



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

Kauan da Silva

Redes Neurais de Grafos para Análise de Partidas de Futebol

Limeira
2022

Kauan da Silva

Redes Neurais de Grafos para Análise de Partidas de Futebol

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

Orientador: Prof. Dr. Ulisses Martins Dias

Este trabalho corresponde à versão final da Monografia apresentada por Kauan da Silva e orientada pelo Prof. Dr. Ulisses Martins Dias.

Limeira
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Luiz Felipe Galeffi - CRB 8/10385

Si38r Silva, Kauan da, 2000-
Redes neurais de grafos para análise de partidas de futebol / Kauan da Silva. –
Limeira, SP : [s.n.], 2022.

Orientador: Ulisses Martins Dias.
Trabalho de Conclusão de Curso (graduação) – Universidade Estadual de
Campinas, Faculdade de Tecnologia.

1. Redes neurais (Computação). 2. Teoria dos grafos. 3. Futebol. I. Dias,
Ulisses Martins, 1983-. II. Universidade Estadual de Campinas. Faculdade de
Tecnologia. III. Título.

Informações adicionais, complementares

Palavras-chave em inglês:

Neural networks (Computer science)

Graph theory

Soccer

Titulação: Bacharel

Banca examinadora:

Ulisses Martins Dias [Orientador]

Plinio Roberto Souza Vilela

Luis Augusto Angelotti Meira

Data de entrega do trabalho definitivo: 19-07-2022

FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de dissertação para o Título de Bacharel em Sistemas de Informação na área de concentração , a que se submeteu o aluno Kauan da Silva, em 18 de junho de 2022 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

Prof. Dr. Ulisses Martins Dias
Presidente da Comissão Julgadora

Prof. Dr. Plínio Roberto Souza Vilela
FT/UNICAMP

Prof. Dr. Luis Augusto Angelotti Meira
FT/UNICAMP

Ata da defesa, assinada pelos membros da Comissão Examinadora, encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Graduação da Faculdade de Tecnologia.

Agradecimentos

Primeiramente, gostaria de agradecer a minha mãe: Tanea Maria Damaceno, que a todo momento me apoiou em todos os meus passos e me incentivou em todos os aspectos que precisei.

Ao meu irmão Jhonatan da Silva, que me apoiou e me tirava de frente do computador quando nada estava avançando durante o trabalho. E ao meu amigo Pedro Augusto de Jesus, que me ajudou lendo este trabalho e me dando consultoria de gramática.

Ao meu orientador Prof. Dr. Ulisses Dias, que me orientou, me ensinou sobre a área de pesquisa do projeto e por sua paciência em me ensinar a respeito da escrita de uma monografia.

Ao meu amigo que conheci durante esses dois semestres de TCC: Leandro Stival, que me ajudou muito durante a codificação da rede neural proposta neste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

GCN (do inglês, *Graph Convolutional Networks*) é um campo de pesquisa relativamente novo, pensando nisso, realizamos um estudo sobre as principais utilizações da GCN no estado da arte da literatura, abordando temas que obtiveram sucesso ao usar a representação em grafos em ambientes de aprendizado de máquina. Uma motivação deste estudo é o fato de essa linha de pesquisa ser promissora, dado que a estrutura de grafos tem sido tradicionalmente usada na computação para representar dados complexos no mundo real. Como estudo de caso, aplicamos GCN em uma situação prática, gerando um modelo para analisar partidas de futebol. Neste modelo, usamos métricas de grafos como: centralidade, entropia e vulnerabilidade da rede para treinar e validar nosso modelo, assim criando uma GCN capaz de prever se uma jogada irá chegar ao quarto final do campo tendo como fonte de informação seus segundos iniciais. Esse estudo possibilita validar algumas hipóteses de pesquisa relacionadas ao quão bem nosso modelo consegue prever o que irá acontecer em cada jogada.

Abstract

GCN (Graph Convolutional Networks) is a relatively new field of research, thinking in this, we carried out a study on the main uses of GCN in the state of the art of literature, approaching themes that were successful when using the representation in graphs in machine learning environments. One motivation of this study is the fact that this line of research is promising, given that the graph structure has traditionally been used in computing to represent complex data in the real world. As a case study, we apply GCN in a practical situation, generating a model to analyze soccer. In this model, we use graph metrics such as: centrality, entropy and network vulnerability to train and validate our model, thus creating a GCN capable of predicting whether a play will reach the final quarter of the field based on information your initial seconds. This study makes it possible to validate some hypotheses of research related to how well our model predicts what will happen in every move.

Lista de Figuras

2.1	Gráfico da ReLU.	14
2.2	Gráfico da LogSoftMax.	15
3.1	Representa o fluxo da metodologia proposta neste trabalho: (1) Separação dos IPB; (2) Divisão das partes de cada IPB; (3) Separação dos frames do FETW; (4) Criação do grafo para cada frame do FETW e população de cada vértice do grafo com suas respectivas métricas; (5) Treinamento e teste da GCN; (6) Verificação dos resultados das classificações.	19
3.2	Demonstração do funcionamento da seleção dos dados utilizados. Uma partida de futebol pode ser dividida em um conjunto de IPBs, cada IPB pode ser subdividido em Target, Lag e FETW. Os grafos são gerados para cada frame presente no FETW. Como temos 167 frames, teremos 167 grafos que comporão a entrada da GCN.	21
3.3	Representa o fluxo da GCN proposta neste trabalho, que inicia passando por 5 camadas convolucionais, todas compostas por função de ativação ReLu e um <i>dropout</i> de 50%, mas que diferem nos valores de entrada, saída e quantidade de tarefas criadas. Os valores de entrada e saída de cada camada são: camada 1 10:256, camada 2 256:512, camada 3 512:1024, camada 4 1024:512 e camada 5 512:256. Após passar pelas camadas convolucionais, se realiza um <i>pooling</i> para formatar a saída da camada 5 na entrada da primeira camada linear. Finalizado o <i>pooling</i> , começa a passagem pelas 3 camadas lineares que, semelhante às camadas convolucionais, tem como função de ativação Relu e aplica um <i>dropout</i> de 50%, e suas diferenças são nos valores de entrada, saída e quantidade de filtros de linearidade, os valores de entrada e saída de cada camada são: camada 1 512:256, camada 2 256:128, camada 3 128:2. Finalizando o fluxo da rede se aplica a função de LogSoftMax.	22

