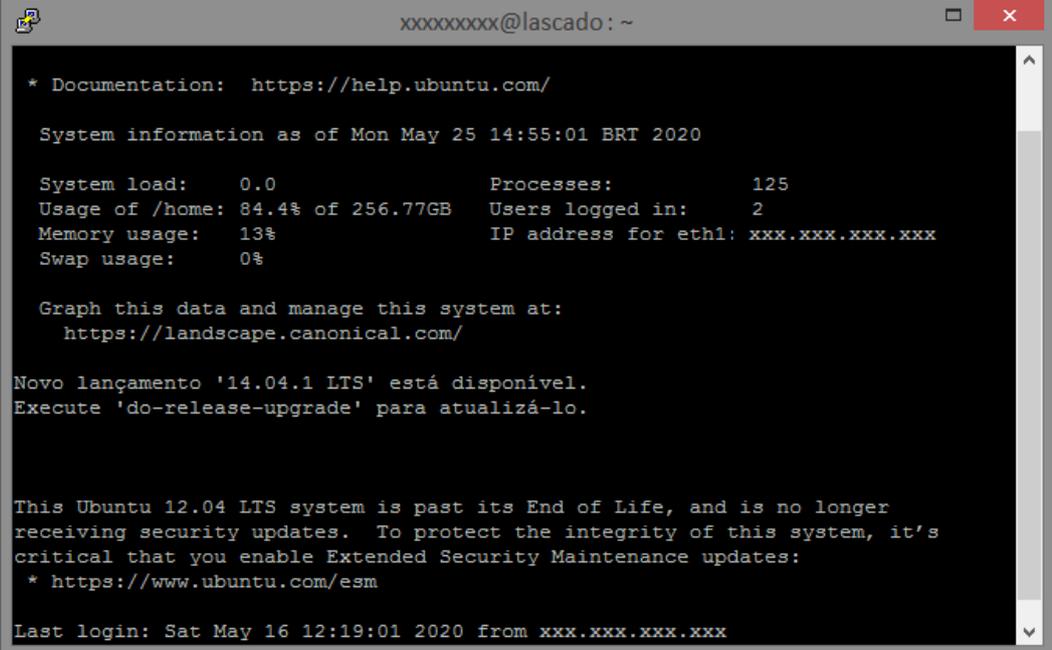
```
putty -D 1234 email-lascado
```

⁴ PuTTY - <https://www.putty.org/>

Em seguida, na página do PuTTY, é solicitada a senha do usuário e, posteriormente, são mostradas as informações do sistema (Figura 8). Além das configurações listadas, foram necessários os seguintes ajustes no navegador:

1. Em Opções;
2. Clicar em Configurar conexão;
3. Escolher a opção “Configuração manual de proxy”;
4. No campo Domínio SOCKS inserir: 127.0.0.1;
5. No campo Porta inserir: 1234;
6. Selecionar a opção SOCKS v4.

FIGURA 8 - INFORMAÇÕES DO SISTEMA ACESSADO PELO PUTTY



```
* Documentation: https://help.ubuntu.com/

System information as of Mon May 25 14:55:01 BRT 2020

System load:      0.0                Processes:        125
Usage of /home:  84.4% of 256.77GB    Users logged in:  2
Memory usage:    13%                IP address for eth1: xxx.xxx.xxx.xxx
Swap usage:      0%

Graph this data and manage this system at:
https://landscape.canonical.com/

Novo lançamento '14.04.1 LTS' está disponível.
Execute 'do-release-upgrade' para atualizá-lo.

This Ubuntu 12.04 LTS system is past its End of Life, and is no longer
receiving security updates. To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
* https://www.ubuntu.com/esm

Last login: Sat May 16 12:19:01 2020 from xxx.xxx.xxx.xxx
```

Fonte: autor.

Por fim, para ter acesso as funcionalidades do DIGTS, basta digitar o endereço do servidor onde ele se encontra, na porta 5000. Por exemplo: xxx.xxx.xxx.xxx:5000⁵. O ambiente do software é mostrado no tópico a seguir.

⁵ Endereço real omitido por segurança da rede.

3.2.2 Determinação dos Parâmetros e Valores

Com o acesso permitido ao DIGITS, a primeira página apresentada é a *Models* onde são mostrados os testes realizados, conforme Figura 9. Além da opção *Dataset* que mostra os grupos de dados criados e a *Pretrained Models* onde são exibidos os modelos que já foram treinados.

FIGURA 9 - TELA MODELS DO DIGITS

The screenshot shows the DIGITS web interface. At the top, there's a navigation bar with 'Home' and '117 GPU available'. Below that, there are tabs for 'Datasets (10)', 'Models (20)', and 'Pretrained Models (1)'. A 'Group Jobs' checkbox is checked. The main content is a table with columns: 'name', 'status', 'hostname', 'state', 'epoch', and 'submitted'. The table lists various training jobs, including 'Case1_OC_1_199', 'Case1_HP_1_199', and 'Case1_TC_1_199', each with multiple sub-jobs. The 'status' column shows 'Done' for all jobs, and the 'epoch' column shows values ranging from 96 to 128. The 'submitted' column shows dates in April 2018.

Fonte: autor.

Na página de classificação de modelos, conforme visto na Figura 6, na opção *Solver Options* observa-se a existência de 10 parâmetros:

- *Server-side file;*
- *Batch size;*
- *Training epochs;*
- *Batch accumulations;*
- *Snapshot interval(in epochs);*
- *Blob format;*
- *Validation interval(in epochs);*
- *Solver type;*
- *Random seed;*
- *Base learning rate.*

Desses parâmetros, foram escolhidos para estudo: *Training epochs* e *Base learning rate*, além do *Standard networks* (Tabela 3), mantendo os outros valores como padrão estabelecido pelo software. Optou-se por essa escolha, pois são os que mais influenciam no resultado do treinamento. O *Training epochs* indica quantos ciclos de treinamento a rede irá fazer, acarretando em mudanças nos pesos da matriz interna. Por se tratarem de modelos estocásticos, o *Learning rate* indica quantas vezes o peso do modelo será atualizado e mostra o quão agressivo será o ajuste da curva de aprendizado. O *Standard networks* mostra os modelos de rede disponíveis no software: LeNet, AlexNet e GoogLeNet, permitindo que o código fonte destes seja alterado nos *frameworks*: *Caffe*, *Torch* ou *Tensorflow*.

