

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

MARCUS VINICIUS CARVALHO MARTINS

**Um modelo para previsão de
resultados em partidas do
Campeonato Brasileiro de Futebol**

Prof. Filipe Braidão do Carmo, M.Sc.
Orientador

Rio de Janeiro, Junho de 2018

Um modelo para previsão de resultados em partidas do Campeonato Brasileiro de Futebol

Marcus Vinicius Carvalho Martins

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

Marcus Vinicius Carvalho Martins

Aprovado por:

Prof. Filipe Braidão do Carmo, M.Sc.

Prof. Leandro Guimarães Marques Alvim, D.Sc.

Prof. Bruno José Dembogurski, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

Junho de 2018

Agradecimentos

Primeiramente gostaria de agradecer aos meu pais (Fani e Eduardo), por tudo que me foi ensinado e todo apoio oferecido. Além disso, agradecer a todos da minha família, pelo companheirismo e solidariedade. Gostaria também de agradecer ao meu orientador, Filipe Braida, por toda paciência, confiança e apoio para a execução do trabalho. Por ultimo, porém não menos importante, agradecer a minha namorada Thaiany, principalmente, por todo o apoio que me deu nessa jornada.

RESUMO

Um modelo para previsão de resultados em partidas do Campeonato Brasileiro de Futebol

Marcus Vinicius Carvalho Martins

Junho/2018

Orientador: Filipe Braidão do Carmo, M.Sc.

Um desejo comum a muitos povos, a previsão se tornou possível a partir de estudos estatísticos e áxilos computacionais, porém muitas áreas tem peculiaridades que tornam mais complexas a previsão de resultados. O futebol vem passando por uma transformação tecnológica, são anos de evolução e armazenamento de dados que hoje podem contribuir para uma previsão do desempenho das equipes. Nesse trabalho apresentaremos um modelo para previsão de partidas de futebol, com experimentação no Campeonato Brasileiro de futebol. O modelo é composto por uma base de dados reduzidas e com dados de fácil acesso (considerados os dados simples sobre uma partida).

ABSTRACT

Um modelo para previsão de resultados em partidas do Campeonato Brasileiro de Futebol

Marcus Vinicius Carvalho Martins

Junho/2018

Advisor: Filipe Braidão do Carmo, M.Sc.

The prediction of sport events through machine learning algorithms, especially in football, is a subject which has been in the spotlight lately, mainly because of the increase in the amount of data available and the popularity of the game. This work presents a study focused on a model propose used to predict the result of football matches from the Campeonato Brasileiro de futebol, an Brazilian tournament. Using data from the seasons comprised in the period of 2003-2017.

Lista de Figuras

Figura 2.1: Simplificação de um neurônio.	6
Figura 2.2: Visualização do modelo MCP	7
Figura 2.3: Representação gráfica de diferentes funções de ativação. (a) função degrau; (b) função linear; (c) função logística; (d) função tangente	9
Figura 4.1: Distribuição de classes	29

Lista de Tabelas

Tabela 3.1: Exemplo de snpashot de uma partida	19
Tabela 3.2: Exemplo de subdivisão mandante e agregado	23
Tabela 3.3: Lista de parâmetros utilizados	24
Tabela 4.1: Exemplo de saída q fornecida pela rede	26
Tabela 4.2: Exemplo de saída p esperada	26
Tabela 4.3: Acurácia de treino e teste	30
Tabela 4.4: Matriz de confusão do experimento I	30
Tabela 4.5: Matriz de confusão do experimento II	30

Sumário

Agradecimentos	i
Resumo	ii
Abstract	iii
Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivo do trabalho	1
1.3 Organização do trabalho	2
2 Redes Neurais Artificiais	3
2.1 Introdução	3
2.1.1 Historia	3
2.1.2 Visão biológica	5
2.1.3 Modelo MCP	6

2.1.4	Funções de ativação	8
2.2	Aprendizado	8
2.2.1	Aprendizado Supervisionado	10
2.3	Perceptron	12
2.4	Multilayer Perceptron	13
2.4.1	Camadas intermediarias	13
3	Proposta	15
3.1	Motivação	15
3.2	Aplicações relacionadas	16
3.2.1	Trabalhos	17
3.3	Modelo de previsão	18
3.3.1	Atributos do modelo	21
4	Experimentos	25
4.1	Configuração do experimento	25
4.1.1	Dados	27
4.2	Resultados	30
5	Conclusões	32
5.1	Considerações	32
5.2	Limitações e trabalhos futuros	32
	Referências	34

Capítulo 1

Introdução

1.1 Contextualização

Com surgimento pouco explicado, porém historia incontestável o futebol segue sua crescente financeira. O esporte que já é praticado em toda parte do mundo com popularidade imensurável segue em transição. Atualmente a tecnologia inunda o esporte, seja na preparação dos jogadores, prevenção de lesões e até em análises pré e pós jogos. Entre essas aplicações da tecnologia no futebol existe um ramo que busca entender e prever o desempenho das equipes em um determinado contexto.

A previsão de resultados em partidas de futebol se popularizou juntamente com a evolução das técnicas de previsão. Hoje são usadas para diversos fins, desde apostas até para o impulsionar resultados de uma equipe em um campeonato.

Nesse contexto o estudo de técnicas para a previsão de resultados em partidas de futebol vem se popularizando e ganhando importância, esse trabalho tem como objetivo propor um modelo para esta previsão aplicado ao Campeonato Brasileiro de Futebol.

1.2 Objetivo do trabalho

Os objetivos desse trabalho são:

- Implementar um modelo estatístico/computacional para a previsão de resultados em partidas de futebol, do Campeonato Brasileiro de Futebol, utilizando redes neurais artificiais.
- Aplicação do modelo em ambiente real para mensuração de resultados.
- Compreender os resultados para extração dos principais fatores históricos, estatísticos, que influenciam em uma partida de futebol.
- Entender as limitações na previsão de partidas.

1.3 Organização do trabalho

- Capítulo 2: apresenta toda a base teórica para a discussão do tema, falando sobre redes neurais artificiais e o modelo de redes recorrentes.
- Capítulo 3: apresenta a proposta de modelo e implementação
- Capítulos 4 e 5: apresentam os experimentos, seus respectivos resultados e suas conclusões.

Capítulo 2

Redes Neurais Artificiais

2.1 Introdução

2.1.1 Historia

O cérebro humano sempre foi considerado uma máquina com grande capacidade de processamento, reconhecimento, associação e generalização de padrões. Por isso, com o desenvolvimento das ciências ao redor do mundo, surgiram diversos estudos que tentaram simular essas capacidades. Em sendo assim, torna-se interessante analisarmos tais estudos e observamos suas repercussões.

O primeiro modelo artificial de um neurônio biológico surgiu em um trabalho publicado em 1943, de Warren McCulloch, um psicólogo e neuro-fisiologista com uma carreira na tentativa de modelar eventos do sistema nervoso, e Walter Pitts, um matemático recém graduado da Universidade de Illinois (LANDAHL; MCCULLOCH; PITTS, 1943). O trabalho teve como grande objetivo definir e discutir um modelo de neurônio biológico artificial, que foi chamado de neurônio MCP (devido a **McCulloch** e **Pitts**).

As técnicas de aprendizado, que foram aplicadas sobre modelos de neurônios artificiais, começaram a ser impulsionadas após o surgimento de um trabalho publicado, em 1949, pelo psicólogo Donald Hebb (HEBB, 1949). Neste trabalho, Donald Hebb

percebeu e demonstrou como a plasticidade de redes neurais é obtida através da variação dos pesos de entrada dos neurônios. Em seu estudo, ele sugere uma teoria para explicar o aprendizado em neurônios biológicos que se baseia no reforço das ligações sinápticas entre neurônios excitados, por isso ele sugeriu a variação dos pesos de entrada dos neurônios, na tentativa de simular esse fenômeno biológico.