Sumário

1	Introdução	9
1.1	Contexto	9
1.2	Motivação	10
1.3	Objetivos	10
1.4	Organização do Texto	11
2	Levantamento bibliográfico	12
2.1	Conceitos Básicos	12
2.1.1	Definições para Criação da GCN	12
2.1.2	Definições para Criação dos Grafos	15
2.2	Revisão da Literatura	17
3	Desenvolvimento	19
3.1	Definição do Banco de Dados	20
3.2	Arquitetura da GCN	21
3.3	Discussão dos Resultados	22
4	Conclusões	24
	Referências bibliográficas	25

Capítulo 1

Introdução

1.1 Contexto

Neste trabalho, iremos abordar o tema GCN (do inglês, *Graph Convolutional Networks*), um campo de pesquisa relativamente novo que surgiu a partir da união de outros dois ramos de estudo, o estudo de Grafos e IA (Inteligência Artificial). Esta união surge a partir do desejo de retirar os dados de entrada do espaço euclidiano, tradicionalmente utilizado a partir da extração de *features* de imagens, áudios e textos para serem utilizados como entrada para o treinamento e validação de redes neurais, podendo ser tanto CNN (do inglês, *Convolutional Neural Networks*) ou as tradicionais ANN (do inglês, *Artificial Neural Networks*). Para realizar essa mudança, os pesquisadores começaram a procurar por uma estrutura de dados flexível o suficiente para suportar diversos tipos de análises (ZHANG et al., 2019).

Para atender a esses requisitos, foi escolhido o grafo como modelo de dado, modelo este composto de um conjunto não vazio de vértices e um conjunto de arestas. Por conta desta simplicidade e capacidade de representação, é suficiente para armazenar informações complexas e também representar situações do mundo real com seus vértices e relacionamentos.

A escolha de grafos como fonte de dados traz novos desafios, por exemplo, como os grafos possuem estruturas irregulares, torna-se mais difícil generalizar algumas funções matemáticas para grafos, como as operações da camada de Convolução e da camada de Agregação (do inglês, Pooling). Outro problema seria como criar modelos escaláveis, pois como estamos na era do *big-data*, existe uma grande variedade de dados que precisam ser processados e armazenados.

Os grafos possuem uma estrutura irregular, fazendo com que tenham tipos, propriedades e tarefas variadas, assim criando uma necessidade de uma metodologia diferente para tratar problemas específicos, como grafos que contém dados que precisam de conhecimento interdisciplinar para realizar sua análise.

1.2 Motivação

O trabalho foi incentivado pelo fato de estudar e aplicar uma rede neural, que utiliza de grafos como dado de entrada, por serem uma estrutura de dados simples e com capacidade de representar situações do mundo real.

Para exemplificar nossa motivação, elaboramos uma rede neural utilizando os conceitos de GCN e abordaremos um estudo de caso relacionado ao futebol, que tradicionalmente tem sido resolvido com CNN (STIVAL, 2022), para prever se ao final de um IPB (Intervalo de Posse de Bola), intervalo de tempo que decorre desde o momento em que um time de futebol rouba a bola do outro time até o momento em que ele perde a posse de bola, a bola chegou ou não no quarto final do campo.

Neste trabalho, vamos utilizar uma GCN para prever qual foi o desfecho do IPB, usando como métrica de validação a acurácia balanceada da GCN após o treinamento.

1.3 Objetivos

O objetivo deste trabalho é realizar um estudo de GCN, o que inclui revisar conceitos e definições, e aplicar em um estudo de caso. Neste estudo de caso, estamos interessados nos seguintes objetivos específicos.

- Uma situação do mundo real, ao ser representada em grafos, pode ser usada diretamente como entrada em uma rede?
- O desempenho da GCN faz com que ela seja uma opção em relação aos métodos tradicionais de aprendizado de máquina?

1.4 Organização do Texto

No Capítulo 2, serão apresentados e explicados brevemente os conceitos sobre os quais foi construído este trabalho, como: grafos, GCN, *pooling*, convolução, fase de combinação, intervalo de posse de bola e as métricas utilizadas para compor os vértices dos grafos. Também será realizada a revisão da literatura apresentando trabalhos que obtiveram sucesso aplicando a GCN.

No Capítulo 3, é relatado o desenvolvimento da GCN utilizada, mostrando como foram selecionados e tratados os dados, a separação dos dados em treino e teste, o funcionamento da GCN e, por fim, como foi realizada a captação dos resultados.

No Capítulo 4, será realizada a conclusão deste trabalho, apresentando os resultados obtidos a partir do treinamento e teste da GCN e mostrando qual foi o desfecho em relação aos objetivos deste trabalho, além de situar o leitor em relação aos próximos passos.

Capítulo 2

Levantamento bibliográfico

Neste capítulo, serão apresentados os conceitos e a revisão da literatura, com a seguinte divisão de subseções: a Seção 2.1 apresentará os conceitos de grafos e de GCN, logo após serão apresentadas as métricas utilizadas, com base no trabalho de Stival (2022) e para finalizar será apresentado as definições da GCN. A Seção 2.2 apresenta trabalhos que utilizaram GCN e obtiveram sucesso em sua pesquisa.

2.1 Conceitos Básicos

Grafo é uma estrutura de dados composta por um conjunto não vazio de vértices $V = \{v_1, v_2, \dots, v_i\}$ e um conjunto de arestas $A = \{a_1, a_2, \dots, a_3\}$, onde cada aresta representa a ligação entre um par de vértices, e pode possuir uma direção, especificando qual é o vértice de saída e o de chegada. Para este trabalho, uma direção não foi especificada pela nossa representação não exigir essa funcionalidade. (STIVAL, 2022).

2.1.1 Definições para Criação da GCN

Nesta subseção, serão apresentadas as seguintes definições utilizadas na criação da GCN:

Convolutional Neural Network: são redes neurais que realizam o aprendizado com algumas camadas de convolução, camadas estas responsáveis por extrair características dos dados fornecidos.

