

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
FACULDADE DE VETERINÁRIA**

Gabriel Steinhauer Torres Rodrigues

**USO DE INTELIGÊNCIA ARTIFICIAL COM FOCO EM VISÃO
COMPUTACIONAL NA PRODUÇÃO DE BOVINOS E SUÍNOS**

Porto Alegre

2019/2

Gabriel Steinhauer Torres Rodrigues

**USO DE INTELIGÊNCIA ARTIFICIAL COM FOCO EM VISÃO
COMPUTACIONAL NA PRODUÇÃO DE BOVINOS E SUÍNOS**

**Trabalho de conclusão de curso de graduação
apresentado à Faculdade de Veterinária da Universidade
Federal do Rio Grande do Sul como requisito parcial
para a obtenção do título de Graduação em Medicina
Veterinária.**

Orientador: Dr. André Gustavo Cabrera Dalto
Co-orientador: Dr. Rafael da Rosa Ulguim

Porto Alegre

2019/2

RESUMO

O presente trabalho teve como objetivo realizar uma revisão sobre a história das redes neurais artificiais, assim como analisar diferentes trabalhos acadêmicos de áreas com e sem relação com as ciências veterinárias, para verificar o atual estado das redes neurais no meio acadêmico. Foi determinado, por meio da análise de quarenta artigos referentes a soluções de problemas na produção animal, através do uso de redes neurais, que o maior número de trabalhos da área abrangem a espécie bovina, que o número de dados de treinamento nos trabalhos é pequeno, se comparado com bases de dados de artigos de outras áreas do conhecimento, que a maioria dos trabalhos não entram em detalhes sobre o *software* utilizado para implementação das redes neurais e que a maioria dos autores principais dos trabalhos atuavam em departamentos de ciência animal. Outro objetivo do trabalho foi determinar as mais eficazes arquiteturas de redes neurais, cuja tarefa compreende a predição da ocorrência de descarte de matrizes suínas provenientes de granjas de quarto sítio e unidades produtoras de leitões. Uma base de dados com 5.013 fêmeas descartadas, classificadas em sete diferentes categorias de motivos de descarte, foi utilizada para o treinamento das redes neurais. Além disso as mais eficazes arquiteturas de redes neurais, cuja tarefa compreende a predição da ocorrência de descarte de matrizes suínas provenientes de granjas de quarto sítio e unidades produtoras de leitões. Uma base de dados com 5.013 fêmeas descartadas, classificadas em sete diferentes categorias de motivos de descarte, foi utilizada para o treinamento das redes neurais. A base foi filtrada e os dados de diferentes índices produtivos das fêmeas foram utilizados para a realização de quatro experimentos. O primeiro experimento tinha o objetivo de testar a eficácia de uma rede neural em classificar o motivo de descarte das fêmeas suínas nas sete diferentes categorias. A acurácia máxima alcançada foi de 56,35%. O segundo experimento visava determinar a eficácia de uma rede neural em estimar a probabilidade de descarte das fêmeas em apenas uma categoria. A acurácia máxima alcançada foi de 99,78%. O terceiro experimento tinha como objetivo avaliar a eficácia de uma rede neural em prever as variáveis da vida produtiva do parto seguinte das fêmeas, baseada em dados dos dois partos anteriores. O erro médio absoluto mínimo alcançado foi de 1,777 para a variável de número de desmamados. O quarto e último experimento tinha o intuito de testar a capacidade de uma rede neural em determinar se uma fêmea seria ou não descartada no parto seguinte, baseada em dados de dois partos anteriores. De maneira geral, as redes neurais demonstraram adequado desempenho em encontrar padrões em diferentes dados da vida produtiva de matrizes suínas e predizer a ocorrência e a classificação de seus descartes.

Palavras-chave: redes neurais artificiais, matrizes suínas, descarte de matrizes, aprendizado de máquina.

ABSTRACT

The objective of this work was to review the history of artificial neural networks, as well as to analyze different academic works in areas related and unrelated to the veterinary sciences, to verify the current state of neural networks in the academia. It was determined, through the analysis of forty research works referring to solutions of problems in animal production, through the use of neural networks, that the largest number of works in the area are referring to the bovine species, that the number of training data in the research works is small, compared to databases of articles from other areas of knowledge, that most of the papers do not go into detail about the software used to implement neural networks and that most of the main authors of the papers work in animal science departments. Besides that, another objective of this work was to determine the most effective neural network architectures, whose task includes the prediction of the occurrence of culling of female swine breeders from four-site units and piglet-production units. A database of 5,013 culled gilts, classified into seven different culling reason categories, was used for the neural networks training. The base was filtered and the data from different productive indexes of the females were used to perform four experiments. The first experiment aimed to test the effectiveness of a neural network in the task of classifying the reason for culling swine gilts into seven different categories. The maximum accuracy achieved was 56.35%. The second experiment aimed to determine the effectiveness of a neural network in estimating the probability of gilt culling in only one category. The maximum accuracy achieved was 99.78%. The third experiment aimed to evaluate the effectiveness of a neural network in predicting the productive life variables of the breeder's next delivery, based on data from the previous two parities. The minimum mean absolute error reached was 1.777 for the weaned number variable. The fourth and final experiment was designed to test the ability of a neural network to determine whether or not a gilt would be culled in the next parturition based on data from two previous births. In general, neural networks have shown adequate performance in finding patterns in different data of the productive life of swine breeders and predicting the occurrence and classification of their culling events.

Keywords: *artificial neural networks, gilts, culling, machine learning.*

AGRADECIMENTOS

Agradeço aos meus pais, Clacy e Ivan, que durante esses vários anos de curso me deram suporte não só financeiro, mas também emocional, para enfrentar os desafios da vida acadêmica. Eles, que ao longo de incontáveis noites acordaram involuntariamente ao som da porta de meu quarto abrindo de madrugada, para que eu pudesse fazer o “lanchinho da madrugada” e continuar estudando para as provas algumas horas antes de sair para a faculdade. Obrigado mãe, você que durante anos acordou mais cedo que do precisava para preparar um café gostoso do jeito que só tu sabes fazer, que várias vezes assumiu sozinho tarefas dentro de casa para que eu pudesse me concentrar nos estudos, que apesar de voltar cansada do trabalho sempre ofereceu ajuda no que for que eu precisasse. Obrigado pai, você que sempre foi companheiro e compreensível, que apesar de eu ter causado muitos prejuízos, nunca reclamou. Você que sempre perguntou se eu “tinha dinheiro na carteira” e sempre puxou minha orelha, perguntando se eu “estava estudando”, mesmo depois de um exaustivo dia de serviço. Não tenho como resumir em uma ou duas páginas o quanto vocês me apoiaram durante esses anos e muito menos resumir o quanto eu sou grato por tudo o que vocês fizeram e fazem por mim. Esse trabalho de conclusão de curso e todos os outros que eu fiz durante a faculdade são mais de vocês do que meus, sem vocês eu não teria conseguido. Eu amo muito vocês.

Agradeço à minha namorada, Íris Beatriz, que sem dúvidas foi a melhor coisa que aconteceu comigo durante a faculdade (e não é porque o curso é traumatizante). Eu me apaixonei no momento em que te conheci e me apaixono cada vez mais a cada dia que convivo contigo. Você tornou os dias na faculdade mais leves e agradáveis e se eu tinha forças para levantar de manhã todos os dias e me deslocar por uma hora até a FAVET, é porque eu sabia que encontraria lá uma mulher linda, com um sorriso doce e olhos brilhantes como uma noite estrelada me esperando para irmos à aula juntos. Não conheço superlativos exagerados o suficiente no dicionário para descrever o quão grande é meu amor por ti. Obrigado por tudo amor da minha vida <3.

Agradeço também aos amigos que fiz durante o início de minha trajetória acadêmica. Obrigado Anderson por sua sagacidade e seus trocadilhos que sempre me arrancavam risadas. Obrigado Láisa pelas bobagens e discussões nerds que tivemos. Obrigado Maria Vic, pelas risadas, voluptuosidades, assédios descarados à Íris e informações inúteis sobre astrologia. Obrigado Renan pelas vitórias na sinuca e pela parceria. Obrigado Bianca pelas brincadeiras e conversas filosóficas que tivemos. Obrigado outra Bianca, Gianini e Ana Paula por me ensinar a ter mais paciência com pessoas de “personalidade forte” e pelas ótimas conversas que

tivemos. Obrigado Lucas pelo carinho e pelos ótimos papos que tivemos, você é incrível! Obrigado Luíza pelas discussões acaloradas e interessantes e pelos ensinamentos de coisas extremamente específicas que só tu saberias. Obrigado Thamiris pelas zoeiras e pelas gírias de Alvorada que aprendi contigo. Obrigado Mateus pelas partidas de futebol e de nada pelas caronas. Obrigado amigos do SGR (Carol, Rogan, Gabi Velho, Jasmyne, Paola, Gabriely, etc.) pela parceria, foi ótimo passar esses anos convivendo com vocês. Se o SGR é o melhor setor da faculdade, é por causa de vocês, que compõem ele. Obrigado amigos que eu conheci no começo do curso e que aqui não citei porque tenho memória curta e porque estou escrevendo este texto por último e de última hora, como sempre. Obviamente essas frases não são o suficiente para descrever o quanto eu gosto de vocês e o quanto a nossa amizade significa pra mim.

Obrigado amigos que conheci já no final do curso, mas que são tão especiais para mim quanto os que citei acima. Obrigado Josnei, Rayssa, Kevin, Rafa, Mônica, Samuel, Pietra e talvez mais alguém que eu tenha esquecido, mas que provavelmente está no famigerado grupo do Whatsapp, cuja obscenidade no nome me impede de citá-lo aqui. Espero continuar convivendo e zoando com vocês pelos anos que seguirem.

Obrigado André, pela parceria de anos e principalmente pelos ensinamentos não só sobre medicina de ruminantes, mas também sobre como lidar com pessoas e alunos, pois nisso tu é realmente um pós-doc. Tu foste não somente um tutor pra mim, mas um grande amigo que tornou meus dias na faculdade muito mais divertidos e significantes. Eu sei que posso contar contigo para o que for e espero que tu saibas que o mesmo vale pra ti. A paixão que tu tens pela profissão (de veterinário e educador) é admirável e é algo que eu almejo um dia alcançar.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquematização de estrutura básica de uma rede neural profunda.....	16
Figura 2 – Demonstração do processo de seleção de amostras dos pares de partos de fêmeas suínas com dois ou mais partos, para o Experimento 3.....	51
Figura 3 – Demonstração do processo de seleção de amostras dos pares de partos de fêmeas suínas com dois ou mais partos, para o Experimento 4.....	52
Figura 4 – Resumo da base de dados, dos critérios utilizados para filtragem dos dados, do tipo de tarefa das redes neurais e da quantidade de valores de saída de cada uma.....	53
Gráfico 1 – Espécies abordadas nos trabalhos analisados.....	42
Gráfico 2 – Média do número de amostras dos trabalhos, separadas por espécie.....	43
Gráfico 3 – Número de amostras das bases de dados de treinamento dos trabalhos analisados na presente revisão.....	44
Gráfico 4 – Número de autores principais e suas áreas de atuação.....	44
Gráfico 5 – Softwares utilizados para a implementação das redes neurais nos trabalhos analisados.....	45
Gráfico 6 – Acurácia por época de cada arquitetura testada.....	61
Gráfico 7 – Erro médio absoluto (MAE) por época de cada arquitetura testada.....	63
Gráfico 8 – Acurácia por época da arquitetura Arq_lin_1.....	65
Gráfico 9 – Acurácia por época de arquiteturas das funções de ativação escolhidas para experimento 1.....	65
Gráfico 10 – Acurácia por época de arquiteturas com diferentes funções de otimização, utilizadas no experimento 1.....	69
Gráfico 11 – Erro médio absoluto (MAE) por época de arquiteturas com diferentes funções de otimização, utilizadas no experimento 3.....	70
Gráfico 12 – Acurácia por época das redes neurais utilizadas no experimento 1.....	71
Gráfico 13 – Acurácia por época da rede neural Arq_class_2.....	71
Gráfico 14 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos reprodutivos (C1) ou não (C2 a C7).....	73
Gráfico 15 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C2 ou não (C1 e C3 a C7).....	73
Gráfico 16 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C3 ou não (C1 a C2 e C4 a C7).....	74

Gráfico 17 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C4 ou não (C1 a C3 e C5 a C7).....	74
Gráfico 18 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C5 ou não (C1 a C4 e C6 a C7).....	75
Gráfico 19 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C6 ou não (C1 a C5 e C7).....	75
Gráfico 20 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C7 ou não (C1 e C3 a C7).....	76
Gráfico 21 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável duração da lactação.....	78
Gráfico 22 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável duração da gestação.....	78
Gráfico 23 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável intervalo desmame cio.....	79
Gráfico 24 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável número de nascidos totais.....	79
Gráfico 25 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável número de nascidos vivos.....	80
Gráfico 26 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável número de nascidos totais.....	80
Gráfico 27 – Acurácia por época das redes neurais utilizadas no experimento 4.....	81

SUMÁRIO

1 INTRODUÇÃO	8
1.1 HISTÓRIA DAS REDES NEURAIIS	8
1.3.1 Biomimética	8
1.3.2 Neurônios	9
1.3.3 Rosenblatt e o “Perceptron”	10
1.3.4 ADALINE e MADALINE	11
1.3.5 O inverno da inteligência artificial	12
1.3.6 O retorno das redes neurais	13
2 ESTRUTURA DAS REDES NEURAIIS	15
2.1 ESTRUTURA BÁSICA	15
2.2 TIPOS DE REDES NEURAIIS	16
2.2.1 Redes neurais supervisionadas	16
2.2.2 Redes neurais não-supervisionadas	17
2.2.3 Redes neurais do tipo <i>feed-forward</i>	17
2.2.4 Redes neurais do tipo <i>recurrent</i>	17
2.2.5 Redes neurais convolucionais	18
2.3 FUNÇÕES DE CUSTO	18
2.3.1 Função de custo <i>mean squared error</i>	19
2.3.2 Função de custo <i>mean absolute error</i>	19
2.3.3 Função de custo <i>binary cross-entropy</i>	20
2.3.4 Função de custo <i>multi-class cross-entropy</i> ou <i>category cross-entropy</i>	20
2.4 FUNÇÕES DE ATIVAÇÃO	20
2.4.1 Função de ativação <i>step</i>	21
2.4.2 Função de ativação <i>sigmoid</i>	21
2.4.3 Função de ativação <i>tanh</i>	22
2.4.4 Função de ativação <i>softmax</i>	22
2.4.5 Função de ativação <i>ReLU</i>	22
2.4.6 Função de ativação <i>SELU</i>	23
2.4.7 Função de ativação <i>Leaky ReLU</i>	24
2.5 ALGORITMOS DE OTIMIZAÇÃO	24