TABELA 3 - DESCRIÇÃO DOS PARÂMETROS ESCOLHIDOS PARA TESTE

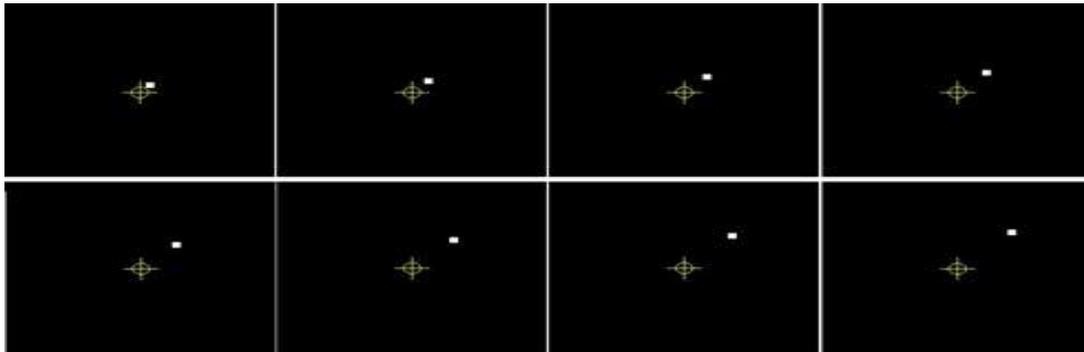
Parâmetro	Descrição	Valores para Teste
<i>Training epochs</i>	Número de épocas que serão utilizadas para treinamento da rede.	100, 200, 500 e 1.000
<i>Base Learning Rate</i>	Afeta a velocidade com que a rede aprende.	0.01, 0.1 e 1.00
<i>Standard Networks</i>	Redes existentes no DIGITS.	AlexNet e GoogLeNet
<i>Frameworks</i>	Frameworks disponíveis no DIGITS.	<i>Caffe</i>
Modelos	Modelos de geração de sequência de imagem.	<i>Moving Point, Rotating Point, Triple Cloud e Gaussian Cloud</i>
Quantidade de imagens	Total de imagens utilizadas do conjunto.	100, 500, 1000 e 2000
Fração de treinamento	Porcentagem de imagem utilizada para treinamento.	25%, 50%, 75% e 100%

Fonte: autor.

Visto que, alguns parâmetros têm valores contínuos, foi necessário limitá-los para que fosse possível a realização dos testes. Deste modo, foi selecionada uma faixa discreta entre 3 e 4 valores para cada um escolhido. Para a base de dados de imagens utilizadas no treinamento, foram feitas as seguintes configurações:

- Foi utilizado o programa *Fisgen*⁶ (*Fast Image Sequence*), onde foi criado um conjunto de imagens sintéticas. Essas imagens são criadas seguindo modelos que definem padrões: distância euclidiana de um ponto até o centro da imagem; medida do ângulo de arco; medida do raio de um círculo e taxa de cobertura. Uma sequência de imagens gerada pode ser vista na Figura 10.

FIGURA 10 - IMAGENS SINTÉTICAS GERADAS PELO *FISGEN* COM O MODELO *MOVING POINT*



Fonte: *Fisgen*.

O *Fisgen* possui nove parâmetros como, resolução da imagem, quantidade de imagens do conjunto e nível de ruído. Todos os parâmetros estão listados na Tabela 4.

TABELA 4 - PARÂMETROS DE CONFIGURAÇÃO DO *FISGEN*

Parâmetro	Descrição
Hr	Altura da imagem; resolução em pixel
Vr	Largura da imagem; resolução em pixel
S	Indica a quantidade de imagens do conjunto
O	Indica a taxa de deslocamento temporal
N	Nome do grupo de imagens; nome do diretório
G	Classe Java que implementa o modelo
Rg	Número de arquivos agrupados em um diretório; valor padrão é 10
Rn	Indica o nível de ruído em percentual
Rl	Indica se adiciona uma linha de referência desde o centro da imagem

Fonte: autor.

⁶ *Fisgen* – Programa de computador destinado à geração rápida de grandes conjuntos de imagens de tamanho arbitrário para suporte ao treinamento e validação de desempenho de redes neurais de aprendizagem profunda. Foi escrito em Linguagem Java e está em sua versão 1.10. Deve ser registrado brevemente junto ao INPI para proteção de propriedade intelectual de seus autores, André Franceschi de Angelis e Thais Rocha.

Para efetuar os testes foram utilizados os modelos descritos na Tabela 5. A partir desses modelos, o *Fisgen* gera uma sequência de imagens com desenho de um ponto ou de um conjunto de pontos. Deve se considerar as imagens como uma animação, visto que cada uma tem a posição do ponto um pouco deslocada em relação à anterior de tal forma que, ao observar em sequência se tem a impressão de movimento.

TABELA 5 - DESCRIÇÃO DOS MODELOS DE GERAÇÃO DE SEQUÊNCIA DE IMAGENS

Modelo	Desenho	Movimento	Medida
<i>Moving Point (MP)</i>	Ponto	Cruzar a diagonal da imagem da esquerda para direita	Distancia Euclidiana do ponto até o centro
<i>Rotating Point (RP)</i>	Ponto	Caminhar ao longo de um traçado circular fixo desenhado	Ângulo de arco
<i>Gaussian Cloud (GC)</i>	Conjunto de Pontos	Cruzar a diagonal da imagem, em formato de uma nuvem	Taxa de cobertura de um retângulo centralizado fixo
<i>Triple Cloud (TP)</i>	Três Conjuntos de Pontos	Cruzar a diagonal da imagem, em formato de agregação de nuvens	Taxa de cobertura de um retângulo centralizado fixo

Fonte: ANGELIS *et al.*, 2019 - adaptada.

Posteriormente, foi necessário fazer a configuração da ferramenta DIGITS criando o grupo de dados e os modelos, conforme descrito a seguir.

3.2.3 Configuração de Treinamento

Esta etapa divide-se em duas atividades:

1. Criar grupo de dados para treinar a CNN:

Cada grupo possui especificações relacionadas ao nome do conjunto e foi definida a percentagem de imagens de validação, mantendo os parâmetros *Image Type*, *Image size*, *Resize Transformation* e *Minimum samples per class* com os valores padrões, conforme Figura 11.

FIGURA 11 - TELA DE CADASTRO DE GRUPO DE DADOS NO DIGITS

Fonte: autor.

As proporções utilizadas entre os treinamentos e validações foram 25%, 50%, 75% e 100% de imagens, dando um total de 64 grupos de dados.

2. Criar modelo de classificação:

FIGURA 12 - INDICAÇÃO DOS PARÂMETROS INALTERADOS

Fonte: autor.

Foi determinado que, para cada modelo criado, os valores dos parâmetros não testados se manteriam no padrão estabelecido pelo software. Veja os parâmetros indicados na Figura 12.

Na última etapa do projeto foram realizados os testes e os cálculos que são mostrados a seguir.

3.2.4 Análise dos Resultados

Ao longo do projeto foram executados 1.152 testes, alternando os parâmetros citados anteriormente. Observou-se que a variação entre os valores de 500 e 1.000 épocas se mostram pequenas, então decidiu-se que os testes com 1.000 épocas de treinamento não fossem mais realizados.

Todos os testes foram realizados no DIGITS e, posteriormente, os TOP-1 Accuracy (resultados) que representam o quanto a rede conseguiu absorver com o treinamento, foram armazenados em planilhas do Excel, conforme Figura 13.

FIGURA 13 - EXEMPLO DOS RESULTADOS PLANILHADOS

M	N	O	P	Q	R	S	T	U	V
Snapshot Validation	Blob	Solver	Rate	alexnet					
1	1	NVCaffe	SGBD	0.01					
25x75 - TOP 1									
MP					RP				
Epocas	Imagens				Epocas	Imagens			
	100	500	1000	2000		100	500	1000	2000
100	58.33%	53.23%	42.97%	47.9%	100	60.0%	34.4%	19.52%	8.76%
200	50.0%	45.16%	49.8%	54.31%	200	60.0%	33.6%	18.33%	7.17%
500	25.0%	54.03%	59.04%	57.72%	500	56.0%	33.6%	15.54%	6.77%
1000	41.67%	50.0%	59.04%	61.32%	1000				
TC					GC				
Epocas	Imagens				Epocas	Imagens			
	100	500	1000	2000		100	500	1000	2000
100	81.33%	92.8%	74.7%	70.94%	100	75.0%	83.87%	89.16%	87.58%
200	89.33%	96.8%	79.12%	74.95%	200	83.33%	84.68%	87.95%	87.98%
500	81.33%	96.0%	80.72%	84.57%	500	83.33%	84.68%	93.57%	90.98%
1000					1000				

Fonte: autor.