Graph Convolutional Networks: são uma evolução das CNNs, utilizando grafos como entrada para generalizar as convoluções. Como nas CNNs, as GCNs utilizam camadas de convolução para realizar o aprendizado, assim conseguindo propagar pesos entre as

camadas e diminuindo a dependência em relação às camadas totalmente conectadas (LIU; ZHOU, 2020).

Camadas de Convolução: são camadas que utilizam correlações cruzadas, operações matemática semelhantes às convoluções, que conectam uma função a outra e medem a integral de sua multiplicação (GÉRON, 2019). Estas camadas funcionam da seguinte maneira: cada camada cria filtros com pesos aleatórios, seguindo o procedimento de classificação descrito no trabalho de You, Ying e Leskovec (2020). Após a criação dos filtros, estes são aplicados na matriz de adjacências do grafo, que contém em suas linhas são representações dos jogadores de futebol e suas colunas são representações das métricas. Depois da aplicação de cada filtro, é gerada a saída da camada de forma normalizada. A quantidade de filtros utilizados é igual à quantidade de saídas de cada camada, por exemplo, se a primeira camada de convolução tem uma entrada de tamanho 10 e precisa ter uma saída de tamanho 256, então aquela camada da GCN cria a quantidade necessária de filtros para atender a essa necessidade.

Camadas Lineares: aplica transformação linear aos dados recebidos, dada pela Equação 2.1. Estas camadas foram utilizadas para que no final da execução da rede ela tenha como saída um vetor linear, visto que a GCN trabalhará com duas classes, portanto tendo que fazer escolhas binárias. Estas camadas funcionam da seguinte maneira: cada camada utiliza de pesos aleatórios, inicializados neste trabalho com o procedimento de inicialização Glorot, como descrito no trabalho de McCaffrey (2017).

$$Linear(x) = xA^T + b \quad (2.1)$$

Mapa de Características: é o resultado de uma camada de convolução ou uma camada linear, que servirá de entrada para a próxima camada.

Algoritmo de Backpropagation: este algoritmo é dividido em duas etapas: a *forward propagation* e a *backward propagation*. A *forward propagation* corresponde ao processo de passar os valores computados pelas camadas da rede, enquanto a *backward propagation* corresponde à tarefa de retropropagação até a camada de entrada do gradiente do erro, que atualizará os pesos (FREITAS, 2021).

Função de Dropout: esta função consiste em trocar aleatoriamente alguns valores dos tensores de entrada para zero, seguindo uma probabilidade definida pelo criador da rede, assim regularizando e prevenindo a co-adaptação de neurônios.

Funções de ativação: a propagação de informações do mapa de características na GCN depende de qual função de ativação foi utilizada nas camadas da rede, adicionando complexidade não-linear e ajudando a aumentar a representatividade da GCN. As funções de ativação utilizadas foram a ReLU e a LogSoftMax, dadas pelas equações 2.2 e 2.3. E os gráficos das funções estão representados nas figuras 2.1 e 2.2

A função ReLU foi escolhida, pois reduz o desaparecimento dos gradientes ao atribuir zero para valores negativos, assim conseguindo valores mais estáveis para os gradientes durante a fase de treinamento (GÉRON, 2019).

$$ReLU(x) = (x)^+ = \max(0, x) \quad (2.2)$$

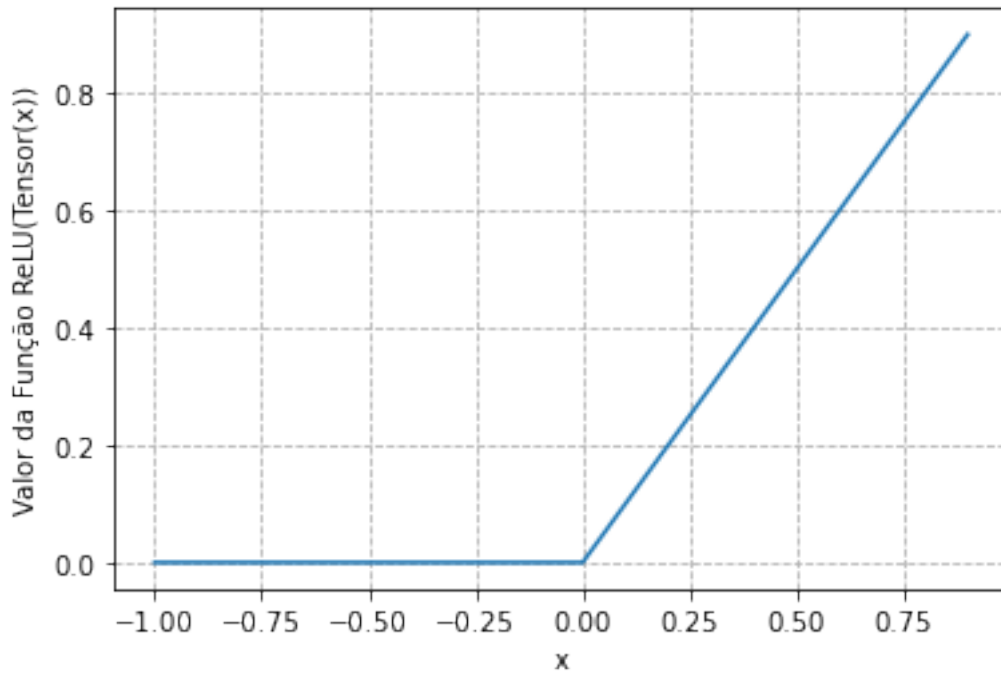


Figura 2.1: Gráfico da ReLU.

