

Estudo do impacto da variação de parâmetros em uma rede neural artificial aplicado a bases com diferentes características

Guilherme Fontana Kilson¹, Felipe Roncalli De Paula Carneiro¹

¹Centro Universitario Presidente Antônio Carlos (UNIPAC)
Rua Dr. Antonino Sena Figueiredo, n° 807 - Santa Tereza I, Barbacena - MG - Brasil

guilhermekilson@yahoo.com.br, felipecarneiro@unipac.br

Resumo. *No decorrer do crescimento tecnológico desde o século passado, a computação foi uma das áreas mais afetadas, apresentando avanços significativos em todos os seus ramos, inclusive o da inteligência artificial. Essa área por sua vez busca desenvolver sistemas capazes de simular o raciocínio humano. Dentro desse campo, destaca-se a inteligência computacional, que utiliza técnicas computacionais para resolver problemas complexos, um exemplo promissor dessa área são as redes neurais artificiais, inspiradas no funcionamento do cérebro humano, que têm aplicações em diversos setores da sociedade. O propósito deste estudo é avaliar a sensibilidade da variação dos parâmetros em uma rede neural artificial do tipo multilayer perceptron (MLP), investigando seu impacto em problemas distintos. Para isso, foi realizado uma análise de desempenho diversificando os parâmetros de um algoritmo específico de redes neurais.*

Palavras-chave: *Redes Neurais Artificiais; Neurônio; Múltiplas Camadas; Inteligência Artificial.*

Abstract. *In the course of technological growth since the last century, computing has been one of the most affected areas, showing significant advances in all its branches, including artificial intelligence. This area, in turn, seeks to develop systems capable of simulating human reasoning. Within this field, computational intelligence stands out, which uses computational techniques to solve complex problems, a promising example of this area are the artificial neural networks, inspired by the functioning of the human brain, which have applications in various sectors of society. The purpose of this study is to evaluate the sensitivity of parameter variations in an artificial neural network of the multilayer perceptron (MLP) type, investigating its impact on different problems. For this, a performance analysis was performed by diversifying the parameters of a specific neural network algorithm.*

Keywords: *Artificial Neural Networks; Neuron; Multiple Layers; Artificial Intelligence*

1. Introdução

Desde o século passado, a pesquisa e o desenvolvimento em tecnologias para computadores e seus mais diversos ramos expandiu, pois essa área foi impulsionada pelo crescimento tecnológico presente, principalmente, durante a Segunda Guerra Mundial e de maneira subsequente, no período da Guerra Fria. Algumas inovações nesse segmento

foram o transistor, micro chip, consolidação de centros de produção, pesquisa para computadores (como exemplo: o Vale do Silício), construção dos semicondutores e do circuito integrado por Robert Noyce e Jack Kilby, surgimento de mainframes e o desenvolvimento do microprocessador [Junior 2012].

Nos primeiros anos de seu surgimento, o uso era destinado a cálculos complexos, problemas específicos e estreitamente definidos e a fim de pesquisas diversas. Foi em 1981 com o lançamento do computador pessoal, que a popularização do mesmo aconteceu, devido a pesquisas em circuitos integrados, os chips, que foram responsáveis pela crescente miniaturização dos equipamentos eletrônicos. Desde então, houve um aumento no desenvolvimento tecnológico com inovações em softwares, processadores, arquiteturas, assim como em disquetes e discos rígidos [GADELHA 2009].

As primeiras pesquisas com inteligência artificial apontam à década de 50 com John McCarthy definindo como, "Fazer a máquina comportar-se de tal forma que seja chamada inteligente caso fosse este o comportamento de um ser humano" [McCarthy 2007]. Desse modo, a inteligência artificial (IA) é um campo multidisciplinar que busca desenvolver sistemas capazes de imitar a capacidade humana de aprendizado, raciocínio e tomada de decisões.

Embora muito ampla, uma das vertentes da inteligência artificial é a inteligência computacional, essa área refere-se ao conjunto de técnicas computacionais capazes de resolver problemas complexos nos quais as abordagens tradicionais são ineficazes. Essas técnicas utilizam o poder de processamento dos computadores em cálculos de operações matemáticas, para permitir que os algoritmos consigam desempenhar tarefas de classificação de padrões, de aproximação de funções, de previsão de séries temporais e de otimização. Cada ferramenta tem um processo de aprendizagem computacional, na qual o sistema se apropria de informações dos dados para conseguir aprender [Kruse et al. 2013].

Ainda na área de inteligência artificial, um dos campos mais conhecidos são as redes neurais artificiais (RNAs), que são inspiradas no funcionamento do cérebro humano. Uma RNA é um modelo computacional que foi desenvolvido inspirado no funcionamento do cérebro humano, elas são compostas por neurônios artificiais interconectados entre si, eles são capazes de processar informações, realizar cálculos e transmitir sinais através de conexões ponderadas, denominadas sinapses artificiais. Existem vários modelos de RNAs, e um dos mais difundidos é o de redes *Multilayer Perceptron* (MLP). O *perceptron* multicamadas é uma importante arquitetura de redes neurais, ele é formado por uma camada de entrada, camada intermediária e a camada de saída [Araújo 2015].

As redes neurais artificiais representam um importante setor da inteligência artificial presente hoje em muitos segmentos da sociedade e contendo também diversas aplicações no dia a dia, dada a sua capacidade de aprender a partir de dados inseridos e treinados, a cada iteração o resultado desejado é refinado. Desse modo, suas aplicações podem ser em diversos setores da sociedade, como: saúde, economia, transporte, educação, indústria de máquinas, isso deve-se também ao fato de poder induzi-la a agir em prol da solução de casos em um nicho ou problema específico. Alguns trabalhos como exemplo, são: Sistema hipermídia sobre câncer de colo de útero com interface adaptativa usando Redes Neurais Artificiais MLP e Sistema Especialista

[Malinverni et al. 2006]; utilizando Redes Neurais Artificiais e Regressão Logística na Predição de Hepatite A [Santos et al. 2005]. A IA está sendo usada para otimizar processos, melhorar a tomada de decisões, automatizar tarefas e até mesmo auxiliar na criação de novos produtos e serviços.

Os sistemas que utilizam redes neurais são uma propriedade fundamental. A habilidade de aprender a partir de seu ambiente e, com isso, extrair conhecimento. Isso deve-se a um método de treinamento, consistindo num processo iterativo de ajustes, aplicando o algoritmo de treinamento a seus pesos sinápticos [Guimarães et al. 2008].

1.1. Objetivos

O artigo em questão tem como objetivo analisar o algoritmo usado quanto a sua sensibilidade e variação dos parâmetros do código em bases de dados diferentes, e como isso impacta no resultado final.

Quanto aos objetivos específicos, são eles:

1. Analisar o impacto da variação de parâmetros como: número de *folds*, taxa de aprendizagem, número de épocas de treinamento e número de neurônios na camada oculta; no resultado final. Além de esclarecer se para aquela base de dados o algoritmo teve um bom resultado ou não. Esses conceitos serão explicitados no referencial teórico.
2. Esclarecer para cada base de dados se o algoritmo atingiu um resultado aceitável, já que as bases apresentam dados que estão inseridos em contextos distintos.
3. Comparar os resultados do estudo em questão com outros que tiveram a mesma proposta utilizando as mesmas bases de dados usadas nesse estudo.