Os dados planilhados foram transcritos para uma nova planilha intermediária, em um formato que permite automação dos processos (Figura14). Em seguida, eles foram

agrupados seguindo os critérios apresentados na Tabela 6. Em princípio, foram feitos testes com tipos variados de gráficos e, ao final, decidiu-se por usar conjuntos de gráficos do tipo mapa de calor. Para tanto, foi desenvolvido um programa em Java para transformar a planilha intermediária em *scripts* do *software* GNUPlot. Os gráficos foram exportados no formato jpg para análise visual.

FIGURA 14 - EXEMPLO PLANILHA INTERMEDIÁRIA

```
Grupo, Framework, Network, Top, Fraction, Rate, Model, Epocas, 100img, 500img, 1000img, 2000img

1, NVCaffe, Alexnet, 1, 25x75, 0.10, MP, 100, 37.50, 12.10, 12.85, 13.83
1, NVCaffe, Alexnet, 1, 25x75, 0.10, MP, 200, 16.67, 12.10, 12.85, 13.83
1, NVCaffe, Alexnet, 1, 25x75, 0.10, MP, 500, 33.33, 15.32, 12.85, 15.23
1, NVCaffe, Alexnet, 1, 25x75, 0.10, TC, 100, 69.33, 68.00, 10.44, 22.44
1, NVCaffe, Alexnet, 1, 25x75, 0.10, TC, 200, 70.67, 5.60, 33.33, 21.84
1, NVCaffe, Alexnet, 1, 25x75, 0.10, TC, 500, 54.67, 5.60, 10.44, 22.44
1, NVCaffe, Alexnet, 1, 25x75, 0.10, RP, 100, 60.00, 34.40, 16.33, 7.17
1, NVCaffe, Alexnet, 1, 25x75, 0.10, RP, 200, 72.00, 34.67, 13.94, 7.17
1, NVCaffe, Alexnet, 1, 25x75, 0.10, RP, 500, 36.00, 28.00, 13.15, 7.17
1, NVCaffe, Alexnet, 1, 25x75, 0.10, GC, 100, 62.50, 81.45, 95.58, 90.78
1, NVCaffe, Alexnet, 1, 25x75, 0.10, GC, 200, 62.50, 81.45, 95.58, 90.78
1, NVCaffe, Alexnet, 1, 25x75, 0.10, GC, 500, 62.50, 81.45, 95.58, 90.78

2, NVCaffe, GoogleNet, 1, 25x75, 0.10, MP, 100, 16.67, 15.32, 12.85, 13.83
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, MP, 200, 16.67, 15.32, 12.85, 15.23
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, MP, 500, 16.67, 15.32, 12.85, 15.23
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, TC, 100, 54.67, 5.6, 10.44, 21.84
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, TC, 200, 54.67, 5.6, 10.44, 21.84
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, TC, 500, 54.67, 5.6, 10.44, 21.84
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, RP, 100, 36, 29.07, 14.34, 7.17
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, RP, 200, 36, 29.07, 14.34, 7.17
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, RP, 500, 36, 29.07, 14.34, 7.17
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, GC, 100, 62.5, 81.45, 95.58, 90.78
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, GC, 200, 62.5, 81.45, 95.58, 90.78
2, NVCaffe, GoogleNet, 1, 25x75, 0.10, GC, 500, 62.5, 81.45, 95.58, 90.78
```

Fonte: autor.

No processo de análise, cada grupo de quatro gráficos forma uma figura a ser examinada, na maioria das vezes com variação de um único parâmetro. A comparação visual permitiu entender o comportamento de cada um dos parâmetros individualmente para as redes analisadas, conforme apresentado a seguir.

4. RESULTADOS E DISCUSSÃO

Os gráficos contidos no presente trabalho, foram agrupados de acordo com a Tabela 6, onde o símbolo de “*” indica que todos os valores estabelecidos para cada parâmetro foram utilizados. A fração de treinamento (*Fraction*), na maioria das comparações, encontra-se na percentagem 75, visto que é a mais utilizada. A taxa de aprendizagem foi fixada no valor 0.10, na maioria das vezes, pois é o intermédio dentre os valores escolhidos para teste.

TABELA 6 - CONFIGURAÇÃO DOS CONJUNTOS DE GRÁFICOS

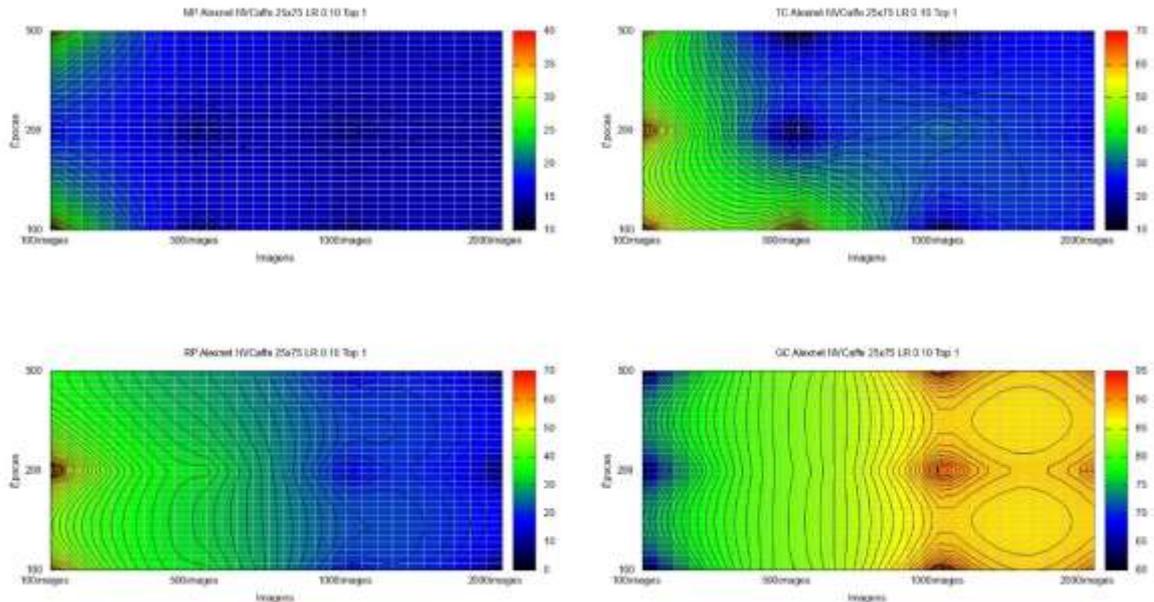
Comparação	Parâmetros							
	Framework	Network	Top	Fraction	Rate	Model	Gráficos	Grupo
1	NVCaffe	Alexnet	1	25x75	0,10	*	4	01
2	NVCaffe	GoogleNet	1	25x75	0,10	*	4	02
3	NVCaffe	Alexnet	1	75x25	0,10	*	4	03
4	NVCaffe	GoogleNet	1	75x25	0,10	*	4	04
5	NVCaffe	Alexnet	1	*	0,10	GC	4	05
6	NVCaffe	GoogleNet	1	*	0,10	GC	4	06
7	NVCaffe	Alexnet	1	75x25	0.01, 1.00	MP, GC	4	07
8	NVCaffe	GoogleNet	1	75x25	0.01, 1.00	MP, GC	4	08
9	NVCaffe	*	1	75x25	0,10	MP, GC	4	09
10	NVCaffe	*	1	25x75, 75x25	0,10	GC	4	10

Fonte: Produção própria em conjunto com o orientador.