Em 1958, Frank Rosenblatt apresentou um novo modelo, o *perceptron* (ROSENBLATT, 1958). A partir disto, ele demonstrou que se as RNAs, redes neurais artificiais, com neurônios MCP fossem acrescidas de "sinapses ajustáveis", poderiam ser treinadas para identificar certos tipos de padrões. Além de apresentar o modelo, Rosenblatt propôs um algoritmo para treinar a rede e executar funções simples, por isso seu trabalho tornou-se tão popular.

Rosenblatt descreveu o perceptron com 3 camadas, sendo a primeira responsável por receber as entradas do exterior e tendo conexões fixas (retina). A segunda camada recebe impulsos da primeira, por meio de conexões, cuja eficiência de transmissão (peso) é ajustável e envia suas saídas para a terceira camada (resposta). Esse modelo elementar de RNA é capaz de trabalhar como um classificador de padrões, ainda que simples. Todavia, mesmo com toda a fundamentação teórica, aplicação na solução de problemas de classificação e popularidade, o modelo perceptron não foi tão bem recebido pela comunidade científica.

Em Minsky e Papert (1969), o modelo *perceptron simples*, como ficou conhecido o perceptron de Rosenblatt, não teria mais aplicação prática, levando em conta sua limitação de linearidade entre as classes, segundo eles essa limitação praticamente inutilizava o uso do modelo, vez que é muito difícil que problemas reais respeitem a limitação.

Devido, em sua maior parte, a grande repercussão do trabalho de Minsky e Papert, durante a década de 70, o assunto esfriou e pouco evoluiu, tendo em vista a dificuldade em lidar com "problemas difíceis de aprender" (*hard learning problems*) que são problemas não-linearmente separáveis, e o crescimento rápido da complexidade e do tempo para o treinamento da rede. Para muitos estudiosos, o assunto não teria mais futuro e poucos continuaram a pesquisar.

Em 1982, depois de mais de dez anos sem avanços significativos na área, um estudo de Hopfield colocou novamente assunto em alta. Nesse estudo foi apresentado o que ele nomeou de rede de Hopfield, chamando muita atenção para as propriedades associativas das RNAs Hopfield (1982). Além disso, a descrição de um novo modelo de treinamento, o *back-propagation*, ajudou a provar que a visão da alta escalabilidade da complexidade e tempo de treinamento, feita por Minsky e Papert, foi um tanto quanto pessimista, ao passo que o *back-propagation* se mostrou uma ótima solução para o treinamento de RNAs e, certamente, colaborou substancialmente para a evolução da tecnologia.

Após o ressurgimento do grande interesse pelo estudo de RNAs, em 1980 até os dias de hoje, a área passou por muitas variações. Após a popularização do algoritmo *back-propagation*, sugeriram diversas variações do mesmo que buscavam otimizar o processo de treinamento, e assim resultar em maior velocidade de execução. Paralelamente a isto, sugeriram diversas pesquisas de aceleração do treinamento com o uso de *hardware* especializado para RNAs. Assim, até então a área continua em mutação constante, evoluindo para novos modelos de redes, com mais camadas e complexidade, e simultaneamente, procurando formas de acelerar o treinamento, bem como torná-lo, assim, mais simples.

2.1.2 Visão biológica

O processamento de informação no cérebro efetua-se em cerca de 10^{11} unidades, denominados *neurônios*, que têm uma estrutura relativamente simples, porém são fundamentais para o funcionamento cerebral. Biologicamente os neurônios são compostos de três seções: *corpo celular*, *dendritos* e *axônios*, cada um com uma função própria, porém complementar. Essa estrutura pode ser vista na Figura 2.1. Toda a estrutura, topologia de suas conexões e o comportamento em rede dos neurônios são base para os estudos de RNAs.

Os dendritos tem como função básica receber impulsos nervosos advindos de outros neurônios, e com isso a informação contida no impulso é conduzida até o corpo celular. Tal corpo tem como função básica processar impulsos nervosos recebidos

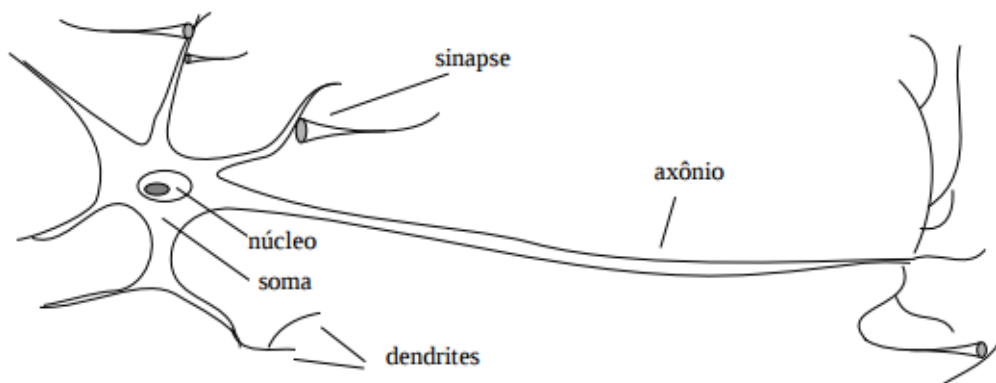


Figura 2.1: Simplificação de um neurônio.

e gerar novos impulsos que serão direcionados a outros neurônios, passando pelo axônio para chegar aos dendritos do neurônio de destino.

O ponto de contato entre a terminação de um axônio de um neurônio e o dendrito de outro é denominado de *sinapse*. Nesse sentido, destaca-se que sinapses são de suma importância no processo cerebral, na medida em que elas controlam o fluxo de toda informação e são partes fundamentais na formação da redes neurais biológicas.

Um dos principais pontos dos neurônios biológicos é a forma como se comunicam. E como já citado anteriormente, o processo todo ocorre devido às sinapses nervosas. Para que estas ocorram antes, os neurônios executam uma função, que é a combinação dos sinais recebido por neurônios anteriores. Dessa maneira, combinação resulta em um novo sinal. Toda essa forma de comunicação entre neurônios é parte fundamental no estudo de redes neurais artificiais.

2.1.3 Modelo MCP

O modelo de neurônio artificial proposto por McCulloch e Pitts (LANDAHL; MCCULLOCH; PITTS, 1943) (MCP) consiste em uma representação matemática do que se sabia sobre neurônios biológicos naquele momento. Em sua descrição, é exposto um modelo com n terminais de entrada (representando dendritos) que recebem os valores de $x_1, x_2, x_3, \dots, x_n$ (que representam as ativações dos neurônios

anteriores) e um terminal de saída y (representando o axônio). São utilizados também os pesos acoplados aos terminais de entrada $w_1, w_2, w_3, \dots, w_n$ para representar o comportamento das sinapses.

Cumprindo-se observar que o efeito de uma sinapse em particular i no neurônio pós-sináptico é dado por $x_i w_i$. Em uma descrição simples, os pesos determinam o "grau" em que o neurônio deve "considerar", ou dar importância, para sinais vindos daquela conexão. É possível ver uma representação desse modelo na Figura 2.2.

