



UFOP

Universidade Federal
de Ouro Preto

Universidade Federal de Ouro Preto - UFOP

Escola de Minas - EM

Departamento de Engenharia de Controle e Automação - DECAT



**PREVISÃO DA QUANTIDADE DE REFEIÇÕES DE UM RESTAURANTE
UNIVERSITÁRIO VIA REDES NEURAS ARTIFICIAIS**

Gabriel Silveira Camba

**Ouro Preto
Minas Gerais - Brasil
2023**



Universidade Federal de Ouro Preto - UFOP

Escola de Minas - EM

Departamento de Engenharia de Controle e Automação - DECAT



Gabriel Silveira Camba

PREVISÃO DA QUANTIDADE DE REFEIÇÕES DE UM RESTAURANTE UNIVERSITÁRIO VIA REDES NEURAIAS ARTIFICIAIS

Monografia apresentada à Escola de Minas da Universidade Federal de Ouro Preto, como requisito parcial para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Agnaldo José da Rocha Reis, Dr.

Ouro Preto
Minas Gerais - Brasil
2023



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Gabriel Silveira Camba

PREVISÃO DA QUANTIDADE DE REFEIÇÕES DE UM RESTAURANTE UNIVERSITÁRIO VIA REDES NEURAIAS ARTIFICIAIS

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 31 de março de 2023

Membros da banca

Dr. Agnaldo José da Rocha Reis - Orientador (Universidade Federal de Ouro Preto)
Dra. Adrielle de Carvalho Santana - Examinadora (Universidade Federal de Ouro Preto)
Dr. Alan Kardek Rêgo Segundo - Examinador (Universidade Federal de Ouro Preto)

Agnaldo José da Rocha Reis, orientador do trabalho, aprovou a versão final e autorizou o seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 26/04/2023.



Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis, PROFESSOR DE MAGISTERIO SUPERIOR**, em 26/04/2023, às 13:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0515249** e o código CRC **68BA8255**.

AGRADECIMENTOS

A UFOP pelo ensino de qualidade e o amparo na pesquisa.

A Escola de Minas e seus professores que me proporcionaram os conhecimentos para chegar até aqui.

Ao meu orientador Dr. Agnaldo, pelos ensinamentos durante o curso e pelo grande auxílio neste trabalho.

A minha família que me deu todo apoio e suporte para concluir mais esta etapa na minha vida. Em especial aos meus pais Domingos e Iris que sempre me deram todo amor e amparo. Aos meus irmãos Natália e João que sempre me incentivaram a continuar firme na minha caminhada. Ao meu padrinho Miguel por ter me encorajado a seguir meus sonhos. E a minha sobrinha Estela que me deu motivação para buscar um mundo melhor para ela. A Raquel, que se manteve ao meu lado me apoiando.

E a todos que de alguma forma me ajudaram e estiveram comigo durante toda minha caminhada

RESUMO

Os Restaurantes Universitários (RU) são Unidades de Alimentação e Nutrição (UAN) presentes na maioria dos Institutos Federais de Ensino Superior (IFES), tendo como intuito oferecer uma opção nutritiva e acessível para a comunidade acadêmica. Para suprir essa demanda com eficiência, ou seja, evitando falta de refeições e/ou desperdício, uma estimativa do número de refeições de elevada acurácia se faz necessária. O modelo atualmente empregado no estudo aqui apresentado vale-se de métodos estatísticos lineares para a realização da previsão do número diário de refeições a serem preparadas. A fim de se investigar a possibilidade de melhorar a qualidade dessa previsão em um restaurante universitário, foi realizada uma análise a partir do uso de inteligência artificial. Nesse sentido, as Redes Neurais Artificiais (RNA) se mostram como uma poderosa ferramenta capaz de extrair conhecimento a partir do treinamento em uma base de dados. Desse modo, investigou-se neste trabalho a sua aplicabilidade, comparando o seu desempenho àquele obtido com o modelo linear atualmente empregado. O modelo apresentou um R^2 de 62%, um Erro Médio Absoluto de 162,9 refeições e uma Raíz Quadrada do Erro Médio absoluto de 224,4. Em comparação com o modelo atual, este teve um erro médio 1% maior. Apesar de um desempenho aquém do modelo atual, a utilização de RNA para predição do número de refeições diárias do restaurante possui um potencial de apresentar melhores resultados e mais consistentes, caso uma base de dados maior esteja disponível.

Palavras-chave: Restaurante Universitário; Redes Neurais Artificiais; Inteligência Artificial; Aprendizado de Máquina; Previsão do número de Refeições.

ABSTRACT

University Restaurants (UR) are Food and Nutrition Units (FNU) present in most Federal Institutes (FI), in order to offer a nutritious and accessible option for the academic community. To meet this demand efficiently, that is, avoiding lack of meals and/or waste, a highly accurate estimate is necessary. The model currently used in the study presented here uses linear statistical methods to predict the daily number of meals to be prepared. To investigate whether it can improve the quality of this prediction in a university restaurant, an analysis using artificial intelligence was performed. In this sense, Artificial Neural Networks (ANN) are shown to be a powerful tool capable of extracting knowledge from training in a database. Thus, in this work its applicability was investigated and its results were compared to that obtained with the linear model currently in use.

Keywords: University Restaurant; Neural Network; Artificial Intelligence; Machine Learning; Number of Forecasted Meal.

LISTA DE FIGURAS

Figura 1: Esquema representativo de um neurônio típico.....	5
Figura 2: Modelo de neurônio artificial	5
Figura 3: Gráfico de uma função limiar	6
Figura 4: Gráfico de uma função sigmoide.....	7
Figura 5: Gráfico de uma função ReLU	8
Figura 6: Modelo esquemático do perceptron de Rosenblatts	9
Figura 7: Estrutura básica de um MLP.....	10
Figura 8: Gráfico representativo de uma função de perda ideal.....	12
Figura 9: Pipeline metodológico	15
Figura 10: Curva do número de refeições de um período letivo.....	18
Figura 11: Média de refeições por dia da semana.....	21
Figura 12: Média de refeições por semana.....	21
Figura 13: Esquema representativo da arquitetura final da rede.....	24
Figura 14: Curva da comparação entre os dados reais e previstos pelo RNA	25
Figura 15: Curva da função de perda obtida	25
Figura 16: Gráfico do erro causado pela previsão.....	26

LISTA DE TABELAS

Tabela 1: Hiper parâmetros utilizados nos testes	22
Tabela 2: Neurônios e camadas ocultas nos testes realizados	23
Tabela 3: Hiper parâmetros utilizados na execução final.....	24
Tabela 4: Distribuição do erro cometido pelo modelo proposto	27
Tabela 5: Distribuição do erro cometido pelo modelo atual	27

LISTA DE ABREVIATURAS

GPU	<i>Graphics Processing Unit</i> (Unidade de Processamentos Gráficos)
IA	Inteligência Artificial
IFES	Institutos Federais de Ensino Superior
MAE	<i>Mean Absolut Error</i> (Erro Médio Absoluto)
R²	Coefficiente de Determinação
ReLU	<i>Rectified Linear Unit</i>
RMSE	<i>Root Mean Square Error</i> (Raiz Quadrada do Erro Médio)
RNA	Redes Neurais Artificiais
RU	Restaurante Universitário
UAN	Unidade de Alimentação e Nutrição
UFOP	Universidade Federal de Ouro Preto
UFV	Universidade Federal de Viçosa

SUMÁRIO

2	INTRODUÇÃO	1
1.1	Objetivos gerais e específicos	2
1.2	Estrutura do trabalho	3
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	História das Redes Neurais Artificiais	4
2.2	Inspiração no neurônio biológico	4
2.3	Neurônio artificial	5
2.3.1	Função limiar	6
2.3.2	Sigmoide	7
2.3.3	ReLU	7
2.4	Tipos de treinamento de uma Rede Neural	8
2.4.1	Supervisionado	8
2.4.2	Não Supervisionado	8
2.4.3	Aprendizagem por Reforço	8
2.5	Perceptrons	9
2.6	Perceptrons Multicamadas	9
2.7	Hiperparâmetros	10
2.8	Avaliação do modelo	12
2.9	Trabalhos relacionados	13
3	METODOLOGIA	15
3.3	Descrição dos Métodos	15
3.4	Ferramentas utilizadas	16
3.4.1	Linguagem de Programação	16
3.4.2	Google Colaboratory	16
3.4.3	PyTorch	17
3.5	Coleta de dados	17
3.6	Exploração e transformação dos dados	18
3.7	Arquitetura da rede	19
3.8	Métricas de Avaliação	19
4	RESULTADOS E DISCUSSÃO	20
4.1	Análise dos dados	20
4.2	Desempenho do modelo	25
4.3	Comparação com o modelo atual	27
5	CONCLUSÃO	28
	REFERÊNCIAS	29

2 INTRODUÇÃO

A entrada em uma universidade pública representa um desafio não só intelectual, como também uma mudança, muitas vezes, drástica no estilo de vida dos jovens. De acordo com Alves e Boog (2007), os comportamentos alimentares são expressivamente influenciados por conta dessa mudança na rotina.

Seja pela nova rotina intensa de estudos ou uma abertura de um ciclo social diferente do vivido anteriormente, muitos jovens optam por alimentos menos saudáveis ou por ‘pular’ uma refeição. Essas opções alimentares coincidem com a diminuição do consumo de frutas e hortaliças e com a ampliação do consumo de carboidratos simples e lipídeos (MONTEIRO et al., 2009)

Além disso, grande parte dos discentes que se encontram nas Instituições Federais de Ensino Superior (IFES) é de baixa renda. A propósito, como mostra a V Pesquisa Nacional de Perfil Socioeconômico e Cultural dos graduandos das IFES (ANDIFES, 2018), 70,2% dos graduandos possuem renda mensal familiar per capita de até 1,5 salário mínimo, limitando as opções para uma alimentação de qualidade para este grupo, fazendo com que estas instituições tenham como razão não só a questão educacional como também a social.

Os hábitos alimentares têm uma relação importante com o desempenho acadêmico. O desequilíbrio nutricional além de causar alterações na saúde global interfere na aprendizagem, memória e tem efeitos no comportamento do indivíduo (YEHUDA et al., 2006). ‘Saltar’ refeições ou comer em horários irregulares também pode afetar negativamente o desempenho acadêmico, pois pode causar fadiga, dificuldade de concentração e dificuldade de aprendizado.