Os gráficos foram dispostos em 10 figuras e, para compreendê-los, é necessário se atentar às seguintes observações:

1. Cada gráfico possui em seu eixo x os valores de imagem (100, 500, 1.000, 2.000) e em seu eixo y os valores de épocas de treinamento (100, 200 e 500);
2. A coloração ao se aproximar das cores quentes (vermelha, laranja e amarela), indica que se obtiveram melhores resultados, em comparação às cores frias (verde e azul) e neutra (preta);
3. São identificados da esquerda para direita, onde o primeiro da esquerda é o A, o primeiro da direita B, o segundo da esquerda é o C, e o último é o D;
4. Na faixa de cores à direita de cada, observa-se de forma crescente os resultados arredondados obtidos no TOP-1 *Accuracy*, de acordo com as configurações de cada treinamento.

FIGURA 15 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA ALEXNET



Fonte: Produção própria em conjunto com o orientador.

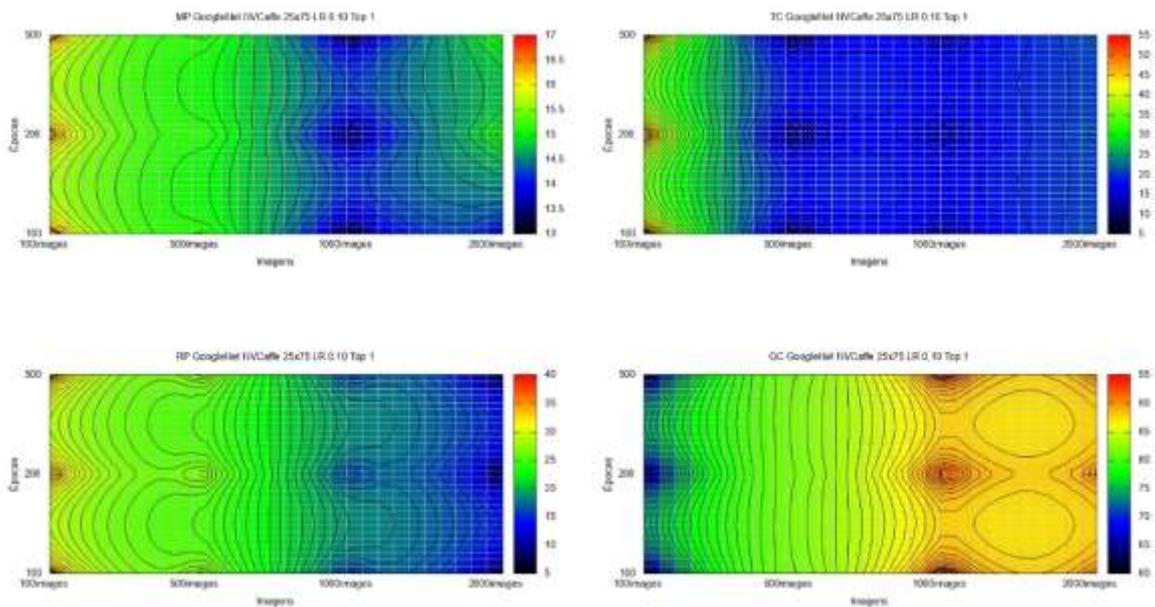
Na Figura 15, observa-se a influência do modelo para os seguintes parâmetros: rede (AlexNet), fração de treinamento (25%) e taxa de aprendizagem (0.10). Na FIG. 15-A, encontra-se o modelo *Moving Point* (MP), que avalia a distância euclidiana de um ponto até o centro. Para este modelo, observou-se que os melhores resultados foram obtidos com poucas imagens e épocas. No máximo local, notou-se que poucas imagens e mais épocas geram também bons resultados.

Na FIG. 15-B, encontra-se o modelo *Triple Cloud* (TC), que possui três conjuntos de pontos, diferente do GC que possui apenas um. Ambos calculam a taxa de cobertura de um retângulo. Por haver três conjuntos, há um componente de variação muito grande, por esse motivo ao utilizar muitas imagens, confunde-se o treinamento, prejudicando o resultado.

Na FIG. 15-C, encontra-se o modelo *Rotating Point* (RP) responsável pela medição do ângulo de arco da circunferência. Nessas configurações, realizar o treinamento com poucas imagens e mais épocas, se mostrou mais vantajoso, pois assim, o algoritmo reconhece melhor o padrão.

Na FIG.15-D, onde o modelo utilizado é *Gaussian Cloud* (GC), é notório que o número de imagens ao se aproximar de 1.000, acarreta em melhores resultados, se comparado com os valores distantes dessa faixa. Entretanto, as épocas se mostraram flutuantes. Logo, utilizar poucas épocas torna a acurácia mais eficiente.

FIGURA 16 - INFLUÊNCIA DO MODELO NOS PARÂMETROS GOOGLÉNET



Fonte: Produção própria em conjunto com o orientador.

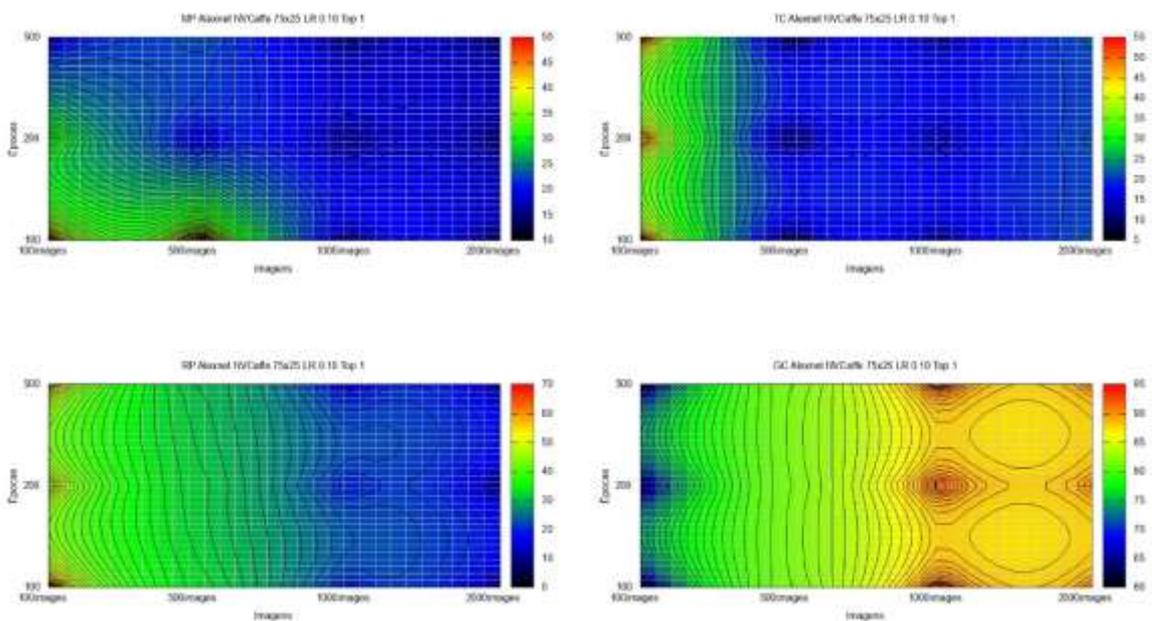
Na Figura 16, mantiveram-se as mesmas configurações citadas na Figura 15, alterando apenas a rede para GoogLeNet. Nota-se que o modelo MP representado na FIG. 16-A, e o modelo RP representado no gráfico na FIG. 16-C, mostraram-se eficazes com poucas imagens e épocas.