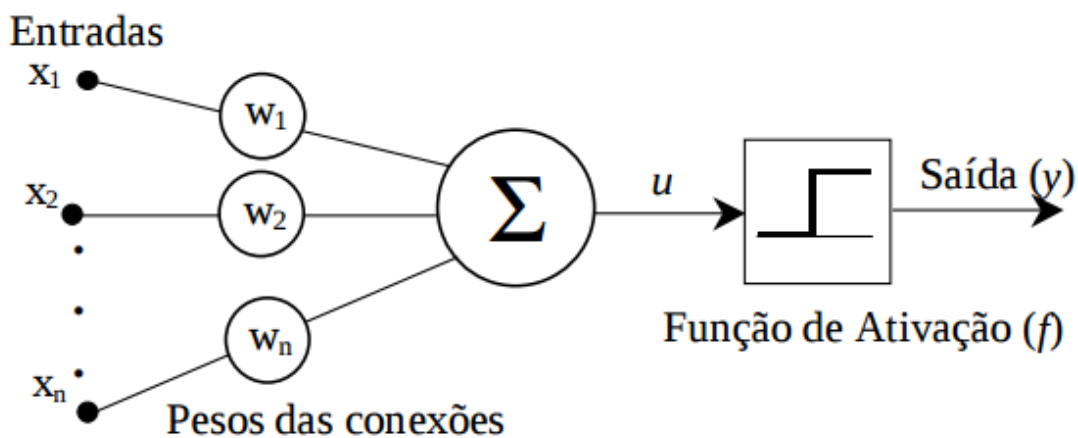


Figura 2.2: Visualização do modelo MCP

Um neurônio biológico, que foi grande referência para o trabalho de McCulloch e Pitts, dispara quando a soma dos impulsos exercidos sobre ele ultrapassa o seu limiar de excitação, *threshold*. Esse comportamento é recriado no neurônio MCP por um mecanismo simples que faz a soma dos valores $x_i w_i$ recebidos pelo neurônio (soma ponderada) e, comparando a soma obtida ao limiar (threshold) do neurônio, decide se o neurônio deve ou não disparar (saída 0 ou 1). Essa decisão feita pelo modelo é obtida a partir da aplicação de uma função de ativação, que efetivamente ativa ou não, a saída dos neurônios.

Sabendo que em neurônios biológicos não existe um mecanismo para a sincronização das ações, nem há restrição para que suas saídas sejam necessariamente ativadas em tempos discretos, resta claro que o modelo MCP é uma simplificação de um neurônio biológico dentro das limitações matemáticas. Independentemente das simplificações feitas por McCulloch e Pitts, o modelo foi um sucesso e teve êxito em, quando conectados em forma de rede, gerar alto poder computacional e solucionar problemas

muito complexos.

2.1.4 Funções de ativação

Como dito anteriormente, a função de ativação é o mecanismo utilizado para determinar a saída y de um determinado neurônio, considerando-se os valores dos vetores $x = (x_1, x_2, x_3, \dots, x_n)^T$ e $w = (w_1, w_2, w_3, \dots, w_n)^T$, sendo tais vetores as representações dos valores de entradas e valores de peso, respectivamente.

É importante entender que, intuitivamente, as funções de ativação introduzem um componente não linear nas redes neurais, que faz com que elas possam "aprender" mais do que relações lineares entre as variáveis e, assim sendo, sejam capazes de resolver problemas não-linearmente separáveis, que são os problemas difíceis de aprender.

A função de ativação de neurônios MCP é do tipo degrau deslocada do limiar de ativação θ em relação a origem, ou seja, sua saída será 1 para $\sum_{i=1}^n x_i w_i \geq \theta$ e será 0 para $\sum_{i=1}^n x_i w_i < \theta$, como pode ser visto na equação (talvez criar a equação).

Além da função degrau, existem muitas outras funções de ativação. Nesse âmbito, ressalta-se que essas funções têm um papel essencial na modelagem de redes neurais artificiais. Ao introduzir uma ativação não linear, o custo da rede neural aumenta e a otimização fica mais complicada. Na imagem 2.3, podemos ver as mais utilizadas.

2.2 Aprendizado

A capacidade de aprender por meio de exemplos é considerada uma das principais características das RNAs e um dos grandes motivos das RNAs serem alvo de tantos estudos, posto que essa capacidade pode ser utilizada para resolver problemas complexos e forma relativamente simples. Sendo assim, um neurônio é considerado ser um elemento adaptativo e seus pesos sinápticos são modificáveis dependendo do algoritmo de aprendizado.

Em RNAs, o "conhecimento" não é obtido através de regras explícitas, como em outras abordagens de inteligência artificial. Em redes neurais artificiais, o co-

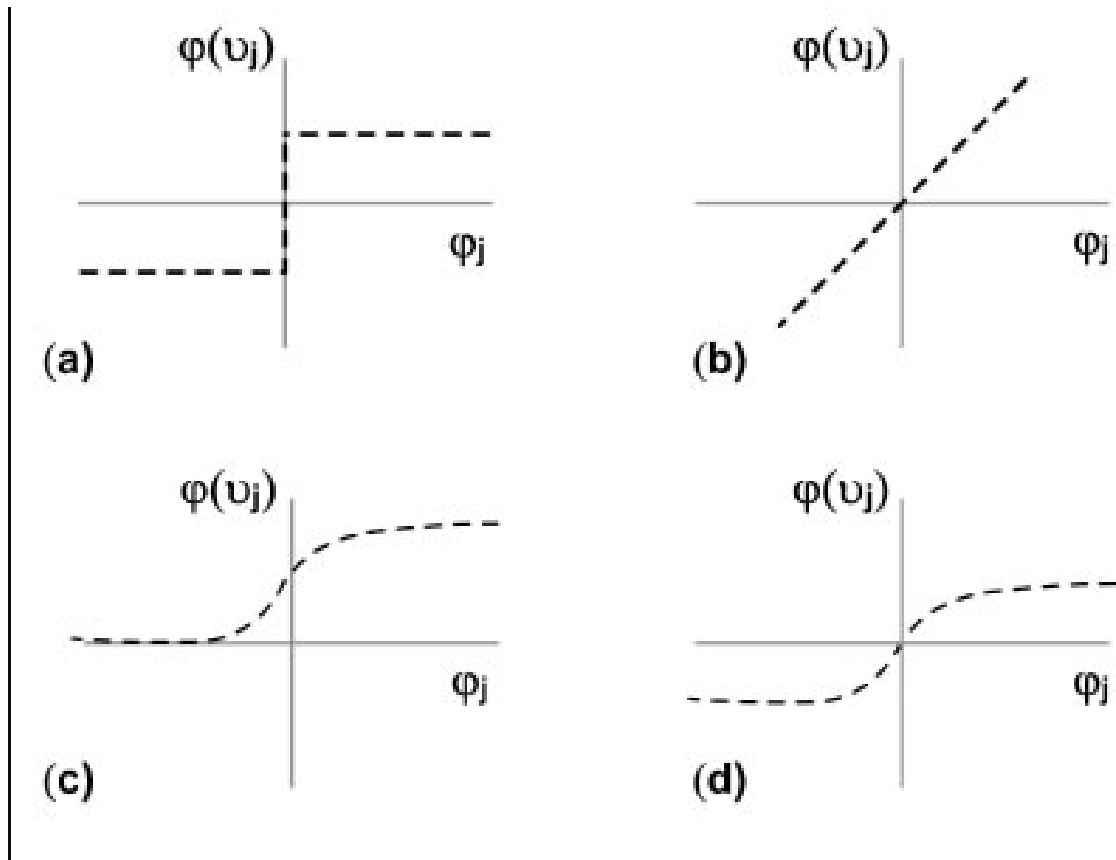


Figura 2.3: Representação gráfica de diferentes funções de ativação. (a) função degrau; (b) função linear; (c) função logística; (d) função tangente

nhecimento é absorvido através da variação das intensidades das conexões entre os neurônios. Sendo mais específico, o processo consiste no ajuste iterativo dos pesos da rede (que guardam ao final do processo o conhecimento que a rede adquiriu) baseado do limiar do erro da rede. Dessa forma, quanto mais o erro vai diminuindo, mais próxima da iteração final a rede se encontrará.

O aprendizado sempre esteve em pauta em estudos sobre RNAs. Nessa perspectiva, pode-se observar que, em 1949, o trabalho de Donald Hebb (HEBB, 1949) motivou um dos métodos pioneiros para aprendizado em RNAs, que ficou conhecido como *Aprendizado Hebbiano*. O aprendizado Hebb, como também é chamado, propõe que o peso de uma conexão deve ser ajustado caso haja sincronismo entre os "níveis de atividade" de saída e entrada. De forma simples, isso garante que, caso um primeiro neurônio tenha grande influência na ativação de um segundo neurônio, o peso da conexão entre eles seja reforçado.