A função LogSoftMax foi utilizada para definir a probabilidade de cada classe em uma escala logarítmica negativa, pois como esta função tem seu resultado em valores logaritmos, consegue uma maior variedade de representação dos dados, melhor desempenho nos cálculos e maior otimização do valor do gradiente.

$$LogSoftMax(x_i) = \log\left(\frac{e^{(x_i)}}{\sum_j e^{(x_j)}}\right) \quad (2.3)$$

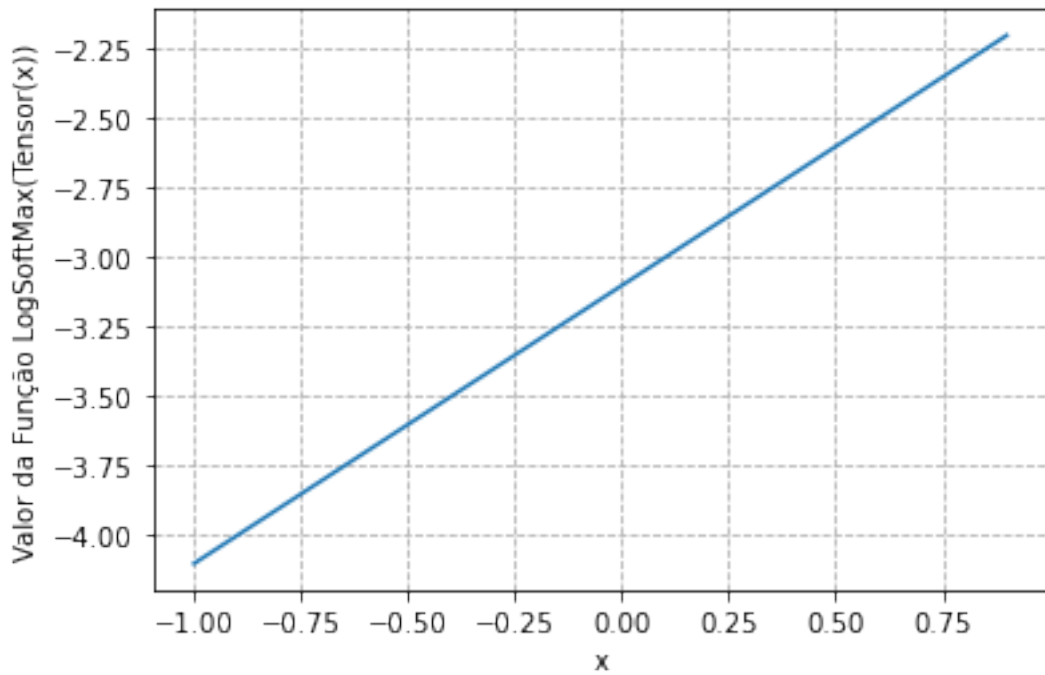


Figura 2.2: Gráfico da LogSoftMax.

Função de Perda: esta função indica quão satisfeito o algoritmo está com o resultado, comparando as pontuações da categoria certa com as demais categorias. A função de perda aplicada neste trabalho foi a entropia cruzada, dada pela equação 2.4. Esta função foi escolhida por permitir a atribuição de pesos (w) para as classes (C) de previsão, sendo útil para treinar um conjunto de dados desequilibrados. Na equação 2.4, os valores de y representam os resultados corretos rotulados e os valores de X representam as características utilizadas durante o treinamento.

$$l(x, y) = \sum_{n=1}^N -w_{y_n} * \log\left(\frac{e^{x_{n,y_n}}}{\sum_{c=1}^{|C|} e^{x_{n,c}}}\right) \quad (2.4)$$

Pooling: realiza a condensação das informações, com o objetivo de subamostrar os dados de entrada para reduzir a carga computacional, o uso de memória e o número de parâmetros que precisam ser treinados na rede (GÉRON, 2019).

2.1.2 Definições para Criação dos Grafos

Nesta subseção, serão apresentadas as definições utilizadas na criação dos grafos, baseadas no trabalho de Stival (2022).

No contexto de futebol, teremos as seguintes definições:

Intervalo de Posse de Bola (IPB): o IPB é o tempo em que a bola fica em posse de um time até: (i) ir para fora do campo, ou (ii) outro time retomar a posse, ou (iii) ser revertida em chute a gol (com ou sem sucesso).

Centralidade de Intermediação (CI): a CI indica a importância de um vértice no contexto do grafo, que para nosso trabalho será um jogador.

Excentricidade: a excentricidade é o valor da distância máxima entre dois vértices de um grafo. Neste trabalho, o valor da excentricidade determinará a facilidade da bola em chegar a um jogador, pois quanto menor for o valor da excentricidade, mais perto um jogador está do outro, considerando toda a rede.

Eficiência Global (EG): a EG indica como está o posicionamento dos vértices levando em conta a disponibilidade de um jogador para receber e realizar passes.

Eficiência Local (EL): assim como a EG, a EL também representa como estão posicionados os vértices, mas agora somente os vértices próximos serão impactados.

Vulnerabilidade: a Vulnerabilidade representa o impacto que a falta de um vértice causa na rede, interpretando para o nosso trabalho, ela será o impacto que a falta de um jogador faz para o desempenho da equipe, onde valores negativos irão indicar que a eficiência do time decai sem o jogador.

Coeficiente de Clusterização (CC): a CC mostra a importância de um vértice para gerar triangulações com seus vizinhos por todo o grafo. Em nosso trabalho, mostra a importância de um jogador para realizar a triangulação de passes entre o time, pois estas triangulações permitem que a equipe com posse de bola se movimente com mais velocidade.

Entropia: a Entropia representa o quanto um vértice consegue influenciar na situação do grafo, para nosso caso de uso, quanto maior for o valor desta métrica, mais oportunidades para um jogador (vértice) participar de uma jogada.

PageRank: o PageRank indica a popularidade de um vértice em participar da situação do grafo, essa métrica será utilizada para avaliar a popularidade de um jogador (vértice) em participar da construção de uma jogada, quanto maior for o valor desta métrica, mais participações um determinado jogador teve com outros jogadores também participativos.

No contexto de validação da informação, utilizaremos a métrica **BA (do inglês, *Balanced Accuracy*)**: a BA é uma maneira de analisar os resultados de classificadores booleanos, sendo aplicada quando a distribuição dos dados das classes não estão proporcionais, existindo mais amostras de uma classe do que da outra.

2.2 Revisão da Literatura

Utilizações que mais obtiveram sucesso com GCN ocorreram nos ramos de classificação de documentos (DEFFERRARD; BRESSON; VANDERGHEYNST, 2016), aprendizagem não supervisionada (ZHAO et al., 2019) e recomendação de sistemas (ZHANG et al., 2019).