1.2. Justificativa

Percebe-se como exposto anteriormente, um aumento expressivo no uso da inteligência artificial como um todo e também na conjuntura do uso das redes neurais nas mais diversas áreas.

Entretanto, alguns estudos apontam que as redes do tipo MLP apresentam alguns inconvenientes nas tarefas de classificação de padrões. Dentre eles é a longa duração do período de treinamento que exige um grande esforço computacional. Logo em seguida, tem-se o problema de se determinar os parâmetros de treinamento, como a taxa de aprendizagem, o tamanho da camada oculta e a escolha dos pesos sinápticos iniciais, pois, tais parâmetros apresentam um papel significativo no desempenho das redes do tipo MLP. Apesar de conter diversos estudos que apontam processos e metodologias para identificação de parâmetros, ainda não existe um método exato para uma boa escolha de parâmetros [Manzan et al. 2016]. Alguns dos estudos são de: [Isa and Mamat 2011] e [Kim 2005].

Este trabalho realiza uma pesquisa e elabora uma discussão a respeito dos parâmetros escolhidos em uma rede neural artificial do tipo MLP usando do *error Back Propagation* como método de treino. E também analisa o impacto da diversificação dos parâmetros (número de *folds*, taxa de aprendizado, número de épocas e número de neurônios na camada oculta) em 3 bases de dados diferentes e como isso afeta o resultado final.

2. Referencial Teórico

O referencial teórico tem como objetivo embasar teoricamente a pesquisa, fornecendo um contexto teórico e conceitual para o estudo.

2.1. Redes Neurais Artificiais (RNA)

Raymund Kurzweil, inventor, futurista e autor de diversos livros, estabeleceu uma das diversas definições de inteligência artificial acerca do seu comportamento: “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas” [Kurzweil et al. 1990]. Além disso, os novos sistemas que surgiam tinham por finalidade pensar como seres humanos. “O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal” [Haugeland 1989].

De acordo com [Fleck et al. 2016], tem sido observado o desenvolvimento de processos industriais cada vez mais complexos, onde o avanço das tecnologias que conduzem as tomadas de decisões tornou um fator decisivo para a inserção e manutenção de um produto ou serviço no mercado moderno. Os autores ainda destacam que o uso de Redes Neurais Artificiais (RNAs) está sendo cada vez mais uma opção atrativa, tendo em vista a atual ampliação da capacidade computacional.

Uma das diversas definições de redes neurais foi feita por Simon Haykin: “Uma rede neural é maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso” [Haykin 2001]. Ademais, o autor diz que o conhecimento é obtido pela rede por meio de um processo de aprendizagem, e a força de conexão entre os neurônios (pesos sinápticos), são usados para armazenar o conhecimento adquirido.

As redes neurais são, portanto, uma ferramenta de inteligência artificial que evidencia a melhor forma de dados serem processados de uma maneira ágil, inspirados no funcionamento do cérebro humano. Desta forma, as RNAs são modelos matemáticos que se assemelha às estruturas neurais biológicas e adquirem capacidade por meio de aprendizado e posterior generalização [Nunes 2003].

2.2. Modelo de um neurônio artificial

Um neurônio artificial é a unidade fundamental de uma rede neural artificial, como o neurônio biológico [Amancio 2013]. De acordo com [Haykin 2001], existem três elementos que são a base de um modelo neuronal, são eles: um conjunto de sinapses ou elo de conexão, cada uma caracterizada por um peso ou força própria; um somador para somar os sinais de entrada ponderando pelas respectivas sinapses do neurônio e uma função de ativação para restringir a amplitude de saída de um neurônio. A figura 1 evidencia e exemplifica os elementos citados anteriormente.

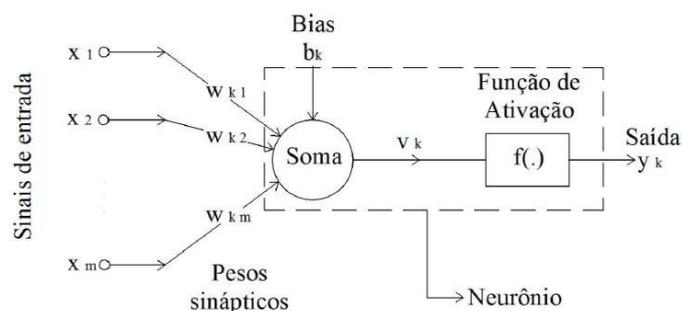


Figura 1. Neurônio artificial proposto por McCulloch e Pitts em 1943.
Fonte: Araújo (2013).

Dessa forma, [Nunes 2003] exhibe que a operação de um neurônio da rede neural se resume em quatro passos: primeiro os sinais são apresentados a entrada, cada sinal é multiplicado por um peso que indica sua influência na saída da unidade, é feita a soma ponderada dos sinais que produz um nível de atividade e se este nível produz um limite (*threshold*) a unidade produz a saída.

2.3. Arquitetura da rede neural artificial

Conforme [Amancio 2013], a arquitetura de uma rede neural é a maneira como estão estruturados os neurônios na rede. Ao se desenvolver uma rede neural artificial, uma das escolhas mais difíceis é a da arquitetura. Essa arquitetura irá afetar o desempenho da rede bem como as aplicações para o qual ela é desejada e sua estrutura está intimamente ligada ao algoritmo de aprendizado usado para o seu treinamento [Raia Junior 2000]. O autor pode escolher dentre alguns modelos, mas no geral elas são subdivididas em 3 classes: rede neural *Feedforward* de 1 camada, rede neural *Feedforward* Multicamadas e redes recorrentes ou realimentadas [Furtado 2019]. Para este artigo científico, a rede neural utilizada foi a *multilayer perceptron* (MLP), ela pertence a segunda classe: rede neural *Feedforward* Multicamadas.

Diferentemente de uma rede neural de camada única, a arquitetura de uma rede neural de múltiplas camadas contém uma ou mais camadas ocultas, que também são chamadas de neurônios ocultos ou unidades ocultas. A função dos neurônios ocultos é intervir entre a camada externa e a saída da rede de uma maneira útil [Haykin 1994]. Ainda de acordo com Haykin, o diferencial desse tipo de rede é a presença de uma ou mais camadas ocultas que possibilitam o aprendizado de tarefas mais complexas.

A figura 2 ilustra a arquitetura de uma rede neural de múltiplas camadas. Nela por sua vez observa-se uma camada de entrada contendo três nós conectados com a camada oculta que possui duas camadas, essas por sua parte ligadas a camada de saída composta por N neurônios.

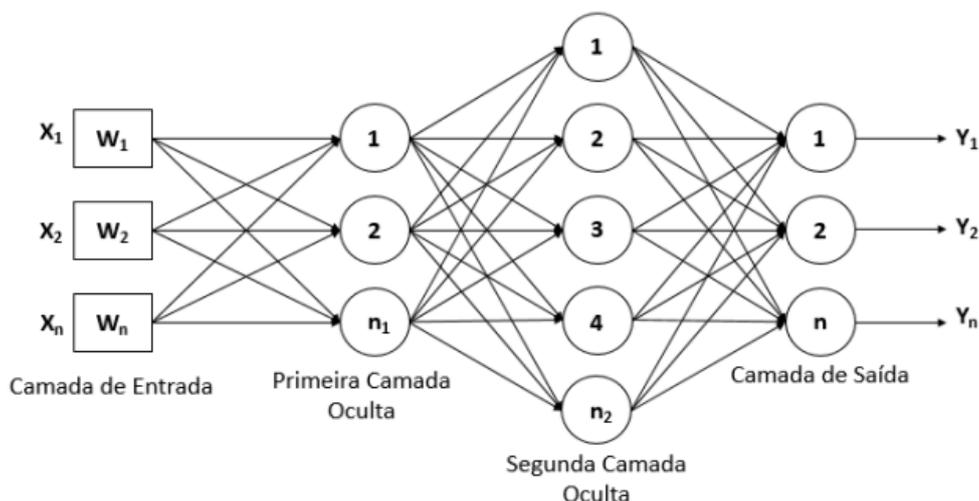


Figura 2. Rede neural de múltiplas camadas.
Fonte: Haykin (1994).