Um Restaurante Universitário (RU) funciona como uma Unidade de Alimentação e Nutrição (UAN) dentro das Universidades Federais. Segundo Lanzillotti et al. (2004), UANs são espaços voltados para preparação e fornecimento de refeições equilibradas em nutrientes, segundo o perfil da clientela. Nesse cenário, essas unidades visam atender à toda comunidade universitária, abrangendo docentes, discentes e técnico-administrativos, com a finalidade de oferecer uma alimentação balanceada e nutritiva, possibilitando assim a segurança alimentar dessa comunidade.

Os RUs são administrados por empresas terceirizadas e subsidiadas pelas IFES de forma a disponibilizar as refeições a um custo mais acessível, possibilitando abranger uma gama maior de pertencentes dessa comunidade. Servindo almoço e jantar durante os dias letivos, essas UANs acabam se tornando a fonte principal de alimentação de muitos

estudantes, por permitirem que pessoas de baixa renda tenham uma boa alimentação a um baixo custo (ou até mesmo nenhum custo em caso de estudantes contemplados com bolsas de permanência) ou por alunos que não possuem tempo ou disponibilidade para produzir a própria refeição em suas casas.

Um dos problemas enfrentados pela administração dos RUs se dá na estimação do número de refeições diárias necessárias. A subestimação pode acarretar longas filas e demora no atendimento dessa demanda, já que precisam produzir mais refeições que o planejado para o dia. Isto desincentiva ou até impossibilita pessoas que necessitam dessa refeição em sua rotina de usufruir desse serviço. Por outro lado, a superestimação da quantidade de alimentos necessária pode gerar desperdício, sendo um problema grave, ainda mais em se tratando de um país no qual muitas pessoas estão em condições de insegurança alimentar. Além do mais, o gasto desnecessário do dinheiro público, gerado pelo desperdício, poderia estar sendo investido de uma outra maneira dentro da instituição.

Tendo em vista os problemas citados anteriormente, a busca por uma melhor alternativa para previsão da quantidade de frequentadores dos Restaurantes Universitários se torna um tópico de grande relevância. Prever com maior acurácia a quantidade diária de refeições possibilita atender a toda comunidade acadêmica que necessita desse serviço de forma otimizada.

Dentre as diversas formas de se prever um valor baseado em dados passados, as Redes Neurais Artificiais (RNA) se apresentam como uma ferramenta muito poderosa. O uso de Redes Neurais em previsão tem se mostrado útil, principalmente quando a série de dados usada tem comportamento não linear (BRANCO; SAMPAIO, 2008).

1.1 Objetivos gerais e específicos

O objetivo geral deste trabalho é realizar a previsão da quantidade de refeições do RU da Universidade Federal de Ouro Preto (UFOP) via RNA e avaliar a sua aplicabilidade.

Além disso, como objetivos específicos, destacam-se:

- Contextualizar a realidade de um RU de modo a entender melhor o problema;
- Descrever a base de dados e estudar os métodos que podem ser empregados no seu pré-processamento;
- Buscar RNAs que possam ser usadas para o problema proposto;
- Avaliar os resultados e a aplicabilidade dos métodos testados para previsão de refeições diárias no RU;

1.2 Estrutura do trabalho

O presente trabalho está estruturado em 5 capítulos. O Capítulo 2 apresenta uma revisão da literatura a respeito da história das Redes Neurais Artificiais, sua inspiração da biologia e a descrição dos componentes presentes na arquitetura da rede. O Capítulo 3 apresenta a descrição das ferramentas utilizadas para o desenvolvimento do algoritmo e a metodologia utilizada para se obter os resultados. No Capítulo 4 são apresentados os resultados obtidos e a comparação com o modelo de previsão atualmente empregado. Por fim, as principais conclusões deste estudo e uma sugestão para trabalho futuro são apresentadas no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 História das Redes Neurais Artificiais

RNA são um modelo que utiliza algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede de uma forma ordenada para alcançar um objetivo de projeto desejado (HAYKIN, 2001). Esta é uma das diversas ferramentas no campo de Inteligência Artificial, sendo muito útil para solução de diversos problemas como reconhecimento de objetos, reconhecimento da fala, diagnósticos médicos, entre outros (ALOM et al., 2018).

A inspiração desse modelo vem do próprio cérebro humano e da forma como funciona. Segundo Hebb (1949), as bases fisiológicas para a aprendizagem se dão pela excitação de conjuntos de células, que é a manifestação ativa do processamento cognitivo natural.

Os primeiros trabalhos a respeito deste assunto foram propostos por Mcculloch e Pitts (1943), em que consideraram os neurônios como elementos lógicos com saídas binárias, de forma com que todos os elementos eram atualizados de forma síncrona, em tempos discretos e regularmente espaçados (CLARK, 1991).

2.2 Inspiração no neurônio biológico

O cérebro humano é composto por uma série de redes neurais. Segundo Moreira (2013), neurônios são células nervosas altamente estimuláveis, que processam e transmitem informações através de sinais eletroquímicos. A maioria dos neurônios, tipicamente, possui o corpo celular e dois tipos de prolongamentos citoplasmáticos, os dendritos e os axônios.

Vê-se na Figura 1 um esquema típico de um neurônio humano. Os dendritos (A), recebem como entrada os impulsos nervosos. O corpo celular (B), é onde está localizado o núcleo e grande parte das organelas. O axônio (C) é o prolongamento que faz a condução do impulso nervoso, sendo geralmente mais longo que o dendrito.

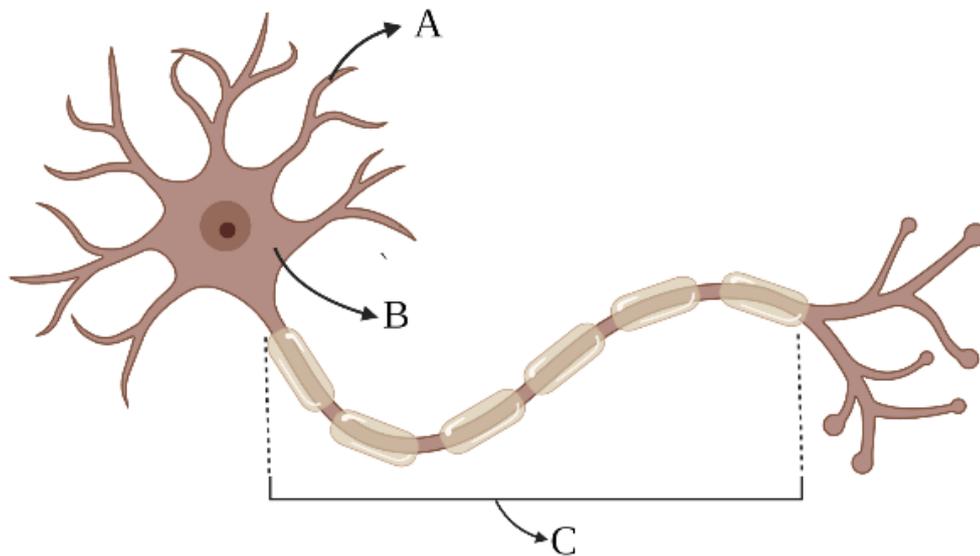


Figura 1: Esquema representativo de um neurônio típico
 Fonte: Elaborado pelo autor. Criado com BioRender

2.3 Neurônio artificial

Nied et. al (2007) descrevem, de forma geral, um neurônio artificial como um conjunto de n entradas ao qual cada entrada é multiplicada por um determinado peso e, em seguida, os resultados são somados e comparados a um limiar. Vê-se na Figura 2 o neurônio artificial proposto por McCulloch e Pitts (1943):

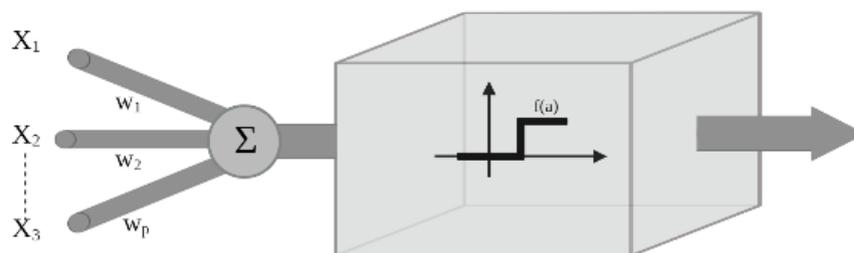


Figura 2: Modelo de neurônio artificial
 Fonte: Adaptado de McCulloch; Pitts, (1943). Criado com BioRender

Normalmente o tipo de processamento de um único neurônio é a combinação linear das entradas com os respectivos pesos, seguida pela passagem da combinação linear por uma

função de ativação (RAUBER, 2023). Sendo assim, o modelo pode ser descrito matematicamente pela seguinte equação:

$$y = FA \left(\sum_{j=1}^n (w_n x_n) + b \right) \quad (1)$$

Haykin (2001) define a função de ativação como uma função que restringe o intervalo permissível de amplitude do sinal de saída a um valor finito, sendo este, unitário fechado $[0,1]$ ou $[-1, 1]$. Ao passar pela função de ativação, esta pode aplicar uma transformação não-linear nos dados de entrada, o que possibilita o modelo aprender e executar tarefas mais complexas (DATA SCIENCE ACADEMY, 2022). Existem diversas opções de função de ativação, sendo ainda uma área de pesquisa ativa no campo de Inteligência Artificial. Algumas das principais funções de ativação presentes na literatura são: a limiar, a sigmoide e a ReLu.