O trabalho de Zhao et al. (2019) utiliza a GCN com RNN (Rede Neural Recorrente) para prever o tráfego urbano, criando assim uma variação da rede neural chamada de T-GCN (do inglês, *Temporal Graph Convolutional Network*).

A T-GCN utiliza a GCN para conseguir capturar os dados espaciais das vias urbanas, visto que esses dados oferecidos pelo tráfego urbano estão naturalmente em forma de grafo, então a utilização de grafos seria mais promissora em comparação à utilização de imagens, uma das possíveis entradas para uma CNN. A composição dos grafos utilizados no trabalho de Zhao et al. (2019) usa vértices para representar as estradas e arestas representam as relações de conexão entre estradas. As informações de tráfego das estradas são descritas como atributos dos vértices.

O resultado que Zhao et al. (2019) obteve utilizando a GCN superou as expectativas, atendendo a todas necessidades do projeto. Com essa ótima performance, a GCN obteve forte recomendação para trabalhos futuros no seguimento de pesquisa a respeito de previsão espaço-temporal.

O trabalho de Zhang et al. (2019) trata de uma outra variação da GCN, utilizando um codificador gráfico e um decodificador para tratar as estruturas gráficas semânticas em todos seus vértices, para assim utilizar estes dados como entrada da GCN. Essa nova versão foi chamada de STAR-GCN (do inglês, *Stacked and Reconstructed Graph Convolutional Networks*), criada para a recomendação de sistemas com base nas preferências dos usuários.

A GCN foi criada para substituir o modelo de recomendação de sistemas antigo baseado em FM (Fatoração de Matrizes). Essa mudança decorreu da premissa de que, para utilizar a FM, era necessário ter uma grade regular de dados que muitas vezes não correspondiam ao que eram coletados, então neste ponto a representação em grafos permite a utilização de grade irregular. Visto que GCN consegue suportar a utilização desse tipo de grade, pois sua estrutura consegue gerar representações de um vértice por meio do compartilhamento de parâmetros de seus vizinhos utilizando de um operador chamado agregador gráfico. E com a utilização da GCN é possível empilhar diversos agregadores e funções não lineares, possibilitando assim criar uma rede profunda, capaz de extrair características em alcances distantes de um grafo.

O resultado que [Zhang et al. \(2019\)](#) obteve utilizando a GCN atendeu as necessidades do trabalho, e assim a utilização de GCN recebeu boas motivações para trabalhos futuros na seguintes linhas de pesquisa: detecção de comportamento anormal, previsão espaço-temporal, previsão de popularidade do segmento, entre outros temas.

Seguindo o conceito de criar uma CNN para grafos, criando assim uma GCN, o trabalho do [Defferrard, Bresson e Vandergheynst \(2016\)](#) elabora uma GCN para realizar a classificação de documentos, enfrentando a dificuldade de definir filtros para grafos localizados que sejam eficientes para avaliar e aprender. No desenvolvimento da pesquisa, foi introduzido um modelo cuja complexidade computacional é linear com a dimensionalidade dos dados, confirmando que a qualidade do grafo de entrada é de suma importância, que por fim mostrou que os dados estatísticos de estacionariedade local e composicionalidade feitas pelo modelo são verificadas para documentos de texto desde que o grafo esteja bem construído.

O trabalho de [Defferrard, Bresson e Vandergheynst \(2016\)](#) concluiu um modelo que classifica documentos utilizando como estrutura de dados os grafos, por meio fundamentos matemáticos e computacionais, mostrando que o modelo conseguiu extrair feições locais e estacionárias através de camadas convolucionais de grafos, assim com este modelo novos trabalhos serão realizado em campos importantes de pesquisa onde os dados são normalmente representado em forma de grafos, podendo assim incorporar informações externas sobre a estrutura dos dados, em vez de grafos criados artificialmente, que podem sofrer variações em sua qualidade, como foi abordado durante a realização da pesquisa.

Capítulo 3

Desenvolvimento

Neste capítulo, será apresentado o desenvolvimento da GCN, desde a seleção dos dados até a apresentação dos resultados, como ilustrado na Figura 3.1, dividida em 6 etapas: (i) separação dos IPB's, (ii) composição de IPB, (iii) delimitação do FETW (*do inglês, Feature Extraction Time Window*), região utilizada para a criação dos grafos usados como entrada para a GCN, (iv) composição de um grafo e algumas das métricas de seus vértices, (v) fase de treino e teste da GCN e (vi) fase de verificação dos resultados obtidos pela GCN.