Na FIG. 16-B, o modelo TC indica que o uso de grandes quantidades de imagens, ocasiona em resultados ineficaz. O uso de poucas épocas, melhora sua eficácia. A FIG. 16-D, indica que para o modelo GC não importa a rede utilizada, os resultados serão similares.

Observando as Figuras 15 e 16, pode-se concluir que independente das redes AlexNet e GoogLeNet, ao utilizar o modelo MP, não é recomendado o uso de grandes quantidades de imagens e épocas. No RP os resultados para as redes foram semelhantes, sendo melhor utilizar poucas imagens e mais épocas. O TC se mostrou flutuante com bons resultados

quando se atribui poucas imagens e épocas, ou poucas imagens e mais épocas. O GC também mostrou essa mesma variação, porém com mais imagens. Observando a faixa de cores da Figura 15, é notório que em relação aos resultados, o modelo MP, não se mostrou bom com esses valores; os modelos RP e TC se mostraram similares; e o modelo GC é o que possui melhor acurácia. O mesmo ocorre com a Figura 16, apesar das semelhanças de comportamento, com essas configurações é aconselhável utilizar a rede AlexNet, ao invés da GoogLeNet, pois mostrou melhores resultados, exceto o modelo GC que não se alterou.

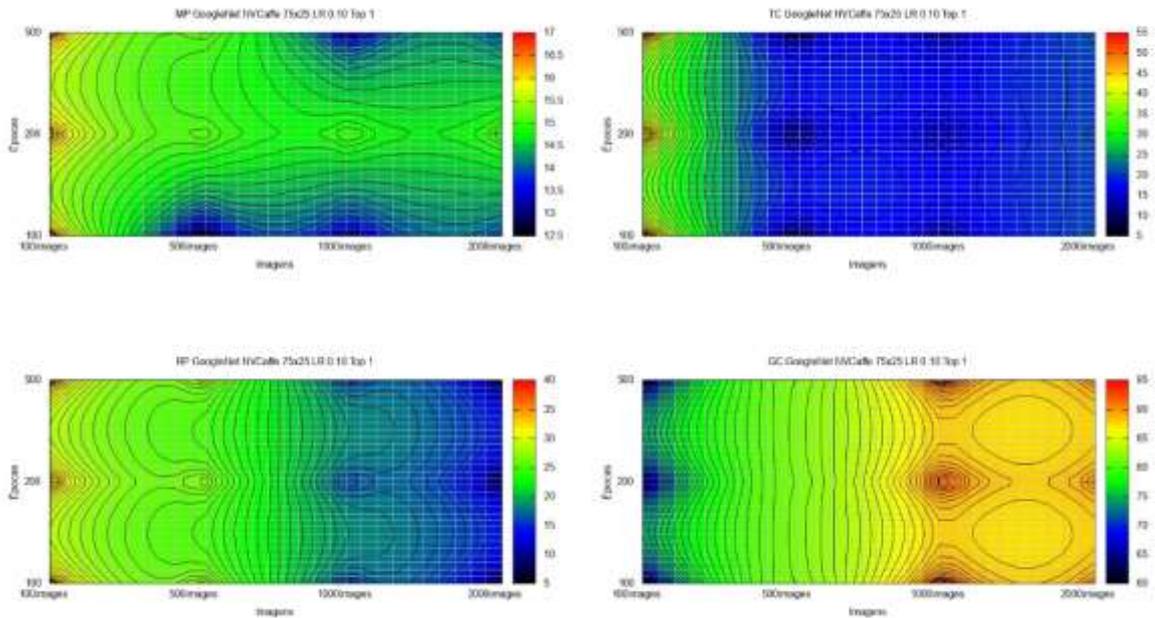
FIGURA 17 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA ALEXNET E FRAÇÃO 75%



Fonte: Produção própria em conjunto com o orientador.

Na Figura 17, foi fixada a fração de treinamento mais usual quando o DIGITS é utilizado (75%), mantendo a rede (AlexNet) e a taxa de aprendizagem (0.10). Para o modelo MP, observou-se que os melhores resultados foram obtidos com poucas imagens e épocas. Em relação aos outros três modelos, notou-se que houve variações semelhantes quando comparado com as Figuras 15 e 16, mantendo as mesmas observações.

FIGURA 18 - INFLUÊNCIA DO MODELO NOS PARÂMETROS PARA GOOGLNET E FRAÇÃO 75%

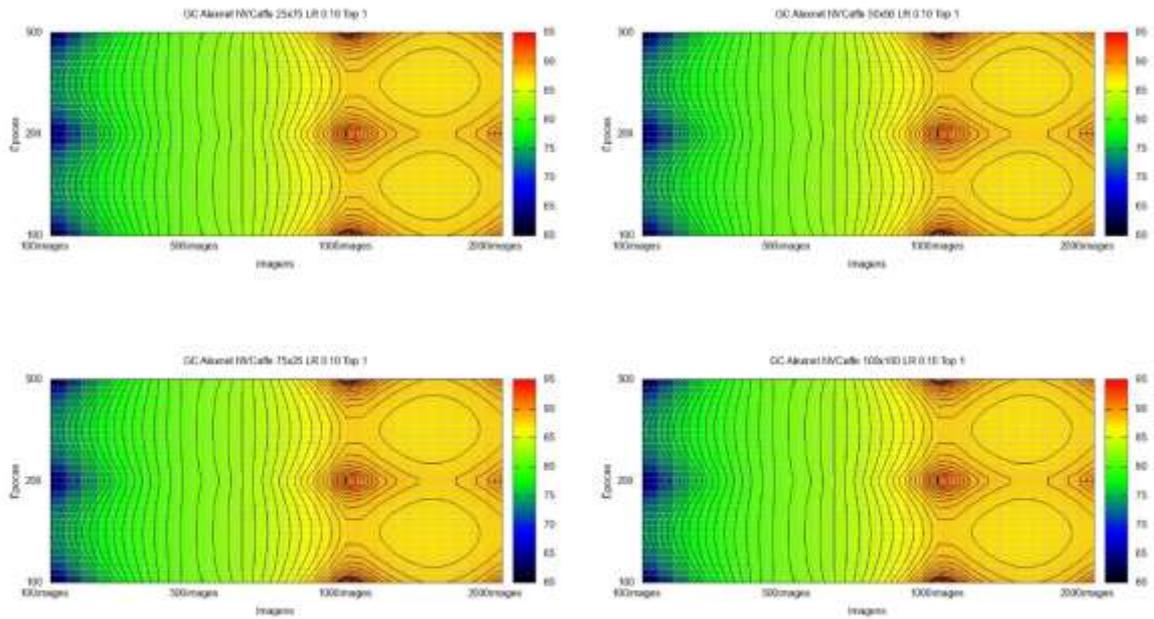


Fonte: Produção própria em conjunto com o orientador.