2.3.1 Função limiar

A função limiar (Figura 3) refere-se ao modelo McCulloch-Pitts, no qual a saída recebe valor 0 caso o resultado do produto interno das entradas pelos respectivos pesos mais os bias seja menor que um limite determinado e 1 caso seja maior. Ela é dada pela seguinte equação:

$$f(u) = \begin{cases} 1 & \text{se } u \geq a \\ 0 & \text{se } u < a \end{cases} \quad (2)$$

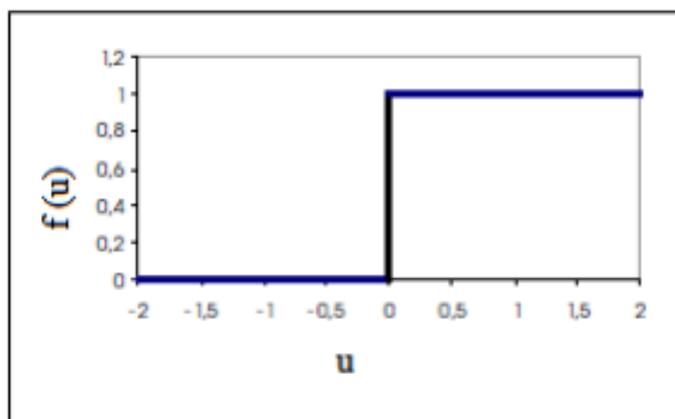


Figura 3: Gráfico de uma função limiar
Fonte: Fleck et al. (2016)

2.3.2 Sigmoide

Segundo Fleck et al. (2016), a função Sigmoide (Figura 4) pode ser definida como uma função crescente, podendo ter um comportamento linear ou não linear, assumindo valores em um intervalo entre 0 e 1. A inclinação da função é determinada pelo parâmetro a , sendo mais inclinada à medida que o valor de a aumenta. Ela é dada pela seguinte equação:

$$f(\mu) = \frac{1}{1+e^{-a\mu}} \quad (3)$$

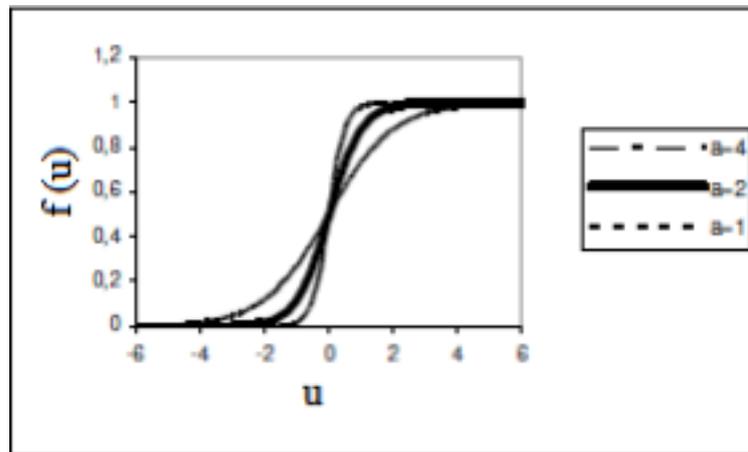


Figura 4: Gráfico de uma função sigmoide
Fonte: Fleck et al. (2016)

2.3.3 ReLU

Segundo Rasamoelina et. al (2020), *Rectified Linear Unit* (ReLU) é a função de ativação mais utilizada, tendo uma performance de treinamento superior em relação às outras funções. Como mostrado na Figura 5, para valores negativos a saída recebe valor zero, e para valores positivos, retorna um valor x maior que zero.

Sua principal vantagem é não ativar o neurônio caso a entrada seja negativa, dessa forma, apenas alguns neurônios são ativados, tornando a rede esparsa, eficiente e fácil para a computação (DATA SCIENCE ACADEMY, 2022). Ela é dada pela seguinte equação:

$$f(x) = \max(0, x) \quad (4)$$

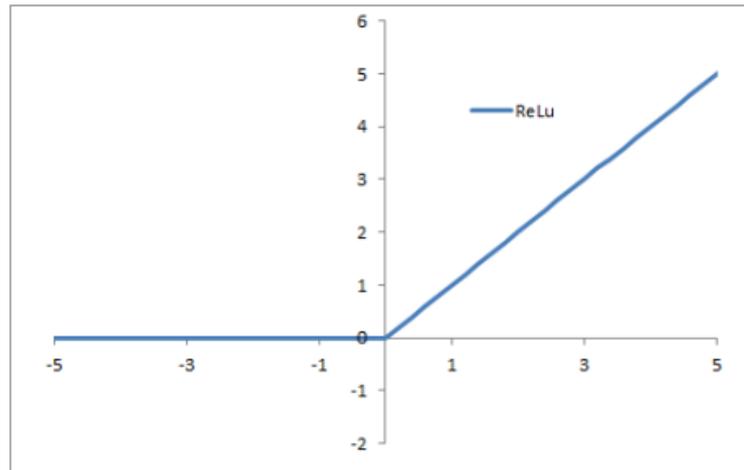


Figura 5: Gráfico de uma função ReLU
Fonte: Farias (2018)

2.4 Tipos de treinamento de uma Rede Neural

Gupta et. al. (2020) definem o processo de aprendizado no contexto de RNA como o problema de atualizar os pesos das conexões de uma rede neural para que ela possa executar uma tarefa específica com eficiência.

2.4.1 Supervisionado

Gallant (1990) descreve que no aprendizado supervisionado tanto as amostras da base de treino (entradas) quanto a variável alvo (saídas) são apresentados para o algoritmo, que deve então mapear as entradas dadas para as saídas desejadas.

2.4.2 Não Supervisionado

No aprendizado não supervisionado o modelo não conhece a resposta para cada amostra, sendo assim, o que o este pode fazer é agrupar os dados, assumindo similaridade de características entre eles (GALLANT, 1990).

2.4.3 Aprendizagem por Reforço

Sutton e Barto (2018) descrevem o aprendizado por reforço como um aprendizado no qual o modelo precisa aprender o que fazer para maximizar um sinal numérico de recompensa, sendo que este não recebe nenhum tipo de ação a ser tomada, precisando descobrir como maximizar a recompensa através de tentativas. Os maiores desafios no aprendizado por reforço é definir como serão atribuídas as recompensas para cada ação

(SCHMIDHUBER, 2015). Géron (2022) pontua que este tipo de aprendizado é muito utilizado em jogos e controle de máquinas.

2.5 Perceptrons

O Perceptron é o algoritmo mais simples de uma Rede Neural Artificial. Frank Rosenblatt no final da década de 1950 propôs o modelo Perceptron de uma camada, ou seja, composto por um conjunto de entradas, um processador e uma saída (SHIFFMAN; FRY; MARSH, 2012).

Coppin (2010) descreve o funcionamento do algoritmo Perceptron da seguinte forma:

1. Atribuição de pesos aleatórios para as entradas;
2. Dados de treinamento são inseridos e processados;
3. Verificam-se se os resultados condizem com o erro esperado;
4. Caso a saída não satisfaça a condição, os pesos são reajustados de acordo com a taxa de aprendizado.

a taxa de aprendizado.

O processo é realizado de forma contínua, sendo realizado até satisfazer as condições das saídas. A Figura 6 representa um modelo Perceptron.

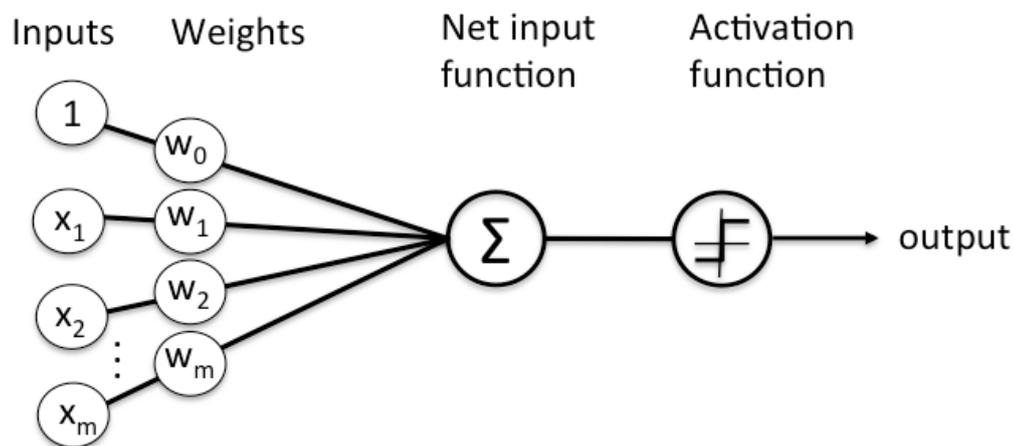


Figura 6: Modelo esquemático do perceptron de Rosenblatts
Fonte: Raschka (2015)

2.6 Perceptrons Multicamadas

O Perceptron Multicamada é o mais conhecido tipo de rede neural (POPESCU et al., 2009). Noriega (2005) pontua que o principal ponto negativo do perceptron é que este só pode solucionar problemas que são linearmente separáveis, não sendo possível resolver grande parte dos problemas mais desafiadores na Inteligência Artificial. Dessa forma, a implantação

de múltiplas camadas torna a rede mais poderosa e abrangente na questão de resolução de problemas. Um exemplo de uma rede neural multicamada é representado pela Figura 7.

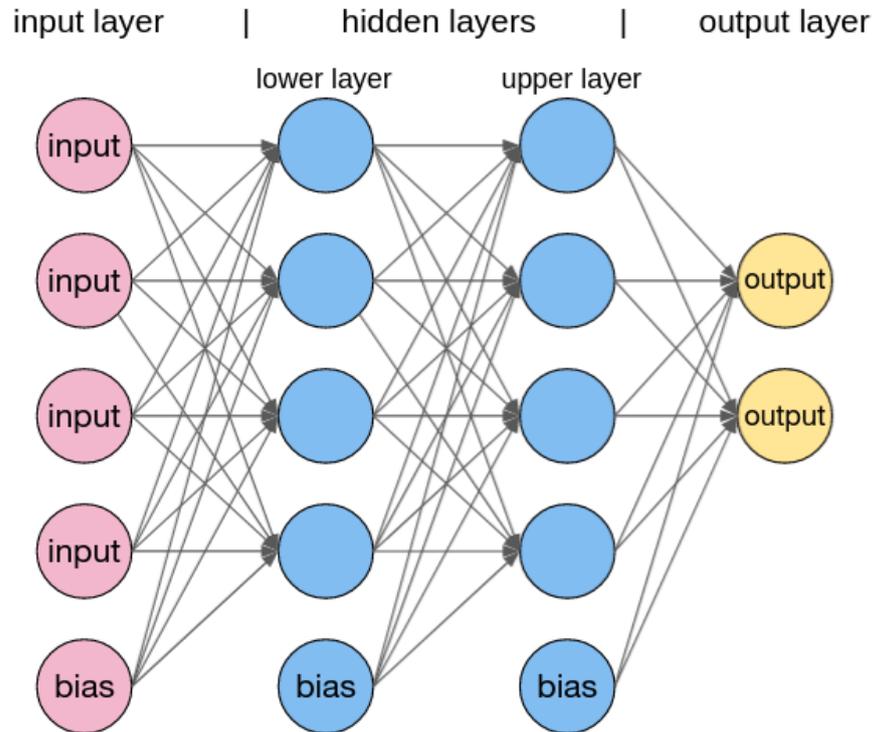


Figura 7: Estrutura básica de um MLP
Fonte: AN (2021)

No treinamento da rede multicamada, se dá utilizado o algoritmos de retropropagação, que consiste em dois passos: o processamento direto e processamento reverso.