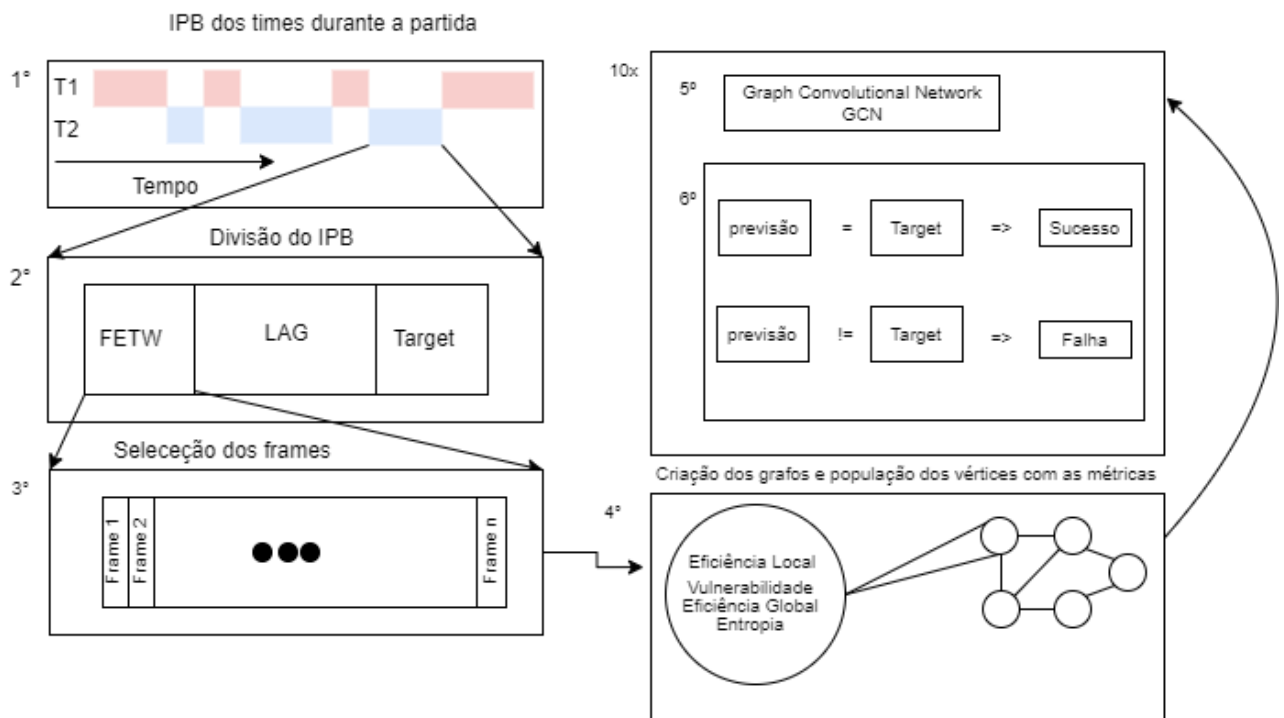


Figura 3.1: Representa o fluxo da metodologia proposta neste trabalho: (1) Separação dos IPB; (2) Divisão das partes de cada IPB; (3) Separação dos frames do FETW; (4) Criação do grafo para cada frame do FETW e população de cada vértice do grafo com suas respectivas métricas; (5) Treinamento e teste da GCN; (6) Verificação dos resultados das classificações.

3.1 Definição do Banco de Dados

Os dados foram selecionados seguindo a divisão das partidas de futebol em intervalos de posse de bola, como apresentado na Figura 3.2, que mostra em seu lado direito um fluxo de intervalos de posse de bola intercalado entre as equipes 1 e 2. Na parte central da figura, se escolhe um IPB e o amplia para mostrar um IPB subdividido em 3 regiões: FETW, *lag*, *target*, e por fim, no lado esquerdo da figura, é ampliada a região FETW, para mostrar que para cada frame desta região será criado um grafo correspondente. Esta divisão feita em cada IPB foi proposta por Stival (2022).

FETW: corresponde aproximadamente a 5 segundos iniciais de um IPB (167 frames). Deste intervalo de tempo, foram criados grafos para cada um dos frames e populados seus vértices com as métricas necessárias para o treinamento da GCN, que foram as seguintes: CI, Excentricidade, EG, EL, Vulnerabilidade, CC, Entropia e PageRank.

Lag (Atraso): intervalo de tempo variável entre o FETW e o *target* (apresentado em sequência). Correspondendo assim à parte central do IPB.

Target: corresponde aproximadamente a 5 segundos finais de um IPB. Nesta fase, é verificado se a equipe teve sucesso ou fracasso em chegar à zona de interesse, assim gerando o rótulo do IPB para compor o banco de dados, juntamente com as métricas extraídas no FETW.

Para este trabalho, foram utilizados os dados de 10 jogos (20 tempos) como amostra para treinar e validar nossa GCN. Destas partidas foram catalogados seus eventos, como: passes, chutes e faltas, além das posições dos jogadores em cada instante de tempo. Estes dados foram obtidos por especialistas a partir de um vídeo das partidas utilizando o *software* de extração semi-automático Dvideo (LEITE DE BARROS et al., 2006).

Após analisar os jogos, foram encontrados 4.448 IPBs, que variam em relação ao tempo que uma equipe passou com a bola. Assim, é necessário verificar se o tamanho de um determinado IPB seria suficiente para realizar uma análise. Foram selecionados os IPBs com uma duração maior ou igual a 15 segundos, pois IPBs inferiores a esse tamanho não apresentam uma movimentação grande o suficiente em campo, ou não possuem nenhum passe realizado. Levando em consideração os 15 segundos, foram selecionados 1.034 IPBs viáveis, abordagem semelhante ao trabalho de Stival (2022).

A partir dos IPBs selecionados foi gerado, para cada frame, um grafo $G = (V, A)$, onde o conjunto V são os jogadores, foram populados com as métricas necessárias, e o conjunto A são as possibilidades de passes entre os jogadores. Finalizado a geração dos grafos foi realizada a