2.5.1	<i>Gradient Descent</i>	24
2.5.2	<i>Stochastic Gradient Descent</i>	25
2.5.3	<i>Adagrad</i>	25
2.5.4	<i>Adam</i>	25
2.5.5	<i>RMSProp</i>	26
2.6	PROBLEMAS COMUNS.....	26
2.6.1	Problema da dissipação de gradiente.....	26
2.6.2	Problema da explosão do gradiente.....	27
2.6.3	<i>Overfitting</i>	27
3	Uso de redes neurais na atualidade.....	27
3.1	MÉTODO DE BUSCA DOS ARTIGOS.....	27
3.2	EXEMPLOS DE REDES NEURAI APLICADAS A DIVERSAS ÁREAS.....	28
3.3	EXEMPLOS DE REDES NEURAI APLICADAS À ÁREA DE PRODUÇÃO ANIMAL.....	29
3.3	ANÁLISE DOS TRABALHOS PUBLICADOS NA ÁREA DA PRODUÇÃO ANIMAL.....	37
4	ARTIGO – PREVISÃO DE MOTIVO DE DESCARTE DE MATRIZES SUÍNAS utilizando redes neurais artificias	45
4.1	RESUMO.....	46
4.2	ABSTRACT.....	46
4.3	INTRODUÇÃO.....	47
4.4	MATERIAIS E MÉTODOS.....	48
4.4.1	Origem da Base de Dados Utilizada.....	48
4.4.2	Normalização e Verificação dos dados.....	48
4.4.3	Organização dos dados para Experimento 1.....	48
4.4.4	Organização dos dados para Experimento 2.....	50
4.4.5	Organização dos dados para Experimento 3.....	50
4.4.6	Organização dos dados para Experimento 4.....	51
4.4.7	Determinação das melhores arquiteturas para as redes neurais.....	53
4.4.8	Arquitetura das redes neurais para os experimentos 1, 2 e 4.....	58
4.4.9	Arquitetura das redes neurais para o experimento 3.....	59
4.4.10	Software e hardware utilizados para a implementação das redes neurais.....	59
4.5	RESULTADOS.....	60

4.5.1 Resultados das arquiteturas testadas para determinar o efeito de uma função <i>dropout</i> na camada de entrada das redes neurais.....	60
4.5.2 Resultados das arquiteturas testadas para determinar o efeito das funções de ativação das redes neurais	63
4.5.3 Resultados das arquiteturas testadas para determinar o número de camadas e de nódulos nas camadas escondidas.....	66
4.5.4 Resultados das arquiteturas testadas para determinar a função de otimização utilizada	68
4.5.5 Resultados do experimento 1	70
4.5.6 Resultados do experimento 2	72
4.5.7 Resultados do experimento 3	76
4.5.8 Resultados do experimento 4	81
4.6 DISCUSSÃO	82
4.5 CONCLUSÃO.....	84
5. CONCLUSÕES.....	85
REFERÊNCIAS	86

1 INTRODUÇÃO

Redes neurais e outros algoritmos de aprendizado de máquinas não são algo novo, essas ferramentas estão presentes na literatura científica de 7 décadas atrás, porém, suas aplicações para problemas do mundo real somente obtiveram atenção do público geral nos últimos anos (AGGARWAL, 2018). As áreas onde um médico veterinário pode atuar, em sua maioria, possuem como alguns requisitos o raciocínio lógico e reconhecimento de padrões (FEITOSA, 2014). Essas habilidades, até certo ponto, podem ser exercidas com excelência por uma rede neural, oferecendo aos profissionais uma ferramenta valiosa de auxílio na tomada de decisões (SANTOS; SEIXAS; PEREIRA; MEDRONHO, 2005). A tecnologia ainda não é avançada o suficiente para substituir completamente um profissional do ramo, mas é certamente capaz de realizar tarefas que, devido ao tempo necessário, à complexidade e à dificuldade, seriam inviáveis ou demasiadamente custosas para seres humanos (DYBOWSKI; GANT, 2007).

O aumento da eficácia no exercício da profissão de médicos veterinários é importante tanto do ponto de vista econômico, visto que a produção e clínica animal movimentam grandes quantidades de dinheiro, quanto do ponto de vista social, visto que esses profissionais lidam diariamente com a vida de animais e questões de saúde pública (KAHN, 2006).

1.1 HISTÓRIA DAS REDES NEURAIS

Para entendermos as motivações e processos de criação por trás dos diversos algoritmos de redes neurais existentes nos dias de hoje, devemos olhar para o passado e tentar compreender os problemas que estavam tentando ser resolvidos e o contexto histórico em que se inseriam os pesquisadores e profissionais por trás do desenvolvimento dessa tecnologia.

1.3.1 Biomimética

Biomimética é o estudo das estruturas biológicas e suas funções, com o objetivo de otimizar estratégias e soluções para problemas de diversas áreas do conhecimento. Esse termo foi cunhado apenas recentemente por Janine Benyus (1997) em seu livro “Biomimicry: Innovation Inspired by Nature”, porém a biomimética tem sido utilizada na história humana há muito mais tempo (BENYUS, 1997). Um exemplo bastante palpável é o uso da seda há mais

de 5000 anos e como os humanos se inspiraram na técnica empregada pelo bicho-da-seda para a produção de diferentes tecidos (CHERRY, 1987).

Eisner e Nowicki (1983) publicaram na revista *Science* um trabalho científico relatando um incrível fenômeno da natureza e da seleção natural: algumas aranhas “decoram” suas teias com regiões mais espessas, chamadas *estabilimenta*, para que pássaros consigam enxergá-las e não as destruam ao levantar voo. Na época, os pesquisadores pensaram que os pássaros evitavam as teias devido à sua estrutura mais grossa, porém Craig e Bernard (1990) descobriram que essas teias possuíam um alto índice de reflexão para a luz ultravioleta, que coincidentemente faz parte do espectro de absorção de cones e bastonetes da retina da maioria dos pássaros. Alguns anos depois, tentando solucionar o que viria a ser um grande problema para pessoas residindo e trabalhando em arranha-céus, a empresa Arnold Glas lançou o vidro Ornilux, que possui em sua estrutura faixas reflexivas à luz UV, evitando que pássaros colidam com as janelas que utilizem esse tipo de vidro.

McVay (1990) propôs um novo método de síntese de materiais que buscava imitar um processo observado em moluscos, na etapa de formação de suas conchas. Liu *et al.* (2007) utilizou a estrutura de folhas da planta *Nelumbo nucifera*, conhecida popularmente como lótus, para criar um tecido de algodão com propriedades hidrofóbicas. Ethan *et al.* (2014) provou que materiais com uma microestrutura que mimetiza a pele de tubarões são menos propensos à contaminação bacteriana, sem nenhum uso de agentes antibióticos ou condições que desfavoreçam seu crescimento.

As estruturas biológicas acabam sendo ótimos exemplos de soluções inteligentes e simples para diversos problemas presentes na sociedade humana, provavelmente porque essas estruturas foram selecionadas naturalmente por milhares de anos ao ambiente em que estamos inseridos (BENYUS, 1997). Seres vivos que conseguiram sobreviver aos desafios da natureza de maneira eficiente, sem utilizar muita energia, como tubarões cuja pele possui propriedades antibacterianas ou aranhas que evitam o risco de terem suas teias destruídas por pássaros desavisados, sobrevivem até hoje e nos mostram o caminho para o futuro.

1.3.2 Neurônios

Existem aproximadamente 100 bilhões de neurônios no sistema nervoso de humanos e de dez a cinquenta vezes mais células gliais, que em conjunto formam complexas redes responsáveis pela memória, raciocínio, reconhecimento de padrões e outras funções do sistema

neuronal. Esses neurônios funcionam de forma conjunta através da transmissão de potenciais de ação em estruturas chamadas axônios, desencadeados por mudanças no potencial de membrana que ultrapassam um certo limite. Esses potenciais de ação carregam informação e o potencial de membrana é causado por diferentes concentrações de íons dentro e fora das células neuronais (CUNNINGHAM, 2013).

1.3.3 Rosenblatt e o “Perceptron”

Rosenblatt (1958) publicou em um artigo um modelo que é hoje considerado como o precursor das redes neurais artificiais. O modelo criado pelo pesquisador seguia princípios de aprendizagem e reconhecimento de padrões há não muito tempo descobertos por cientistas do ramo da neurociência. De maneira resumida, uma das teorias que buscavam elucidar o funcionamento do cérebro humano considerava que:

- As conexões entre os neurônios associados aos processos de aprendizagem e reconhecimento de padrões são diferentes de um indivíduo para outro e a construção dessas conexões após o nascimento é, em sua essência, randômica, sendo pouco afetada por fatores genéticos.
- Essas conexões são sujeitas a pequenas mudanças causadas ao longo do tempo e, portanto, possuem uma plasticidade inerente.
- Estímulos similares ativam grupos de neurônios similares, sendo que o conceito de similaridade, nesse contexto, não necessariamente é relacionado a condições similares no ambiente em que o indivíduo está inserido, mas sim a situações que disparem o mesmo grupo de neurônios relacionados à transcrição de estímulos externos em correntes elétricas, sejam essas quais forem.

Rosenblatt (1958) teorizou e construiu um modelo computacional que seria capaz de simular um neurônio biológico e conseqüentemente reproduzir algumas de suas capacidades de aprendizagem e reconhecimento de padrões. O *Mark I Perceptron* utilizava valores de entrada, que seriam análogos aos estímulos do meio em diferentes tipos de neurônios sensitivos responsáveis por traduzir essas informações em estímulos nervosos, e valores de saída, análogos ao resultado de determinado estímulo elétrico no cérebro de um indivíduo. O processo pelo qual

um reconhecimento de padrões acontecia utilizava de operações matemáticas para encontrar relações entre os valores de entrada e saída. Essas operações matemáticas consistiam no seguinte:

- Os valores de entrada eram multiplicados por um número, chamado de “peso”.
- Como os valores de entrada eram variados, o número passado para o *perceptron* consistia em uma soma de todos esses valores de entradas multiplicados pelos seus respectivos pesos.
- Esse somatório de valores de entradas multiplicados por seus pesos era então subtraído de um número, chamado de “bias” em inglês e traduzido aqui de forma não literal para “limiar”, de modo a preservar seu sentido no modelo proposto.
- Esse limiar tinha o propósito de tentar simular o que acontecia em um neurônio biológico, onde um potencial de ação só seria desencadeado quando o potencial de membrana fosse modificado o suficiente para ultrapassar o limiar do neurônio.
- Para que o *perceptron* “aprendesse”, os pesos e o limiar eram modificados um por um a cada tentativa, de modo que, eventualmente, o resultado fosse satisfatório o suficiente e conseguisse generalizar os diferentes valores de saída para os diferentes valores de entrada.

Através dessas regras, o *Mark I Perceptron* era capaz de resolver problemas de classificação, porém com apenas duas classes no máximo, devido à natureza binária de seu valor de saída. Outra maneira de descrever as capacidades desse modelo computacional é dizer que o mesmo possui a habilidade de aprender padrões linearmente separáveis.

Com isso, o *perceptron* revolucionaria o campo da inteligência artificial. Ao mimetizar os neurônios biológicos, era possível alcançar e se utilizar das capacidades de um cérebro humano, através do uso de um computador (YADAV, YADAV, KUMAR, 2015).

1.3.4 ADALINE e MADALINE

O sucesso do *perceptron* fez com que o interesse geral pela inteligência artificial crescesse e diversas pessoas tentassem criar modelos que partissem dos mesmos princípios. Um

exemplo é o *ADALINE*, criado por Bernard Widrow e Ted Hoff na Stanford University, no ano de 1960. *ADALINE* significa *Adaptive Linear Element* e a única diferença marcante para o *perceptron* era o fato de que os pesos eram ajustados de forma automática baseados na soma dos valores de entrada multiplicados pelos seus pesos e não por um ser humano se baseando nos valores de saída (YADAV; YADAV; KUMAR, 2015; WIDROW; HOFF, 1960).

Outra diferença marcante do modelo de Widrow e Hoff para o de Rosenblatt foi o emprego de mais de um neurônio artificial na estrutura da rede neural. Ao implementar em seu modelo várias unidades *ADALINE*, em 3 diferentes camadas (camada de entrada, camada escondida e camada de saída), os pesquisadores alcançaram resultados surpreendentes e o modelo começou a ser utilizado para remover ecos de ligações telefônicas. Essa nova estrutura foi chamada de *MADALINE*, pois utilizava muitas unidades *ADALINE* (YADAV; YADAV; KUMAR, 2015; WIDROW; HOFF, 1960).

1.3.5 O inverno da inteligência artificial

O chamado “inverno da inteligência artificial” foi uma época onde os investimentos e interesse geral pela tecnologia entraram em declínio, portanto desenvolver qualquer pesquisa durante a época era algo muito desafiador. Existem vários motivos para isso ter ocorrido, mas alguns acontecimentos tiveram um grande impacto na visão que o público geral e principalmente os investidores tinham sobre o potencial da inteligência artificial (YADAV; YADAV; KUMAR, 2015; NEURAL NETWORKS – HISTORY, 2019).

Na década de 50, existia uma urgência para o desenvolvimento de pesquisas que auxiliassem os Estados Unidos a adquirir uma vantagem tecnológica sobre a Rússia, devido à guerra fria (YADAV; YADAV; KUMAR, 2015; BRIEF HISTORY OF NEURAL NETWORKS, 2019). Em 1954, o Conselho Nacional de Pesquisa (*National Research Council*) dos Estados Unidos começou a investir fortemente em pesquisas voltadas à tradução por máquinas, para a tradução de documentos e pesquisas russas. Porém, muitas expectativas foram criadas e o poder computacional dos equipamentos da época e o conhecimento sobre algoritmos eficientes de inteligência artificial não eram vastos o suficiente para que fosse possível alcançar algum progresso nessa área (BRIEF HISTORY OF NEURAL NETWORKS, 2019).

Em 1964, foi criado um comitê para avaliar o problema da tradução por máquinas e procurar por soluções para a falta de avanço nas pesquisas. Esse comitê, chamado de *Automatic Language Processing Advisory Committee* (ALPAC), concluiu seu trabalho em 1966, quando

publicou um relatório sobre o estado das pesquisas na época, onde relatou que a tradução com utilização de máquinas não só era mais cara, mas também menos precisa e mais demorada do que uma tradução feita por humanos (PIERCE *et al.*, 1966). Então, depois de 10 anos de investimento e mais de 20 milhões de dólares, o apoio do Conselho Nacional de Pesquisa dos Estados Unidos às pesquisas foi encerrado (BRIEF HISTORY OF NEURAL NETWORKS, 2019).

Minsky e Papert (1969) publicaram o livro “Perceptrons”, que apresenta uma visão bastante pessimista do potencial das redes neurais utilizando múltiplos *perceptrons*, que era o que se tinha de mais avançado naquele período. Isso, somado às diversas frustrações encontradas pelos pesquisadores devido às altas expectativas geradas por declarações exageradas na época, causou um intenso declínio no investimento de pesquisas desse ramo nas décadas 70 e 80 (NEURAL NETWORKS – HISTORY, 2019).

Em 1973, o parlamento do Reino Unido pediu para Sir Michael James Lighthill escrever um relatório sobre a situação do progresso nas pesquisas de inteligência artificial no Reino Unido. Assim como o relatório da ALPAC, o relatório Lighthill relatava que as expectativas criadas para a inteligência artificial não se concretizariam tão cedo e que muitas habilidades que teriam sido atribuídos aos algoritmos não seriam aplicáveis aos problemas complexos do mundo real, apenas a versões simplificadas que estariam sendo usadas para testá-los (YADAV; YADAV; KUMAR, 2015; BRIEF HISTORY OF NEURAL NETWORKS, 2019).

Esses relatórios pessimistas e as constantes decepções com os resultados de diversas pesquisas que tiveram um alto investimento tanto de empresas quanto de programas de governos contribuíram para o período que chamamos hoje de inverno da inteligência artificial.

1.3.6 O retorno das redes neurais

Da mesma forma que a impressão pública sobre a inteligência artificial foi influenciada por alguns pesquisadores e profissionais do ramo da ciência da computação com expectativas exageradas seguidas por uma óbvia decepção e visão pessimista, o ressurgimento do interesse por essa área de pesquisa também foi, de certa forma, devido às pessoas envolvidas nela. Na década de 80, John Joseph Hopfield publicou diversos trabalhos abordando a utilidade das redes neurais artificiais e sua aplicabilidade a problemas do mundo real. John era didático e carismático, trazendo bastante atenção de diversos investidores e profissionais para a tecnologia que apresentava. Além disso, o fato do Japão ter anunciado, em 1982, o famoso *Fifth*

Generation Computer Systems (FGCS), que seria um supercomputador com um grande número de processadores paralelos e que teria o intuito de rodar diversos algoritmos de inteligência artificial que não atingiram seu potencial no passado devido à falta de poder computacional dos equipamentos da época, fez com que os governos de diversos países que tentavam se manter no topo de uma corrida tecnológica começassem a investir em programas semelhantes (BRIEF HISTORY OF NEURAL NETWORKS, 2019; NEURAL NETWORKS – HISTORY, 2019).