Assim como na Figura 17, os dados contidos na Figura 18 obtiveram as mesmas configurações mudando apenas a rede para GoogLeNet. Nessas condições, os gráficos indicam que quanto menor a quantidade de imagens, melhores são os resultados para o MP e RP. Observa-se que não houve alterações nos modelos TC e GC, se comparados ao conjunto de gráficos anterior.

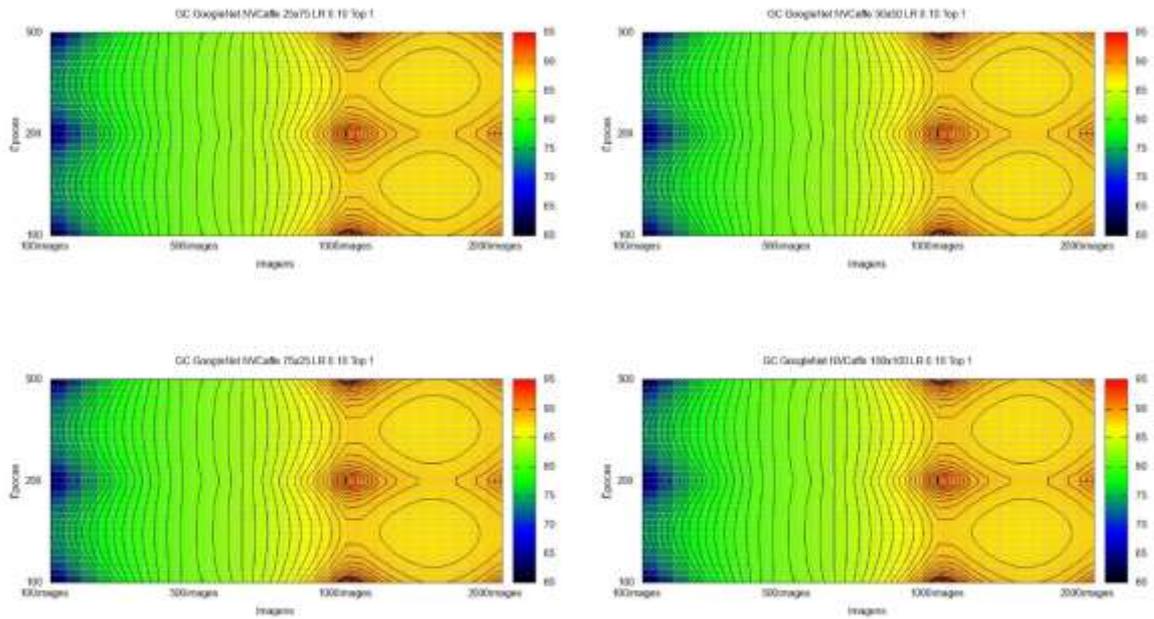
Observando as Figuras 17 e 18, pode-se concluir que o TOP-1 *Accuracy* dos modelos TC e GC não se alteraram com a mudança da rede. Os modelos MP e RP utilizando a rede GoogLeNet, mostraram uma grande variação nos resultados comparados com o modelo usado na Figura 17. O modelo MP com a rede AlexNet obteve um resultado 33 pontos percentuais maior se comparado com a rede GoogLeNet, o mesmo ocorreu com o modelo RP com um resultado 30 pontos percentuais maior.

FIGURA 19 - INFLUÊNCIA DA FRAÇÃO DE TREINAMENTO PARA A REDE ALEXNET



Fonte: Produção própria em conjunto com o orientador.

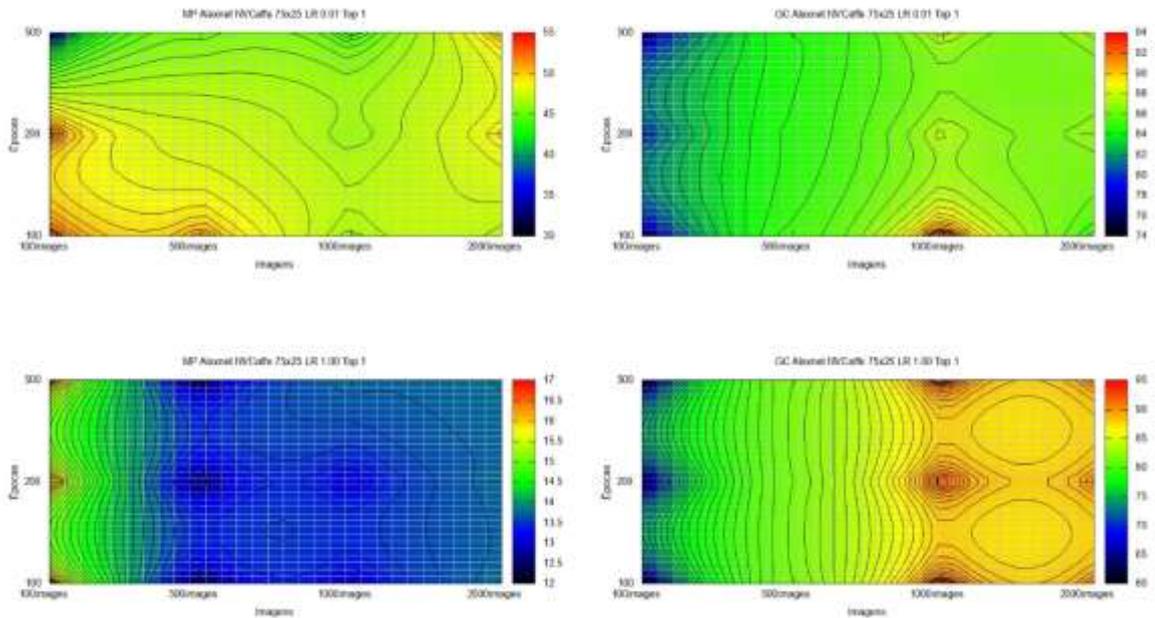
Na Figura 19, foi usada a rede AlexNet, a taxa de aprendizagem 0.10, o modelo GC e frações de validações variadas. Nota-se que treinamentos com poucas épocas e 1.000 imagens obtiveram bons resultados, independente da fração de treinamento. Portanto, utilizar 25% do conjunto de dados é suficiente para treinar a rede. Como observado, os resultados se mostraram flutuantes em relação às épocas para este modelo.

FIGURA 20 - INFLUÊNCIA DA FRAÇÃO DE TREINAMENTO PARA A REDE GOOGLNET

Fonte: Produção própria em conjunto com o orientador.

Na Figura 20, foi analisada a rede GoogLeNet com os mesmos parâmetros contidos na Figura 19 e, os resultados obtidos em ambas foram idênticos, reforçando que a quantidade de fração de treinamento não influencia no resultado. Portanto, utilizar a fração de treinamento baixa é suficiente para este modelo.