A maneira como o ambiente influencia a rede durante o aprendizado é parâmetro para a definição de diferentes paradigmas de aprendizado, sendo eles:

- **aprendizado supervisionado**, que é baseado em um conjunto de exemplos constituídos de estímulos e respostas, é utilizado um agente externo que indica à rede a resposta desejada para determinada entrada.
- **aprendizado não-supervisionado**, que é baseado apenas nos estímulos recebidos pela própria rede, que deve aprender automaticamente a “categorizar” os estímulos. Nesse sentido, cumpre dizer que não existe um agente externo indicando a resposta desejada para os padrões de entrada.
- **aprendizado por reforço**, o comportamento da rede é avaliado apenas com base em alguma critério numérico, que é fornecido em instantes espaçados de tempo.

2.2.1 Aprendizado Supervisionado

Popularmente utilizado para classificação e reconhecimento de padrões, esse tipo de aprendizado implica na adição de um supervisor (ou treinador) ao treinamento. Esse elemento será responsável por estimular as entradas da rede (com uma massa de dados pré-definida), observar a saída fornecida pela rede e compará-la com a saída desejada. Neste processo, o observador obtém um δ do erro atual da rede e utiliza esse valor para ajustar os pesos de forma iterativa.

Esse tipo de aprendizado pode ser implementado de duas formas: *off-line* e *on-line*. Na rede on-line, que é a mais popular atualmente, a rede está em constante processo de aprendizado e mutação. Na rede off-line, os dados utilizados para o treinamento são fixos, desta forma, ao obter uma solução para a rede, esta permanece fixa. Para acrescentar dados é preciso adicionar os novos dados aos antigos e treinar a rede novamente.

Um ótimo exemplo para aprendizado supervisionado é o aprendizado por correção de erros. Neste modelo, as iterações visam minimizar o erro da saída y da rede.

Comumente, usa-se a expressão $e(t) = \gamma_d(t) - \gamma(t)$ para representar o erro no instante de tempo t , em que $\gamma_d(t)$ é a saída desejada neste mesmo instante de tempo e $\gamma(t)$ é a saída atual fornecida pela rede. Utilizando o valor do erro no instante t , o algoritmo corrige os pesos da rede e prepara a próxima iteração.

De forma genérica, a atualização dos pesos da rede, baseada no erro, é descrita na equação 2.1, considerando que $w_i(t)$ é o peso da entrada i no instante t , $x_i(t)$ é a entrada do neurônio i no instante t , η que é uma taxa de aprendizado e $e(t)$ é o erro no instante t .

$$w_i(t + 1) = w_i(t) + \eta e(t) x_i(t) \quad (2.1)$$

O entendimento da equação 2.1 é de que o ajuste dos pesos deve ser proporcional ao produto do erro pelo valor atual da entrada (que produziu o erro) naquele instante de tempo. Além disso, fica evidente a possibilidade de (utilizando as entradas e variando os pesos) determinar cada valor possível para o erro e dessa forma buscar o conjunto de entradas que fornece o menor erro possível (mínimo global).

Durante todo o processo de aprendizagem, a rede estará constantemente sendo exposta a estímulos oriundos das bases utilizadas para treino e para teste, porém não é incomum que, com a fixação da base de treino e de teste, a rede acabe em um processo de *overfitting* ou sobre-ajustamento. Isso acontece quando a rede está tão treinada para uma determinada base de dados que só é capaz de acertar exemplos dessa base, ou seja, a rede ajusta-se muito bem ao conjunto de dados anteriormente observado e mostra-se ineficaz para prever novos resultados.

Em 2014, um artigo ganhou popularidade por oferecer uma forma simples de solucionar o overfitting (SRIVASTAVA et al., 2014). O *droupout* é uma técnica utilizada durante o treinamento que funciona zerando aleatoriamente uma percentagem (definida pelo utilizador) de neurônios nas camadas da rede neural. Resumidamente, podemos dizer que o dropout é uma técnica de regularização que descorrelaciona a saída dos neurônios, e por isso gera mutação na rede.

2.3 Perceptron

O trabalho de Rosenblatt, em 1958, (ROSENBLATT, 1958) apresentou o que ficou conhecido como *perceptron simples*. A estrutura que é composta por uma rede de neurônios (MCP) e por uma regra de aprendizado foi a primeira a atribuir "conhecimento" à rede. Alguns anos depois, Rosenblatt demonstrou o teorema da convergência do perceptron, provando que um neurônio MCP treinado com o algoritmo de aprendizado perceptron sempre converge (caso o problema seja linearmente separável), ou seja, é sempre capaz de encontrar um conjunto de pesos que classifica corretamente os dados.

A descrição inicial de Rosenblatt era composta por uma unidade de entrada (retina), por um nível intermediário (associação) e um nível final de saída (resposta). Mesmo possuindo três camadas, este modelo ficou conhecido como perceptron de uma camada, basicamente pelo fato de que somente o nível de saída (resposta) possuía propriedades adaptativas. As unidades de entrada eram essencialmente sensoriais e o nível intermediário de associação possuíam pesos fixados antes do treinamento. Um dos principais motivos para somente uma camada ser adaptativas era a dificuldade atual para treinar mais de uma camada de uma rede simultaneamente, haja vista que até então não existia nenhum método que possibilitasse isso.

Mesmo com todos os avanços do perceptron, o mesmo foi duramente criticado por uma parte da comunidade estudiosa. Basicamente, o algoritmo era duramente contestado por ter sua capacidade computacional limitada aos problemas linearmente separáveis.

Minsky e Papert analisaram matematicamente o perceptron e demonstraram, em seu trabalho, que redes de uma camada não seriam capazes de solucionar problemas que não fossem linearmente separáveis, e como não acreditavam na possibilidade de se construir um método de treinamento para redes com mais de uma camada, concluíram que as redes neurais seriam sempre suscetíveis a essa limitação Minsky e Papert (1969).

2.4 Multilayer Perceptron

Durante o treinamento de redes com única camada, como *perceptron simples*, é utilizado um algoritmo de aprendizado supervisionado e correção de erros (como descrito anteriormente). Durante esse processo, os pesos são atualizados de acordo com o erro relativo a cada iteração da rede, para obter esse erro simplesmente é possível comparar o valor da saída fornecida pela rede com a saída desejada (a considerada correta). Todavia, ao adicionarmos mais camadas ao processo, teríamos camadas intermediárias, isto é, camadas cujo valor da saída desejada não é conhecido. Por isso, o grande problema para trabalharmos com redes múltiplas camadas é: Como calcular ou estimar o erro das camadas intermediárias?

A solução surgiu depois de quase 10 anos, sem muitas evoluções nos estudos de redes neurais artificiais, quando em um estudo apresentou a descrição do algoritmo de retro-propagação de erros, ou *back-propagation* (WILLIAMS; HINTON, 1986). O algoritmo funciona com o princípio de estimar o erro das camadas intermediárias, utilizando o gradiente descendente, baseado no efeito que causam na camada de saída. É importante frisar que, na camada de saída, é possível identificar o erro, pois existe um *valor esperado*.

A função das múltiplas camadas, em redes como a *multilayer perceptron*, é transformar o conjunto de dados no espaço da entrada de forma que possa ser tratável para a saída da rede. De forma resumida, um problema não linearmente separável será transformado em um problema linearmente separável pelas camadas intermediárias e, assim, poderá ser solucionado pela camada de saída da rede.

Para possibilitar o uso do gradiente descendente, é necessário utilizar funções de ativação contínuas e diferenciáveis, desta forma não é possível utilizar a função degrau (como no perceptron simples) e comumente serão utilizadas funções sigmoidais.

2.4.1 Camadas intermediárias

As camadas intermediárias de uma MLP (Multilayer Perceptron) são responsáveis por codificar internamente os padrões de entrada e, desse modo, essa codificação

é utilizada pela camada de saída. Com uma grande quantidade de unidades intermediárias divididas entre as camadas, é possível formar representações internas para qualquer conjunto de padrões de entrada. Ficou provado em um trabalho de Cybenko que uma camada intermediária é suficiente para aproximar qualquer função contínua e duas camadas são suficientes para aproximar qualquer função (RESEARCH; DEVELOPMENT; CYBENKO, 1988).