Após esse súbito aumento no auxílio financeiro de diversas pesquisas no ramo da inteligência artificial, diversos algoritmos e modelos interessantes começaram a surgir. Rumelhar, Hinton e Williams (1986) descreveram um novo método de aprendizagem nas redes neurais: o *back propagation*, onde os valores de saída de cada neurônio são utilizados para calcular a performance da rede neural em geral e são então utilizados para fazer modificações nos neurônios anteriores, de modo a melhorar o aprendizado. Esse método de aprendizagem já havia sido descrito por Werbos (1974), porém, durante uma época onde as redes neurais não possuíam a devida atenção, o algoritmo permaneceu dormente por alguns anos (NEURAL NETWORKS – HISTORY, 2019). Yann LeCun *et al* (1989) demonstraram um modelo de rede neural que conseguia identificar códigos postais escritos à mão, através de imagens. Essa rede neural possuía uma diferença marcante dos modelos que a antecederam: ela era convolucional, o que significa que ao invés de designar um peso para cada pixel da imagem, diversos grupos de pixels eram formados e analisados individualmente, reduzindo o poder computacional necessário para o aprendizado da rede. Schmidhuber (1992) descreveu o processo que hoje é chamado de *max-pooling*, que torna mais fácil o reconhecimento de imagens, visto que diminui o tamanho e extrai informações importantes destas, reduzindo o poder computacional necessário para sua análise. Hochreiter e Schmidhuber (1997) publicaram o trabalho “Long Short-Term Memory”, que descrevia um modelo de rede neural onde os neurônios não possuem conexões somente com os neurônios da camada seguinte, mas também com a camada anterior e consigo mesmos, oferecendo mais uma etapa de feedback e melhorando o aprendizado da rede.

Muitas pesquisas foram publicadas nos anos que se seguiram, aumentando ainda mais o repertório de algoritmos e modelos de redes neurais disponíveis para profissionais do ramo do aprendizado de máquinas. Nos dias atuais, a inteligência artificial e principalmente as redes neurais estão no topo dos assuntos de interesse de pesquisadores e diversas inovações surgem de forma extremamente rápida (AGGARWAL, 2018).

2 ESTRUTURA DAS REDES NEURAIS

Os modelos de redes neurais encontrados atualmente possuem grande variação e bastante espaço para ajuste de parâmetros. As opções de customização são muitas e, por isso, as funcionalidades são muito versáteis. Apesar disso, a construção da arquitetura das redes neurais segue uma lógica e pode ser explicada de forma simplificada. (VASILEV *et al.*, 2019).

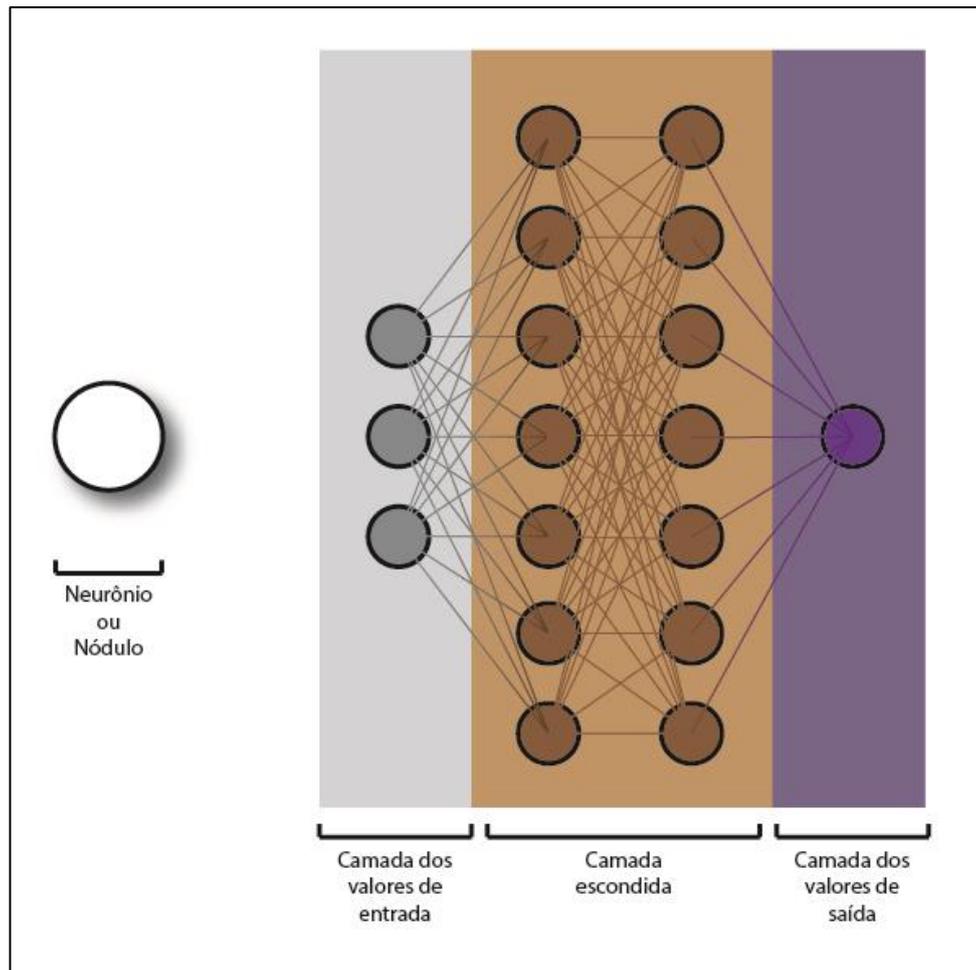
2.1 ESTRUTURA BÁSICA

A unidade básica de uma rede neural é um neurônio, também chamado de nódulo. Esses nódulos podem ser de 3 tipos:

- De entrada, se receberem os valores de entrada da base de dados;
- De saída, se fizerem parte da última camada de nódulos da rede neural e retornar um valor de saída;
- Da camada escondida, se não fizerem parte das duas categorias acima e estiverem entre elas.

À medida que as estruturas dos modelos ficam mais complexas, esses neurônios acabam apresentando diferenças, porém a ideia se mantém a mesma: eles se conectam entre si para formar relações entre os valores de entrada e saída, mimetizando uma rede de neurônios biológicos (VASILEV *et al.*, 2019).

Figura 1 – Esquemática de estrutura básica de uma rede neural profunda.



Fonte: o próprio autor.

2.2 TIPOS DE REDES NEURAIAS

Como dito anteriormente, a variabilidade das redes neurais é gigantesca, mas ainda assim é possível classificá-las de acordo com certos parâmetros. Alguns desses parâmetros são a necessidade ou não de supervisão durante o treinamento, a estrutura das conexões dos nós das redes neurais e a quantidade de camadas nos modelos (DATA SCIENCE ACADEMY, 2019). Usando essas características, as seguintes classificações são possíveis:

2.2.1 Redes neurais supervisionadas

Essas redes neurais aprendem através da comparação dos resultados gerados pelos nós da camada de saída com os valores reais dessas variáveis, ou seja, a base de dados

utilizada precisa ter valores de entrada, que são utilizados para a rede neural encontrar os padrões de conexão escondidos entre esses valores, e valores de saída, para que a rede neural avalie sua eficácia utilizando-os. A “supervisão”, então, não se trata de uma constante avaliação feita pelo usuário da rede neural na etapa de treinamento, mas sim de uma característica da base de dados utilizada (GOYAL; PANDEY; JAIN, 2018).

2.2.2 Redes neurais não-supervisionadas

Essas redes neurais possuem uma aplicabilidade um pouco mais limitada que o grupo anterior, sendo utilizadas para classificar diversas variáveis em diferentes grupos, de acordo com a semelhança entre si. Assim, os dados utilizados não necessariamente precisam ter valores de entrada e saída, apenas precisam possuir variáveis que tenham uma certa relação entre si. O único parâmetro referente à base de dados que precisa ser definido pelo usuário é o número de grupos classificatórios desejado e isso nem sempre é necessário em todas as redes neurais não-supervisionadas (GOYAL; PANDEY; JAIN, 2018).

2.2.3 Redes neurais do tipo *feed-forward*

Essa categoria de redes é uma das mais simples e nela se enquadram os modelos que apresentam nódulos que apenas causam mudanças nos nódulos da camada seguinte, sem presença de loops ou conexões com os nódulos da camada anterior. Geralmente, os pesos dos neurônios são ajustados utilizando o algoritmo de *back propagation*, ou alguma variação deste (GOYAL; PANDEY; JAIN, 2018).

2.2.4 Redes neurais do tipo *recurrent*

Essa categoria abrange as redes que possuem nódulos que realizam feedback consigo mesmos ou com os neurônios de camadas anteriores. Essas conexões em loops podem acontecer a cada ciclo de treinamento ou em um intervalo fixo de ciclos, dependendo do modelo da rede. Essas redes neurais são normalmente utilizadas quando o contexto é importante para os valores de saída, ou seja, um valor de entrada isolado não é suficiente para uma análise adequada dos padrões escondidos que geram o valor de saída. Um exemplo disso é o reconhecimento de texto e posterior tradução deste, onde uma palavra isolada como valor de entrada não é o suficiente

para determinar a tradução desta palavra, no valor de saída, sendo necessária uma correlação com as outras palavras da frase para uma adequada tradução (DATA SCIENCE ACADEMY, 2019).

2.2.5 Redes neurais convolucionais

Redes neurais convolucionais são parecidas com as do tipo *feed-forward*, no sentido de passarem os valores de saída de cada nódulo apenas para os nódulos seguintes, mas se diferenciam delas por ter algumas camadas extras após a camada de valores de entrada. Essas camadas podem ser compostas por *kernels* ou por camadas de *pooling*. Essas redes neurais são muito utilizadas para reconhecimento de imagem, identificação de objetos em vídeos e outras funções que requeiram uma base de dados que contenham imagens ou um grande número de valores de entrada (VASILEV *et al.*, 2019).

Kernels são algoritmos que processam os valores de entrada, para extrair informações que tenham relação com os valores de saída desejados. Eles são necessários, pois quando são utilizadas bases de dados com um grande número de valores de entrada (pixels de uma imagem, por exemplo), não só o poder computacional necessário se torna muito grande, como também os valores de entrada não significativos atrapalham o desempenho da rede neural, então ter um algoritmo que extraia as informações de interesse dessa base de dados é fundamental (AGGARWAL, 2018).

Camadas de *pooling* simplificam os valores de saída de cada camada, tentando preservar suas características a ponto de não diminuir o desempenho da rede neural. Nessas camadas, normalmente são utilizados algoritmos com funções não-lineares e que tentam manter as informações extraídas por um *kernel* na camada anterior (DATA SCIENCE ACADEMY, 2019). Camadas de *pooling* são essenciais em modelos de redes neurais que usam os pixels de imagens de alta resolução como valores de entrada (AGGARWAL, 2018).

2.3 FUNÇÕES DE CUSTO

Funções de custo são essenciais para o funcionamento de uma rede neural, pois são responsáveis por avaliar o quão longe os valores de saída gerados na última camada estão dos valores de saída reais presentes na base de dados. Essa avaliação é necessária para verificar o desempenho da rede neural e direcionar a modificação dos pesos dos nódulos para uma

configuração que alcance um melhor desempenho no próximo ciclo de treinamento. Existem várias funções de custo e cada uma é adequada a um tipo de base de dados e finalidade da rede neural (AGGARWAL, 2018). As mais comuns e normalmente utilizadas estão listadas abaixo.

2.3.1 Função de custo *mean squared error*

Essa função de custo é normalmente utilizada em redes neurais de regressão e/ou que utilizam bases de dados com valores de saída grandes e não-binários. O resultado dessa função consiste numa média da diferença do valor de saída previsto pela rede neural e o valor de saída real presente na base de dados de treinamento e validação, elevada ao quadrado. O valor perfeito para essa função seria zero, significando que não existe diferença entre o valor previsto e o valor real, porém isso não é viável numa situação real. A propriedade da diferença ser elevada ao quadrado é útil, pois valores de saída previstos que se distanciam demais dos valores reais são exacerbados e representados em números elevados, mostrando que a rede neural não apresenta um desempenho adequado (AGGARWAL, 2018). A fórmula da função *mean squared error* é:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2$$

Onde n é o número de valores de saída, y é o valor de saída real e y' é o valor de saída previsto.

2.3.2 Função de custo *mean absolute error*

Essa função de custo é muito semelhante à função *mean squared error*, se diferenciando apenas pelo fato da diferença do valor de saída previsto pela rede neural e o valor de saída real presente na base de dados de treinamento e validação não ser elevada ao quadrado. Desta forma, a função é definida matematicamente assim:

$$MAE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)$$

É uma função útil para situações onde é necessário saber exatamente quão distante estão os resultados previstos dos valores reais, porém ela não é muito utilizada como função de custo.

2.3.3 Função de custo *binary cross-entropy*

A *binary cross-entropy* é uma função muito utilizada para problemas de classificação binários, por exemplo, uma rede neural responsável por identificar se uma imagem contém ou não um cachorro, ou seja, que contém um valor de saída de um ou zero (VASILEV *et al.*, 2019). A função é definida da seguinte forma:

$$BNE = -\frac{1}{n} \sum_{i=0}^n [y_i * \log(y'_i) + (1 - y_i) * \log(1 - y'_i)]$$

Onde n é o número de valores de saída, y é o valor de saída real e y' é o valor de saída previsto.

Podemos notar, então, que os valores que a função retornará estarão entre 0 e 1, sendo que 0 significa o menor valor possível de diferença entre o valor de saída previsto e o valor de saída real e 1 significa o contrário, com o maior erro possível (VASILEV *et al.*, 2019).

2.3.4 Função de custo *multi-class cross-entropy* ou *category cross-entropy*

Essa função é muito parecida com a *binary cross-entropy*, com a diferença de que ela consegue avaliar não somente um valor de saída binário, mas vários deles, fazendo uma média do erro de cada um. É extremamente útil para redes neurais classificatórias que possuem como valores de saída mais do que uma opção de classificação, por exemplo, um modelo que tem como o objetivo classificar a espécie de um animal em uma imagem. A fórmula matemática para essa função de custo é basicamente a mesma da *binary cross-entropy*, com a diferença de que é feita uma média com os diferentes resultados para cada valor de saída (AGGARWAL, 2018).

2.4 FUNÇÕES DE ATIVAÇÃO

Funções de ativação são responsáveis por transformar os valores de saída dos nódulos das redes neurais, sejam eles da camada de entrada, da camada escondida ou da camada de saída. Essas funções são importantes para atribuir propriedades não lineares à rede neural. A matemática por trás dos nódulos, sem uma função de ativação presente, é uma função polinomial de grau um e, portanto, pode não ser capaz de representar as complexas conexões que as variáveis possuem no mundo real. Além disso, uma rede neural composta somente por

diversas camadas de funções lineares nada mais é do que uma função linear extensa, sendo nada diferente dos modelos de regressão já existentes na literatura (VASILEV *et al.*, 2019).

Sendo assim, as funções de ativação são responsáveis por conferir às redes neurais a habilidade de encontrar padrões complexos em bases de dados extensas, multidimensionais e não-lineares (VASILEV *et al.*, 2019).