De acordo com o autor [Haykin 2001], os nós de fonte da camada de entrada da rede fornecem os respectivos elementos do padrão de ativação, que constituem os sinais de entrada aplicados ao neurônio na segunda camada. Ainda assim, os sinais de saída da segunda camada são utilizados como entrada para a terceira camada e assim por diante por toda a rede. Consequentemente, as entradas dos neurônios em cada camada da rede serão os sinais de saída da camada precedente.

Segundo [Silva et al. 2010], a topologia de uma RNA está ligado às diferentes formas de composições estruturais que as mesmas podem assumir. Sendo assim, a topologia de uma RNA condiz com a forma com que as conexões estão organizadas, levando em conta o número de neurônios nas camadas ocultas e as funções de ativação de cada neurônio.

2.3.1. Perceptron Multicamadas - MLPs

Rosenblatt em 1958 introduziu o *perceptron* como uma forma simples de RNA cuja principal aplicação se dá nos problemas de classificação de padrões, assim como o *perceptron* de camada única só é capaz de classificar padrões linearmente separáveis [Fleck et al. 2016].

O *perceptron* multicamadas é uma rede neural multicamadas alimentada adiante, constituído por três camadas: o primeiro conjunto é formado por unidades sensoriais situados na camada de entrada, cuja função é a de receber as informações externas; uma ou mais camadas intermediárias ou também denominadas ocultas, responsáveis pelo aumento da capacidade da rede neural artificial em extrair o comportamento mais complexo a partir dos dados disponíveis utilizados no processo de treinamento; e a camada de saída, cuja função é a de dar as respostas corretas para os estímulos do modelo [Araújo 2015]. De acordo com [Barros et al. 2018], os sinais de entrada são propagados camada a camada pela rede em uma direção positiva, ou seja, da entrada, passando pela camada oculta até a camada de saída.

Segundo [Haykin 2001], uma rede do tipo MLP tem três características que a distinguem dos demais tipos de rede:

- Para cada neurônio da rede há uma função de ativação não linear, cuja curvatura é suave.
- A rede contém uma ou mais camadas ocultas, que são diferentes da camada de entrada e de saída. Os neurônios dessas camadas ocultas são responsáveis pela capacidade de aprendizagem de problemas complexos.
- Existe um alto grau de conectividade entre os neurônios, isso significa que um neurônio de qualquer camada da rede está conectado a todos os neurônios da camada anterior.

Desse modo, essa rede neural tem sido utilizada com frequência para solução de problemas complexos, através de seu treinamento supervisionado, utilizando um algoritmo conhecido como Algoritmo de Retropropagação do Erro (*Error Back Propagation*) que é baseado na regra de aprendizagem por correção do erro [Haykin 2001].

2.4. Algoritmo de Retropropagação do Erro - *Error Back Propagation*

Existem diversos tipos de algoritmos de aprendizagem para uma rede MLP, porém um dos mais conhecidos é chamado de *backpropagation* que em português, corresponde a retropropagação. Esse nome é justificado pelo fato de que o proveniente da diferença entre a saída estimada da rede com a saída esperada é usado no ajuste de todos os pesos sinápticos da rede. Desse modo, cada padrão tem seu sinal propagado da entrada até a saída, e o sinal proveniente do erro é retropropagado da saída para a entrada [Manzan et al. 2016].

[Raia Junior 2000], afirma que o termo *backpropagation* surge do fato de que o algoritmo se baseia na "retropropagação" dos erros para os ajustes de pesos das camadas intermediárias, isto é, quando um padrão é apresentado à rede pela primeira vez, esta produz uma saída aleatória - a diferença entre esta saída e a desejada constitui o erro.

O processo de aprendizagem de um algoritmo através do algoritmo de retropropagação do erro consiste em dois passos através de todas as camadas da rede, são eles: passo para frente, a propagação, e passo para trás, a retropropagação. No passo para frente, o padrão do vetor de entrada é aplicado aos nós sensoriais da camada de entrada, propagando-se camada por camada, por meio dos pesos sinápticos que são fixos, até a camada de saída produzindo uma resposta. Logo após no passo para trás, os dados produzidos pela camada de saída são comparados à saída desejada para o estímulo dado à rede, conseqüentemente gerando-se um erro que se propaga desde a camada de saída até a camada de entrada, os neurônios então farão o ajuste dos seus pesos sinápticos para reduzir os seus erros [Haykin 2001].

Conforme [Manzan et al. 2016], nesse algoritmo de aprendizagem algumas variáveis são definidas, como: a taxa de aprendizagem da rede, o número de neurônios da camada oculta e o critério de parada que pode ser, por exemplo, ao se atingir um certo número de ciclos (ou épocas) ou pela determinação de um erro máximo admissível.

Em relação a definição do número de neurônios artificiais que compõem uma camada intermediária ou oculta de uma rede, [de Pádua Braga et al. 2007] afirma que esse número depende de vários fatores, como: a variação dos dados de treinamento, quantidade de ruído ou viés contido nos dados de treinamento, complexidade da função a ser aprendida e a proximidade a uma distribuição estatística dos dados para treinamento da rede.

Já em relação a taxa de aprendizado, de acordo com [Manzan et al. 2016], ela representa o tamanho do passo que a rede toma em busca do ajuste ideal dos pesos sinápticos. Passos muito pequenos podem fazer com que o treinamento fique lento. Passos muito longos podem conduzir o ajuste a direções incorretas em relação à direção ideal do treinamento, o que significa que o próximo passo irá requerer correção do erro anterior.

No que se refere ao critério de parada, muito se discute sobre isso referente ao momento ideal para encerrar treinamento de uma rede do tipo MLP. Sob condições normais, a curva do erro quadrático médio (MSE) diminui ao longo da evolução do número de ciclos. Vale ressaltar que o MSE é uma medida que avalia o desempenho da rede neural ao quantificar a diferença entre as saídas previstas e os valores reais. Entende-se por ciclo a apresentação de todo o conjunto de amostras de treinamento durante a fase de aprendizagem da rede MLP. Normalmente, os padrões usados no treinamento possuem

ruídos. Se a rede treinar muito, ela pode aprender com os ruídos dos padrões do conjunto de treinamento. Isso leva à perda de generalidade, fenômeno esse denominado *overfitting* [Manzan et al. 2016].

Ainda assim, de acordo com [Haykin 1994], existe uma técnica chamada *early stopping*, a fim de evitar esse problema. Além do conjunto de treinamento, também é usado um conjunto de validação. Periodicamente, o conjunto de validação é submetido à classificação pela rede com os pesos sinápticos ajustados naquele ciclo de treinamento. O erro quadrático médio do conjunto de validação é calculado. Se o erro começa a aumentar, a rede está começando a aprender com os ruídos do conjunto de treinamento, ou seja, está caminhando para uma condição de *overfitting*. Esse é o momento de se encerrar o treinamento. A isso é chamado *early stopping*.