Em exemplos reais, usa-se, de modo geral, somente uma camada intermediária. Geralmente, utiliza-se duas camadas intermediárias quando a função a ser aproximada apresenta descontinuidade. Vale destacar que as camadas intermediárias adaptam-se por meio do algoritmo *back-propagation*, que aproxima o erro. E caso tenhamos muitas camadas intermediárias, isso resultará em muita aproximação e baixo desempenho.

Capítulo 3

Proposta

3.1 Motivação

Com o aumento da capacidade de processamento e armazenamento de dados, a tecnologia se aproximou do mundo dos esportes, ao passo que hoje já é fundamental para grande parte das modalidades, incluindo o futebol. É muito comum que os clubes analisem estáticas do próprio clube e dos adversários na preparação para uma partida, inclusive alguns clubes têm áreas de inteligência totalmente voltadas para essas análises. Tal fator passou a ser fundamental para o alto desempenho de uma equipe.

Uma das vertentes que vem se tornando comum dentro desse cenário é a utilização desses dados para a previsão de resultados em partidas de futebol. Essas previsões são majoritariamente utilizadas por casas de apostas para definir o valor inicial das apostas em um determinado jogo, porém são também utilizadas por sites esportivos, visando a informação e entretenimento do seu público. Todo esse cenário, sobretudo, voltado para apostas, gera uma grande importância nesse tipo de estudo, pois estão diretamente ligados com faturamento de grandes empresas.

Atualmente, houve um crescimento na área de estudo, o que levou a criação de conferências de estudos voltados para aplicações matemáticas nos esportes, como grande exemplo, a *MathSport International* que ocorre a cada dois anos. Contudo, a

maior parte dos estudos direcionados à área de predição esportiva ainda se concentra no setor privado. Em razão disto, os estudos dificilmente são divulgados.

Um grande desafio da predição de resultados, no mundo dos esportes, é definir um conjunto de atributos (*features*) que contextualizam o cenário desse esporte, de maneira que tais atributos estão diretamente ligados ao resultado da partida. Por exemplo, se estamos tentando prever resultados de partidas de futebol, pode ser interessante saber quantos gols cada um dos times faz por jogo. Esse parâmetro de média de gols por partida possivelmente é um atributo correlacionado com o resultado do jogo de futebol. Por isso, é interessante ter um nível bom de entendimento sobre o esporte que se está estudando.

Baseado neste cenário, este trabalho visa apresentar um modelo para previsão de resultados em partidas de futebol utilizando redes neurais artificiais com múltiplas camadas, tendo enfoque no campeonato brasileiro de futebol, utilizando dados considerados básicos, no sentido de serem de fácil acesso e sem subjetivismos.

3.2 Aplicações relacionadas

Os sites de aposta são uma grande referência de aplicações praticas que utilizam sistemas de previsão. Entre os mais populares estão: bet365¹ e bets99². Ambos utilizam das previsões para definir os valores que servem como base de cálculo para as recompensas dos apostadores.

Os sites de aposta têm grande comprometimento com as previsões feitas, pois estas tem influência no resultado financeiro dos mesmos. Esse é mais um dos motivos pelo qual não são divulgadas muitas informações quanto aos modelos utilizados e também não ficam claros os resultados obtidos por muitas empresas que pesquisam nessa área.

Alguns sites de apostas tem modalidades de aposta fora do comum, como por exemplo:

¹<https://www.bet365.com>

²<https://www.bets99.com>

- qual das equipes obterá o primeiro lateral
- qual das equipes terminará vencendo a primeira metade do jogo
- qual das equipes cometerá a primeira falta

Para este tipo de modalidade fica difícil acreditar que os sites possuam algoritmos para previsão.

Além dos sites de aposta, os sites de predição são outra forma de negócio, vez que eles têm seus próprios algoritmos para provisões de resultados e as vendem. Esses sites costumam ter assinaturas que dão direito a visualizar uma lista de partidas que foram previstas pelo site. Os usuários tendem a assinar serviços visando obterem um melhor desempenho nos sites de aposta. Além de informar as suas previsões, esses sites costumam ter dados históricos sobre o confronto.

Como exemplos desse tipo de site, pode-se citar: [soccerplatform](https://soccerplatform.me)³ e [statarea](http://www.statarea.com/)⁴. Ambos apresentam previsões gratuitas (para jogos considerados de menor importância), porém as principais previsões ficam limitadas aos assinantes dos sites.

3.2.1 Trabalhos

Devido a criação de conferências internacionais, tornou-se comum encontrar trabalhos que fazem predições nos esportes, de forma que em algumas modalidades existem trabalhos muito bem consolidados e com resultados expressivos. No caso do futebol, em específico, o estudo segue em evolução. Com a entrada de grandes instituições na "disputa", empresas como Microsoft e Google já têm seus próprios algoritmos.

Um exemplo de estudo relacionado tenta prever a classificação final de um campeonato simulando cada rodada do mesmo (MIN et al., 2008). Neste trabalho foi proposto um *framework* feito com a linguagem de programação *java* para previsão de esportes (principalmente futebol) utilizando inferência bayesiana. Vale dizer que esse *framework* foi testado na copa do mundo de 2002.

³<https://soccerplatform.me>

⁴<http://www.statarea.com/>

Entre os atributos de entrada para a predição deste trabalho estão, por exemplo, o índice para moral do time, índice de trabalho em equipe, dentre outros. Esses são exemplos de atributos inferidos humanamente (provavelmente por especialistas), sendo, portanto, subjetivos. Desse modo, utilizar atributos subjetivos implica em depender de pessoas que tenham vasto conhecimento no esporte, além de poder facilmente sofrer influência do viés.

O algoritmo foi bem sucedido ao acertar o campeão e vice-campeão da copa do mundo de 2002. Porém, essas foram as únicas posições em que o algoritmo foi preciso.

Outro estudo interessante que pode ser destacado foi publicado no ano de 2017, em uma conferência da MathSport International, onde foi apresentado um trabalho com modelos de predições, utilizando *machine learning* (GUHA et al., 2017). Este modelo foi utilizado para prever resultado de partidas que terminaram empatadas ao fim do primeiro tempo, desta forma o objetivo era acertar o resultado de uma partida a partir de informações obtidas no primeiro tempo de jogo.

3.3 Modelo de previsão

Visando prever o resultado de uma partida de futebol, modelamos o problema como um "retrato" das estatísticas das equipes que se enfrentarão. O objetivo é ser capaz de contextualizar uma partida somente com dados históricos e objetivos, o que torna possível que o modelo seja aplicado para qualquer outro campeonato sem grandes dificuldades, desde que estejam disponíveis informações históricas do mesmo.

Uma possível alternativa para esse modelo de retrato estatístico, seria a tentativa de modelar o problema como uma serie temporal. Dessa forma a ideia seria utilizar na entrada do modelo os atributos de partidas anteriores e com isso prever o resultado da partida atual. Entretanto, não é comum encontrar muita informação sobre as partidas e, para este tipo de modelagem, seriam importantes informações detalhadas, como por exemplo: posse de bola das equipes, finalizações corretas, finalizações erradas, numero de falta, entre outros.

Na prática não é fácil listar todas as informações que tem co-relação com o resultado de um jogo, muitas dessas informações são subjetivas. Isso significa que o nosso modelo é uma simplificação do problema real e que visa criar uma aproximação do contexto de uma partida de futebol.

Porém, na perspectiva de um retrato estatístico, considerando que cada partida prestes a acontecer possa ser representada por um conjunto de informações históricas sobre os clubes que se enfrentarão, algo como uma "fotografia estatística" que descreva o contexto da partida prestes a acontecer, um *snpashot*, um dos desafios do nosso modelo é encontrar as informações mais relevantes que possam ser co-relacionadas com o resultado final da partida. Um exemplo de um possível conjunto de informações para compor esse snapshot pode ser visto na tabela 3.3.