2.4.1 Função de ativação *step*

Essa é uma das funções de ativação mais simples, pois sua execução depende mais de ferramentas de programação do que de artifícios matemáticos. Ela processa os valores de saída e retorna o número um quando o valor ultrapassa um limite pré-estabelecido e zero quando o valor não ultrapassa esse limite, portanto, é uma função de ativação binária. Dito isso, é uma função útil apenas para problemas de classificação que possuem apenas uma classe e as duas possíveis saídas são um ou zero. Essa função é demasiadamente simples e atualmente está em desuso, visto que existem diversas opções superiores à disposição dos usuários de redes neurais (AGGARWAL, 2018).

2.4.2 Função de ativação *sigmoid*

É uma função parecida com a função *step*, mas possui a diferença de não ser binária, podendo retornar qualquer número entre zero e um (AGGARWAL, 2018). A transformação do valor de saída do nóculo em número nesse intervalo se dá pela seguinte expressão matemática:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Essa transformação matemática possui a característica de retornar valores muito próximos de um quando o valor de saída é muito maior que zero e de retornar valores muito próximos a zero quando o valor de saída é muito menor que zero (VASILEV *et al.*, 2019). Valores de saída que se mantêm no intervalo de -2 a 2 são normalizados e transformados em números entre zero e um, seguindo um gradiente suave.

Essa função é mais avançada que a função *step*, porém é pouco utilizada por ocasionar diversos problemas, como o problema da dissipação do gradiente, explicado mais abaixo, e por exigir um grande poder computacional (AGGARWAL, 2018).

2.4.3 Função de ativação *tanh*

É extremamente similar à função *sigmoid*, com a única diferença de retornar valores que variam de -1 a 1. Fora isso, possui as mesmas propriedades matemáticas e ocasiona os mesmos problemas, razão pela qual também não é muito utilizada em redes neurais modernas (AGGARWAL, 2018). A expressão matemática da função é:

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

2.4.4 Função de ativação *softmax*

Essa função é geralmente utilizada na última camada de uma rede neural, pois tem uma propriedade interessante: ela retorna uma distribuição de probabilidade de todos os valores de saída da camada em que é utilizada. Ela faz isso, ao transformar cada valor de saída em um número contido no intervalo de zero a um, sendo que a soma de todos os valores de saída da camada é igual a um. O valor retornado para cada variável de saída é equivalente à proporção daquele número para a soma de todos as variáveis, significando que o maior valor de saída é também o valor de saída mais provável. A função *softmax* é extremamente útil para redes neurais de classificação com múltiplas categorias, visto que, diferente de um modelo que utiliza uma função *step*, por exemplo, ela apresenta a acurácia estimada da classificação (VASILEV *et al.*, 2019). Matematicamente, ela é descrita da seguinte forma:

$$\text{softmax}(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{z_k}} \text{ para } j = 1, \dots, K.$$

Além disso, a função *softmax* não apresenta os mesmos riscos de ocorrência de erros que as funções *sigmoid* e *tanh*, sendo extensamente utilizada nas redes neurais modernas (AGGARWAL, 2018).

2.4.5 Função de ativação *ReLU*

Chamada assim, a função *ReLU*, ou *Rectified Linear Units*, é uma das funções de ativação mais populares da atualidade, sendo utilizada frequentemente tanto em problemas de classificação quanto de regressão. Assim como a função *step*, ela possui propriedades que são mais derivadas da programação de computador do que da matemática. A *ReLU* retorna um valor

igual a zero, se o valor de saída for menor que zero e, para valores que não seguem essa regra, ela não gera nenhuma modificação (AGGARWAL, 2018). Essa função, então, opera de maneira semelhante aos neurônios biológicos, visto que o potencial de membrana de uma célula neuronal não possui nenhum limite, mas, caso não atinja o potencial de ação, nada acontece (GOYAL; PANDEY; JAIN, 2018).

Por ser tão simples, essa função não precisa de um grande poder computacional para ser utilizada, diferente da função *sigmoid* e *tanh*, por exemplo. Além disso, também diferentemente dessas funções citadas, a *ReLU* não costuma propiciar o aparecimento de erros como o problema da dissipação de gradiente e da explosão de gradiente (GOYAL; PANDEY; JAIN, 2018).

Um problema comum na utilização dessa função é o chamado problema do *ReLU* moribundo, onde alguns nódulos, por retornarem um valor igual a zero após a aplicação da função *ReLU*, acabam ficando permanentemente inativos, visto que qualquer peso que multiplique esse neurônio não conseguirá modificar o valor. Esse problema afeta a performance da rede neural e, se um número suficiente de nódulos for desativado, pode fazer com que a acurácia permaneça a mesma independente do número de ciclos de treinamento subsequentes (VASILEV *et al.*, 2019).

2.4.6 Função de ativação *SELU*

A função de ativação *Scaled Exponential Linear Unit (SELU)* é uma das funções mais recentes apresentada nesse trabalho. Ela é definida matematicamente da seguinte forma:

$$SELU(x) = \lambda \begin{cases} x & \text{se } x > 0 \\ \alpha e^x - \alpha & \text{se } x \leq 0 \end{cases}$$

Sendo que:

$$a \approx 1.6732632423543772848170429916717$$

$$\lambda \approx 1.0507009873554804934193349852946$$

Ela possui a particularidade de ser capaz de tornar uma rede neural *self-normalizing* quando é utilizada, ou seja, ela torna desnecessária a etapa de normalização da base de dados antes do treinamento da rede.

A normalização da base de dados nada mais é do que a modificação de cada um dos valores de entrada, de modo que deles seja subtraída a média e o resultado seja dividido pelo desvio padrão dessas variáveis. A normalização deixa cada um dos grupos de variáveis da base de dados com uma distribuição normal (HANSEN, 2020).

2.4.7 Função de ativação *Leaky ReLU*

Essa função foi criada para corrigir o problema do *ReLU* moribundo, citado anteriormente. O funcionamento dela é basicamente o mesmo, com a única diferença sendo o limite mínimo dos valores de saída da função. A *Leaky ReLU*, ao invés de retornar zero para valores abaixo de zero, retorna um número extremamente pequeno, de modo a evitar que aquele nóculo seja desativado permanentemente, independente dos pesos que o multipliquem, mas ainda assim impedindo que os nóculos possuam uma grande influência nos valores de saída da camada final (VASILEV *et al.*, 2019).

2.5 ALGORITMOS DE OTIMIZAÇÃO

Algoritmos de otimização tentam minimizar ou maximizar alguma função. A função utilizada por esses algoritmos, no caso das redes neurais, contém as funções executadas pelos vários nóculos da arquitetura da rede e possui como resultado o custo, logo, os algoritmos de otimização possuem a importante tarefa de minimizar o custo dos modelos onde são aplicados (VASILEV *et al.*, 2019).

2.5.1 *Gradient Descent*

Um gradiente, no contexto de algoritmos de otimização, é um campo de vetores formado por derivadas parciais da função a ser otimizada. Esse gradiente é representado por uma matriz jacobiana. Uma derivada pode ser descrita como a taxa de variação instantânea de y em razão de x , então um gradiente nada mais é do que uma derivada de uma função com várias variáveis.

O algoritmo de otimização *gradient descent* utiliza a informação obtida através de um gradiente para realizar modificações nas funções presentes nos nóculos e causar uma redução no custo do modelo da rede neural, utilizando uma função de custo para verifica-lo. De maneira geral, a cada ciclo de treinamento, o algoritmo modifica a função de modo que as variáveis “se movimentem” pelo gradiente em uma direção fixa e por um valor fixo (chamado de taxa de aprendizado), até que um incremento nessa mesma direção não cause mais uma redução no custo, atingindo um local no plano multidimensional do gradiente que chamamos de *local minima*. Esse local é chamado de *minima*, pois representa um valor de x onde um incremento de qualquer valor menor ou igual que a taxa de aprendizado, para qualquer direção

(modificando positivamente ou negativamente qualquer uma das variáveis da função), causaria um aumento em y (o custo, no caso de uma rede neural). Ele é chamado de *local*, pois não é possível saber se não existe outro valor de x em algum lugar do gráfico que seja correspondente a um y menor, pois as modificações estão limitadas por incrementos (ou decréscimos) do valor da taxa de aprendizado (GOYAL; PANDEY; JAIN, 2018).

2.5.2 *Stochastic Gradient Descent*

Enquanto o algoritmo *gradient descent* utiliza como base o custo médio de todas as predições, o *Stochastic Gradient Descent (SGD)* utiliza o cálculo do custo para cada uma das amostras da base de dados de treinamento, resultando em modificações específicas para cada uma delas. Exatamente por isso, esse algoritmo costuma apresentar uma grande variação no custo das redes neurais que o utilizam, mas acaba aumentando a chance de achar um *local minima* cujo custo seja menor (GOYAL; PANDEY; JAIN, 2018).

2.5.3 *Adagrad*

Nesse algoritmo, a taxa de aprendizado é automaticamente adaptada baseada na frequência dos parâmetros contidas nas funções dos nódulos. Se um parâmetro aparece frequentemente e possui grande influência no custo do modelo, a taxa de aprendizado para a função daquele nódulo será menor do que para uma que possua parâmetros que aparecem poucas vezes. A frequência e influência de cada parâmetro é calculada pela soma de seus gradientes de ciclos de treinamento anteriores, elevados ao quadrado.

Dessa forma, uma das principais vantagens desse algoritmo é a desnecessidade de ajustar a taxa de aprendizado. Devido ao algoritmo ajustar esta taxa para cada uma das amostras utilizadas no treinamento, ele se torna útil em casos onde a base de dados de treinamento não é muito extensa (VASILEV *et al.*, 2019).

2.5.4 *Adam*

Adam significa *Adaptive Moment Estimation* e é chamado assim, pois além de utilizar a frequência e influência dos parâmetros de cada função, ele utiliza a média dos resultados dos

gradientes de ciclos de treinamentos passados para melhorar a taxa de aprendizado para cada amostra. Atualmente é uma das funções de otimização mais utilizadas e uma das que convergem para o menor custo em menos ciclos de treinamento (VASILEV *et al.*, 2019).

2.5.5 *RMSPprop*

Esse algoritmo é bastante semelhante ao *Adagrad*, pois utiliza a soma de gradientes de ciclos de treinamento anteriores, elevados ao quadrado, para adaptar a taxa de aprendizado para cada amostra e cada função. Ele diverge, no entanto, devido a utilização de apenas gradientes correspondentes a um número fixo de ciclos de treinamento anteriores, de modo que essa soma não aumente infinitamente à medida que a rede neural aprende (AGGARWAL, 2018).

2.6 PROBLEMAS COMUNS

Embora os atuais modelos de rede neurais existentes sejam bastante diferentes entre si e apresentem, cada um, uma estrutura e funcionamento únicos, alguns problemas são factíveis de acontecer na maioria deles. Esses contratempos possuem sua origem no modo como as redes neurais funcionam e nos algoritmos utilizados por elas. Atualmente, diversas formas de evitá-los foram descritas, porém o seu aparecimento continua sendo uma preocupação no momento do planejamento da arquitetura de redes neurais (AGGARWAL, 2018).

2.6.1 Problema da dissipação de gradiente

Como os neurônios das camadas mais avançadas dependem de modificações nas camadas que os antecedem, algo comum de acontecer é que, devido ao método de *back propagation*, os neurônios de camadas iniciais sofrem mudanças cada vez menores, pois uma pequena modificação em seus valores de saída gera grandes modificações nos neurônios de camadas avançadas. O problema da dissipação de gradiente ocorre quando as modificações geradas pelo algoritmo de *back propagation* (ou semelhantes) nos neurônios iniciais se tornam tão pequenas que o número de ciclos de treinamento necessários para uma melhora na eficácia se torna grande demais para ser utilizável em um problema na vida real (DATA SCIENCE ACADEMY, 2019).

2.6.2 Problema da explosão do gradiente

Esse problema é o contrário da dissipação de gradiente, fazendo com que a rede neural gere valores de saída estranhos e desconexos a cada ciclo de treinamento. Existem vários motivos para ele ocorrer, mas o mais frequente consiste em pequenas modificações nas camadas iniciais da rede neural, que devido às conexões com as camadas subsequentes, causam grandes modificações nos valores de saída. As funções de ativação *sigmoid*, *tanh*, e suas variações geralmente estão presentes quando esse problema ocorre. Diferente do problema de dissipação de gradiente, o da explosão não é muito frequente (DATA SCIENCE ACADEMY, 2019).

2.6.3 *Overfitting*

Essa complicação, diferente das anteriormente citadas, está relacionada muito mais à base de dados do que à estrutura da rede neural. O *overfitting* ocorre quando a rede neural só apresenta resultados adequados quando aplicada à base de dados utilizada em seu treinamento, não sendo capaz de generalizar o aprendizado para valores de entradas para o qual não foi treinada ainda, mas que ainda seguem o padrão da base de dados de treinamento. Isso geralmente ocorre quando a base de dados de treinamento não possui variabilidade e/ou quando o número de dados é muito pequeno (DATA SCIENCE ACADEMY, 2019).

3 USO DE REDES NEURAIIS NA ATUALIDADE

À medida que a popularidade das redes neurais aumentou nos últimos anos, a tecnologia foi aplicada a áreas cada vez mais diversificadas. Como a inteligência artificial busca, até certo nível, mimetizar a inteligência humana, é de se esperar que as aplicações dessa técnica sejam relacionadas com áreas do conhecimento humano que derivam ou dependem do raciocínio lógico, memória e reconhecimento de padrões.

3.1 MÉTODO DE BUSCA DOS ARTIGOS

Os artigos analisados e descritos a seguir foram obtidos a partir da busca nas seguintes bases de dados eletrônicas: Portal de Periódicos da Coordenação de Aperfeiçoamento de

Pessoal de Nível Superior (CAPES), PubMed, Scielo, Scopus, ScienceDirect e Web of Science, através da utilização de diferentes palavras chaves nos idiomas inglês e português. As palavras chaves utilizadas foram “neural network animal production” “neural network animal science” “neural network animal” “neural network veterinary” “neural network swine OR cattle OR poultry OR broiler OR pork OR chicken OR sheep OR ovine” e os períodos de publicação utilizados compreenderam os anos no intervalo de 1980 a 2019.

3.2 EXEMPLOS DE REDES NEURAIIS APLICADAS A DIVERSAS ÁREAS

Manter estradas em boas condições é algo vital para a segurança dos motoristas que transitam nessas vias. Para que manutenções sejam feitas frequentemente, é necessário que haja o constante monitoramento do estado das estradas, de modo que defeitos sejam detectados o mais cedo possível. Detectar esses defeitos através de mão-de-obra humana é caro e demorado, então tecnologias que automatizem esse processo são essenciais. É o exemplo da rede neural convolucional proposta por Lei Zhang *et al.* (2016), que alcançou uma acurácia de 86,96% em classificar imagens de pedaços de estradas que apresentavam ou não rachaduras, utilizando fotos de um smartphone que poderia, numa situação mais aplicável ao mundo-real, ser um drone.

O monitoramento de condição é utilizado para avaliar o estado de máquinas e detectar componentes que podem estar apresentando defeitos. Esse tipo de monitoramento é essencial para reduzir custos operacionais e aumentar a vida-útil das máquinas. Uma maneira de fazer esse monitoramento é através da análise de vibração de diferentes elementos das máquinas. Para isso, a frequência de vibrações é monitorada e o espectro dessas frequências é analisado por um profissional para que haja a detecção de uma mudança no padrão de vibração da máquina. Esse processo de análise possui vários desafios e diversas situações podem dificultar ou mascarar completamente mudanças nas vibrações, de modo que o profissional não perceba uma falha. Uma maneira de remediar isso é através do uso de uma rede neural treinada com esses tipos de situações, que consiga perceber sutilezas que passariam despercebidas por um ser humano. Foi o que fizeram Olivier Janssens *et al.* (2016), com um modelo que alcançou uma acurácia de 93,61% na tarefa de detectar falhas em mecanismos, através da análise de seu padrão vibracional.