2.5. Validação Cruzada

Primeiramente, divide-se o conjunto de dados em dados de treino e teste, ao invés de usar todo o montante para treino e, em seguida, executar o teste nos mesmos dados. A validação cruzada é um método de re-amostragem usado para avaliação de modelo para evitar o teste de um modelo no mesmo conjunto de dados no qual ele foi treinado. Existem diversos métodos de validação cruzada, entre eles: método *Leave-One-Out* (LOO), método *holdout*, método *K-Fold*, entre outros [Alex 2019b].

2.5.1. Método *k-fold cross validation*

O método para validação do conjunto de dados usado nesse algoritmo foi o *k-fold Cross Validation*.

Sendo assim, no método *k-fold*, "a amostra d é dividida em K partes (d_1, d_2, \dots, d_k) de tamanho parecido m_k . O processo terá K interações onde, em cada interação, a amostra de validação será dada por d_k , com $k = 1, 2, \dots, K$, e a amostra de treino para a criação do preditor será o conjunto das outras $K-1$ partes, ou seja, $d(-k) = d_1, d_2, \dots, d_{k-1}, d_{k+1}, \dots, d_k$. Assim, ao final dos K passos, teríamos usado todos os dados tanto na parte de treino, quanto na parte de validação" [Burman 1989].

Ainda assim, nesse tipo de validação os dados conhecido são divididos em k partes iguais. Em seguida, é executado um processo iterativo que determina $k-1$ partes que definem o conjunto de treinamento e uma parte selecionada que define o conjunto de validação, e a cada passo do processo esse conjunto é alternado. A cada iteração realizada são feitos " k " treinamentos e aplicações de cada RNA treinada ao grupo de validação. Depois é calculado e armazenado o erro médio quadrático referente à cada aplicação de topologias testadas. Então, em cada topologia testada são obtidos " k " vetores contendo " k " erros médios quadráticos (EMQ) computados para cada iteração [Pelli 2013].

Desse modo, a escolha da melhor topologia que se adeque ao problema será determinada pelo menor EMQ médio, associado ao desvio padrão dos erros. Após sua definição, são feitos treinamentos da RNA com ela, sendo necessário escolher qual é a melhor rede treinada. Devido as RNAs serem da categoria de algoritmos chamados de heurísticas, a cada treinamento é obtido um resultado diferente. O que se espera é que seja escolhida a RNA com maior precisão na generalização [Pelli 2013].

3. Trabalhos relacionados

Esta seção destaca alguns trabalhos relacionados com o tema proposto nesse artigo, com a finalidade de mostrar alguns dos estudos feitos e resultados obtidos sobre aplicações parecidas.

[Kim 2005] realizou um estudo no âmbito das redes neurais artificiais, onde expôs que o modelo de retropropagação do erro apresenta um problema que é a sensibilidade ao valor inicial dos pesos. Já o desempenho em termos de precisão de previsão e custo de computação depende muito dos pesos iniciais. Desse modo, é proposto no artigo uma regra de previsão para minimizar o efeito dos pesos iniciais e melhorar a precisão da previsão nos conjuntos de dados de teste. Os resultados experimentais indicaram uma melhoria considerável no desempenho da predição, usando uma regra de predição proposta.

[Souza and Ramos 2021] elaboraram um artigo onde analisam redes neurais convolucionais usadas para reconhecimento de dígitos escritos à mão. Foram desenvolvidos cinco modelos com parâmetros fixos, como filtros convolucionais 3x3, cinco camadas e funções de ativação específicas. Os modelos foram treinados usando um conjunto de dados de imagens e avaliados com base no tempo de treinamento e acurácia. Os resultados mostraram que o parâmetro stride é crucial para a acurácia e o tempo de treinamento dos modelos. Modelos com um valor de stride menor alcançaram maior acurácia, mas exigiram mais tempo de treinamento. O modelo com maior acurácia teve o tempo de treinamento mais longo.

3.1. Trabalhos relacionados à base de dados

Esta seção irá abordar a respeito dos trabalhos encontrados que utilizaram alguma das bases de dados usadas neste estudo, realizando estudos semelhantes ao proposto neste trabalho. Dentro do tempo de pesquisa deste trabalho em questão, não foi encontrado trabalhos relacionados à primeira base de dados. A seção 4.1.1 irá expor os detalhes das bases de dados com mais clareza.

Em se tratando da segunda base de dados(Centro de Serviço de Transfusão de Sangue), [Barhoom 2019] elaborou um estudo onde foi proposto um modelo de RNA para prever se uma pessoa vai doar sangue ou não. O modelo usado foi a de uma rede neural *perceptron* multicamadas(MLP), dessa forma o autor recorreu ao uso de uma ferramenta chamada *JustNN* para construir o modelo de RNA. A arquitetura desse modelo foi determinada para conter: uma camada de entrada, 2 camadas ocultas e uma camada de saída. Em relação à base de dados que foi utilizada, os dados de origem foram retirados do banco de dados de doadores de sangue do Centro de Serviços de Transfusão de Sangue em *Hsin-Chu* Cidade em Taiwan. Ele particionou o conjunto de dados aleatoriamente em conjunto de treinamento e conjunto de validação usando uma partição de treinamento/validação de aproximadamente 70/30. O modelo foi treinado para 80219 ciclos e a precisão da validação obtida foi 99,31%.

Quanto a terceira base de dados (*Haberman's Survival*), [Shawarib et al. 2020] realizou uma pesquisa e elaborou um método através de um modelo de RNA para poder determinar se o paciente depois da cirurgia sobreviveu 5 anos ou mais ou morreu em 5 anos. O modelo usado foi a de uma rede neural *perceptron* multicamadas(MLP), o ambiente utilizado para construí-lo foi o *JustNN*, a arquitetura desse modelo conta com uma

camada de entrada, 2 camadas ocultas e uma camada de saída. Em relação à base de dados, foi utilizado o conjunto de dados de sobrevivência ao câncer de mama de *Haberman*. Ele particionou o conjunto de dados de um total de 306 instâncias sendo 204 amostras de treinamento e 102 amostras de validação. O modelo foi treinado para 31452 ciclos e a precisão da validação obtida foi 88,24%.