“No processamento direto, uma entrada é aplicada à rede neural e seu efeito é propagado pela rede, camada a camada. Durante o processamento direto, os pesos da rede permanecem fixos”. Iyoda (2000). Já no processamento reverso, é computado um sinal de erro na saída da rede que é propagado no sentido contrário e ao final deste processo os pesos são ajustados de acordo com uma regra de correção de erro.

Os pesos são modificados utilizando um processo chamado gradiente descendente. A descida do gradiente é um algoritmo de otimização usado para encontrar os pesos que minimizam a função de perda (KALIRANE, 2023).

2.7 Hiperparâmetros

Para o bom desempenho de um modelo, além da escolha da quantidade de camadas ocultas e seus respectivos neurônios, é importante também a definição de seus parâmetros. Como pontua Karpathy (2016), estes parâmetros ditam a desempenho do modelo, e para o seu

melhor desempenho, ajustes pontuais e manuais devem ser feitos. Taylor (2017) complementa a ideia, pontuando que diferentemente do que é feito com os pesos e bias (ou vieses), os hiperparâmetros não podem ser ajustados automaticamente. Porém, na literatura existem algumas recomendações que podem guiar os estudos para a definição dos hiper parâmetros iniciais.

A taxa de aprendizagem, como define Taylor (2017), é o valor que determina o quão rápido ou devagar o algoritmo aprende. O ajuste desse parâmetro pode impactar na busca pelo mínimo global. Ainda segundo o autor, um valor muito alto pode fazer com que o modelo não consiga encontrar um valor satisfatório, uma vez que o tamanho do passo da descida do gradiente, inviabilize-o de encontrar um valor mínimo na função de perda, e acabe divergindo. Já um valor muito pequeno pode exigir muito tempo para convergir.

A regularização L2, também conhecida como *weight decay*, é o custo adicionado ao valor dos coeficientes dos pesos. Este parâmetro é utilizado para reduzir o sobreajuste do modelo no treinamento, inserindo restrições na complexidade da rede e fazendo com que os pesos assumam valores pequenos, tornando assim a distribuição de valores dos pesos mais regular (CHOLLET, 2021).

O parâmetro de tamanho do lote, ou *batch size*, é o que define a quantidade de amostras que serão vistas pelo modelo antes da atualização dos parâmetros internos (BROWNLEE, 2021). Existem três tipos de abordagens para o tamanho do lote, sendo descritas a seguir:

- *Batch Gradient Descent*: Quando o tamanho do lote é igual à quantidade de amostras na base de treino;
- *Stochastic Gradient Descent*: Batch Size: Quando o tamanho do lote é igual a 1 amostra;
- *Mini-Batch Gradient Descent*: Quando o tamanho do lote é maior que uma amostra e menor que o tamanho da base de treino;

Cada vez que o modelo interage com um lote, é considerada uma iteração. Quando todas as amostras da base de treino passaram pelo treinamento, é contabilizada uma época.

A função de perda é um gráfico utilizado para indicar o quão próximo o modelo está do seu objetivo. Para isso, é necessário especificar qual métrica o algoritmo deve priorizar a melhora, podendo ser, por exemplo, o Erro Médio Absoluto, a Raiz Quadrada do Erro Médio, dentre outros. Chollet (2021) destaca que a escolha da função de perda é extremamente importante, pois o modelo irá fazer de tudo para minimizar esta perda. Logo, caso a métrica

não esteja correlacionada com o objetivo final, o resultado pode acabar não sendo o esperado. A Figura 8 exemplifica o comportamento ideal de uma função de perda. O modelo inicia com um erro muito alto, e à medida que vão se passando as épocas, considerando que o modelo esteja aprendendo, o valor decai até se estabilizar em uma faixa que pode ser um mínimo local ou, idealmente, o mínimo global.

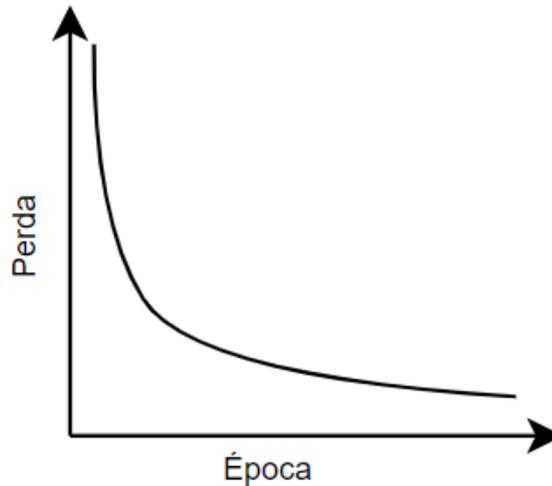


Figura 8: Gráfico representativo de uma função de perda ideal
Fonte: Adaptado Google (2022)

2.8 Avaliação do modelo

Para avaliar o desempenho do modelo, é importante utilizar métricas diferentes, pois cada uma traz informações a respeito de como o modelo se comportou. Dentre as diversas métricas, pode-se citar como as mais utilizadas as seguintes: Coeficiente de Determinação (R^2), Raiz Quadrada do Erro Médio (RMSE) e Erro Médio Absoluto (MAE).

A métrica R^2 (Eq. (5)) é uma medida que indica o quanto da variável dependente é explicada pela variável independente (HAMILTON; GHERT; SIMPSON, 2015). Seu resultado é um valor no intervalo $[0, 1]$, no qual 1 significa que o modelo consegue explicar 100% da variância dos dados. Quanto menor este indicador, menor será a porção dos dados que o modelo é capaz de explicar.

$$R^2 = \frac{\sum_{j=1}^n (P_j - A_j)^2}{\sum_{j=1}^n (A_j - \bar{A})^2} \quad (5)$$

Outras duas métricas muito utilizadas para avaliar modelos de regressão são o RMSE e o MAE. Estas duas métricas estão quase sempre juntas, pois trazem informações complementares a respeito do desempenho do modelo. O Erro Médio Absoluto, mais conhecido como MAE (Mean Absolut Error) (Eq. (6)), retorna a média da diferença entre o valor predito e o real.

$$MAE = \frac{1}{n} \sum_{j=1}^n |e_j| \quad (6)$$

Já o RMSE (Eq. (7)) consiste na raiz quadrada média do erro.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (e_j)^2} \quad (7)$$

Sendo assim, algumas particularidades podem ser tiradas na observação das duas métricas. As duas retornam valores de $[0, \infty]$, sendo este quanto menor, melhor. Porém, os resultados apresentam magnitudes diferentes. Enquanto o MAE traz valores de mesma magnitude de erro, o RMSE penaliza mais os erros elevados, fazendo com que esses erros elevados causem um desempenho muito pior para o modelo na avaliação com essa métrica. Como pontua Willmott e Matsuura (2005), avaliando o RMSE de forma isolada, não é possível dizer o quanto esta métrica reflete o erro médio. Portanto, constata-se que essas duas métricas são complementares.

2.9 Trabalhos relacionados

Nesta parte do capítulo, apresentam-se trabalhos da literatura, em que foram utilizados Redes Neurais Artificiais no processo de predição da quantidade de refeições necessárias por dia em um Restaurante Universitário. Desta forma, estes artigos serviram como base para o presente trabalho.

Em Rocha et al. (2011) os autores utilizaram uma Rede Neural Multicamadas para prever a quantidade de frequentadores diários do Restaurante Universitário alocado na FCL/Assis/Unesp. Foram coletados dados de 254 dias, sendo esta base de dados fornecida pela administradora do local, contando com as seguintes variáveis de entrada: “Nível de

aceitação das refeições”, “Dia da semana”, “Mês”, “Média dos últimos 5 dias”, “Média dos últimos 30 dias”, “Número de refeições do dia anterior”, “Se será feriado no dia anterior” e “Se será feriado no dia posterior”. A rede neural foi projetada com uma estrutura contendo 8 neurônios de entrada, duas camadas ocultas, possuindo 20 e 10 neurônios respectivamente, e um neurônio de saída. O resultado obtido pela RNA foi, em geral, um erro de 9,5%.

Em Pereira et al. (2016), foram cedidas pela administração 155 amostras diárias de frequências no Restaurante Universitário da Universidade Federal de Viçosa (UFV). Foi utilizada uma rede MLP para previsão da variável alvo, sendo que para o treinamento do modelo foram separadas 135 amostras para treino e 20 para teste. Segundo os autores, foram testados diversos tipos de estrutura para se encontrar o número de neurônios na camada oculta que retornasse melhores resultados. Sendo assim, foi definido que a arquitetura da rede teria cinco variáveis de entrada (“Tipo de Carne”, “Dia da Semana”, “1 dia antes”, “2 dias antes”, “3 dias antes”), duas camadas ocultas, sendo a primeira com 20 neurônios e a segunda com 10, e um neurônio de saída. O resultado obtido teve um RMSE de 17,01%.

3 METODOLOGIA

Neste capítulo são apresentadas as etapas que foram realizadas para o desenvolvimento deste trabalho.

3.3 Descrição dos Métodos

A Figura 9 ilustra os passos seguidos na execução da metodologia deste trabalho.