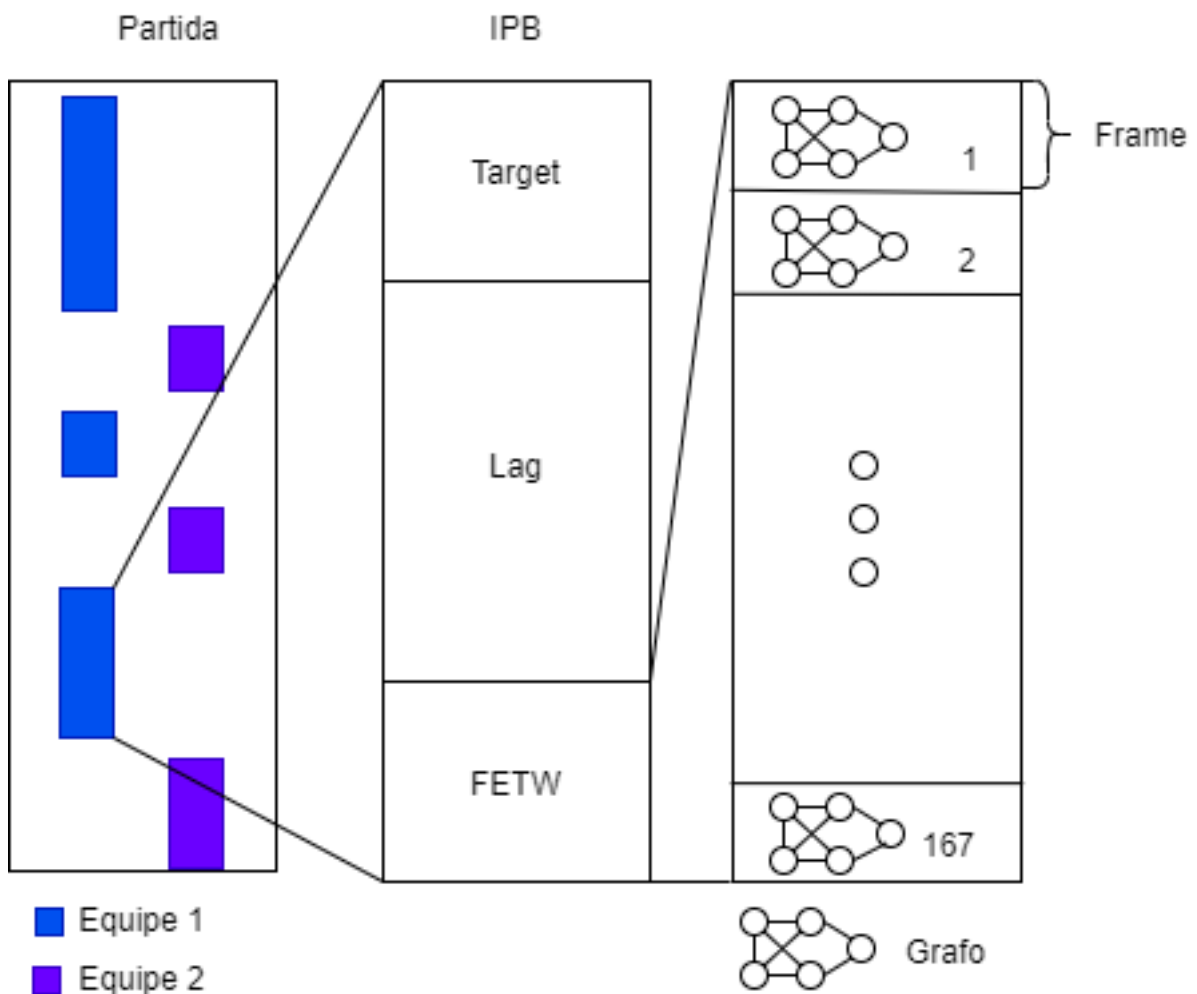


Figura 3.2: Demonstração do funcionamento da seleção dos dados utilizados. Uma partida de futebol pode ser dividida em um conjunto de IPBs, cada IPB pode ser subdividido em Target, Lag e FETW. Os grafos são gerados para cada frame presente no FETW. Como temos 167 frames, teremos 167 grafos que comporão a entrada da GCN.

divisão de quais tempos dos jogos seriam utilizados para treino e validação, entre os jogos que foram disponibilizados como caso de uso.

3.2 Arquitetura da GCN

Com os dados separados, começa a fase de treinamento da GCN. Como ilustrado na Figura 3.3, a GCN é composta por 5 camadas de convolução e 3 camadas lineares. Todas as 8 camadas possuem tamanhos diferentes para os canais de entrada e saída, os tamanhos utilizados para as entradas e saídas das camadas convolucionais (seguindo o formato entrada:saída) foram os seguintes: 10:256, 256:512, 512:1024, 1024:512, 512:256, e os tamanhos

utilizados para as entradas e saída das camadas lineares, seguindo o mesmo formato anterior, foram os seguintes: 512:256, 256:128, 128:2, que utilizam como função de ativação a ReLU e aplicam um *Dropout* de 50%. A rede conta também como uma camada de *pooling* para formatar a saída da última camada convolucional como entrada para a primeira camada linear na fase de *forward propagation*, que para finalizar aplica a função de LogSoftMax.

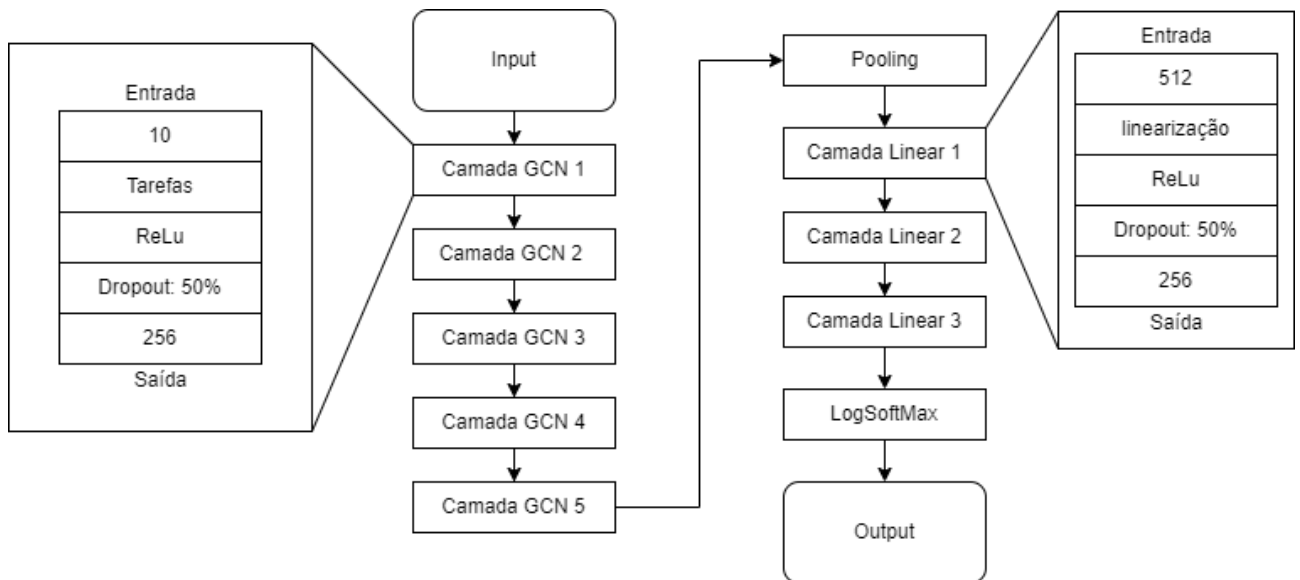


Figura 3.3: Representa o fluxo da GCN proposta neste trabalho, que inicia passando por 5 camadas convolucionais, todas compostas por função de ativação ReLu e um *dropout* de 50%, mas que diferem nos valores de entrada, saída e quantidade de tarefas criadas. Os valores de entrada e saída de cada camada são: camada 1 10:256, camada 2 256:512, camada 3 512:1024, camada 4 1024:512 e camada 5 512:256. Após passar pelas camadas convolucionais, se realiza um *pooling* para formatar a saída da camada 5 na entrada da primeira camada linear. Finalizado o *pooling*, começa a passagem pelas 3 camadas lineares que, semelhante às camadas convolucionais, tem como função de ativação Relu e aplica um *dropout* de 50%, e suas diferenças são nos valores de entrada, saída e quantidade de filtros de linearidade, os valores de entrada e saída de cada camada são: camada 1 512:256, camada 2 256:128, camada 3 128:2. Finalizando o fluxo da rede se aplica a função de LogSoftMax.