4. Metodologia

4.1. Material usado

4.1.1. Base de dados

Nesta pesquisa foram utilizadas 3 bases de dados diferentes, o conjunto de dados são multivariados e os atributos apresentam características de números reais. O problema que abrange a aprendizagem é um problema de classificação. Quanto às bases de dados, são elas:

- A primeira base de dados é composta por 210 instâncias e 7 atributos. Em seu contexto é apresentada medidas geométricas de sementes pertencentes a três diferentes variedades de trigo, são eles: Kama, Rosa e Canadense[Charytanowicz and Lukasik 2012]. A base de dados pode ser acessada através do site: <http://archive.ics.uci.edu/dataset/236/seeds>.
- A segunda base de dados é composta por 748 instâncias e 5 atributos. Seus dados foram retirados do Centro de Serviços de Transfusão de Sangue na cidade de *Hsin-Chu* em Taiwan. Em seu contexto é apresentado dados como recência, frequência, monetária (total de sangue doado), tempo e uma variável binária. Seu objetivo foi com esses dados montar um modelo de marketing de RFMTC (Recência, Frequência, Valor Monetário, Tempo, *Churn rate*)[Yeh 2008]. A base de dados pode ser acessada através do site: <http://archive.ics.uci.edu/dataset/176/blood+transfusion+service+center>.
- A terceira base de dados é composta por 306 instâncias e 3 atributos. Seus dados foram retirados da base de dados do *University of Chicago's Billings Hospital*. Nela é apresentado um estudo conduzido entre 1958 e 1970 a respeito da sobrevivência de pacientes que se submeteram a cirurgia para câncer de mama. Seus atributos são: a idade do paciente no momento da operação, ano de operação do paciente, número de linfonodos axilares positivos detectados e uma variável(1 ou 2) indicando se o paciente sobreviveu mais de 5 anos após a cirurgia ou menos. [Haberman 1999]. A base de dados pode ser acessada através do site: <http://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival>.

Essas bases de dados foram retiradas do site do repositório de aprendizado de máquina da *UC Irvine da University of California, Irvine* [Dua and Graff 2019].

4.1.2. Algoritmo

O algoritmo utilizado para os teste foi uma rede neural artificial *feed-forward* com código em *python*. A validação foi feita com *k-fold cross validation*, o treinamento com *error back propagation*, técnica essa ainda muito utilizada para treinar grandes redes de

aprendizagem profunda. Esse algoritmo conta com uma camada de entrada, uma camada intermediária e uma de saída. O trabalho foi baseado na implementação de Alex Souza [Alex 2019a].

4.2. Preparação dos dados

As informações das 3 bases de dados se encontram em sua forma bruta, ou seja, sem nenhuma alteração. Eles variam entre si em dezenas, centenas ou milhares de unidades. Visto isso, durante a execução do algoritmo, uma parte do mesmo é voltado para uma etapa chamada de normalização das entradas.

Na etapa de normalização cada variável de entrada deve ser pré-processada no intuito de que a média calculada sobre todo o conjunto de treinamento esteja entre 0 e 1, ou que seja pequena em comparação ao desvio-padrão. Sem essa condição, o vetor peso de determinado neurônio pode ter dificuldades para convergir, tornando o treinamento lento [LeCun 1993]. Ainda assim, ele aponta dois passos a serem efetuados na normalização dos dados de entrada, são eles: o conjunto de treinamento contém as variáveis de entrada que deverão ser não correlacionadas, o que pode ser verificado por meio da análise de componentes principais; além de serem descorrelacionadas, as variáveis de entrada devem ser escolhidas de forma que suas covariâncias sejam aproximadamente iguais, a fim de que os diferentes pesos sinápticos da rede sejam treinados com velocidade aproximadamente igual. Entretanto em alguns casos, pode existir algum tipo de correlação entre elas.

4.3. Implementação

A pesquisa trata-se de alterar os quatro parâmetros do código e contabilizar o resultado para cada base de dados, são eles: número de *folders*, taxa de aprendizagem, número de épocas de treinamento e número de neurônios na camada oculta.

Quanto ao número de *folders*, é empregado três variações, contendo 3, 4 e 5 *folders*. A taxa de aprendizagem varia de três formas, são elas: 0,10; 0,20; 0,30. O número de épocas varia de duas formas: 2000 e 10000 épocas. O número de camadas varia de três formas: 4, 5 e 6 camadas.

O estudo contém três bases de dados e cada uma foi executada 162 vezes, sendo que cada uma delas contém 54 cenários diferentes, isso é resultado de cada cenário ter sido executado três vezes. No final, totalizando 486 execuções, 162 vezes cada uma. A tabela onde os dados foram contabilizados contém uma coluna onde mostra o tempo gasto para a execução de cada interação ali presente.

Além disso, as interações foram feitas visando uma maior combinação entre as diversas opções de parâmetros disponíveis. Dessa forma, em um dos cenários onde o número de *folders* é 3, foram feitos o maior número de interações possíveis com as variações dos outros parâmetros possíveis, ou seja, nesse mesmo cenário teve variação de todos os outros parâmetros com todas as formas possíveis (dentro do especificado). Na figura 3, é mostrado um pequeno trecho da tabela utilizada como parâmetro na contabilização dos resultados como exemplificação. Os dados nela contidos são referentes à segunda base de dados.

n_fold	l_rate	n_epoch	n_hidden	Tempo(seg)	Score 1	Score 2	Score 3	Score 4	Score 5	Accuracy
3,00	0,10	2000	4,00	191,87	77,91	81,92	77,91			79,25
3,00	0,10	2000	4,00	185,56	78,31	81,92	78,71			79,65
3,00	0,10	2000	4,00	188,07	77,91	81,92	77,91			79,25
3,00	0,20	2000	4,00	183,18	79,91	82,32	78,31			80,18
3,00	0,20	2000	4,00	187,21	79,91	82,32	78,31			80,18
3,00	0,20	2000	4,00	190,23	79,91	82,32	78,31			80,18
3,00	0,30	2000	4,00	194,20	79,91	81,52	78,71			80,05
3,00	0,30	2000	4,00	191,03	79,91	81,52	78,71			80,05
3,00	0,30	2000	4,00	184,94	79,91	82,51	78,71			80,05
4,00	0,10	2000	4,00	275,70	79,67	78,07	84,49	75,40		79,41
4,00	0,10	2000	4,00	283,97	79,14	78,07	84,49	74,86		79,14
4,00	0,10	2000	4,00	279,18	79,14	78,07	84,49	74,86		79,14
4,00	0,20	2000	4,00	287,13	80,21	76,47	84,49	74,86		79,01
4,00	0,20	2000	4,00	300,98	80,21	76,47	84,49	74,86		79,01
4,00	0,20	2000	4,00	281,93	80,21	76,47	84,49	74,86		79,01
4,00	0,30	2000	4,00	279,50	80,21	79,14	83,93	75,40		79,67
4,00	0,30	2000	4,00	278,29	80,21	79,14	83,95	75,40		79,67
4,00	0,30	2000	4,00	283,41	80,21	79,14	83,95	75,40		79,67

Figura 3. Trecho da tabela de parâmetros.

Fonte: Elaborado pelo autor.

Em cada combinação de parâmetro possível, o código foi executado três vezes e os resultados contabilizados. Isto acontece devido ao computador utilizado para fazer os procedimentos ser um computador para uso pessoal.

Diante disso, no contexto em que o trabalho foi feito e no tempo disponível para a sua conclusão, tornou-se inviável a aplicação desse procedimento em bases de dados com o número de instâncias superior a 800.

5. Resultados

Essa seção apresentará os resultados obtidos em cada base de dados, seguindo a ordem em que elas foram apresentadas na seção 4.1.1. Assim como também uma discussão acerca deles.

5.1. Primeira base de dados

Na presente base de dados é apresentado dados de medidas geométricas de sementes pertencentes a três diferentes variedades de trigo. Nessa base, foram gastos aproximadamente 15 horas em tempo de execução.

A melhor acurácia foi de 96,19%, enquanto a pior foi de 91,90%. A acurácia média entre todas as 162 interações foi de 94,17%.

5.1.1. Dez melhores resultados

O melhor resultado obtido, foi com os seguintes parâmetros: número de *folds* igual a 3, taxa de aprendizado igual a 0,1, número de épocas igual a 10000 e número de neurônios na camada oculta igual a 4; totalizando uma acurácia de 96,19%.