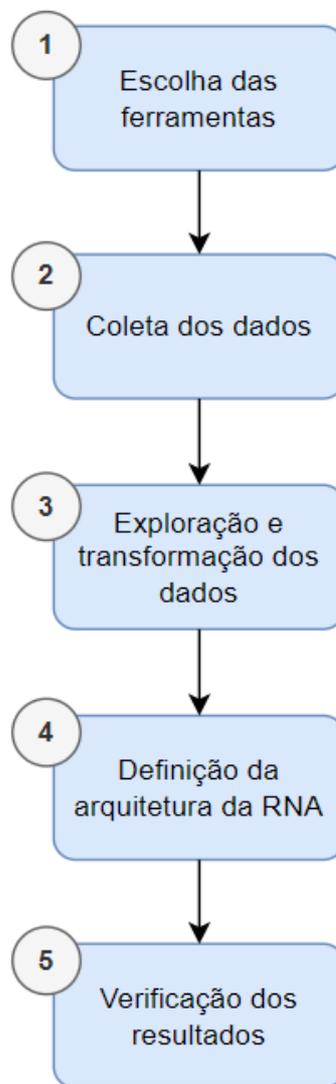


Figura 9: Pipeline metodológico
Fonte: Elaborado pelo autor

3.4 Ferramentas utilizadas

3.4.1 Linguagem de Programação

A linguagem escolhida para o implementar o algoritmo foi o Python. Dentre as suas diversas funcionalidade e utilizações, os motivos de sua escolha se deram por:

- Ser uma linguagem de sintaxe simples e amigável, que permite o rápido aprendizado;
- Ser uma ferramenta voltada para projetos de Inteligência Artificial (IA) e que possui muitas bibliotecas que facilitam a implementação de um código;
- Comunidade muito vasta e ativa, permitindo encontrar de forma mais fácil projetos ou trechos de códigos-base;

3.4.2 Google Colaboratory

O ambiente escolhido para o desenvolvimento do algoritmo deste trabalho foi o Google Colaboratory. Mais conhecido como Colab, é um serviço de nuvem oferecido pelo Google de forma gratuita, permitindo o desenvolvimento de algoritmos em diversas linguagens, como R, Julia, Swift e principalmente, Python (GOOGLE, 2023).

De acordo com o próprio site (GOOGLE, 2023), este é um ambiente que permite escrever e executar códigos no próprio navegador sem nenhuma configuração necessária, acesso sem custos a Unidade de Processamentos Gráficos (Do inglês *Graphics Processing Unit*, GPU) e compartilhamento fácil com a comunidade, sendo indicado para estudantes e pesquisadores de IA desenvolverem seus projetos

Outra vantagem de se utilizar este ambiente é pelo fato de ele já possuir integrado bibliotecas muito utilizadas em Ciência de Dados como:

- *Pandas* para análise e manipulação de dados (MCKINNEY, 2010);
- *Numpy* para realizar operações matemáticas e manipular *arrays* multidimensionais (HARRIS et al., 2020);
- *Matplotlib* para visualização de dados através de gráficos, histogramas e diagramas de dispersão (HUNTER, 2007);
- *Sklearn*, que oferece uma ampla variedade de algoritmos de aprendizado de máquina, incluindo classificação, regressão, agrupamento e redução de dimensionalidade (PEDREGOSA et al., 2011).

Todas as bibliotecas citadas foram utilizadas ao longo da implementação do algoritmo desenvolvido neste projeto.

3.4.3 PyTorch

Para facilitar o desenvolvimento de projetos de RNA, a utilização de bibliotecas são grandes facilitadores.

PyTorch é uma biblioteca de tensores otimizada para aprendizado profundo usando GPUs e CPUs (PASZKE et al., 2019). É especializada em diferenciação automática, cálculos de tensores e aceleração de GPU. Isso a torna mais adequada para aplicativos de aprendizado de máquina de ponta, como aprendizado profundo (ORACLE, 2022)

Stevens et al. (2020) destacam que a biblioteca Pytorch fornece uma excelente introdução ao aprendizado profundo, e ao mesmo tempo, uma poderosa ferramenta para uso tanto no contexto profissional em projetos utilizando dados reais, quanto para pesquisas. Algumas outras bibliotecas muito utilizadas, como TensorFlow (ABADI et al., 2016), foram consideradas, porém o Pytorch se mostrou mais recomendado para iniciantes. Dentre os motivos da escolha estão alguns pontos destacados pelos autores, tais como: a simplicidade para criar códigos em Python usando a biblioteca e o suporte a otimizações numéricas em expressões matemáticas genéricas, importantes para a implementação de projetos de aprendizagem profunda.

3.5 Coleta de dados

O Restaurante Universitário da UFOP, campus Morro do Cruzeiro serve refeições no almoço e no jantar de segunda a sexta-feira durante os dias letivos e no intervalo entre períodos. Os dados utilizados para o desenvolvimento do trabalho foram cedidos pela atual administração do RU e autorizados para serem utilizados no presente trabalho.

Foram coletadas 71 amostras relativas a um período letivo completo. A Figura 10 apresenta a quantidade de refeições por dia da base de dados. Das amostras cedidas, foram informados: o dia da semana, a semana e a quantidade de frequentadores em cada dia da base de dados disponibilizada.

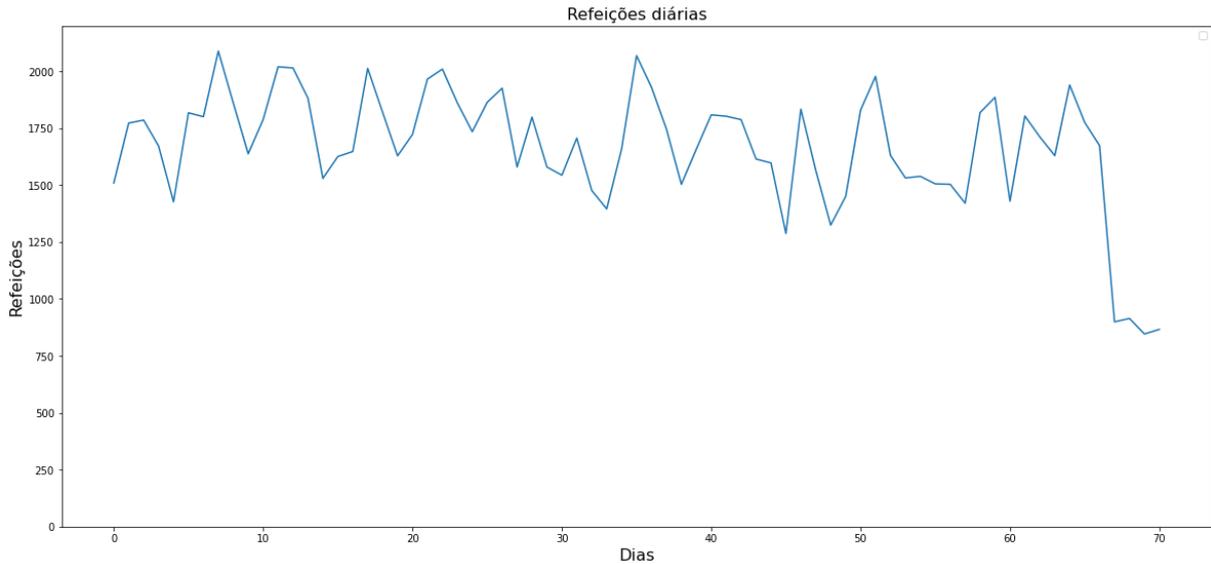


Figura 10: Curva do número de refeições de um período letivo
Fonte: Elaborado pelo autor

3.6 Exploração e transformação dos dados

Nesta etapa do processo são feitas visualizações nas amostras presentes na base de treino a fim de se retirar maiores informações a respeito do problema. Gráficos e métodos estatísticos foram utilizados para gerar esses *insights*. Ademais, são aplicadas transformações de forma a deixar os dados de uma maneira que o modelo possa interpretar melhor, já que segundo Géron (2022), a maior parte dos algoritmos de Aprendizado de Máquina preferem trabalhar com variáveis numéricas.

Primeiramente, das 71 amostras presentes na base, 56 foram separadas para o treinamento do modelo e 15 para teste. Então, através de explorações nos dados de treino, foram criadas cinco variáveis de entrada para o modelo, sendo estas: a quantidade de refeições dos três dias anteriores, se foi feriado no dia anterior e se será feriado no dia seguinte. Além do mais, foram aplicadas as devidas transformações nas variáveis categóricas presentes na base de dados.

Existem diversas técnicas para lidar com variáveis categóricas, dentre elas estão as técnicas de *LabelEncoder* e *OneHotEncoder*. Dahouda et al. (2021) descreve que o primeiro método é utilizado quando a característica categórica é ordinal e a manutenção da ordem é importante, sendo cada característica decodificada para refletir uma sequência.

Já na segunda técnica, a variável é expandida, criando-se uma nova coluna para cada categoria existente dentro dessa variável. A amostra recebe o valor 1 caso pertença a essa classe e 0 caso contrário. Desta forma, cada nível da categoria variável é comparada a um

nível de referência especificado, especialmente quando não há ordenação natural entre as categorias (DAHOUDA et al., 2021).

Para a entrada “Dia da Semana”, aplicou-se a técnica OneHotEncoder, LabelEncoder para “Semana” e para as entradas “Feriado Seguinte” e “Feriado Anterior” foi aplicada a binarização.

3.7 Arquitetura da rede

Para a definição da arquitetura de uma rede multicamadas é preciso fazer a escolha das quantidades de camadas ocultas e suas respectivas quantidades de neurônios. Esta não é uma tarefa simples, pois não há uma definição exata de quais valores trarão um resultado ótimo, sendo necessário executar diversos testes (KARSOLIYA, 2012). Sendo assim, foram testadas arquiteturas com diferentes quantidades de camadas ocultas e suas respectivas quantidades de neurônios foram testadas, verificando-se o desempenho de cada um.

3.8 Métricas de Avaliação

Ao final, a avaliação do desempenho do modelo foi feita por meio das métricas R^2 para que demonstrar o quanto o modelo consegue explicar da base dados, MAE e RMSE para se obter informações a respeito do erro absoluto gerado e a magnitude dos desvios. As mesmas métricas foram aplicadas nos resultados obtidos utilizando modelo atual de previsão da administração do restaurante para os mesmos dados teste, a fim de comparar os dois modelos.

4 RESULTADOS E DISCUSSÃO

A seguir são demonstrados os resultados obtidos pela exploração dos dados e o desempenho final do modelo, bem como as métricas utilizadas para avaliar o desempenho. Por fim, os resultados são comparados com o método atual utilizado pela administração do Restaurante.