Lista de atributos da partida		
atributo	time A	time B
média de gols marcados	1.7	1.5
média de gols sofridos	1.4	0.6
média de vitórias	0.61	0.34
venceu o último jogo?	Sim	Não
time vai jogar completo?	Sim	Não

Tabela 3.1: Exemplo de snpashot de uma partida

Verifica-se na tabela 3.3 o exemplo de um possível *snapshot* para uma determinada partida de futebol, segundo a qual demonstra um conjunto de dados sobre os times que estão prestes a se enfrentar e que (no nosso modelo) definem a partida como um todo. Desta forma, baseado nessas informações e com algum conhecimento sobre futebol, uma pessoa estaria apta a avaliar qual time possui mais chance de vencer.

Os atributos do exemplo são considerados impactantes para o resultado de uma partida. Como dito anteriormente, alguém que tenha conhecimento sobre futebol estaria apto para avaliar qual time tem mais chance de vencer somente utilizando esses parâmetros, ao passo que quantos mais parâmetros (que tenham co-relação com o resultado) tivermos, ficaria mais fácil analisar o possível resultado.

Obviamente, em nosso modelo final, utilizaremos mais atributos do que os descrito acima para descrever uma partida, porém todos os atributos que utilizaremos são

oriundos de análise em jogos que antecedem essa partida que está prestes a acontecer. E cumpre destacar que nenhum de nossos dados foi inferido manualmente.

Existem diversos dados que seriam muito impactantes em partidas de futebol, porém não são de fácil acesso, ou necessitam de inferência humana como:

- **status físico dos jogadores dos times;** sabemos que futebol é um esporte, em que o contato é essencial e, sendo assim, a parte física das equipes impacta diretamente no resultado da partida, porém é muito difícil obter essa informação para todas as partidas do campeonato.
- **status psicológico dos jogadores dos times;** de extrema importância para execução de qualquer atividade o status psicológico, contudo, atualmente, é praticamente impossível ter relatórios psicológicos acerca de todos os jogadores para utilizar essa informação como um dos atributos do modelo.

Além dos aspectos interessantes citados acima, muitos outros são importantes para uma verdadeira descrição de uma partida, bem como a qualidade técnica do time, confiança da torcida, momento interno do clube, qualidade técnica dos jogadores do banco de reservas, dentre muitos outros. Todavia, não existem formas claras de definir atributos como esses, e modelos que fazem uso desse tipo de atributo geralmente usam especialistas de futebol para modelar os dados, o que pode impulsionar os resultados de uma predição.

Mesmo com a capacidade de tornar a predição mais eficaz, utilizar esse tipo de dados (com inferência humana) é problemático. Supondo que estejamos tentando prever jogos do campeonato indiano, utilizando somente dados históricos, tem-se que seria possível prever com o modelo aqui descrito, porém para termos atributos como qualidade técnica dos times e status físico dos jogadores dos times precisaríamos de alguém que regularmente acompanhe os jogos campeonato indiano e possa definir esses valores (com o risco de viés).

3.3.1 Atributos do modelo

Os atributos do nosso modelo podem ser divididos em duas grandes classes, sendo essas:

- **históricos globais**; os dados que compõem o que definimos aqui como "históricos globais" são os dados dos clubes em toda a história do campeonato. O que significa que são médias que levam em conta todos os jogos, que encontram-se disponíveis na base de dados do clube no campeonato, e que antecedem a partida a ser prevista.
- **históricos locais**; os dados que compõem o que definimos como históricos locais são referentes a um clube em uma determinada edição do campeonato. Dessa maneira, se estamos prevendo um jogo da décima rodada, pode-se concluir que os dados que compõem o histórico local serão compostos com todos os jogos do clube (da primeira até a nona rodada) no campeonato atual.

Essa grande divisão nos atributos ajuda a criar um efeito de desempenho histórico dos clubes e do momento atual dos mesmos. Cumprindo-se ressaltar, ainda, que algumas variáveis do nosso modelo só existem no contexto do histórico local e outras somente no contexto histórico global, porém é relevante dizer que maior parte dos nossos atributos coexiste em ambos os grupos.

As variáveis do histórico local funcionam muito bem para representar a situação atual de um clube em uma competição, tendo em vista que possuir status positivos nesse grupo significa que, atualmente, o clube está em bom momento. Já as variáveis do histórico global funcionam bem para o entendimento do contexto do clube perante a história de um determinado campeonato, posto que ter um bom status global significa que o clube costuma ter um resultado satisfativo na competição.

atributo	time A	time B
media de gols por jogo local	1.89	0.81
media de gols por jogo global	0.68	0.92

O exemplo descrito na tabela 3.3.1 mostra dois atributos. Analisando esses atributos, podemos perceber que o time A, historicamente, não faz muitos gols por

partida (0.68 gols por partida considerando todas as partidas disputadas pelo time A na história do campeonato), por outro lado, na edição corrente do campeonato, está com uma média de 1.89 gols por jogo.

Ademais, é de suma importância ressaltar que quando estamos falando do histórico global, em teoria, estamos nos referindo à história de um clube em um determinado campeonato. Entretanto, muitas vezes os dados disponíveis não contêm toda a história do campeonato e, por isso, deve-se considerar como toda a história disponível na base de dados. Além disso, quanto mais dados disponíveis, melhor para a predição.

Além dessa grande divisão entre histórico local e histórico global, temos também uma subdivisão interna. Tal subdivisão dos dados é resultado da fase de ida e volta, ou seja, dos times que jogam em casa e fora de casa. Dessa forma, criam-se três subdivisões, sendo elas:

Vale ressaltar que alguns campeonatos trabalham com campo neutro, o que significa que nenhuma dos times é mandante da partida. Isso inviabiliza essa subdivisão para estes tipos de campeonatos.

- **mandante**; nesse primeiro contexto, levando em conta somente jogos como mandante.
- **visitante**; nesse segundo contexto, levando em conta somente jogos como visitante.
- **agregado**; nesse terceiro contexto, levando em conta todas as partidas (como mandante e visitante).

Sendo assim, alguns atributos ganham variações nas subdivisões descritas acima. Nesse ponto, podemos ver um exemplo de variações na tabela 3.3.1, onde é possível observar que o atributo total de gols marcados **local** como **mandante** do time A refere-se à quantidade de gols feita no campeonato corrente (atual) somente em jogos como mandante.

Os atributos que variam entre as subclasses são basicamente:

Atributo	time A
total de gols marcados local como mandante	12
total de gols marcados local como visitante	4
total de gols marcados local agregado	16

Tabela 3.2: Exemplo de subdivisão mandante e agregado

- **média de vitórias**, a soma de da quantidade de partidas vencidas divida pela quantidade de partidas disputadas.
- **média de derrotas**, a soma da quantidade de partidas perdidas divida pela quantidade de partidas disputadas.
- **média de empates**, a soma da quantidade de partidas que terminaram empatadas divida pela quantidade de partidas disputadas.
- **média de gols marcados**, a soma da quantidade de gols marcados divida pela quantidade de partidas disputadas.
- **média de gols sofridos**, a soma da quantidade de gols sofridos divida pela quantidade de partidas disputadas.
- **sequência de vitórias**, a soma da quantidade de vitórias em sequencias (o número é zerado em caso de derrota ou empate).

Levando em conta tudo que foi descrito o nosso modelo apresenta o seguinte conjunto de atributos para a contextualização de uma partida. Tendo em vista todas as subdivisões das variáveis o nosso conjunto de atributos do modelo ficou da seguinte forma descrita na tabela 3.3.1.