Dentre os dez melhores resultados, sete deles apresentam número de *folds* igual a 3; cinco deles apresentam uma taxa de aprendizado igual a 0,10; oito deles apresentam

número de épocas igual a 10000 e cinco deles apresentam número de neurônios igual a 4. Isso comprova que a combinação de parâmetros onde foi obtido o melhor resultado, é a melhor combinação para se obter a maior acurácia. Entretanto, é válido ressaltar que dentre esses resultados cinco deles apresentam número de neurônios igual a 4 e cinco deles apresentam número de neurônios igual a cinco. Porém, os três melhores resultados apresentam o número de neurônios igual a 4, por isso, ele está apontado como melhor.

A acurácia entre esses dez resultados variou de 95,23% até 96,19%, variando 0,96%. Essa variação é dada como grande, visto que a variação total entre as 162 interações foi de 4,29%. Com cálculos básicos chega-se a algumas conclusões: 10 interações correspondem a aproximadamente 6,1% do total de 162 e a variação em questão de 0,96% corresponde a aproximadamente 22,3% da variação total (4,29%). Portanto, em dez interações a acurácia variou um total de 22,3%. Os dados mencionados estão expostos abaixo na figura 4.

n_foli	l_raté	n_epo	n_hidd	Tempo(se)	Score	Score	Score	Score	Score	Accura
3,00	0,10	10000	4,00	327,86	97,14	95,71	95,71			96,19
3,00	0,20	10000	4,00	321,63	97,14	95,71	94,28			95,71
3,00	0,20	10000	4,00	318,87	97,14	95,71	94,28			95,71
3,00	0,20	10000	5,00	376,20	98,57	94,28	94,28			95,71
3,00	0,30	10000	5,00	380,13	98,57	94,28	94,28			95,71
5,00	0,10	10000	5,00	754,91	92,23	90,47	100,00	100,00	92,85	95,71
4,00	0,10	10000	4,00	478,94	100,00	96,15	94,23	92,30		95,67
4,00	0,10	10000	5,00	555,47	98,07	96,15	96,15	92,30		95,67
3,00	0,30	2000	4,00	64,59	97,14	94,28	94,28			95,23
3,00	0,10	2000	5,00	76,47	97,14	94,28	94,28			95,23

Figura 4. Dez melhores resultados da 1º base de dados

Fonte: Elaborado pelo autor.

5.1.2. Cinquenta melhores resultados

Dentre esses resultados, a acurácia varia de 94,76% até 96,19%, ou seja, uma variação de 1,43%. Tal porcentagem de variação é dada como pequena, pois levando-se em conta que aumentaram quarenta interações na tabela em relação às dez primeiras e elas tiveram uma mudança de 0,96%. Ou seja, as cinquenta primeiras interações ainda assim apresentam uma boa média de acurácia e, conseqüentemente suas combinações de parâmetros apresentam resultados satisfatórios.

Analiticamente, os parâmetros que se manifestaram na melhor acurácia da tabela, estão presentes dentre os cinquenta melhores como a maioria, com exceção do número de neurônios na camada escondida, pois ele como 4 está presente em apenas 17 das 50 interações; e da taxa de aprendizado, pois ela como 0,10 está presente em apenas 19 das 50 interações. Entretanto, os outros parâmetros mostraram bons resultados, como: o *fold* como 3 está em 38 das 50 interações, assim como também o número de épocas de treinamento como 10000 está em 28 das 50 interações.

5.1.3. Análise geral - 1º base de dados

O algoritmo foi capaz em todas as 162 interações de resultar em uma acurácia dos dados superior a 91,90%. Visto que a base de dados foi montada com o objetivo de através

das medidas geométricas das sementes de trigo, determinar seu tipo, o algoritmo então se mostrou capaz de apresentar uma solução para este problema. A melhor combinação de parâmetros apresentou uma taxa superior a 96%.

Além disso, as variações de parâmetros no algoritmo demonstraram impactos nos resultados, porém eles não foram relevantes o suficiente para que esses números decaíssem de forma considerável, comprovando então a robustez do algoritmo diante desse problema.

5.2. Segunda base de dados

Na base de dados em questão, é apresentado dados de um centro de transfusão de sangue a respeito de seus doadores, com o intuito de demonstrar o marketing RFMTC. Nessa base foram gastos aproximadamente 43,5 horas em tempo de execução.

O algoritmo apresentou a melhor acurácia como 80,50%, enquanto a de 77,40% inferior. A acurácia média entre todas as 162 interações foi de 79,29%.

5.2.1. Dez melhores resultados

O melhor resultado obtido foi com os seguintes parâmetros: número de *folds* igual a 4; taxa de aprendizado igual a 0,1; número de épocas igual a 10000 e número de neurônios na camada oculta igual a 5; totalizando uma acurácia de 80,50%.

Em relação ao número de *folds* , apesar de o melhor resultado apresentar o *fold* como 4, dentre os dez primeiros ele aparece apenas duas vezes, em contrapartida, o número de *folds* igual a 3 aparece sete vezes. Quanto a taxa de aprendizagem, ela como 0,10 aparece 4 vezes dentre as dez. Já o número de épocas de treinamento como 2000 aparece em 6 das 10 interações, e por fim o número de neurônios como 5 aparece somente apenas 1 vez no melhor resultado, se opondo a 7 vezes que o número como 4 neurônios aparece.

A acurácia entre esses dez resultados variou de 80,50% até 80,05%, ou seja, uma variação de 0,5%. Essa é uma variação grande para os dez primeiros resultados, visto que a variação total entre as 162 interações foi de 3,1%. Com alguns cálculos básicos pode-se chegar a alguns valores: 10 interações correspondem a aproximadamente 6,1% do total de 162 e a variação em questão de 0,5% corresponde a 16% da variação total(3,1%). Logo, com dez interações a acurácia variou um total de 16%. Os dados mencionados estão expostos abaixo na figura 5.

n_foli	l_rate	n_epoc	n_hidd	Tempo(se)	Score	Score	Score	Score	Score	Accura
4,00	0,10	10000	5,00	1623,30	81,28	79,14	84,49	77,10		80,50
5,00	0,30	2000	6,00	488,16	81,87	77,85	79,86	85,90	75,83	80,26
4,00	0,10	10000	4,00	1388,80	80,74	79,67	85,56	74,86		80,21
3,00	0,20	2000	4,00	183,18	79,91	82,32	78,31			80,18
3,00	0,20	2000	4,00	187,21	79,91	82,32	78,31			80,18
3,00	0,20	2000	4,00	190,23	79,91	82,32	78,31			80,18
3,00	0,10	10000	4,00	924,22	79,51	82,30	78,31			80,18
3,00	0,10	10000	6,00	1239,95	79,51	83,13	77,91			80,18
3,00	0,30	2000	4,00	194,20	79,91	81,52	78,71			80,05
3,00	0,30	2000	4,00	191,03	79,91	81,52	78,71			80,05

Figura 5. Dez melhores resultados da 2º base de dados

Fonte: Elaborado pelo autor.

5.2.2. Cinquenta melhores resultados

Dentre os cinquenta primeiros, a acurácia varia de 79,67% até 80,50%, ou seja, uma variação de 0,83%. Em relação ao montante total de interações, a variação que é dada até os 50 primeiros melhores resultados é pequena, já que a partir desse ponto a acurácia decái de uma forma mais acentuada.