4.1 Análise dos dados

Baseado nos trabalhos relacionados e na observação dos dados, percebe-se uma certa relação entre uma determinada amostra e as que a antecederam. Em Rocha et al. (2011), os autores utilizaram as médias do últimos 5 e 30 dias para criação de novas *features*. Já em Pereira et al. (2016) foi utilizado o valor dos 3 dias anteriores para criar três colunas. Sendo assim, optou-se por seguir a ideia do segundo trabalho, devido a pouca quantidade de amostras disponíveis, inviabilizando utilizar dados muito anteriores. Ademais, através de entrevistas com os administradores e, comprovado na exploração dos dados, foi detectado que há uma queda na frequência nos dias que antecedem e que sucedem os feriados. Sendo, portanto, uma informação relevante, e, por conseguinte, também adicionada como entrada para o modelo.

Desta forma, as variáveis de entrada “Dia da Semana”, “Semana”, “Feriado Anterior” e “Feriado Seguinte”, dadas como variáveis categóricas, ou seja, não possuem valores quantitativos, necessitam passar por um processo de transformação para serem interpretadas pelo modelo. Através de observações dos dados, como a média de cada dia da semana (Fig. 11) e informações cedidas pela administradora, os dias da semana se mostram relevantes na quantidade de frequentadores. Nos dados semanais (Figura 12) nota-se a presença de uma certa sazonalidade na demanda do Restaurante.

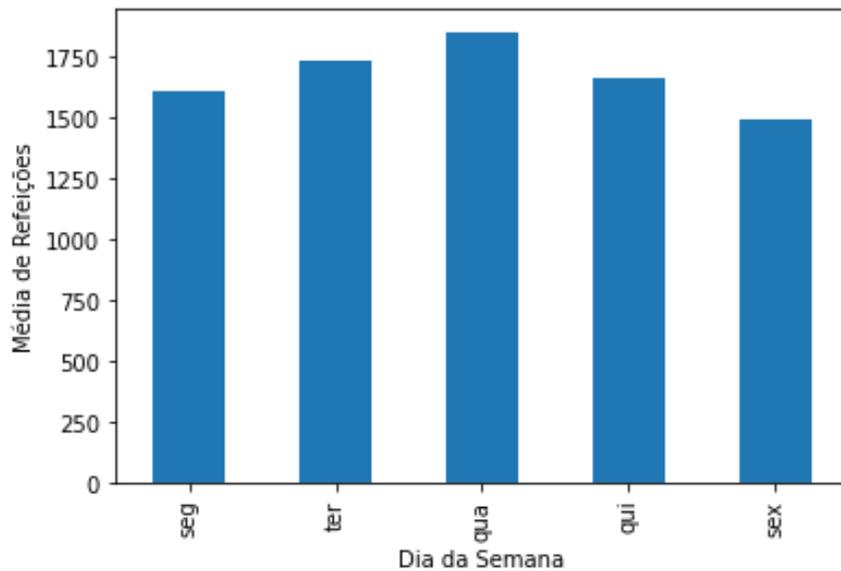


Figura 11: Média de refeições por dia da semana
Fonte: Elaborado pelo autor

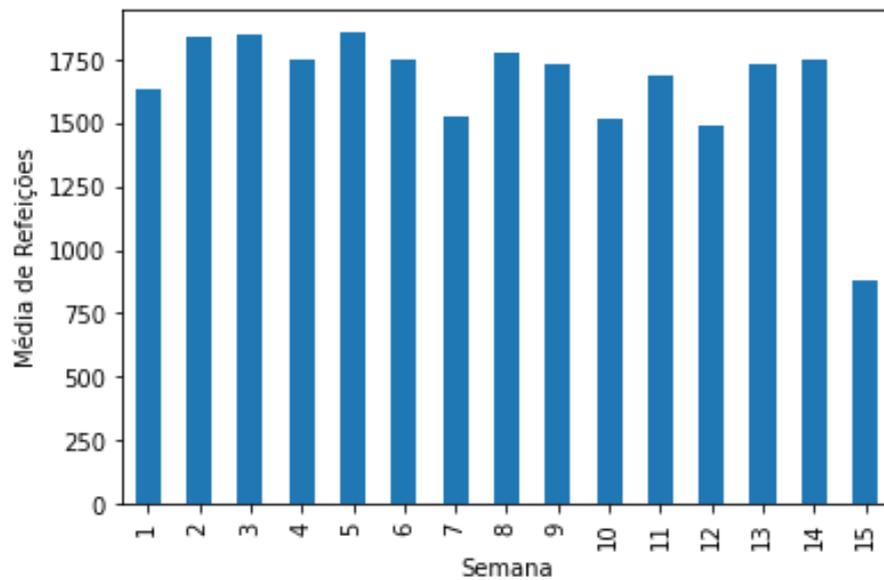


Figura 12: Média de refeições por semana
Fonte: Elaborado pelo autor

Foi escolhido para a codificação do dia da semana o *LabelEncoder*, a fim de demonstrar ao modelo uma certa relevância para cada dia. Os dias foram substituídos e ordenados por valores de 1 a 5, sendo 1 o dia de menor frequência média e o 5 o de maior. Para a variável “Semana”, foi utilizado o *OneHotEncoder*, pois apesar da sazonalidade, esta ordem não é tão bem definida. Sendo assim, optou-se por criar uma coluna para cada semana, e as amostras recebendo o valor 1 caso correspondessem a esta determinada semana e 0 caso contrário. Desta forma permite-se que o modelo aprenda a relevância de cada semana por si

só. Para as variáveis “Feriado Anterior” e “Feriado Seguinte”, optou-se pela binarização, recebendo valor 1 caso positivo e 0 caso negativo para esta informação.

Foram realizados diversos testes em arquiteturas diferentes com o objetivo de se encontrar uma melhor composição. Apesar de não existir algo de concreto que defina como uma arquitetura deve ser montada, existem algumas sugestões que podem guiar o estudo para qual direção de experimentos seguir.

Goodfellow et al. (2016), abordam que de acordo com a teoria da aproximação universal, uma rede composta por uma camada de saída linear e que tenha ao menos uma camada oculta com qualquer função de ativação não linear pode aproximar qualquer função, desde que a rede receba unidades ocultas suficientes.

Karsoliya (2012), afirma que um número muito grande de neurônios nas camadas ocultas pode causar problemas como *overfitting* (sobreajuste), sendo o modelo extremamente adaptado às amostras atuais, porém com baixa generalização em novas amostras. Desta forma, como sugere Géron (2022), optou-se por iniciar os testes com arquiteturas mais simples, com apenas uma camada oculta e fazer incrementações no número de camadas e de seus neurônios. A tabela 1 apresenta os testes, a quantidade de camadas ocultas e neurônios. Os hiperparâmetros iniciais utilizados para os testes estão demonstrados na Tabela 2.

Com relação à função de ativação presente na camada oculta, dentre as funções revisadas, optou-se pela ReLU por ser a mais comum, pelo fato de superar outras populares funções como a sigmoide e tangente hiperbólica (BROWNLEE, 2021).

Tabela 1: Hiper parâmetros utilizados nos testes

Otimizador	Adam
Taxa de Aprendizado	0,001
Regularização L2	0,001
Métrica	Erro médio Absoluto (MAE)
Tamanho do lote	15
Épocas	400
Porcentagem de validação	20%

Fonte: Elaborado pelo autor

Para os diversos testes realizados, o Teste 3 apresentou o melhor resultado, com apenas uma camada oculta e 15 neurônios, que corrobora com a literatura, uma vez que

problemas menos complexos e com poucos dados tendem a ter melhor desempenho em arquiteturas mais simples.

Tabela 2: Neurônios e camadas ocultas nos testes realizados

Teste	Camadas Ocultas	Neurônios
1	1	5
2	1	10
3	1	15
4	1	20
5	1	25
6	2	5
7	2	10
8	2	15
9	2	20
10	2	25
11	3	5
12	3	10
13	3	15
14	3	20
15	3	25

Fonte: Elaborado pelo autor

Sendo assim, a arquitetura da rede ficou desenhada, como exemplificado na Figura 13, com 21 neurônios de entrada, uma camada oculta com 15 neurônios e um neurônio de saída.