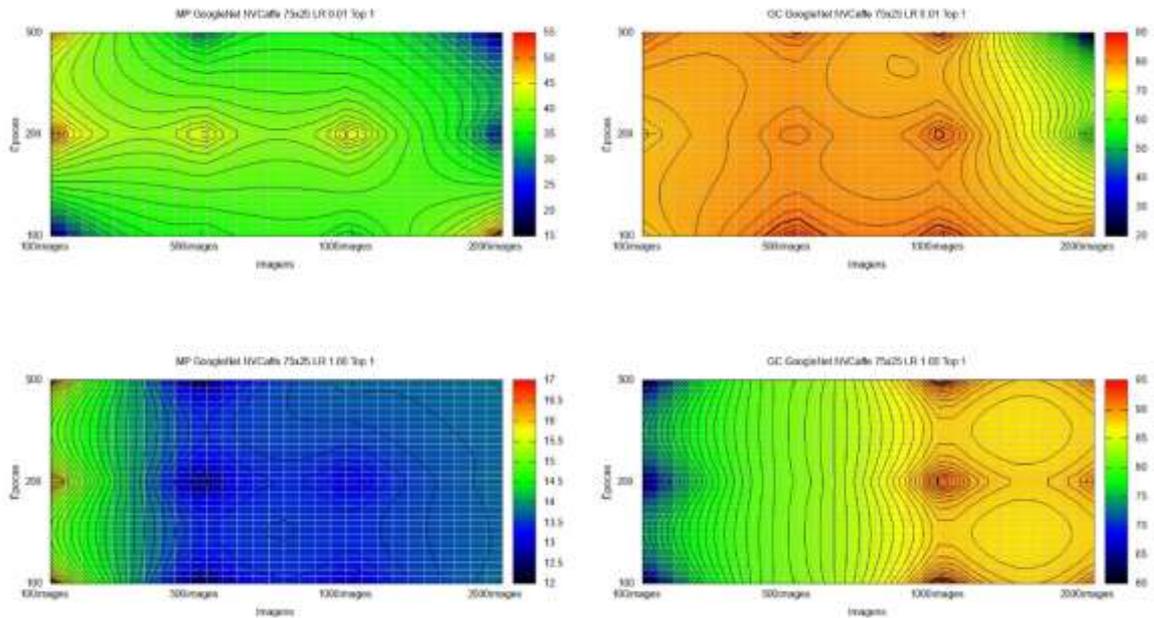
FIGURA 21 - ANÁLISE DA TAXA DE APRENDIZAGEM PARA A REDE ALEXNET



Fonte: Produção própria em conjunto com o orientador.

Na Figura 21, ocorreu a análise da taxa de aprendizagem (0.01 e 1.00) e dos modelos (MP e GC) em relação à fração de treinamento (75%) e rede (AlexNet). Nesta fração de treinamento, o MP apresentou uma dispersão elevada no gráfico, com a taxa de aprendizagem 0.01. Observa-se que, neste modelo houve melhor resultado com taxa mais baixa, poucas imagens e épocas. No GC, nota-se o contrário: quanto maior a taxa, melhor é o resultado do treinamento. Portanto, quando se trata da área de cobertura (GC), é preferível mais imagens, poucas épocas e taxa de aprendizagem mais elevada.

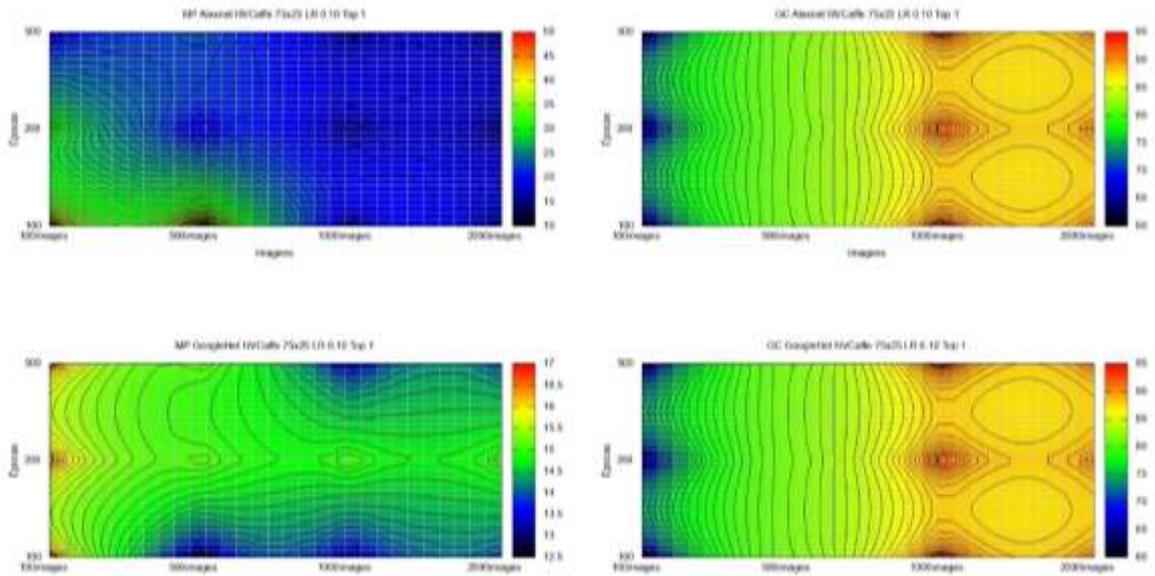
FIGURA 22 - ANÁLISE DA TAXA DE APRENDIZAGEM PARA A REDE GOOGLNET



Fonte: produção própria em conjunto com o orientador.

Na Figura 22, as configurações se mantiveram as mesmas da Figura 21, alterando apenas a rede para GoogLeNet. Nota-se que, em ambas as figuras apenas o modelo GC com taxa de aprendizagem 0.01 obteve alterações em seus resultados. Este modelo se mostrou eficaz com a taxa baixa se comparado aos outros, porém ainda é melhor se utilizado com uma taxa maior. O modelo MP para ambas a redes indica que melhores resultados aparecem com poucas imagens, épocas e taxa de aprendizagem.

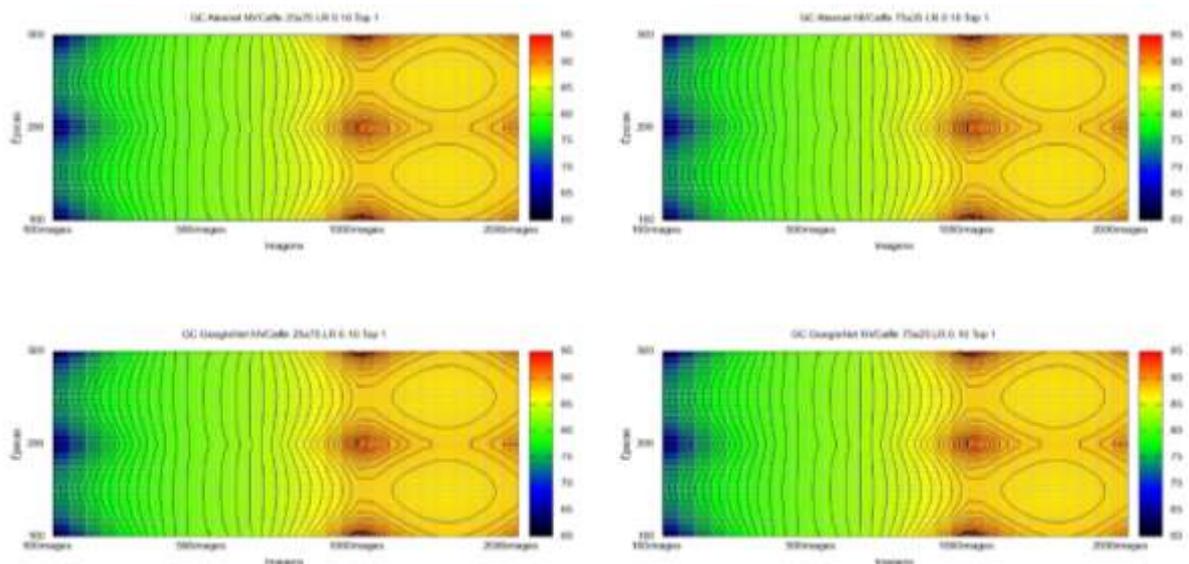
FIGURA 23 - ANÁLISE DAS REDES PARA OS MODELOS MP E GC



Fonte: Produção própria em conjunto com o orientador.