Analisando os parâmetros, vale ressaltar as épocas de treinamento, pois dentre os cinquenta melhores resultados, quarenta contém o número de épocas como 2000. Logo, o treinamento com um número menor de épocas nessa base de dados com mais instâncias se demonstrou mais promissor, em oposição à primeira base de dados que apresentou resultados melhores com um número de épocas maior. Em relação aos outros parâmetros, eles foram bem distribuídos dentre as cinquenta melhores interações.

5.2.3. Análise geral - 2º base de dados

Sendo que a base de dados foi montada com o objetivo de demonstrar o marketing RFMTC com os dados de doadores de sangue, o algoritmo nesse sentido atingiu uma acurácia máxima de 80,5%. Assim, ele através da variação de parâmetros demonstrou capacidade de oferecer uma solução para essa questão específica, comprovando sua habilidade para lidar com esse cenário distinto.

Entretanto, no estudo citado na seção 3.1. de [Barhoom 2019], utilizando da mesma base de dados, ele atingiu com seu modelo uma marca de 99,31%. Realizando uma comparação entre esse resultado com o exposto acima, o ruído apresentado é de 18,81%, sendo esse um valor alto de variação. O algoritmo desta pesquisa no contexto da base de dados do Centro de Serviço de Transfusão de Sangue, demonstra-se então ineficaz se comparado a esse estudo.

5.3. Terceira base de dados

Na base de dados em questão, é apresentado números de um estudo a respeito da sobrevivência de pacientes que se submeteram a cirurgia de câncer de mama. Nessa base de dados foram gastos aproximadamente 17 horas em tempo de execução.

O algoritmo apresentou a melhor acurácia dos dados como 75,68%, quando a pior foi de 68,30%. A acurácia média entre todas as 162 interações foi de 72,67%.

5.3.1. Dez melhores resultados

O melhor resultado obtido foi com os seguintes parâmetros: número de *folds* igual a 4, taxa de aprendizado igual a 0,2, número de épocas de treinamento igual 2000 número de neurônios na camada oculta igual a 5; totalizando uma acurácia de 75,68%.

Quantos aos *folds* é notório a presença dele como 4 em seis das dez interações, valendo ressaltar também que o número de *folds* como 5 aparece em quatro das dez

interações. Já a taxa de aprendizado, é observado uma distribuição por igual dentre os dez melhores resultados, tendo a taxa como 0,10 e 0,20 aparecendo cinco vezes cada uma de maneira consecutiva. É possível destacar a presença da taxa como 0,20, taxa essa que na qual está presente na melhor combinação, nos 5 melhores resultados e, conseqüentemente a taxa como 0,10 nos cinco próximos. O número de épocas se sobressai, pois ele como 2000 está presente em 8 dos dez resultados. Já o número de neurônios como 5 está em apenas quatro dos dez, não sendo a maioria, porém, está presente na melhor combinação.

Dentre os dez, a acurácia variou de 75,68% a 74,75%, ou seja, uma variação total de 0,93%, que representa uma diferença grande nos dez primeiros resultados, considerando a variação total entre as 162 interações foi de 7,38%. Por meio de alguns cálculos simples, é possível obter alguns valores: 10 interações correspondem a aproximadamente 6,1% do total de 162 e a variação em questão de 0,93% corresponde a 12,6% da variação total(7,38%). Assim, em dez interações a acurácia variou um total de 12,6%. Os dados estão expostos abaixo na figura 6.

n_folds	l_rate	n_epochs	n_hidden	tempo(s)	Score	Score	Score	Score	Score	Accuracy
4,00	0,20	2000	5,00	127,29	72,36	81,57	68,42	80,26		75,68
4,00	0,20	2000	5,00	127,90	72,36	81,57	68,42	80,26		75,65
4,00	0,20	2000	6,00	144,92	73,68	78,94	68,42	80,26		75,32
4,00	0,20	2000	6,00	143,92	73,68	78,94	68,42	80,26		75,32
5,00	0,20	2000	5,00	166,42	75,40	75,40	77,04	70,49	77,04	75,08
4,00	0,10	10000	6,00	733,25	69,73	80,26	69,73	80,26		75,00
4,00	0,10	10000	6,00	742,79	69,73	80,26	69,73	80,26		75,00
5,00	0,10	2000	4,00	147,82	75,40	72,13	78,68	68,85	78,68	74,75
5,00	0,10	2000	4,00	153,65	75,40	72,13	78,68	68,85	78,68	74,75
5,00	0,10	2000	5,00	171,75	75,40	75,40	75,40	68,85	78,68	74,75

Figura 6. Dez melhores resultados da 3º base de dados

Fonte: Elaborado pelo autor.

5.3.2. Cinquenta melhores resultados

A acurácia teve uma variação total dentre os cinquenta melhores de 2%, variando com o melhor resultado de 75,68% até 73,68%.

Ao examinar os parâmetros, é importante enfatizar as interações das épocas de treinamento, uma vez que, dos cinquenta resultados bem classificados, trinta e seis deles apresentam o valor de 2000. Quanto aos *folds*, ele como 4 está presente em 22 das 50 interações, por outro lado, ele como 5 está em 26 das 50 interações, representando a maioria. A taxa de aprendizagem e o número de neurônios se encontram bem distribuídos ao longo desses resultados.

5.3.3. Análise geral - 3º base de dados

Considerando que essa é uma base de dados fomentada com informações de mulheres que sobreviveram à cirurgia de câncer de mama, a acurácia máxima que o algoritmo conseguiu atingir foi de 75,68%. Devido a natureza dos dados e a situação problema, é necessário uma acurácia maior dos dados para que as informações tenham maior grau de

assertividade na pesquisa. Dessa forma, o algoritmo mesmo com a variação de parâmetros feita de diversas formas não foi capaz de chegar a um resultado aceitável para tal situação.

Além disso, no estudo citado na seção 3.1. de [Shawarib et al. 2020], usando a mesma base de dados, ele chegou a uma precisão de 88,24%. Ao comparar esse resultado com a informação fornecida anteriormente, o ruído apresentado é de 12,56%, sendo esse um valor alto de variação. Portanto, no atual artigo e no contexto da base de dados *Haberman's Survival*, o algoritmo utilizado mostrou-se ineficaz.

6. Conclusão

O objetivo específico inicial do trabalho era analisar o impacto da variação dos parâmetros: número de *folds*, taxa de aprendizagem, número de épocas de treinamento e número de neurônios na camada oculta. Ele foi atendido, pois durante a exibição dos resultados do estudo, foram feitas discussões e análises a respeito do impacto causado por cada parâmetro no resultado final. Além disso, foi exposto para cada base de dados, análises referente a eficiência do algoritmo sendo boa ou ruim.

Já o segundo objetivo específico foi esclarecer se o algoritmo atingiu um resultado aceitável. Esse objetivo foi atingido para a primeira e segunda base de dados, porém para a terceira não. A seção de resultados inclui ao final da análise de cada situação, uma análise geral onde é explicitado essa situação exposta em detalhes.

O terceiro objetivo específico foi efetuar comparações dos resultados obtidos para cada base de dados com o de outros estudos efetuados com elas seguindo a mesma proposta de pesquisa. Essa meta foi atingida apenas para as bases: Centro de Serviço de Transfusão de Sangue e *Haberman's Survival*. Ela foi alcançada devido as análises feitas em cada resultado de cada base de dados.