Após a fase treinamento, avançamos para a fase de teste para coletar a acurácia balanceada de cada modelo, para assim conseguirmos analisar o desempenho da GCN.

3.3 Discussão dos Resultados

O presente trabalho foi construído em torno da utilização de grafos como estrutura de dados. Uma vantagem que foi possível de analisar com o modelo proposto foi que avançamos em tornar a estrutura relacional dos grafos em um aliado na tarefa de predição, não sendo necessária a utilização de valores numéricos dentro de um vetor ou uma imagem para

representar as entidades do mundo real (jogadores e passes). Isso possibilita que os resultados possam ser reinterpretados utilizando as entradas do modelo, mais acessíveis aos especialistas da área de futebol.

Tornar os modelos baseados em grafos realmente competitivos em comparação aos modelos tradicionais é um desafio ainda maior, dado que os modelos tradicionais já passaram por vários estudos na literatura que fizeram com que se tornassem muito eficientes na resolução de problemas diversos. Esta eficiência pode ser observada analisando os dados de acurácia balanceada na Tabela 3.1, que foram coletados do trabalho de Stival (2022), que utilizou o método tradicional de rede neural da CNN.

Rodada	1	2	3	4	5	6	7	8	9	10
Acurácia	74%	79%	52%	75%	77%	82%	67%	75%	89%	74%

Tabela 3.1: Acurácia Balanceada da DNN em cada rodada de teste.

A GCN utilizada provou que precisa de um refinamento para conseguir ser uma opção em relação aos métodos tradicionais de aprendizado de máquina. Visto que a utilização de grafos está atualmente na fronteira do aprendizado de máquina, ainda levará algum tempo para que os benefícios de sua utilização resultem em modelos com desempenho mais eficiente, podendo ser percebido quando observamos as acurácias dos testes da GCN obtidos neste trabalho, mostrados na Tabela 3.2, que obtiveram resultados inferiores em relação aos métodos tradicionais.

Rodada	1	2	3	4	5	6	7	8	9	10
Acurácia	53,9%	47,9%	47,5%	47,4%	47,5%	44%	61,2%	26,7%	31,7%	54,3%

Tabela 3.2: Acurácia Balanceada da GCN em cada rodada de teste.

Realizando uma comparação dos resultados da GCN na Tabela 3.2 com os resultados do trabalho de Stival (2022) na Tabela 3.1, podemos perceber que os modelos baseados em grafos tem espaço para se tornarem competitivos em relação aos métodos tradicionais de rede neural, dado que um problema maior que foi identificado é que variam muito entre resultados bons (como a sétima execução) e resultados ruins (como a oitava execução).

Capítulo 4

Conclusões

A motivação deste trabalho consistiu em analisar se os algoritmos já desenvolvidos na área de GCN são capazes de competir contra metodologias tradicionais de modo a compreendermos as atuais vantagens e limitações do que vem sendo desenvolvido. Este estudo permite o desenvolvimento posterior de novas abordagens e metodologias para lidar com essas limitações.

Esta monografia analisou que a maior vantagem dos modelos de GCN é utilizar grafos como estrutura de dados para a entrada de uma rede neural, visto que o modelo proposto, conseguiu utilizar grafos para representar situações do mundo real (jogadores e passes), mesmo obtendo resultados inferiores ao modelo tradicional de predição.

Esta monografia também utilizou partidas de futebol para analisar se um modelo GCN baseado nos cinco primeiros segundos de um intervalo de posse bola consegue prever se, ao final da posse, a bola chegará no quarto final do campo. Os resultados obtidos não foram satisfatórios para responder positivamente a esse objetivo, pois em média o classificador apresentou uma acurácia balanceada de 46,2% para diferentes partidas.

Este trabalho deixa como ponto de estudo futuro o refinamento da GCN, para conseguir ser uma opção frente ao método tradicional de aprendizagem de máquina. O que torna este estudo promissor é que em algumas execuções da fase de testes a GCN obteve resultados interessantes.

Referências bibliográficas

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **Advances in neural information processing systems**, v. 29, 2016.

FREITAS, B. S. d. Otimização de Sistemas OAM de Feixes Laguerre-Gauss utilizando técnicas de Aprendizado de Máquina, 2021.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems**. [S.l.]: "O'Reilly Media, Inc.", 2019.

LEITE DE BARROS, R. M.; GUEDES RUSSOMANNO, T.; BRENZIKOFER, R.; JOVINO FIGUEROA, P. A method to synchronise video cameras using the audio band. **Journal of Biomechanics**, v. 39, n. 4, p. 776–780, 2006. ISSN 0021-9290. DOI: <https://doi.org/10.1016/j.jbiomech.2004.12.025>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0021929005000400>>.

LIU, Z.; ZHOU, J. Introduction to graph neural networks. **Synthesis Lectures on Artificial Intelligence and Machine Learning**, Morgan & Claypool Publishers, v. 14, n. 2, p. 1–127, 2020.

MCCAFFREY, J. D. Neural Network Glorot Initialization, 2017.

STIVAL, L. Análise preditiva de fatores que influenciam na entrada na zona de finalização em partidas de futebol, 2022.

YOU, J.; YING, Z.; LESKOVEC, J. Design space for graph neural networks. **Advances in Neural Information Processing Systems**, v. 33, p. 17009–17021, 2020.

ZHANG, J.; SHI, X.; ZHAO, S.; KING, I. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. **arXiv preprint arXiv:1905.13129**, 2019.

ZHAO, L. et al. T-gcn: A temporal graph convolutional network for traffic prediction. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 21, n. 9, p. 3848–3858, 2019.