Na Figura 23, há variação na rede e nos modelos (MP e GC), fixando a fração de treinamento (75%) e taxa de aprendizagem (0.10). Nota-se que, no modelo MP, a rede AlexNet mostrou um resultado superior em relação a GoogLeNet. Enquanto a primeira atingiu uma acurácia de 55%, a segunda ficou abaixo de 17%. Esses fatores não influenciaram a medida de TOP1- *Acuraccy* no modelo GC.

FIGURA 24 - INFLUÊNCIA DAS REDES E FRAÇÕES DE VALIDAÇÕES PARA O MODELO GC



Fonte: Produção própria em conjunto com o orientador.

Na Figura 24, houve a comparação da fração de treinamento (25% e 75%) e tipo de rede (AlexNet e GoogLeNet) para determinar sua influência no modelo GC com a faixa de aprendizagem 0.10. Nessas configurações os TOP-1 *Acurracy* mantiveram o mesmo comportamento, tendo bons resultados a partir de 1.000 imagens para qualquer número de épocas, indicando que neste modelo, para essas condições a taxa de treinamento e a rede não influenciam no resultado. Portanto, o critério de escolha será tempo de processamento e disponibilidade dessas configurações no software.

TABELA 7 - RESUMO DOS RESULTADOS

	Melhores Resultados				
Modelo	Épocas	Imagens	Fração de treinamento	Taxa de Aprendizagem	Rede
MP	Poucas	Poucas	Superior a metade	Intermediária	AlexNet
RP	Poucas	Poucas	Superior a metade	Intermediária	AlexNet
TC	Poucas	Poucas	Superior a metade	Intermediária	AlexNet
GC	Poucas	Intermediária	Indiferente	Extremo superior	Indiferente

Fonte: autor.

Na Tabela 7, são resumidamente descritos os resultados apresentados neste trabalho. No capítulo a seguir, encontra-se a conclusão.

5. CONCLUSÃO

O presente trabalho teve por objetivo o estudo de alguns parâmetros do DIGITS através da análise de sensibilidade, com o intuito de redução de treinamento e melhoria da acurácia. Para este fim, optou-se pela escolha de três parâmetros (épocas, taxa de aprendizado e rede), além de quantidade de imagens, fração de treinamento e quatro modelos de geração de sequência de imagens.

Os resultados indicaram que em algumas situações o uso de muitas imagens pode confundir o algoritmo, acarretando em acurácias insatisfatórias. Notou-se também que o desempenho do DIGITS se mostrou não linear e dependente do problema. Para os modelos *Moving Point*, *Rotating Point* e *Triple Cloud*, a rede AlexNet foi considerada a melhor para o reconhecimento dos padrões, assim como a fração de treinamento 75%. O modelo *Gaussian Cloud* mostrou-se com comportamento oposto aos outros modelos, sendo indiferente a essas configurações. Foi observado que não é necessário utilizar valores extremos em seus parâmetros, poucas épocas, poucas imagens e taxa de aprendizagem intermediária se mostraram suficientes para os melhores resultados.

Essas conclusões permitem que haja redução no volume de dados, no tempo de processamento, e na sobrecarga de parâmetros, demonstrando que o estudo desenvolvido foi satisfatório ao atingir seu principal objetivo, proporcionando a otimização de futuros trabalhos com o DIGITS.

6. REFERÊNCIAS

ANGELIS, A. F.; ROCHA T. **Evaluating the Deep Learning accuracy in data extraction from synthetic image sequences**. Conference of Computational Interdisciplinary Science (CCIS 2019), Atlanta - USA, March 19 - 22, 2019.

CARVALHO, A.C.P.L. **Redes Neurais Artificiais**. ICMC, sem data. Disponível em < <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/#links> >. Acesso em 08/11/2019.

COPELAND, B.J. **Artificial Intelligence**. Encyclopaedia Britannica, 2019. Disponível em: < <https://www.britannica.com/technology/artificial-intelligence/Evolutio-narycomputing>>. Acesso em 14/08/2019.

ESCOLANO, F.; CAZORLA, M.A.; ALFONSO, M.I.; COLOMINA, O.; LOZANO, M.A. **Inteligencia artificial: modelos, técnicas y áreas de aplicación**. Madrid: Thomson, 2003. 370p.

FERNANDES, A.M.R. **Inteligência Artificial: noções gerais**. Florianópolis: VisualBooks, 2005. 160p.

GOLDSCHMIDT, R.R. **Uma Introdução à Inteligência Computacional: fundamentos, ferramentas e aplicações**. Rio de Janeiro: IST-Rio, 2010. 142p.

JANAKIRAM, M.S.V. **Nvidia DIGITS - An Easy Way to Get Started with Deep Learning**. The New Stack, 2018. Disponível em: < <https://thenewstack.io/nvidia-digits-an-easy-way-to-get-started-with-deep-learning/> >. Acesso em 25/09/2019.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. Nature, 2015. Disponível em: < <https://www.nature.com/articles/nature14539> >. Acesso em 28/08/2019.

MACHADO, V.P. **Inteligência Artificial**. Apostila do Curso de Computação da Universidade Estadual do Ceará, sem data.

MONARD, M.C.; BARANAUKAS, J.A. **Aplicações de Inteligência Artificial: Uma Visão Geral**. Instituto de Ciências Matemáticas e de Computação de São Carlos, São Carlos, 2000.

NVIDIA. **Deep Learning**. Disponível em < <https://developer.nvidia.com/deep-learning> >. Acesso em 28/08/2019.

NVIDIA. **NVIDIA Titan Xp**. Disponível em < <https://www.nvidia.com/pt-br/geforce/products/10series/titan-x-pascal/> >. Acesso em 12/11/2019.

NVIDIA. **Deep Learning DIGITS Documentation**. Disponível em < <https://docs.nvidia.com/deeplearning/digits/digits-user-guide/index.html> >. Acesso em 12/11/2019.

PARKER, M. **Digital Signal Processing 101: Everything You Need to Know to Get Started**. Estados Unidos: Elsevier, 2017.

REZENDE, S.O. **Sistemas Inteligente: Fundamentos e Aplicações**: 1.ed. São Paulo: Manole, 2005. 525p.

RUSSEL, S; NORVING, P. **Inteligência Artificial**: 3.ed. Rio de Janeiro: Elsevier, 2013. 988p.

SILVA, L.N.C. **Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais**. Tese (Mestrado em engenharia elétrica). Faculdade de Tecnologia da Universidade Estadual de Campinas. Campinas, 1998.

TURING, A. M. **Computing machinery and intelligence**. Mind, 59, 433-460,1950.

VARGAS, A. C. G.; CARVALHO, A.M.P; VASCONCELOS, C. N. **Um estudo sobre redes Neurais Convolucionais e sua aplicação em detecção de pedestres**. In: XXIX Conference on Graphics, Patterns and Images (SIBGRAPI). São José dos Campos, 2016.

VARELLA, M. Neurônio. Drauzio, sem data. Disponível em < <https://drauziovarella.uol.com.br/corpo-humano/neuronio/> >. Acesso em 20/10/2019.