Uma demonstração das habilidades de reconhecimento de padrões é encontrada na rede neural criada por Dan Ciresan *et al.* (2012), que alcançou uma acurácia de 98,80% na tarefa de

classificar 43 tipos de placas de trânsito. Outra interessante aplicação de redes neurais é para o reconhecimento de padrões de ondas de eletrocardiograma. Osowski e Linh (2001) conseguiram uma acurácia combinada de 96,06% na classificação de sete padrões diferentes de ondas de ECG, utilizando uma rede neural inovadora.

O uso de redes neurais que lidam com base de dados de imagens é extremamente útil na área médica. Por exemplo, durante um diagnóstico de citometria, um citopatologista olha uma amostra por um microscópio e faz o diagnóstico baseado em um processo de reconhecimento de padrão que acontece no nível subconsciente. Logo, passar o conhecimento que permite esse reconhecimento de padrões para profissionais iniciantes na área é complicado. Uma rede neural que aprende utilizando uma base de dados de imagens de citometria e discrimina entre lesões benignas e malignas é possível e foi implementada por pesquisadores, alcançando uma taxa de acerto de 98% (KONTORAVDIS *et al.*, 1998).

3.3 EXEMPLOS DE REDES NEURAI APLICADAS À ÁREA DE PRODUÇÃO ANIMAL

O uso de redes neurais aplicado à produção animal não é algo recente. Os trabalhos na área datam desde a década de 90, o que mudou com o passar do tempo foram os modelos utilizados e a eficácia de tais modelos (BRETHOUR, 1994; KORTHALS; HAHN; NIENABER, 1994; ROUSH; KIRBY; CRAVENER; WIDEMAN, 1996; ROUSH; KIRBY; CRAVENER; WIDEMAN, 1997)

Em 1994, uma rede neural foi desenvolvida com o propósito de estimar o escore de marmoreio em gado vivo, utilizando imagens de ultrassom que eram submetidas a uma análise em uma rede neural do tipo *feedforward*. A rede neural em questão utilizava um sistema de classificação em duas categorias e atingiu uma acurácia de 85,7% (BRETHOUR, 1994). Ainda no mesmo ano, foi demonstrado que uma rede neural era capaz de regular a temperatura ambiente de um criadouro de suínos de modo a manter uma taxa de ganho de peso corporal diário constante, na faixa dos 0,78 kg/dia, porém apresentava problemas na faixa dos 0,70 kg/dia. Os autores definiram que uma base de dados mais detalhada seria necessária para melhorar as previsões do modelo (KORTHALS; HAHN; NIENABER, 1994).

Num ambiente controlado e utilizando quinze variáveis fisiológicas como entrada, cientistas tentaram prever a presença ou não de ascite em frangos de corte, utilizando uma rede neural com apenas uma camada escondida de dezesseis nódulos. O número de amostras utilizada no treinamento foi de 51 animais, o que é considerado baixo, mas apesar disso, para

os 97 animais utilizados para a verificação da acurácia da rede, apenas três tiveram resultados não condizentes com a realidade, sendo dois falsos positivos e um falso negativo (ROUSH; KIRBY; CRAVENER; WIDEMAN, 1996). Um ano após, com o mesmo propósito, mas dessa vez utilizando fatores fisiológicos minimamente invasivos como valores de entrada, os mesmos pesquisadores do trabalho anterior utilizaram seis variáveis para tentar classificar a presença de ascite em frangos de corte. Com uma base de dados de treinamento maior, totalizando 117 aves, o modelo de rede neural artificial atingiu uma sensibilidade de 95% e uma especificidade de 88%, nas 36 amostras utilizadas para validação (ROUSH; KIRBY; CRAVENER; WIDEMAN, 1997). No mesmo ano, pesquisadores utilizaram imagens multiespectrais, captadas por câmeras com filtros cujos comprimentos de onda variavam de 540 nm a 700 nm, para classificar carcaças de frangos quanto à presença ou não de lesões. O modelo com uma camada escondida contendo 512 nódulos, treinado com uma base de treinamento contendo 60 amostras e validado utilizando uma base com 30 amostras, apresentou uma acurácia máxima de 93,3% (PARK; CHEN; NGUYEN, 1997).

No final da década de 90, dois pesquisadores publicaram um trabalho descrevendo uma rede neural capaz de prever a concentração de aminoácidos em cinco diferentes tipos de alimentos, utilizando como valores de entrada seis variáveis de uma análise proximal. A acurácia da rede neural foi comparada com um modelo clássico de regressão linear e em todas as tentativas de predição, a acurácia da rede neural foi maior. Além disso, os coeficientes de determinação para os sete aminoácidos avaliados, na maioria dos casos, ultrapassavam 0,9 (CRAVENER; ROUSH, 1999). Tentando verificar a eficácia das redes neurais para análises de imagem e classificação de conforto térmico em suínos, H. Xin utilizou diferentes algoritmos de processamento de imagem e, com a combinação destes, desenvolveu um modelo que alcançava uma acurácia de 90% na tarefa de classificar o conforto térmico de suínos em “com frio”, “confortável” ou “aquecido” (XIN, 1999).

Com a finalidade de classificar diferentes tipos de mastite, a partir de informações individuais de cada vaca leiteira, coletadas pela *Dairy Herd Improvement Association* (DHIA), além de um questionário abordando dados de manejo e higiene aplicado a funcionários e veterinários em contato com os animais, Heald *et al.* (2000) desenvolveu um modelo de rede neural que alcançou acurácias variando entre 57 a 71%, na tarefa de classificar os animais em quatro categorias: animais sem mastite, animais com mastite contagiosa, animais com mastite ambiental e animais com outro tipo de mastite. Realizando algo que não havia ainda sido explorado nos trabalhos de inteligência artificial voltados à produção animal, Yang, Lacroix e Wade (2000), decidiram utilizar uma rede neural não para tentar prever a ocorrência de mastite

em um animal, mas sim definir quais variáveis possuíam maior influência no resultado desta. Ao retirar, uma por uma, as variáveis utilizadas como valores de entrada, os pesquisadores conseguiram encontrar quais delas possuíam a maior conexão com o valor de saída, que nesse caso era a presença ou não de mastite. Os valores de entrada utilizados faziam parte de dois distintos grupos: um grupo de variáveis associadas ao valor genético e conformação do animal e de seus progenitores e um grupo de fatores associados ao leite. Os pesquisadores descobriram que os fatores associados ao leite possuíam muito mais influência na predição dos diferentes modelos de redes neurais testados e que, quanto ao grupo de valores associados à conformação dos animais, as variáveis referentes ao animal afetavam mais negativamente a acurácia das predições do que as variáveis genéticas referentes aos seus pais. Além disso, os pesquisadores concluíram que, devido à alta acurácia alcançada quando utilizados os dois grupos de variáveis como valores de entrada, a aplicação de uma rede neural para prever a presença ou não de mastite em um rebanho era algo viável e que deveria ser explorado. No mesmo ano, dessa vez para aplicações voltadas à avicultura, um trabalho publicado relatava uma rede neural que tinha o objetivo de prever a temperatura corporal de aves submetidas a diferentes mudanças de temperatura do ambiente, mas apenas com uma correlação de 69% com os valores reais, o que, segundo os próprios autores, era esperado, visto que foram utilizadas apenas três aves com quinze medições cada (LACEY; HAMRITA; MCCLENDON, 2000).

Utilizando uma abordagem diferente das de trabalhos anteriores, Sanzogni e Kerr (2001) utilizaram dados gerais de diferentes fazendas de bovinos leiteiros da Austrália, recolhidos através de questionários feitos aos produtores, para estimar a produção total de leite das propriedades. Com uma correlação que chegou a 90,9% para as fazendas da região Darling Downs da Austrália, a rede neural se provou uma ótima alternativa para fazendas que não acompanham a produção total de seus animais. As variáveis utilizadas como valores de entrada incluíam dados da alimentação, área disponível e número de animais. As redes neurais já citadas utilizavam números e imagens nas bases de dados para procurar por padrões, mas Chedad *et al.* (2001) decidiram inovar e utilizaram o áudio de um ambiente com porcos para treinar uma rede neural artificial na tarefa de identificar tosses. A justificativa do trabalho seria a possibilidade de utilizar a presença das tosses para identificar doenças respiratórias. Para isso, os pesquisadores utilizaram vários sons produzidos por diversos porcos em um ambiente controlado e classificaram eles mesmos a origem do som. A rede neural apresentou uma ótima performance, com uma chance de 91,9% de classificar corretamente os sons, a partir de um arquivo de áudio. Com o intuito de verificar o número de amostras necessárias na base de dados para alcançar uma acurácia de predição de 100% na tarefa de determinar a ocorrência de

síndrome de hipertensão pulmonar em frangos de corte utilizando dados de velocidade de crescimento, pesquisadores utilizaram de 46 a 102 animais, em quatro diferentes grupos, cada um correspondente a uma raça de frango de corte, que tiveram seus pesos mensurados uma vez ao dia. Os pesquisadores concluíram que utilizando uma rede neural, era possível diagnosticar corretamente a ocorrência de síndrome de hipertensão pulmonar utilizando dados de apenas 4 a 6 dias de crescimento, porém existia uma certa variação na acurácia para diferentes raças (ROUSH *et al.*, 2001).

Utilizando um nariz elétrico, chamado AromaScan, Sohn *et al.* (2003) desenvolveram uma rede neural capaz de não só identificar, mas também quantificar odores de efluentes provenientes de sítios de criação de suínos. A rede neural criada pelos pesquisadores usava dados capturados com o nariz elétrico para estimar a concentração de odores provenientes de dejetos suínos, alcançando valores de saída que apresentavam uma correlação de 0,984 com os valores reais. Em um trabalho que buscava elucidar o efeito de diferentes tipos de alimentação na composição da microbiota do trato digestivo de bovinos, uma rede neural artificial foi utilizada como uma ferramenta para verificar a correlação entre a descrição da colônia bacteriana em uma placa de ágar Rogossa MRS e a espécie da bactéria, que foi verificada através de uma análise de 16S rDNA. Os pesquisadores descobriram que era possível prever a espécie apenas pela descrição da colônia em mais de 80% dos casos, sendo que a acurácia de predição para colônias de *Streptococcus bovis* chegou a 96,1% (KRAUSE *et al.*, 2003).

Torres, Hervás e Amador (2005) desenvolveram uma rede neural capaz de prever a curva de lactação de ovelhas, através de somente 15 amostras das primeiras lactações. A aplicação de redes neurais na produção animal não é limitada somente a espécies terrestres. Em 2005, pesquisadores utilizaram uma rede neural para prever a curva de crescimento de camarões, utilizando uma base de dados contendo 18 variáveis referentes aos camarões e ao ambiente em que estavam inseridos, durante uma semana. A acurácia da curva prevista por essa rede para uma base de dados de validação de 249 amostras foi de 98,3%, demonstrando o incrível potencial dessa tecnologia para a carcinicultura (YU; LEUNG; BIENFANG, 2005). Wang e Samarasingue (2005) descreveram em seu trabalho dois modelos de redes neurais que visavam a detecção de mastite, utilizando dados coletados por uma ordenhadeira robótica. Esses dados consistiam em mensurações da temperatura, volume e condutividade elétrica do leite. Se implementado no código fonte de um software de ordenhadeiras robóticas, esse algoritmo poderia auxiliar os produtores na detecção de mastites clínicas e subclínicas, de forma automatizada e pouco custosa. A impressionante acurácia alcançada por um dos modelos foi de 95% de detecção para vacas com mastite.

Semelhante ao trabalho publicado por Chedad *et al.* (2001), um grupo de pesquisadores decidiram testar a eficácia de uma rede neural em identificar a ocorrência de vocalizações em uma baía contendo uma porca e leitões se amamentando desta. Eles descobriram que a rede neural proposta era capaz de identificar vocalizações tanto da porca quanto dos leitões, com uma soma quadrática de erro menor que 0,1, oferecendo uma ferramenta valiosa para identificação de situações inerentes ao risco de redução do bem-estar de leitões (NÄÄS; CAMPOS; BARACHO; TOLON, 2008). Coletar dados do peso de suínos é um processo demorado e trabalhoso, que requer que os animais fiquem parados sobre uma balança por determinado tempo. Foi tentando encontrar uma solução para essa problemática que Wang, Yang, Winter e Walker (2008) desenvolveram uma rede neural capaz de estimar o peso de suínos através de filmagens que não necessitavam que os animais estivessem parados. O processo de coleta de dados consistia em um algoritmo desenvolvido pelos autores que automaticamente selecionava segmentos dos vídeos que não continham artefatos e eram confiáveis o suficiente para estimar o peso dos animais. O erro médio relativo alcançado pela rede neural foi de aproximadamente 3%, utilizando uma amostra de apenas 500 imagens provenientes de 39 suínos.

Em 2009, três pesquisadores publicaram um trabalho sobre uma rede neural capaz de prever com mais de 87% de certeza a curva de lactação de vacas holandesas baseado nos dados genéticos e produtivos de suas progenitoras. A base de dados utilizada possuía 6.095 registros de lactação, provenientes de 2.267 vacas (NJUBI; WAKHUNGU; BADAMANA, 2009). Nem sempre algoritmos de inteligência artificial demonstrarão bons resultados de predição, mesmo que a base de dados seja grande. Um exemplo disso pode ser encontrado no trabalho publicado por Ghandi, Raja, Ruhil e Kumar (2009), onde um índice de correlação de apenas 29,81% foi alcançado pela rede neural proposta pelos mesmos, que utilizava cinco variáveis referentes à vida produtiva de vacas da raça Sahiwal para estimar a produção de leite total, durante toda a vida desses animais. Unindo uma tecnologia de captura de imagens e uma de ciência computacional, pesquisadores da Universidade de León, na Espanha, utilizaram dados de uma espectroscopia de infravermelho por transformada de Fourier de gordura perirenal e gordura proveniente do omento da carcaça de ovinos para classificar, através do uso de uma rede neural, a alimentação que esses animais receberam em vida (OSORIO *et al.*, 2009). As redes neurais também são deveras úteis para o diagnóstico bacteriológico, como podemos constatar na publicação de Salle *et al.* (2010), que utilizou valores de entrada referentes a genes associados à virulência, lesões provocadas por inoculação em pintos, o índice de patogenicidade das amostras e a resistência antimicrobiana a 14 antibióticos de bactérias *Escherichia coli* isoladas

camas de aviários e lesões de frangos de corte para estimar o resultado de sete diferentes provas bioquímicas dessas bactérias.

Com o intuito de prever o peso dos leitões ao nascimento e o número de leitões mumificados, através da temperatura do ar e da frequência respiratória das matrizes, Pandorfi *et al.* (2011) desenvolveram uma rede neural artificial que utilizava o clássico método de *backpropagation*. A rede neural, apesar de somente utilizar duas variáveis de entrada e uma base de dados que continha dados de apenas 27 matrizes primíparas, obteve uma eficácia surpreendente, alcançando um índice de correlação de 93,89% na tarefa de prever o peso dos leitões e 95,26% na de prever o número de leitões mumificados. Em mais um trabalho relacionado a frangos de corte, pesquisadores propuseram e testaram uma rede neural que visava prever o conteúdo de energia e a composição de carcaças, utilizando apenas dados da alimentação dos animais enquanto estavam vivos (FARIDI *et al.*, 2011). Esse é um exemplo de situação onde as variáveis, pelo sabemos sobre a fisiologia dos animais em questão, estão obviamente relacionadas, mas definir quanta influência cada nutriente possui na composição final da carcaça é uma tarefa extremamente complexa, que necessitaria de uma grande dedicação se feita por humanos utilizando puramente matemática, porém que é feita em questão de minutos ou segundos por um algoritmo de inteligência artificial. Savegnano *et al.* (2011) testaram a eficácia de uma rede neural, comparando-a com modelos logísticos clássicos, na tarefa de prever a curva de produção de ovos média de 1.569 poedeiras White Leghorn, através de somente metade dos dados dessa produção. A rede neural proposta pelos pesquisadores se saiu extremamente bem na tarefa e alcançou uma correlação de 98,58% com a curva dos dados reais.