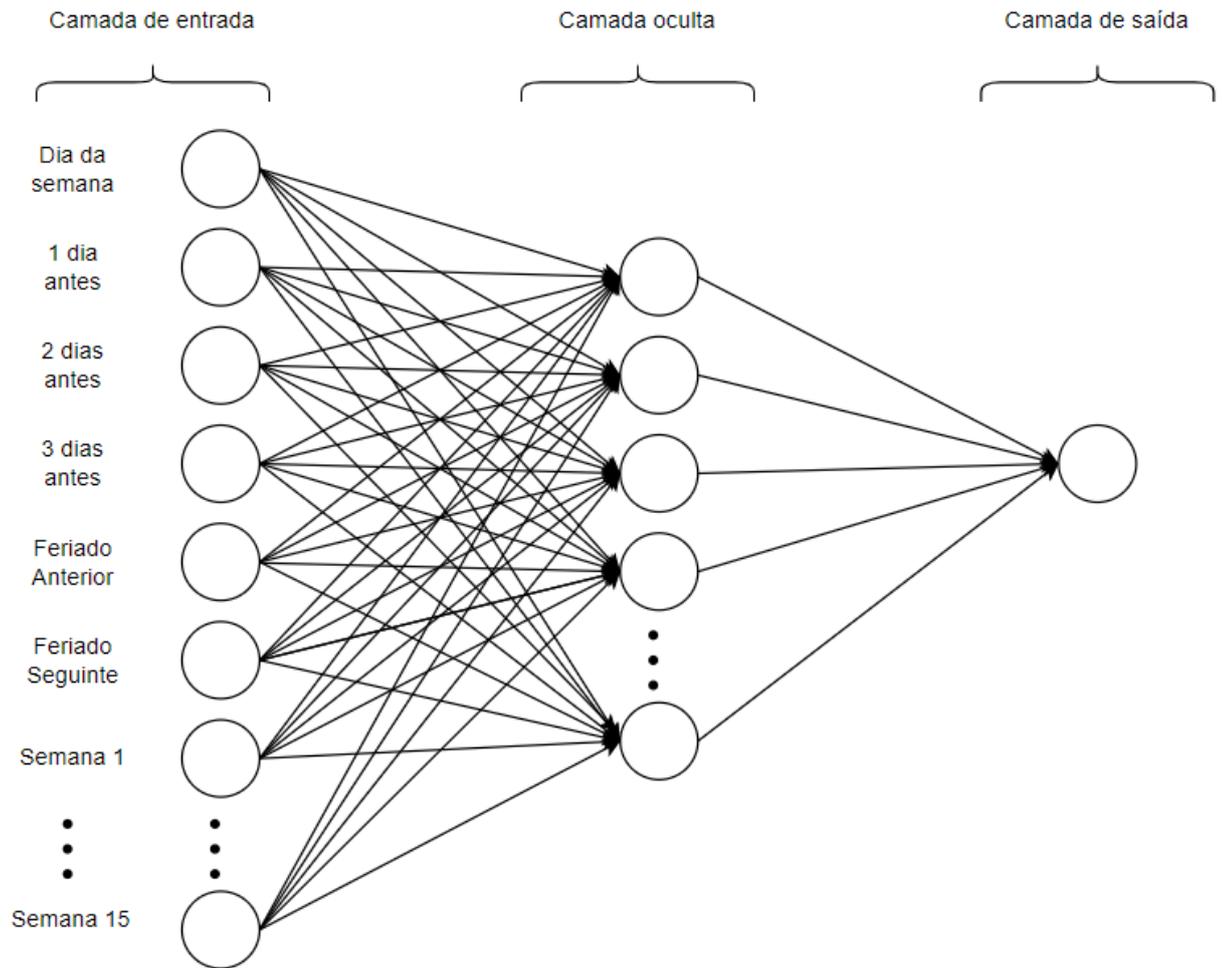


Figura 13: Esquema representativo da arquitetura final da rede
 Fonte: Elaborado pelo autor

Definida a arquitetura da rede, foram feitos novos testes variando os hiperparâmetros e a composição que retornou os melhores resultados estão mostrados na Tabela 3.

Tabela 3: Hiper parâmetros utilizados na execução final

Otimizador	Adam
Taxa de Aprendizado	0,00005
Regularização L2	0,00005
Métrica	Erro médio Absoluto (MAE)
Tamanho do lote	1
Épocas	400
Porcentagem de validação	20%

Fonte: Elaborado pelo autor

4.2 Desempenho do modelo

A Figura 14 mostra o desempenho do modelo na base de teste. É possível perceber que na maioria das previsões, o modelo conseguiu seguir a tendência do valor real.

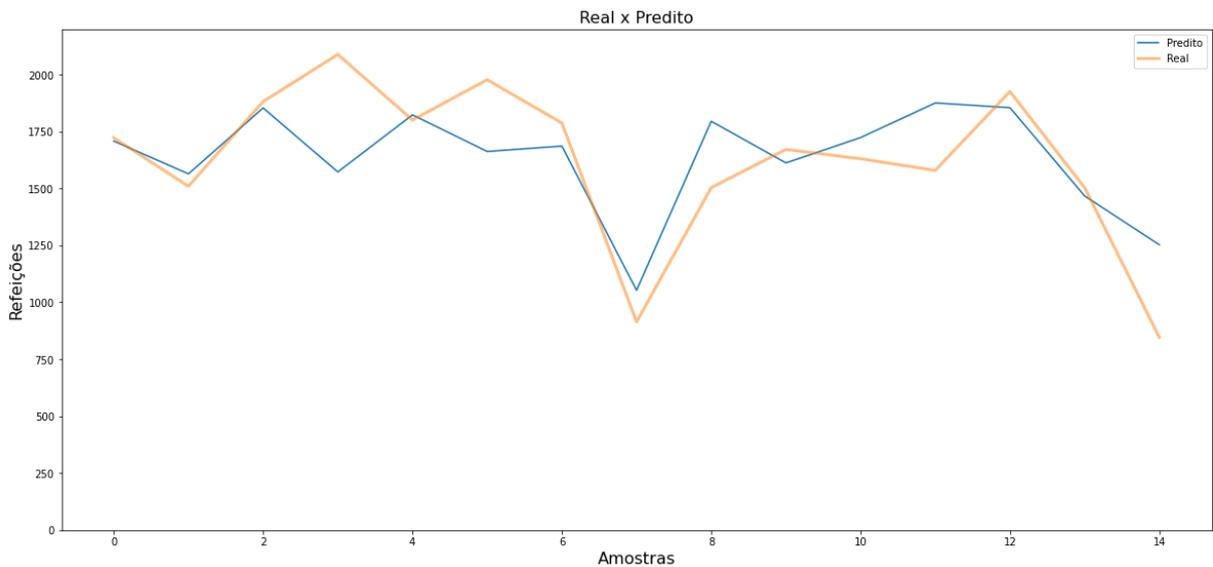


Figura 14: Curva da comparação entre os dados reais e previstos pelo RNA
Fonte: Elaborado pelo autor

A função de perda obtida é apresentada na Figura 15. Como a métrica que o modelo deve minimizar é o Erro Médio Absoluto, é possível observar que este valor reduz ao longo das épocas até oscilar entre um valor mínimo encontrado.

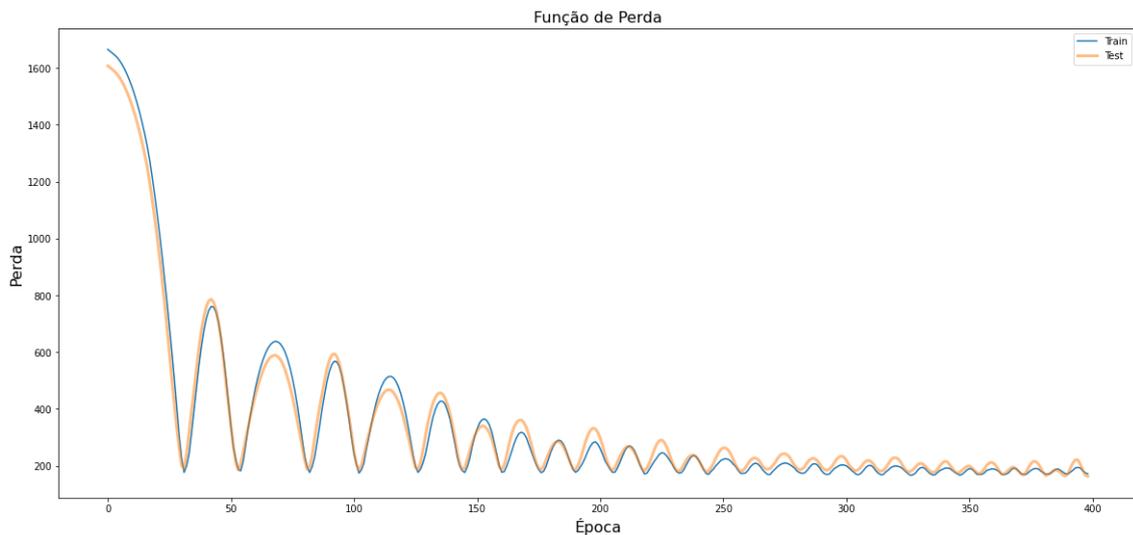


Figura 15: Curva da função de perda obtida
Fonte: Elaborado pelo autor

Realizada as medições para avaliar o desempenho das redes treinadas, obtiveram-se os resultados descritos a seguir.

O R^2 obtido de 56,1% indica que o modelo consegue explicar um pouco mais da metade da variância do problema. Esta é uma métrica difícil de ser interpretada, segundo Hamilton et al. (2015), pois o maior problema é que a variância na população estudada pode ter forte influência na magnitude desta métrica, fazendo com que valores altos não sejam necessariamente bons e valores baixos necessariamente ruins. Sendo assim, a melhor forma de se avaliar esta métrica é comparando modelos de mesma área de estudo, o que é feito no subcapítulo seguinte ao comparar o desempenho do modelo de previsão atual.

O modelo apresentou um MAE de 162,9 refeições, o que resulta em uma média de erro de 10% no total testado. Com relação ao RMSE, o modelo apresentou um valor de 224,4, o que sugere que existam certos desvios elevados de previsão.

A Figura 16 apresenta a distância da previsão em relação aos dados reais. Com isto, é possível visualizar algumas informações. A maior parte das previsões tiveram uma boa aproximação da quantidade de refeições servidas no dia. Porém, um ponto de atenção é o fato de 20% terem sido acima de 20%, o que ocasionaria em situações de grande desperdício ou falta de alimentos. A Tabela 4 apresenta com mais detalhes a faixa de erro do modelo.

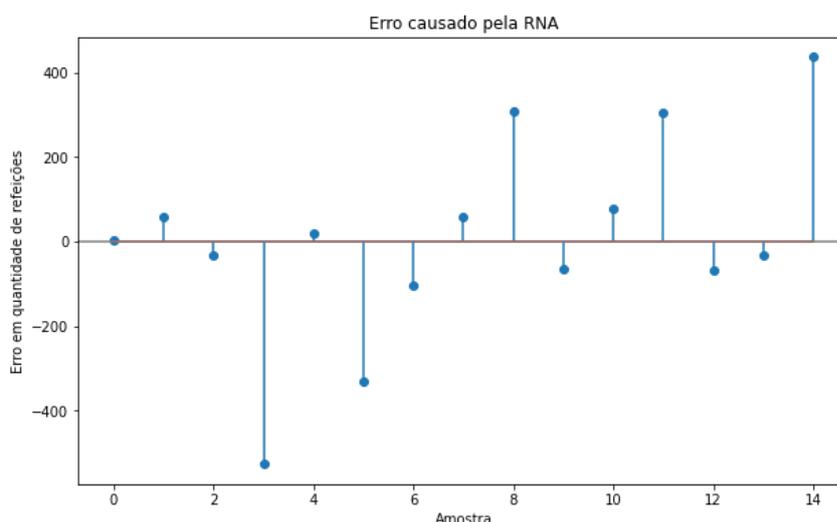


Figura 16: Gráfico do erro causado pela previsão
Fonte: Elaborado pelo autor

Tabela 4: Distribuição do erro cometido pelo modelo proposto

Menor que 5%	53%
Entre 5% e 10%	13%
Entre 10e 15%	13%
Maior de 20%	20%

Fonte: Elaborado pelo autor

4.3 Comparação com o modelo atual

Seguindo um dos objetivos do trabalho, nesta parte do capítulo é feita a comparação dos resultados obtidos com o modelo utilizado no momento pela gerência do RU. Em entrevista com a empresa prestadora dos serviços do RU, constatou-se que o modelo de previsão da quantidade de refeições do dia é feito pela média da soma do dia anterior e do último dia relativo ao mesmo dia da semana. Como exemplo, caso queira saber a quantidade de refeições em uma quarta-feira, é tirada a média da soma do dia anterior e da quarta-feira da semana anterior.