Lista de atributos
média vitórias global do mandante agregado
média derrotas global do mandante agregado
média empates global do mandante agregado
média gols marcados global do mandante agregado
média gols sofridos global do mandante agregado
média vitórias local do mandante agregado
média derrotas local do mandante agregado
média empates local do mandante agregado
média gols marcados local do mandante agregado
média gols sofridos local do mandante agregado
sequencia de vitórias local do mandante agregado
média vitórias local do mandante como mandante
média derrotas local do mandante como mandante
média empates local do mandante como mandante
média gols marcados local do mandante como mandante
média gols sofridos local do mandante como mandante
sequencia de vitórias local do mandante como mandante
média vitórias global do visitante agregado
média derrotas global do visitante agregado
média empates global do visitante agregado
média gols marcados global do visitante agregado
média gols sofridos global do visitante agregado
média vitórias local do visitante agregado
média derrotas local do visitante agregado
média empates local do visitante agregado
média gols marcados local do visitante agregado
média gols sofridos local do visitante agregado
sequencia de vitórias local do visitante como visitante
média vitórias local do visitante como visitante
média derrotas local do visitante como visitante
média empates local do visitante como visitante
média gols marcados local do visitante como visitante
média gols sofridos local do visitante como visitante
sequencia de vitórias local do visitante como visitante

Tabela 3.3: Lista de parâmetros utilizados

Capítulo 4

Experimentos

Neste capítulo apresentaremos uma avaliação experimental do modelo proposto neste trabalho. Organizaremos o capítulo da seguinte forma: descrição dos experimentos realizados, metodologia utilizada, configuração do experimento, resultados objetivos e discussão pertinente quanto aos resultados.

O experimento realizado foi a produção de uma rede neural com múltiplas camadas treinada para a previsão de resultados de partidas do campeonato brasileiro de futebol. Essa rede deve ser capaz de recebendo informações sobre os clubes de uma determinada partida prever o resultado dessa partida entre: vitória do mandante, vitória do visitante ou empate.

4.1 Configuração do experimento

Como descrito no capítulo anterior nosso modelo trabalha com 36 atributos, logo, esse será o número de neurônios da nossa rede na camada inicial. Na segunda camada, a camada oculta, teremos o número da primeira camada ao quadrado, ou seja, 1296 neurônios na segunda camada. Na saída da segunda camada utilizaremos a técnica do *scr* para evitar *overfitting* dos dados. Por fim, a camada de saída da rede conta com 3 neurônios, um para cada possível resultado (vitória do time mandante, vitória do time visitante e empate).

Vale ressaltar, que mais especificamente na camada de saída, utilizamos a função de ativação *softmax*, uma alternativa diferenciável da função máximo que dessa forma garante que as saídas forneçam uma distribuição de probabilidade válida (onde a soma das saídas é 1) (MIKOLOV et al., 2011).

Outro ponto de suma importância é a utilização da entropia cruzada (*cross-entropy*) como função de erro (ou custo), levando em conta que temos uma classificação com três classes para uma partida (vitória mandante, empate, vitória visitante). Essa função de custo mede a diferença entre distribuições de probabilidade e pode ser calculada discretamente pela função descrita em 4.1.

$$H(p, q) = - \sum_x p(x) \log q(x) \quad (4.1)$$

Levando em conta que que p representa a probabilidade correta (0 ou 1) e q a probabilidade informada pela rede. Dessa forma, para a previsão de exemplo de q fornecida pela tabela 4.1 e o resultado p esperado da tabela 4.2 teríamos o seguinte erro descrito em 4.2.

vitória mandante	empate	vitória visitante
0.31	0.68	0.01

Tabela 4.1: Exemplo de saída q fornecida pela rede

vitória mandante	empate	vitória visitante
0.0	1.0	0.0

Tabela 4.2: Exemplo de saída p esperada

$$H = -(0.0 \log(0.31) + 1.0 \log(0.68) + 0.0 \log(0.01)) = 0.385 \quad (4.2)$$

No exemplo de resultado descrito na tabela 4.1 vemos que seguindo a rede a partida teria maior chance de resultar em empate, além disso vemos na tabela 4.2 que o resultado da partida foi empate. Mesmo o resultado da rede estando correto ainda existe um erro que foi calculado acima.

Além disso, visando evitar experimentos demasiadamente longos (em redes que

não apresentam mais progresso), utilizamos a técnica de *early stopping* (parada prematura). Esta técnica é baseada na definição de um limite máximo de épocas sem evolução no aprendizado da rede (WAGNER et al., 2001). Este limite é chamado de paciência (*patience*). Para avaliar se houve ou não melhora, compara-se o erro obtido no conjunto de treino na época atual com o obtido na época anterior. Em nossos treinamentos utilizamos uma paciência de 10 épocas sem melhora.

Tendo em vista toda a configuração da rede, vale dizer que os experimentos foram realizado de três formas, sendo elas:

- **Experimento I**, a base de dados é separada em treino e teste (aleatoriamente) e medimos a acurácia do modelo.
- **Experimento II**, a base de teste é composta por jogos de um único ano e o teste é feito por rodada. Dessa forma tentaremos avaliar a influência do histórico local (dados sobre a edição corrente de um campeonato) no algoritmo.

O **experimento I** será útil para o entendimento do modelo. Selecionando os elementos do teste de forma aleatória será possível entender a acurácia do algoritmo em uma situação regular.

Considerando esse cenário, o **experimento II** terá como proposito entender a evolução da acuarria do algoritmo com relação a rodada corrente. Sabendo que quanto maior é a rodada, mais informação acumulada temos disponível, será possível entender de que forma isso afeta o modelo. Considerando que esse experimento comparará rodadas, a partir da sexta, de uma única edição a base de teste terá 330 partidas.

4.1.1 Dados

Os dados que utilizamos são provenientes da primeira divisão do Campeonato Brasileiro de futebol (série A) e foram concedidos por uma instituição que armazena as informações de todos os jogos do campeonato. Cumprindo-se ressaltar que esses dados não são públicos.

A Liga brasileira de futebol profissional, como também é chamado o Campeonato Brasileiro, é a principal competição futebolística do país. Esta possui quatro divisões, quais sejam: Série A (primeira divisão), Série B (segunda divisão), Série C (terceira divisão) e Série D (quarta divisão). Neste ponto, tem-se que, para todas as divisões, os quatro últimos colocados perdem o direito de disputar a divisão atual no ano subsequente e darão suas vagas para os quatro primeiros da divisão inferior. Tal fator garante uma rotação dos clubes entre as divisões.

Em uma edição do Campeonato Brasileiro de futebol, cada clube joga duas vezes contra os demais, sendo uma vez em seu estádio (como mandante) e outra na estádio do seu adversário (como visitante), totalizando 38 jogos. Os clubes recebem três pontos por vitória e um por empate, o que configura o sistema de pontos corridos (adotado em 2003).

No presente trabalho, foram utilizados dados compreendidos no intervalo de 2003 até 2017 do Campeonato Brasileiro de futebol (série A). Ressalta-se que, para cada ano de campeonato, temos 380 jogos, logo, para nosso trabalho utilizaremos informações de 5320 partidas.

Para cada partida das 5320, tivemos acesso as seguintes informações:

- **rodada**, identificador da rodada entre 1 e 38.
- **equipe mandante id**, identificador único da equipe mandante (que joga na própria casa)
- **equipe visitante id**, identificador único da equipe mandante (que joga na casa do adversário).
- **placar oficial visitante**, quantidade de gols da equipe visitante.
- **placar oficial mandante**, quantidade de gols da equipe mandante.

Utilizando os dados citados acima para cada uma das partidas foi possível obter todos os atributos citados na tabela 3.3.1 com um pré processamento feito com a

linguagem de programação *python*. Durante o pré processamento foram gerados os dados de histórico local e histórico global respeitando a data dos jogos.

Entretanto, é importante ressaltar, que o nosso modelo tem uma grande dependência de dados históricos. Devido a isso, somente utilizamos os dados de 2013 e 2014 para compor o histórico, o que significa, que os dados dessas partidas não fizeram parte da base de treino e teste. Isso acontece pois, caso estejamos tentando prever um jogo do ano de 2013, os atributos que tem dependência global estariam com o valor nulo (já que 2013 é o primeiro ano do conjunto de dados).