No ano de 2012, dois grupos diferentes de pesquisadores desenvolveram redes neurais que tinham basicamente o mesmo objetivo: prever valores genéticos de animais, utilizando como valores de entrada os dados produtivos dos animais. Um dos trabalhos utilizava dados recolhidos de rebanhos bovinos da raça Tabapuã e tentava prever, a partir das variáveis peso ao nascer, peso aos 205 dias e idade da mãe ao parto, o valor genético da característica VG_P205, referente ao potencial genético associado ao peso do animal aos 205 dias de idade (VENTURA *et al.*, 2012). O outro trabalho lidava com vacas da raça Holandesa e possuía uma imensa base de dados com 119.899 registros de lactação, que foi utilizada para tentar prever os valores genéticos para características de produção leiteira e produção de gordura através de dados como idade do animal, produção leiteira, média de produção do rebanho, valor genético para produção leiteira da mãe, entre outras variáveis, somando no total quinze valores de entrada (SHAHINFAR *et al.*, 2012). A rede neural de Ventura *et al.* (2012) não obteve muito sucesso,

alcançando um índice de correlação de somente 74% com os valores gerados pelo método BLUP, enquanto o modelo de Shahinfar *et al.* (2012) alcançou a impressionante acurácia de 93%, possivelmente devido à sua extensa base de dados e ao maior número de variáveis utilizadas como entrada. Tentando prever a produção leiteira de bovinos da raça pardo-suíça de um período de 305 dias, utilizando dados de apenas alguns dias de lactação, Gorgulu (2012) propôs uma rede neural que utilizava como valores de entrada a produção média de períodos de 305 dias, a idade, o número de lactações, a época do ano e a produção leiteira do primeiro ao quarto dia de lactação dos animais para estimar a produção leiteira para os 301 dias restantes. Com um modelo que possuía apenas uma camada escondida com somente 8 nódulos, foi possível alcançar uma correlação de 0,9 com os dados reais.

Buscando estimar a perda de calor durante um processo de compostagem, Boniecki *et al.* (2013) desenvolveu uma rede neural que utilizava como valores de entrada variáveis associadas à matéria orgânica utilizada na compostagem, que nesse caso seriam dejetos provenientes de sistemas de criação de suínos. Os dados analisados da matéria no biorreator foram temperatura, níveis de oxigênio, níveis de gás carbônico, níveis de minerais da matéria orgânica, volume, duração do processo, matéria seca e concentração de substâncias orgânicas. Com esses valores de entrada, a rede neural foi capaz de prever com uma acurácia de 98,71% a quantidade de calor perdido durante o processo de compostagem. Pesquisadores desenvolveram uma rede neural capaz de estimar o peso corporal de coelhos, através de medições de diferentes partos do corpo desses animais, alcançando uma correlação de até 71% com os valores reais e ultrapassando a performance de um modelo linear generalizado (SALAWU *et al.*, 2014).

O equivalente a uma impressão digital de seres humanos, pensando em individualidade e identificação, poderia ser o focinho, no caso dos animais. Foi pensando nisso que El-Henawy, Bakry e Hadad (2016) propuseram um modelo de rede neural capaz de identificar bovinos através de seus focinhos, utilizando apenas um certo número de fotos desses para previamente treinar a rede. Os resultados se mostraram promissores quando a rede neural foi treinada com poucos animais (três) e um grande número de fotos para cada um (60), atingindo uma probabilidade de identificação correta de 99,97%, porém essa acurácia baixava quando o número de animais era aumentado, sendo equivalente, por exemplo, a 72,38% quando utilizados 30 bovinos com 60 fotos cada um (EL-HENAWY; BAKRY; HADAD, 2016). Com o mesmo objetivo, sendo esse a identificação de bovinos através de técnicas de inteligência artificial e visão computacional, um grupo de pesquisadores desenvolveu uma rede neural capaz de reconhecer os animais utilizando vídeos contendo o torso destes (LI; GE; SHEN; LI, 2018).

Dos vídeos, eram extraídas aproximadamente 21.600 fotos, onde os animais apareciam em diferentes posições. A essas fotos eram aplicados diversos *kernels* para a extração de informações importantes, além de elas passarem posteriormente por camadas de *pooling*. A acurácia de identificação da rede neural, para o grupo de 30 animais testados, foi de 93,3%. Similarmente ao trabalho anterior, pesquisadores utilizaram não imagens do corpo, mas sim da face de bovinos, objetivando sua detecção e identificação. O trabalho de Yao *et al.* (2019) demonstrou que é possível alcançar uma acurácia de detecção e identificação de 98,3% e 94,1%, respectivamente, em uma base de dados contendo 300 animais. Os pesquisadores utilizaram várias arquiteturas de redes neurais convolucionais, como LeNet-5, AlexNet, VGGNet, ResNet50 e PnasNet-5, sendo essa última a que apresentou melhor acurácia. Combinando três tipos de redes neurais, duas classificatórias e uma de regressão, pesquisadores desenvolveram um complexo sistemas de monitoramento de comportamento alimentar de bovinos (KUAN *et al.*, 2019). Os autores utilizaram uma rede neural convolucional para confirmar a presença do animal no local de alimentação, pela detecção da face. Fotos da face do animal eram então utilizadas como valores de entrada para outra rede neural convolucional, com a função de identificar o animal. A rede neural de regressão era utilizada para estimar o tempo de permanência do animal no local de alimentação e o comportamento de se alimentar. Os resultados previstos pelas redes foram comparados com observações manuais e a rede neural artificial de detecção dos bovinos apresentou uma acurácia de 97,1%, a rede neural de identificação uma acurácia de 85% e a rede neural de estimação do tempo de alimentação um fator de correlação de 78,02%.

Mwanga *et al.* (2019) utilizou uma rede neural e outros diversos modelos estatísticos para uma tarefa não-convencional e inovadora: identificar e prever decisões de fazendeiros quanto ao método de reprodução empregado em suas propriedades, através da análise de variáveis que podem ser adquiridas por meio de questionários ou pura observação. O modelo alcançou acurácias de até 92,2% e pode ser usado para identificar a necessidade de fazendeiros de suporte e assistência na implementação de um novo método de reprodução para seus animais. A estimativa de valores genéticos usando redes neurais, já descrita nos trabalhos anteriormente citados, também foi testada em ovelhas da raça Kermani, por um grupo de pesquisadores. Eles utilizaram diversos dados fisiológicos e informações referentes à produção de 867 animais para estimar o valor genético da característica peso aos seis meses de idade, alcançando um coeficiente de correlação de 0,864 (GHOTBALDINI *et al.*, 2019). Han *et al.* (2019) utilizaram uma rede neural do tipo *feedforward* para classificar a ocupação de um comedouro de porcos em terminação. Ao colocar três possíveis categorias de saída na rede neural (comedor ocupado

com um porco, comedor desocupado, comedor ocupado com mais de um pouco) e utilizar imagens de uma câmera Intel RealSense D435 fixada ao teto da baia, o modelo alcançou 100% em sensibilidade, especificidade e acurácia. Saxena e Parashe (2019) demonstraram o potencial do uso de algoritmos de inteligência artificial, ao utilizar redes neurais artificiais na avaliação de diferentes dietas para animais. A dupla de pesquisadores treinou uma rede neural para avaliar três fatores relacionados a uma dieta, que podem ajudar na sua formulação: preço, concentração de nutrientes e umidade. Normalmente, as expressões matemáticas utilizadas para calcular a melhor combinação de ingredientes visando certos critérios são complexas e específicas para cada caso, então ter um algoritmo que consiga seguir esses critérios sem necessidade de alterar parâmetros é uma grande vantagem. Além disso, o modelo utilizado no trabalho demonstrou superioridade a outras técnicas já descritas.

3.3 ANÁLISE DOS TRABALHOS PUBLICADOS NA ÁREA DA PRODUÇÃO ANIMAL

Os artigos descritos no item anterior foram analisados e avaliados quanto aos seguintes critérios: objetivo da implementação de uma rede neural, espécie alvo do trabalho, ano de publicação, área acadêmica do autor principal, *software* utilizado, número de animais na base de dados de treinamento, melhor resultado obtido com a rede neural e ao tipo de rede neural (regressão ou classificação). Ao total, foram reunidos 40 artigos, descritos nas Tabelas 1 e 2, que também apresentam os critérios citados anteriormente.

Tabela 1 – Lista de artigos com seus respectivos anos de publicação, objetivo do trabalho, espécie abordada e área de atuação do autor principal (continua)

Nº	Nome do artigo	Ano	Objetivo	Espécie	Área do autor
1	Evaluation of Neural Networks as a Tool for Management of Swine Environments	1994	Controle de temperatura	Suíno	NA
2	Estimating Marbling Score in Live Cattle from Ultrasound Images Using Pattern Recognition and Neural Network Procedures	1994	Estimar escore de marmoreio	Bovino	Agricultura
3	Artificial Neural Network Prediction of Ascites in Broilers	1996	Classificar ascite	Aves de corte	Ciência animal
4	Probabilistic Neural Network Prediction of Ascites in Broilers Based on Minimally Invasive Physiological Factors	1997	Classificar ascite	Aves de corte	Ciência animal

(continua)					
Nº	Nome do artigo	Ano	Objetivo	Espécie	Área do autor
5	Multi-spectral Image Analysis using Neural Network Algorithm for Inspection of Poultry Carcasses	1997	Classificar defeitos em carcaça	Aves de corte	Agricultura
6	Improving Neural Network Prediction of Amino Acid Levels in Feed Ingredients	1999	Estimar níveis de aminoácidos	ND	Ciência animal
7	Assessing Swine Thermal Comfort by Image Analysis of Postural Behaviors	1999	Classificar conforto térmico	Suíno	Agricultura / Engenharia
8	A Computerized Mastitis Decision Aid Using Farm-Based Records: An Artificial Neural Network Approach	2000	Classificar bactéria causadora de mastite	Bovino	Ciência animal
9	Investigation into the production and conformation traits associated with clinical mastitis using artificial neural networks	2000	Verificar influência de variáveis na predição de mastite	Bovino	Ciência animal
10	Feasibility of Using Neural Networks for Real-Time Prediction of Poultry Deep Body Temperature Responses to Stressful Changes in Ambient Temperature	2000	Predizer temperatura corporal após mudanças de temperatura	Aves de corte	Engenharia
11	Milk production estimates using feed forward artificial neural networks	2001	Estimar produção leiteira	Bovino	Administração
12	Recognition System for Pig Cough based on Probabilistic Neural Networks	2001	Reconhecer tosses	Suíno	Agricultura
13	Minimal Number of Chicken Daily Growth Velocities for Artificial Neural Network Detection of Pulmonary Hypertension Syndrome (PHS)	2001	Verificar o número mínimo de amostras para melhor eficácia da rede neural	Aves de corte	Ciência animal
14	Quantification of Odours from Piggery Effluent Ponds using an Electronic Nose and an Artificial Neural Network	2003	Estimar níveis de odor de efluentes	Suíno	Engenharia
15	Diet influences the ecology of lactic acid bacteria and Escherichia coli along the digestive tract of cattle: neural networks and 16S rDNA	2003	Predizer a espécie baseado na descrição da colônia bacteriana	Bovino	Ciência animal
16	Approximating the sheep milk production curve through the use of artificial neural networks and genetic algorithms	2005	Estimar a curva de produção leiteira	Ovino	Economia
17	Predicting shrimp growth: Artificial neural network versus nonlinear regression models	2005	Predizer curva de crescimento	Camarão	Ciência animal / Engenharia
18	On-Line Detection of Mastitis in Dairy Herds Using Artificial Neural Networks	2005	Predizer ocorrência de mastite	Bovino	Ciência da computação
19	Uso de Redes Neurais Artificiais na Identificação de Vocalização de Suínos	2008	Identificar vocalização	Suíno	Engenharia

(continua)

Nº	Nome do artigo	Ano	Objetivo	Espécie	Área do autor
20	Walk-through weighing of pigs using machine vision and an artificial neural network	2008	Estimar peso por imagem	Suíno	Ciência animal
21	Milk yield prediction in Kenyan Holstein-Friesian cattle using computer neural networks system	2009	Predizer curva de lactação das filhas baseado na curva de lactação da mãe	Bovino	Ciência animal
22	Prediction of lifetime milk production using artificial neural network in Sahiwal cattle	2009	Predizer produção de leite total	Bovino	Ciência animal
23	Differentiation of perirenal and omental fat quality of suckling lambs according to the rearing system from Fourier transforms mid-infrared spectra using partial least squares and artificial neural networks analysis	2009	Predizer sistema de criação baseado em análise da gordura de carcaça	Ovino	Ciência animal
24	Utilização de inteligência artificial (redes neurais artificiais) para a classificação do comportamento bioquímico de amostras de Escherichia coli isoladas de frangos de corte	2010	Estimar resultado de provas bioquímicas de E. Coli	Aves de corte	Ciência animal
25	Uso de redes neurais artificiais para predição de índices zootécnicos nas fases de gestação e maternidade na suinocultura	2011	Estimar índices zootécnicos através de variáveis térmicas e fisiológicas	Suíno	Engenharia
26	Predicting carcass energy content and composition in broilers using the group method of data handling-type neural networks	2011	Predizer composição e conteúdo de energia da carcaça através de dados da alimentação dos animais	Aves de corte	Ciência animal
27	Comparison of logistic and neural network models to fit to the egg production curve of White Leghorn hens	2011	Estimar curva de produção de ovos	Aves poedeiras	Agricultura / Ciência animal
28	Prediction of 305-day milk yield in Brown Swiss cattle using artificial neural networks	2012	Predição produção de leite em todo período de lactação	Bovino	Ciência animal
29	Prediction of Breeding Values for Dairy Cattle Using Artificial Neural Networks and Neuro-Fuzzy Systems	2012	Predizer valor genético	Bovino	Ciência animal
30	Uso de redes neurais artificiais na predição de valores genéticos para peso aos 205 dias em bovinos da raça Tabapuã	2012	Predizer valor genético para uma característica	Bovino	Ciência animal
31	Neural prediction of heat loss in the pig manure composting process	2013	Predizer perda de calor em processo de compostagem	Suíno	Engenharia

N°	Nome do artigo	Ano	Objetivo	Espécie	(conclusão)
					Área do autor
32	Using Artificial Neural Network to Predict Body Weights of Rabbits	2014	Predizer peso corporal	Coelho	Ciência da computação
33	Cattle Identification Using Segmentation-based Fractal Texture Analysis and Artificial Neural Networks	2016	Identificar animal pelo focinho	Bovino	Ciência da computação
34	Cow Individual Identification Based on Convolutional Neural Network	2018	Identificar animal por foto 360° do animal	Bovino	Biotecnologia
35	Machine learning models for predicting the use of different animal breeding services in smallholder dairy farms in Sub-Saharan Africa	2019	Predizer decisões de fazendeiros quanto ao uso de IA ou touro	Bovino	NA
36	Cow Face Detection and Recognition Based on Automatic Feature Extraction Algorithm	2019	Identificar animal pela face	Bovino	Ciência da computação
37	An Imaging System Based on Deep Learning for Monitoring the Feeding Behavior of Dairy Cows	2019	Identificar animal e avaliar comportamento alimentar	Bovino	Engenharia
38	Predicting breeding value of body weight at 6-month age using Artificial Neural Networks in Kermani sheep breed	2019	Predizer valor genético de peso corporal aos 6 meses	Ovino	Engenharia
39	Using 3D images and deep learning to predict feeder occupancy in grow-finish pigs	2019	Classificar ocupação de comedouro	Suíno	Ciência animal
40	Application of Artificial Neural Network (ANN) for Animal Diet Formulation Modeling	2019	Predição de valores de dieta	ND	Matemática

ND: não definido.