Aplicando este modelo nas amostras de testes, foram obtidos os seguintes valores para as métricas de avaliação. O R^2 foi de 62%, sendo 3,9% maior que aquele obtido com as RNA. Com o MAE de 145,7 refeições, o modelo atual teve um erro médio de 17,2 refeições, o que resultou em um erro médio 1% menor se comparado ao modelo neural. O RMSE 208,6 demonstra que o erro teve um desvio menor do que das RNA (Tabela 5).

Tabela 5: Distribuição do erro cometido pelo modelo atual

Menor que 5%	47%
Entre 5% e 10%	27%
Entre 10e 15%	13%
Maior de 20%	13%

Fonte: Elaborado pelo autor

5 CONCLUSÃO

Avaliando-se os resultados obtidos é possível perceber que o modelo atual de previsão das refeições se mostra ligeiramente melhor que o proposto, mesmo que com pouca diferença, para o conjunto de dados testados. Tal resultado não era necessariamente esperado. Porém, se se tivesse uma quantidade de dados maior, acredita-se que os resultados poderiam ter sido melhores que aqueles obtidos com a metodologia atual empregada pela gerência do RU.

Tendo em vista que o principal objetivo do trabalho - implementar um algoritmo de Redes Neurais Artificiais para prever a quantidade de refeições do Restaurante Universitário da UFOP – foi cumprido, conclui-se que o trabalho se mostrou viável, embora fique claro que melhorias devem ser feitas na sua metodologia. Além da aproximação do valor real, é preciso que o modelo tenha um desvio padrão baixo, já que o intuito de se melhorar a previsão é reduzir o desperdício e a falta de refeições. Para que isso aconteça, além de um melhor desempenho, é necessário que o modelo tenha experimentado uma base de dados mais ampla, a fim de permitir que a validação dos resultados obtidos seja feita de forma mais abrangente.

Como sugestão para trabalhos futuros, recomenda-se obter uma base de dados com um número maior de instâncias e/ou características (*features*) e repetir a metodologia aqui empregada com o intuito de se reavaliar a aplicabilidade das redes neurais para o problema aqui apresentado. Obtendo um espaço amostral maior será possível fazer um novo treinamento do modelo e comparar o seu desempenho novamente com o modelo vigente.

REFERÊNCIAS

- ABADI, M. et al. **Tensorflow: a system for large-scale machine learning**. Osd. **Anais...**Savannah, GA, USA, 2016.
- ALOM, M. Z. et al. The history began from alexnet: A comprehensive survey on deep learning approaches. **arXiv preprint arXiv:1803.01164**, 2018.
- ALVES, H. J.; BOOG, M. C. F. Comportamento alimentar em moradia estudantil: um espaço para promoção da saúde. **Revista de Saúde Pública**, v. 41, n. 2, p. 197–204, abr. 2007.
- AN, S. **Introduction to how an Multilayer Perceptron works but without complicated math**. Disponível em: <<https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-but-without-complicated-math-a423979897ac>>. Acesso em: 19 mar. 2023.
- ANDIFES. **V-Pesquisa-Nacional-de-Perfil-Socioeconomico-e-Cultural-dos-as-Graduandos-as-das-IFES-2018**. Brasil: [s.n.].
- BRANCO, S. T.; SAMPAIO, R. J. B. **Aplicação de redes neurais artificiais em modelos de previsão de demanda para equipamentos de infra-estrutura de telecomunicações**. (XXVIII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, Ed.)Rio de Janeiro - Brasil: 2008.
- BROWNLEE, J. **How to Choose an Activation Function for Deep Learning**. Disponível em: <<https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/#:~:text=The%20rectified%20linear%20activation%20function,such%20as%20Sigmoid%20and%20Tanh.>>. Acesso em: 14 mar. 2023.
- CHOLLET, F. **Deep Learning with Python, Second Edition**. [s.l.] Manning, 2021.
- COPPIN, B. **Inteligência Artificial**. [s.l.] LTC, 2010.
- DAHOUDA, M. K.; JOE, I. A Deep-Learned Embedding Technique for Categorical Features Encoding. **IEEE Access**, v. 9, p. 114381–114391, 2021.
- DATA SCIENCE ACADEMY. **Deep Learning Book**. Disponível em: <<https://www.deeplearningbook.com.br/>>. Acesso em: 12 mar. 2023.
- FARIAS, N. F. Desenvolvimento de analisador virtual para predição da pressão de fundo em poços de petróleo utilizando rede neural. 2018.
- FLECK, L. et al. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016.
- GALLANT, S. Perceptron-based learning algorithms. **IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council**, v. 1, p. 179–191, 1 fev. 1990.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. [s.l.] O'Reilly Media, 2022.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [s.l.] MIT Press, 2016.

GOOGLE. **Machine Learning: Interpreting Loss Curves**. Disponível em: <<https://developers.google.com/machine-learning/testing-debugging/metrics/interpretic?hl=en>>. Acesso em: 15 mar. 2023.

GOOGLE. **Collaboratory**. Disponível em: <<https://colab.research.google.com/>>. Acesso em: 12 mar. 2023.

GUPTA, M.; BHARGAVA, L.; SREEDEVI, I. **Artificial Neural Network based Task Scheduling for Heterogeneous Systems**. [s.l: s.n.].

HAMILTON, D. F.; GHERT, M.; SIMPSON, A. H. R. W. Interpreting regression models in clinical outcome studies. **Bone & Joint Research**, v. 4, n. 9, p. 152–153, 2015.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357–362, 2020.

HAYKIN, S. **Redes neurais: princípios e prática**. [s.l.] Bookman Editora, 2001.

HEBB, D. O. **The organization of behavior; a neuropsychological theory**. Oxford, England: Wiley, 1949.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in science & engineering**, v. 9, n. 03, p. 90–95, 2007.

IYODA, E. M. Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas. **FEEC/UNICAMP**, 2000.

J W CLARK. Neural network modelling. **Physics in Medicine & Biology**, v. 36, n. 10, p. 1259, 1991.

KALIRANE, M. **Gradient Descent vs. Backpropagation: What's the Difference?** Disponível em: <<https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/>>. Acesso em: 21 mar. 2023.

KARPATY, ANDREJ. **CS231n Winter 2016**. Disponível em: <<https://www.youtube.com/watch?v=NfnWJUyUJYU&list=PLkt2uSq6rBVctENoVBg1TpC7OQi31AIC&index=1>>>. Acesso em: 12 mar. 2023.

KARSOLIYA, S. Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. **International Journal of Engineering Trends and Technology**, v. 3, n. 6, p. 714–717, 2012.

LANZILLOTTI, H. S. et al. Aplicação de um modelo para avaliar projetos de unidades de alimentação e nutrição. **Nutrição Brasil**, v. 3, n. 1, p. 11–17, 2004.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1943.

MCKINNEY, W. **Data structures for statistical computing in python**. Proceedings of the 9th Python in Science Conference. **Anais...Austin, TX**, 2010.

MONTEIRO, C. A. et al. Causas do declínio da desnutrição infantil no Brasil, 1996-2007. **Revista de Saúde Pública**, v. 43, n. 1, p. 35–43, fev. 2009.

- MOREIRA, C. Neurônio. **Revista de Ciência Elementar**, v. 1, n. 1, 30 dez. 2013.
- NIED, A. et al. On-line neural training algorithm with sliding mode control and adaptive learning rate. **Neurocomputing**, v. 70, n. 16, p. 2687–2691, 2007.
- NORIEGA, L. Multilayer perceptron tutorial. **School of Computing. Staffordshire University**, v. 4, p. 5, 2005.
- ORACLE. **O que é PyTorch: um guia completo**. Disponível em: <<https://developer.oracle.com/pt-BR/learn/technical-articles/what-is-pytorch>>. Acesso em: 13 mar. 2023.
- PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. **Advances in neural information processing systems**, v. 32, 2019.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **the Journal of machine Learning research**, v. 12, p. 2825–2830, 2011.
- PEREIRA, H. et al. **Predição Do Número De Refeições Utilizando Redes Neurais Artificiais**. [s.l.: s.n.].
- POPESCU, M.-C. et al. Multilayer perceptron and neural networks. **WSEAS Transactions on Circuits and Systems**, v. 8, n. 7, p. 579–588, 2009.
- RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. **A Review of Activation Function for Artificial Neural Network**. 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI). **Anais...2020**.
- RASCHKA, S. **Single-Layer Neural Networks and Gradient Descent**.
- RAUBER, T. Redes neurais artificiais. 13 mar. 2023.
- ROCHA, J. C.; MATOS, F. D.; FREI, F. Utilização de redes neurais artificiais para a determinação do número de refeições diárias de um restaurante universitário. **Revista de Nutrição**, v. 24, n. 5, p. 735–742, out. 2011.
- SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85–117, 2015.
- SHIFFMAN, D.; FRY, S.; MARSH, Z. **The Nature of Code**. [s.l.] D. Shiffman, 2012.
- STEVENS, E.; ANTIGA, L.; VIEHMANN, T. **Deep Learning with PyTorch**. [s.l.] Manning Publications, 2020.
- SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning, second edition: An Introduction**. [s.l.] MIT Press, 2018.
- TAYLOR, M. **Neural Networks: A visual introduction for beginners**. [s.l.] Blue Windmill Media, 2017.
- WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. **Climate Research**, v. 30, n. 1, p. 79–82, 2005.

YEHUDA, S.; RABINOVITZ, S.; MOSTOFSKY, D. I. Nutritional Deficiencies in Learning and Cognition. **Journal of Pediatric Gastroenterology and Nutrition**, v. 43, 2006.