Um problema apresentado pelos dados durante o pré processamento foi o grande desbalanceamento das classes. Ocorre que, nas partidas disponíveis no conjunto de dados, a maior parte dos resultados é vitória do mandante, isso causa o que chamamos de desbalanceamento entre as classes. Uma base de dados é dita desbalanceada quando existem muito menos casos de algumas classes do que de outras (CHAWLA; JAPKOWICZ; KOTCZ, 2004). Abaixo apresentamos um gráfico para ilustrar a relação entre as classes (vitória do mandante, vitória do visitante e empate)

```
[ symbolic x coords=mandante, visitante, empate, xtick=data ] [ybar,fill=black]
coordinates (mandante, 2974) (visitante, 1388) (empate, 958) ;
```

Figura 4.1: Distribuição de classes

Esse grande desbalanceamento das classes, exposto no gráfico 4.1, pode causar um efeito negativo na rede, fazendo com que a rede sempre preveja a classe super dominante (no nosso caso vitória do mandante). Levando em conta esse cenário é preciso avaliar o balanceamento das classes antes da realização do experimento.

Atualmente existem diversas técnicas para aliviar esse efeito, que já vem sendo estudado desde o crescimento inicial da área de mineração de dados (FU; MENDEL, 1970). Em nosso experimento utilizamos um ajuste no peso das classes. Classes predominantes passam a ter peso menor nos exemplos da rede durante o treinamento e isso cria a possibilidade das classes não predominante de terem maior influência na rede.

4.2 Resultados

Nas tabelas seguintes apresentamos os resultados obtidos para os experimentos descritos anteriormente:

experimento	acurácia teste
experimento I	48,47%
experimento II	46,60%

Tabela 4.3: Acurácia de treino e teste

Como podemos ver em 4.2 os experimentos tiveram resultado similar. O experimento I teve a maior acurácia 48,47% e essa é considerada a acurácia média do modelo. Nas tabelas 4.4 e 4.5 temos respectivamente as matrizes de confusão dos experimentos I e II.

	vitória mandante	empate	vitória visitante
vitória mandante	36,6%	6,0%	7,7%
empate	15,8%	4,2%	4,5%
vitória visitante	13,8%	3,3%	7,6%

Tabela 4.4: Matriz de confusão do **experimento I**

	vitória mandante	empate	vitória visitante
vitória mandante	35,4%	4,5%	13,9%
empate	13,9%	2,1%	8,1%
vitória visitante	10,6%	2,1%	9,0%

Tabela 4.5: Matriz de confusão do **experimento II**

Analisando os resultados obtidos, 48,47% e 46,60%, nos experimentos I e II, respectivamente, teremos um resultado maior que a aleatoriedade (que daria aproximadamente 33,3%). Podemos entretanto perceber, nas tabelas 4.4 e 4.5, que ambos os experimentos tiveram a acurácia concentrada na classe vitória do mandante (a classe dominante no dataset disponível).

Dessa forma, vale analisar as matrizes de confusão dos experimentos. Fica claro que, mesmo com os ajustes dos pesos de classe, a classe dominante continua sendo a vitória do mandante, tendo a maior acurácia do experimento. A classe empate permanece com a acuarria baixa e sendo a mais difícil de acertar.

Ainda analisando as matrizes de confusão, pode-se perceber, que em ambos os experimentos, caso direcionássemos todas as previsões para a classe vitória mandante teríamos uma acurácia maior do que a obtida pelo modelo. No experimento I obteríamos uma acurácia de 50,3% e no experimento II 53,8%.

Analisando detalhadamente o experimento II é possível perceber que existe uma pequena evolução na acurácia do modelo nas rodadas finais do campeonato (onde teoricamente já se tem uma boa quantidade de informação acumulada). Caso fossemos comparar o desempenho do algoritmo entre a sexta e a vigésima segunda rodada (6 até 22) teríamos uma acurácia de 41,1%. Porém, analisando da vigésima terceira até a trigésima oitava (23 até 38) teríamos uma acurácia de 52,5%.

O resultado detalhado do experimento II pode ser visto no gráfico a seguir.

Capítulo 5

Conclusões

5.1 Considerações

Este trabalho propôs um modelo, com dados de fácil acesso, para a previsão de partidas. As experimentações feitas com dados do Campeonato Brasileiro de futebol deixaram evidente as limitações do modelo, porém mostram bons resultados iniciais.

Além disso, ficou claro a importância de bases de dados sólidas para este tipo de modelo. Nem sempre é fácil encontrar dados públicos sobre os campeonatos, muitos são feitos de forma amadora o que causa uma certa imprecisão.

5.2 Limitações e trabalhos futuros

Os experimentos mostram uma grande dependência do histórico local bem definido para a contextualização de status e moral dos clubes. Isso gera uma grande problemática para prever jogos iniciais de um campeonato. Muitos trabalhos utilizam atributos artificiais para criarem esses contextos, porém isso requer algum conhecimento prévio dos times participantes do campeonato que se está tentando prever.

Essa dependência de que os campeonatos possuam uma base de dados histórica bem definida dificulta a previsão para campeonatos que sejam novos ou não ocorram

com frequência (como a copa do mundo). Um outro fator, é que a necessidade de formar um histórico local para a contextualização da moral dos times pode ser um grande problema em campeonatos curtos (com poucas rodadas).

Uma limitação do modelo é a dificuldade em prever empates e vitória do visitante. Em nossa base de dados, do Campeonato Brasileiro de futebol, tivemos um grande desbalanceamento das classes que não necessariamente se repete (com a mesma intensidade) em outros campeonatos, por isso, seria muito importante experimentar o modelo em outros campeonatos.

Além disso, um possível problema, é que com o passar de uma edição corrente do campeonato, os dados locais (que são usados para contextualizar o fase/moral atual do clube) perdem seu efeito. Uma alternativa para esse problema seria a utilização de uma variável que contenha o desempenho nas últimas três partidas.

É de suma importância dizer que os clubes estão simultaneamente em mais de uma competição, utilizar informações a cerca de competições paralelas pode ser muito útil para criar a contextualização de moral do clube além de ser uma ótima forma de impulsionar a geração de mais dados. O grande contraponto é a dificuldade de obtenção desses dados.

Referências

CHAWLA, N. V.; JAPKOWICZ, N.; KOTCZ, A. Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, ACM, v. 6, n. 1, p. 1–6, 2004.

FU, K. S.; MENDEL, J. M. *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications*. [S.l.]: Academic Press, 1970.

GUHA, S. et al. Soccer result prediction using data mining through predictive analytics. AIT, 2017.

HEBB, D. O. *The organization of behavior: A neuropsychological approach*. [S.l.]: John Wiley & Sons, 1949.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982.

LANDAHL, H.; MCCULLOCH, W. S.; PITTS, W. A statistical consequence of the logical calculus of nervous nets. *Bulletin of Mathematical Biology*, Springer, v. 5, n. 4, p. 135–137, 1943.

MIKOLOV, T. et al. Extensions of recurrent neural network language model. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. [S.l.], 2011. p. 5528–5531.

MIN, B. et al. A compound framework for sports results prediction: A football case study. *Knowledge-Based Systems*, Elsevier, v. 21, n. 7, p. 551–562, 2008.

MINSKY, M.; PAPERT, S. *Perceptrons*. MIT press, 1969.

RESEARCH, U. of Illinois at Urbana-Champaign. Center for S.; DEVELOPMENT; CYBENKO, G. *Continuous valued neural networks with two hidden layers are sufficient*. [S.l.: s.n.], 1988.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

WAGNER, M. M. et al. The emerging science of very early detection of disease outbreaks. *Journal of Public Health Management and Practice*, ASPEN PUBLICATION, v. 7, n. 6, p. 51–59, 2001.

WILLIAMS, D.; HINTON, G. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–538, 1986.