Fonte: o próprio autor.

Tabela 2 – Lista de artigos com o software utilizado para implementação da rede neural, número de dados da base de treinamento, método de avaliação da performance, valor desse método e tipo de tarefa (classificação ou regressão). (continua)

N°	Software da rede neural	N° de dados	Métrica avaliada	Valor da métrica	Tipo de rede neural
1	ND	ND	ND	ND	Regressão
2	ND	53	Acurácia	85,7	Classificação (2)
3	NeuroShell	51	Acurácia	96,9	Classificação (2)
4	NeuroShell	117	Acurácia	91,6	Classificação (2)
5	ND	60	Acurácia	93,3	Classificação (2)
6	NeuroShell	ND	Vários	Vários	Regressão

(conclusão)

N°	Software da rede neural	N° de dados	Métrica avaliada	Valor da métrica	Tipo de rede neural
7	ND	138	Acurácia	90	Classificação (3)
8	NeuroShell	1253	Acurácia	71	Classificação (4)
9	ND	4610	Acurácia	78,2	Classificação (2)
10	NeuroShell	15	R ²	0,69	Regressão
11	ND	103	Acurácia	90,9	Regressão
12	ND	92	Acurácia	91,9	Classificação (4)
13	NeuroShell	62	Acurácia	93	Classificação (2)
14	MATLAB	246	R ²	0,984	Regressão
15	ND	95	Acurácia	96,1	Classificação (6)
16	ND	55	MSE	0,00001	Regressão
17	NeuralWorks	459	R ²	0,979	Regressão
18	ND	176	Acurácia	94	Classificação (2)
19	MATLAB	21	MSE	0,09	Classificação (3)
20	MATLAB	500	Erro médio relativo	0,03	Regressão
21	Weka	6095	R ²	0,873	Classificação (4)
22	MATLAB	1120	R ²	0,298	Regressão
23	MATLAB	106	Acurácia	100	Classificação (2)
24	NeuroShell	261	Acurácia	98,37	Classificação (7)
25	ND	27	R ²	0,9526	Regressão
26	ND	ND	R ²	0,999	Regressão
27	Statistica	27	R ²	99,58	Regressão
28	MATLAB	1848	R ²	0,9	Regressão
29	NeuroSolutions	7000	Acurácia	0,93	Regressão
30	MATLAB	19240	R ²	0,74	Regressão
31	Statistica	255	R ²	0,9871	Regressão
32	Python	108	R ²	0,71	Regressão
33	ND	60	Acurácia	99,97	Classificação (3 ou mais)
34	ND	19440	Acurácia	93,33	Classificação (30)
35	ND	16000	Acurácia	92,2	Classificação (2)
36	ND	51151	Acurácia	94,1	Classificação (300)
37	Python	3200	Acurácia	0,971	Classificação/Regressão
38	MATLAB	607	R ²	0,864	Regressão
39	ND	414	Acurácia	100	Classificação (3)
40	MATLAB	3500	Acurácia	99,99	Regressão

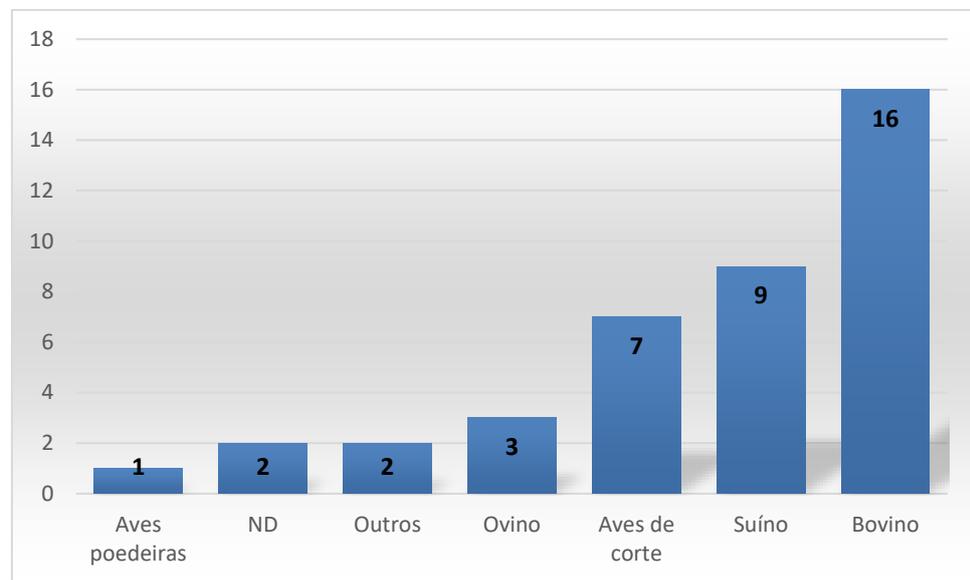
ND: não definido.

Fonte: o próprio autor.

A espécie mais abordada nos trabalhos analisados foi a dos bovinos, com um total de dezesseis artigos lidando com esses animais. Em segundo lugar vieram os suínos, com nove trabalhos. Esses dados podem ser conferidos no Gráfico 1, abaixo.

É possível que os mecanismos de busca tenham adequado os resultados às preferências do autor, baseado no histórico de acesso deste. Em futuros trabalhos de análise literária, cuidados devem ser tomadas para eliminar esse tipo de viés e garantir que os resultados não sejam influenciados.

Gráfico 1 – Espécies abordadas nos trabalhos analisados.



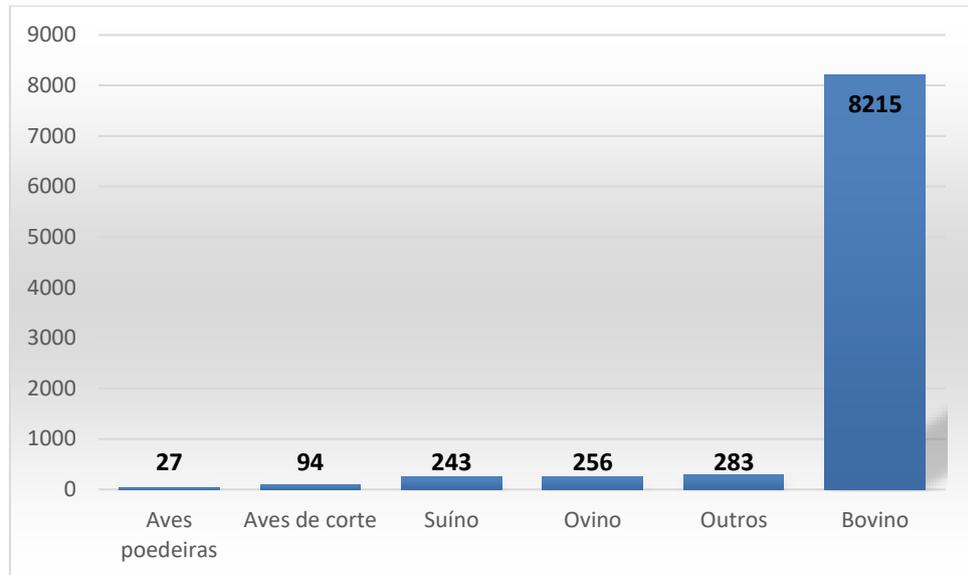
ND: não definido.

Fonte: o próprio autor.

A média de amostras na base de dados de treinamento de trabalhos envolvendo bovinos foi de 8.215, mais de nove vezes mais do que a dos trabalhos envolvendo os outros animais. Isso provavelmente se deve ao fato dos trabalhos de bovinos analisados incluíram vários artigos cujas redes neurais eram do tipo de classificação e lidavam com imagens, necessitando de uma base de dados com um grande número de amostras. Embora a produção de suínos e aves seja de larga escala e bastante tecnificada, o número médio de amostras nas bases de dados dos artigos envolvendo essas espécies não ultrapassaram a marca dos 500 animais. Isso pode ser resultado de um escasso envolvimento dos setores produtivos com os setores de pesquisa ou falta de conhecimento dos autores sobre a importância de uma grande quantidade de dados no treinamento de redes neurais. Outra possível razão é o custo associado à coleta de dados de um grande número de animais, limitando o número de amostras disponíveis para os pesquisadores.

Mesmo com a maior parte dos trabalhos apresentando uma base de dados relativamente pequena, os resultados das redes neurais foram, em sua maioria, satisfatórios, demonstrando o potencial dessa tecnologia até mesmo para produções em pequena escala.

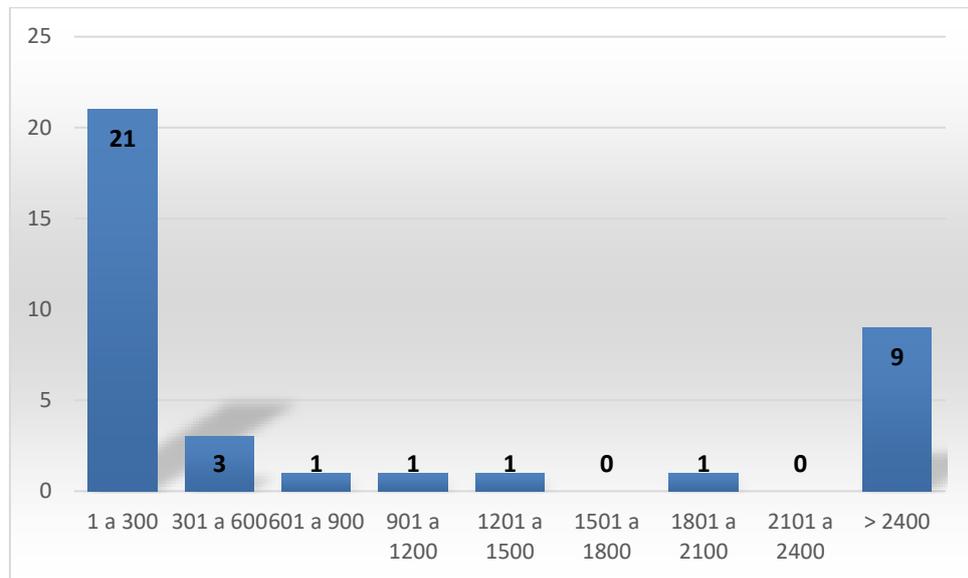
Gráfico 2 – Média do número de amostras dos trabalhos, separadas por espécie.



Fonte: o próprio autor.

A maioria dos trabalhos possuía de um a 300 animais como amostras, demonstrando que as pesquisas da área de produção animal necessitam de maior envolvimento por parte dos produtores e outras pessoas ou entidades que tenham acesso a um grande número de animais. Mesmo com a maior parte dos trabalhos apresentando uma base de dados relativamente pequena, os resultados das redes neurais foram, em sua maioria, satisfatórios, demonstrando o potencial dessa tecnologia até mesmo para produções em pequena escala.

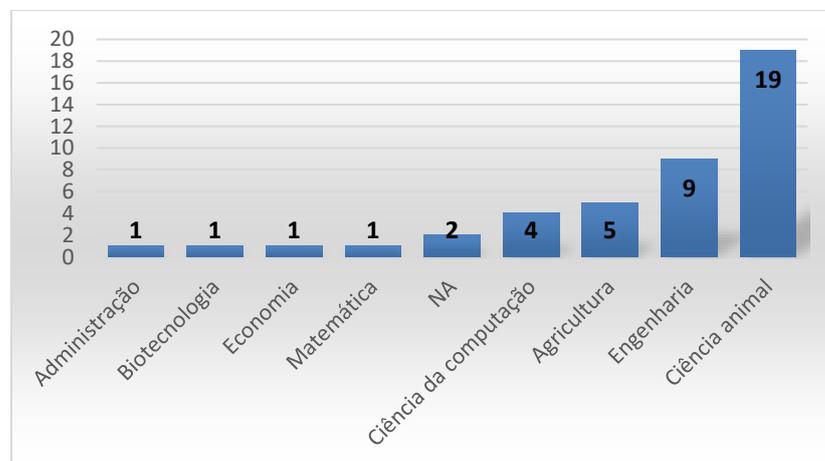
Gráfico 3 – Número de amostras das bases de dados de treinamento dos trabalhos analisados na presente revisão.



Fonte: o próprio autor.

A principal área dos autores principais foi a de ciência animal, com 19 trabalhos representando essa categoria. Em segundo lugar ficou a engenharia, com 9 artigos. É importante ressaltar que em vários trabalhos os autores secundários eram da área de engenharia, ciência da computação ou matemática e que provavelmente eram eles os responsáveis pela programação e implementação das redes neurais. Este fato demonstra a importância da interação entre profissionais de diferentes áreas do conhecimento na implementação desse tipo de tecnologia, tanto no ambiente acadêmico quanto empresarial e produtivo.

Gráfico 4 – Número de autores principais e suas áreas de atuação.

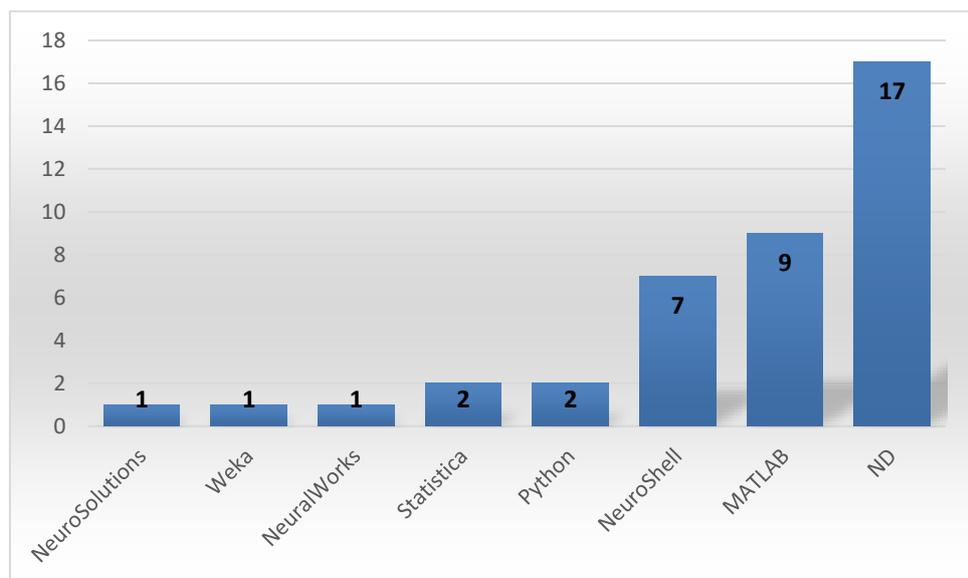


NA: não disponível.

Fonte: o próprio autor.

A maioria dos trabalhos (17) não descrevia o software utilizado (ND), podendo significar que as redes neurais foram implementadas utilizando alguma linguagem de programação, visto que na maioria deles os autores entravam em detalhes sobre a estrutura da rede neural e as funções matemáticas envolvidas na sua construção. Só dois autores citavam a linguagem de programação utilizada, que nesse caso era Python. O software mais citado foi o MATLAB, que possui um pacote específico para implementação de redes neurais. O segundo mais citado foi o NeuroShell, um software que já era utilizado em trabalhos presentes na literatura desde 1996 e aparentemente continua sendo uma ferramenta de eleição para a implementação de redes neurais (ROUSH; KIRBY; CRAVENER; WIDEMAN, 1996; SALLE *et al.*, 2010).