Nesta pesquisa foi realizado apenas uma adaptação no código, sendo ela: a inserção de um método para calcular o tempo de execução do código. Além disso, as outras alterações foram apenas para ele receber as informações de outras bases de dados. Desse modo, não foi realizado nenhuma adaptação ou especialização no algoritmo para resolver algum problema específico. Sendo assim, o objetivo do trabalho foi analisar e avaliar a robustez do algoritmo e como ele responderia a variações de parâmetros e a bases distintas entre si.

Baseado então nos resultados encontrados, o código se adaptou bem a primeira base de dados, atingindo uma acurácia de 96,19%. Ele também se adaptou bem a base de dados do Centro de Serviço de Transfusão de Sangue, pois essa revelou uma acurácia de 80,50%. E por fim a base de dados *Haberman's Survival* apresentou o pior resultado com uma acurácia de 75,68%.

Em se tratando dos trabalhos relacionados às bases de dados, a base do centro de transfusão de sangue foi a que apresentou a pior resposta em comparação com o artigo mencionado, pois na comparação o ruído foi de 18,81%. Por outro lado, a base *Haberman's Survival* revelou um ruído menor de 12,56%, mas ainda sendo um valor ruim. Logo, o algoritmo apresentou bons resultados em dois dos três casos, porém ele não se mostrou efetivo quando comparado aos outros da literatura analisada.

Devido às limitações de tempo e da máquina usada só foi possível utilizar baixos números de épocas de treinamento, percebe-se que nos outros casos relacionados apre-

sentados esses números foram maiores. Logo, em se tratando do presente artigo, poder-se-á ter melhores resultados ao aplicar o algoritmo e cada base de dados nas mesmas circunstâncias de ciclos de treinamento.

Como trabalhos futuros, a metodologia desse estudo aplicado a contextos desse tipo pode ser expandida aumentando o conjunto de teste usado nos parâmetros. Alguns exemplos são: elevar o número de épocas de treinamento, trabalhar também com taxas de aprendizagem entre 0,3 e 0,5 e trabalhar com duas camadas ocultas na arquitetura da RNA.

References

- Alex, S. (2019a). O que é uma rede neural artificial.
- Alex, S. (2019b). Validação cruzada conceito e exemplo.
- Amancio, L. B. (2013). Previsão de recalques em fundações profundas utilizando redes neurais artificiais do tipo perceptron.
- Araújo, C. B. C. d. (2015). Aplicação das redes neurais artificiais do tipo perceptron na estimativa de recalques em estacas.
- Barhoom, A. M. (2019). Blood donation prediction using artificial neural network.
- Barros, V. P. A. et al. (2018). Avaliação do desempenho de algoritmos de retropropagação com redes neurais artificiais para a resolução de problemas não-lineares. Master's thesis, Universidade Tecnológica Federal do Paraná.
- Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514.
- Charytanowicz, Magorzata, N. J. K.-P. K. P. and Lukasik, S. (2012). seeds. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5H30K>.
- de Pádua Braga, A., de Leon Ferreira, A. C. P., and Ludermir, T. B. (2007). *Redes neurais artificiais: teoria e aplicações*. LTC Editora.
- Dua, D. and Graff, C. (2019). Uci machine learning repository. university of california, school of information and computer science, irvine, ca (2019).
- Fleck, L., Tavares, M. H. F., Eyng, E., Helmann, A. C., and Andrade, M. A. d. M. (2016). Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, 1(13):47–57.
- Furtado, M. I. V. (2019). Redes neurais artificiais: uma abordagem para sala de aula. *Ponta Grossa, PR. Atena Editora*, page 19.
- GADELHA, J. (2009). A evolução dos computadores.
- Guimarães, A. M., Mathias, I. M., Dias, A. H., Ferrari, J. W., and Junior, C. R. d. O. C. (2008). Módulo de validação cruzada para treinamento de redes neurais artificiais com algoritmos backpropagation e resilient propagation. *Publicatio UEPG: Ciências Exatas e da Terra, Agrárias e Engenharias-ATIVIDADES ENCERRADAS*, 14(01).
- Haberman, S. (1999). Haberman's Survival. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XK51>.

- Haugeland, J. (1989). *Artificial intelligence: The very idea*. MIT press.
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- Haykin, S. (2001). *Redes neurais: princípios e prática*. Bookman Editora.
- Isa, N. A. M. and Mamat, W. M. F. W. (2011). Clustered-hybrid multilayer perceptron network for pattern recognition application. *Applied Soft Computing*, 11(1):1457–1466.
- Junior, R. L. S. (2012). Análise sobre o desenvolvimento do campo de estudo em informação científica e técnica nos estados unidos e na antiga união soviética durante a guerra fria (1945-1991). *Revista Brasileira de Biblioteconomia e Documentação*, 8(2):130–157.
- Kim, D. (2005). Improving prediction performance of neural networks in pattern classification. *International Journal of Computer Mathematics*, 82(4):391–399.
- Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., Steinbrecher, M., Klawonn, F., and Moewes, C. (2013). *Computational intelligence*. Springer.
- Kurzweil, R., Richter, R., Kurzweil, R., and Schneider, M. L. (1990). *The age of intelligent machines*, volume 580. MIT press Cambridge.
- LeCun, Y. (1993). Efficient learning and second-order methods. *A tutorial at NIPS*, 93:61.
- Malinverni, M. d. S. et al. (2006). Sistema hipermídia sobre câncer de colo de útero com interface adaptativa usando redes neurais artificiais mlp e sistema especialista.
- Manzan, J. R. G. et al. (2016). Análise de desempenho de redes neurais artificiais do tipo multilayer perceptron por meio do distanciamento dos pontos do espaço de saída.
- McCarthy, J. (2007). What is artificial intelligence?
- Nunes, T. V. L. (2003). Método de previsão de defeitos em estradas vicinais de terra com base no uso das redes neurais artificiais: Trecho de aquiraz-ce.
- Pelli, E. (2013). Comparação dos algoritmos máquina de aprendizagem extrema e retropropagação do erro para estimação de altura e volume de árvores.
- Raia Junior, A. A. (2000). *Acessibilidade e mobilidade na estimativa de um índice de potencial de viagens utilizando redes neurais artificiais e sistemas de informações geográficas*. PhD thesis, Universidade de São Paulo.
- Santos, A. M. d., Seixas, J. M. d., Pereira, B. d. B., and Medronho, R. d. A. (2005). Usando redes neurais artificiais e regressão logística na predição da hepatite a. *Revista Brasileira de Epidemiologia*, 8(2):117–126.
- Shawarib, M. Z. A., Latif, A. E. A., Al-Zatmah, B. E. E.-D., and Abu-Naser, S. S. (2020). Breast cancer diagnosis and survival prediction using jnn.
- Silva, I. N. d., Spatti, D. H., and Flauzino, R. A. (2010). Redes neurais artificiais para engenharia e ciências aplicadas.
- Souza, K. and Ramos, F. L. L. (2021). Análise da variação de parâmetros em uma rede neural convolucional para reconhecimento de dígitos escritos à mão. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 13(3).

Yeh, I.-C. (2008). Blood Transfusion Service Center. UCI Machine Learning Repository.
DOI: <https://doi.org/10.24432/C5GS39>.