Gráfico 5 – Softwares utilizados para a implementação das redes neurais nos trabalhos analisados.



ND: Não disponível.

Fonte: o próprio autor.

4. ARTIGO – PREVISÃO DE MOTIVO DE DESCARTE DE MATRIZES SUÍNAS UTILIZANDO REDES NEURAS ARTIFICIAS

Neste item é apresentado o artigo “Previsão de Motivo de Descarte de Matrizes Suínas Utilizando Redes Neurais Artificias” a ser submetido a um periódico.

4.1 RESUMO

Objetivou-se com esse trabalho determinar as mais eficazes arquiteturas de redes neurais, cuja tarefa compreende a predição da ocorrência de descarte de matrizes suínas provenientes de granjas de quarto sítio e unidades produtoras de leitões. Uma base de dados com 5.013 fêmeas descartadas, classificadas em sete diferentes categorias de motivos de descarte, foi utilizada para o treinamento das redes neurais. A base foi filtrada e os dados de diferentes índices produtivos das fêmeas foram utilizados para a realização de quatro experimentos. O primeiro experimento tinha o objetivo de testar a eficácia de uma rede neural em classificar o motivo de descarte das fêmeas suínas nas sete diferentes categorias. A acurácia máxima alcançada foi de 56,35%. O segundo experimento visava determinar a eficácia de uma rede neural em estimar a probabilidade de descarte das fêmeas em apenas uma categoria. A acurácia máxima alcançada foi de 99,78%. O terceiro experimento tinha como objetivo avaliar a eficácia de uma rede neural em prever as variáveis da vida produtiva do parto seguinte das fêmeas, baseada em dados dos dois partos anteriores. O erro médio absoluto mínimo alcançado foi de 1,777 para a variável de número de desmamados. O quarto e último experimento tinha o intuito de testar a capacidade de uma rede neural em determinar se uma fêmea seria ou não descartada no parto seguinte, baseada em dados de dois partos anteriores. De maneira geral, as redes neurais demonstraram adequado desempenho em encontrar padrões em diferentes dados da vida produtiva de matrizes suínas e predizer a ocorrência e a classificação de seus descartes.

4.2 ABSTRACT

The objective of this work was to determine the most effective neural network architectures, whose task includes the prediction of the occurrence of culling of female swine breeders from four-site units and piglet-production units. A database of 5,013 culled gilts, classified into seven different culling reason categories, was used for the neural networks training. The base was filtered and the data from different productive indexes of the females were used to perform four experiments. The first experiment aimed to test the effectiveness of a neural network in the task of classifying the reason for culling swine gilts into seven different categories. The maximum accuracy achieved was 56.35%. The second experiment aimed to determine the effectiveness of a neural network in estimating the probability of gilt culling in only one category. The maximum accuracy achieved was 99.78%. The third experiment aimed to evaluate the effectiveness of a neural network in predicting the productive life variables of the breeder's next

delivery, based on data from the previous two parities. The minimum mean absolute error reached was 1.777 for the weaned number variable. The fourth and final experiment was designed to test the ability of a neural network to determine whether or not a gilt would be culled in the next parturition based on data from two previous births. In general, neural networks have shown adequate performance in finding patterns in different data of the productive life of swine breeders and predicting the occurrence and classification of their culling events.

4.3 INTRODUÇÃO

Anualmente, são descartadas cerca de 40 a 50% das fêmeas suínas de granjas comerciais (PIGCHAMP, 2009). Existem vários modelos de reposição de matrizes suínas descritos na literatura científica e o mais popular parece ser o processo de decisão de Markov. Esses modelos nada mais são do que modelos matemáticos utilizados para auxiliar na tomada de decisão de descartar e substituir matrizes suínas, baseando-se em diferentes variáveis da vida produtiva e reprodutiva destas. Neles, geralmente são incluídas variáveis como dados do parto, tamanho da leitegada, status de saúde da matriz, entre outras (PLÀ-ARAGONÉS, 2015).

Vários trabalhos de pesquisa já demonstraram a superioridade que algoritmos de aprendizado de máquina, na maioria das vezes redes neurais ou variações destas, possuem sobre métodos estatísticos clássicos, como regressão linear, regressão polinomial, métodos de regressão não-lineares (Mitscherlich, Gompertz, Logístico, Brody e von Bertalanffy), entre outros (BRETHOUR, 1994; HEALD *et al.*, 2000; CRANER; ROUSH, 1999; SANZOGNI; KERR, 2001; YU; LEUNG; BIENFANG, 2005; GANDHI; RAJA; RUHIL; KUMAR, 2009; GORGULU, 2012; NJUBI; WAKHUNGU; BADAMANA, 2009; OSORIO *et al.*, 2009; SALAWU *et al.*, 2014; SAVEGNAGO *et al.*, 2011; SEDGHI *et al.*, 2012). As redes neurais com uma ou mais camadas escondidas são, por sua natureza, não lineares e altamente adaptáveis, oferecendo uma ótima alternativa para problemas que envolvem variáveis biológicas e que seguem padrões complexos (AGGARWAL, 2018).

Ao que parece, não há na literatura científica nenhum trabalho utilizando redes neurais artificiais na tentativa de prever diferentes motivos de descarte de matrizes suínas, baseando-se em dados de suas vidas produtivas. Com o intuito de descobrir se esta realização é possível, foi desenvolvida uma rede neural artificial do tipo *feed forward* e nela utilizada uma base de dados contendo diversos dados produtivos de matrizes suínas de granjas de quarto sítio e de unidades produtoras de leitões.

4.4 MATERIAIS E MÉTODOS

4.4.1 Origem da Base de Dados Utilizada

A base de dados utilizada é proveniente de dados coletados por Ulguim (2011) em sua tese de mestrado. A base de dados continha dados produtivos de 5.013 fêmeas suínas, sendo que 2.686 eram provenientes de unidades produtoras de leitões e 2.327 eram de unidades de produção de quarto sítio. Todas as fêmeas cujos dados produtivos foram utilizados foram descartadas por algum motivo e eram provenientes de granjas com inventário superior à 500 fêmeas, que utilizavam softwares de gerenciamento para o armazenamento e análise desses dados.

Os dados produtivos de cada matriz presentes na base de dados utilizada incluíam número de partos, duração da lactação de cada parto, duração de cada gestação, soma dos intervalos desmame cio durante a vida produtiva, número de nascidos totais em cada parto, número de nascidos vivos em cada parto, número de desmamados em cada ciclo e motivo de descarte.

4.4.2 Normalização e Verificação dos dados

A base de dados foi verificada para a presença de valores discrepantes ou ausentes, inconsistências e redundâncias. Amostras que apresentavam os erros citados acima foram retiradas. Os motivos de descarte apresentavam-se descritos em forma de texto e, para poderem ser utilizados como valor de saída da rede neural artificial, foram transformados em números.

4.4.3 Organização dos dados para Experimento 1

Para o Experimento 1, somente as fêmeas com dois ou mais partos foram utilizadas e os motivos foram agrupados em sete distintas categorias, propostas por Lucia *et al.* (1999) e adaptadas para os motivos presentes neste trabalho:

Tabela 3 – Descrição das categorias, motivos incluídos e número de amostras para o Experimento 1. (continua)

Categoria	Descrição	Motivos incluídos	Número de amostras
Categoria 1	Motivos referentes à reprodução	Terceiro Cio	488
		Aborto	
		Falsa Gestação	
		Parto Distócico	
		Anestro	
Categoria 2	Motivo referente à morte	Retorno ao Cio	6
		Morte súbita	
Categoria 3	Motivos referentes a doenças e problemas próximos ao parto	Sacrifício	133
		Abscesso	
		Adenomatose	
		Cistite	
		Constricção Anal	
		Descarga Vulvar	
		Doenças Gastrointestinais	
		Edema	
		Encefalite	
		Hérnia	
Categoria 4	Motivos referentes a problemas locomotores	Mastite	285
		Pneumonia	
		Prolapso Retal	
		Prolapso Vaginal	
		Aprumos	
		Artrite	
Categoria 5	Motivos referentes à performance da leitegada	Calos	702
		Cascos	
		Fraturas	
		Claudicação	
		Baixa Produção Leiteira	
Categoria 5	Motivos referentes à performance da leitegada	Esmagamento	702
		Refugagem	
		Defeitos nos Tetos	

(conclusão)

Categoria	Descrição	Motivos incluídos	Número de amostras
Categoria 6	Motivos referentes à idade avançada	Idade Avançada	609
Categoria 7	Motivos que não se enquadram nas categorias 1 a 6	Atresia Anal Motivo Não Informado	18

Fonte: o próprio autor.

Após essa filtragem, a quantidade de amostras foi reduzida de 12.753 dados de partos de 5.013 fêmeas suínas para 2.241 dados de dois partos de 2.241 fêmeas suínas. O critério de dois partos ou mais foi aplicado devido ao objetivo do experimento, que seria verificar a eficiência de uma rede neural artificial na tarefa de classificar a probabilidade de descarte da fêmea no próximo parto, utilizando os dados referentes aos dois últimos partos.

4.4.4 Organização dos dados para Experimento 2

Para o Experimento 2, somente as fêmeas com dois ou mais partos foram utilizadas e os motivos foram agrupados em duas distintas categorias: uma para motivos específicos de uma categoria do Experimento 1 e uma para os motivos restantes. A motivação para tal divisão foi verificar se ocorreria um aumento de eficácia da rede neural, quando a mesma tivesse apenas dois nódulos na camada de saída. Ao total, foram treinadas sete diferentes redes neurais, devido a presença de sete categorias.

Após essa filtragem, a quantidade de amostras foi reduzida de 12.753 dados de partos de 5.013 fêmeas suínas para 2.241 dados de dois partos de 2.241 fêmeas suínas.

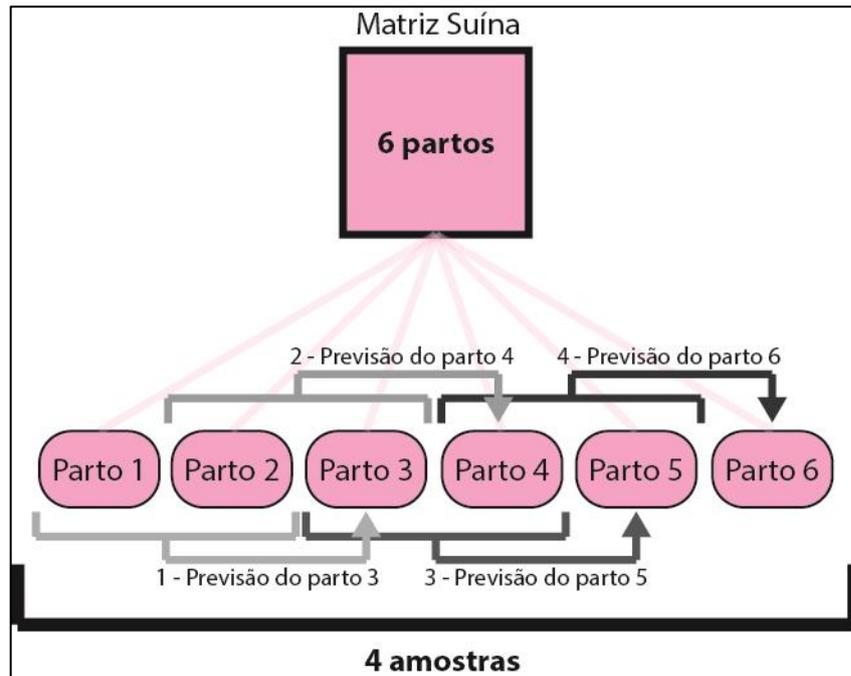
4.4.5 Organização dos dados para Experimento 3

Para o Experimento 3, somente as fêmeas com três ou mais partos foram utilizadas e todos os partos dessas fêmeas foram utilizados para a previsão de alguma variável do parto seguinte, de modo que o número de amostras derivadas de uma fêmea é igual à seguinte equação:

$$N = n - 2$$

Onde N é o número de amostras de cada fêmea e n é o número de partos. Uma melhor representação do processo de escolha dos dados pode ser vista na figura a seguir:

Figura 2 – Demonstração do processo de seleção de amostras dos pares de partos de fêmeas suínas com dois ou mais partos, para o Experimento 3.



Fonte: o próprio autor.

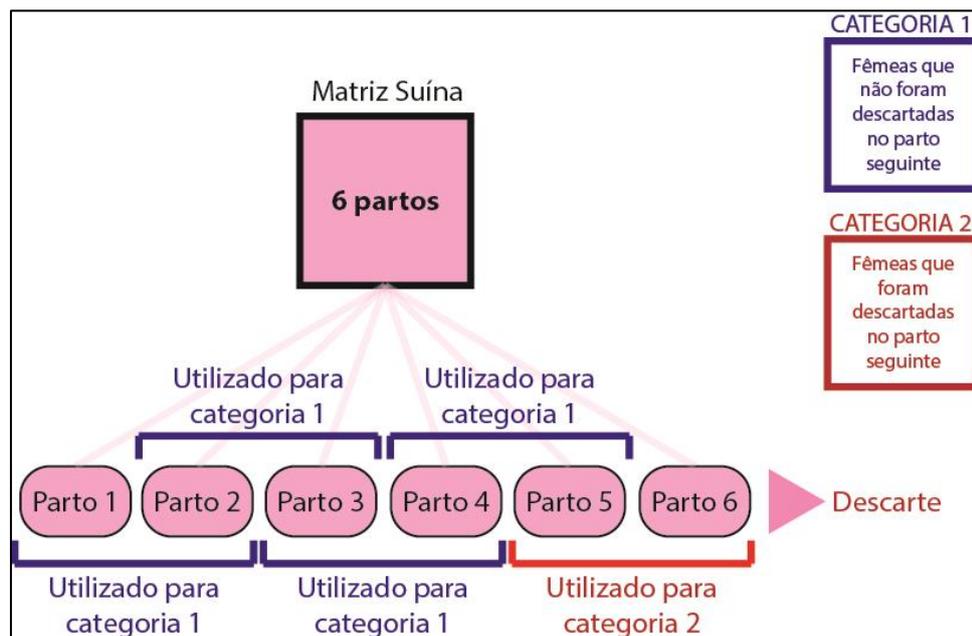
Após essa filtragem, a quantidade de amostras foi reduzida de 12.753 dados de partos de 5.013 fêmeas suínas para 6.708 dados de pares de partos provenientes de 1.802 fêmeas suínas. Dessa forma, serão incluídos não somente dados de partos anteriores ao descarte, mas também dados de partos de fêmeas que continuaram no rebanho. Como o número de variáveis utilizadas como entrada é seis, esse foi o número de redes neurais treinadas para prever uma variável específica do próximo parto.

4.4.6 Organização dos dados para Experimento 4

Para o Experimento 4, somente as fêmeas com dois ou mais partos foram utilizadas, sendo que os últimos dois partos dessas fêmeas foram utilizados como exemplos de partos sucedidos de descarte e o restante dos partos foram utilizados como exemplos de partos que não demonstraram nenhuma razão para descarte das fêmeas. Nesse experimento, não houve a

tentativa de classificar o motivo de descarte das matrizes, mas sim se houve ou não o descarte, após os partos cujos dados foram utilizados como valores de entrada. A ilustração abaixo demonstra o processo de separação dos partos de cada matriz, para ajuste da base de dados:

Figura 3 – Demonstração do processo de seleção de amostras dos pares de partos de fêmeas suínas com dois ou mais partos, para o Experimento 4.

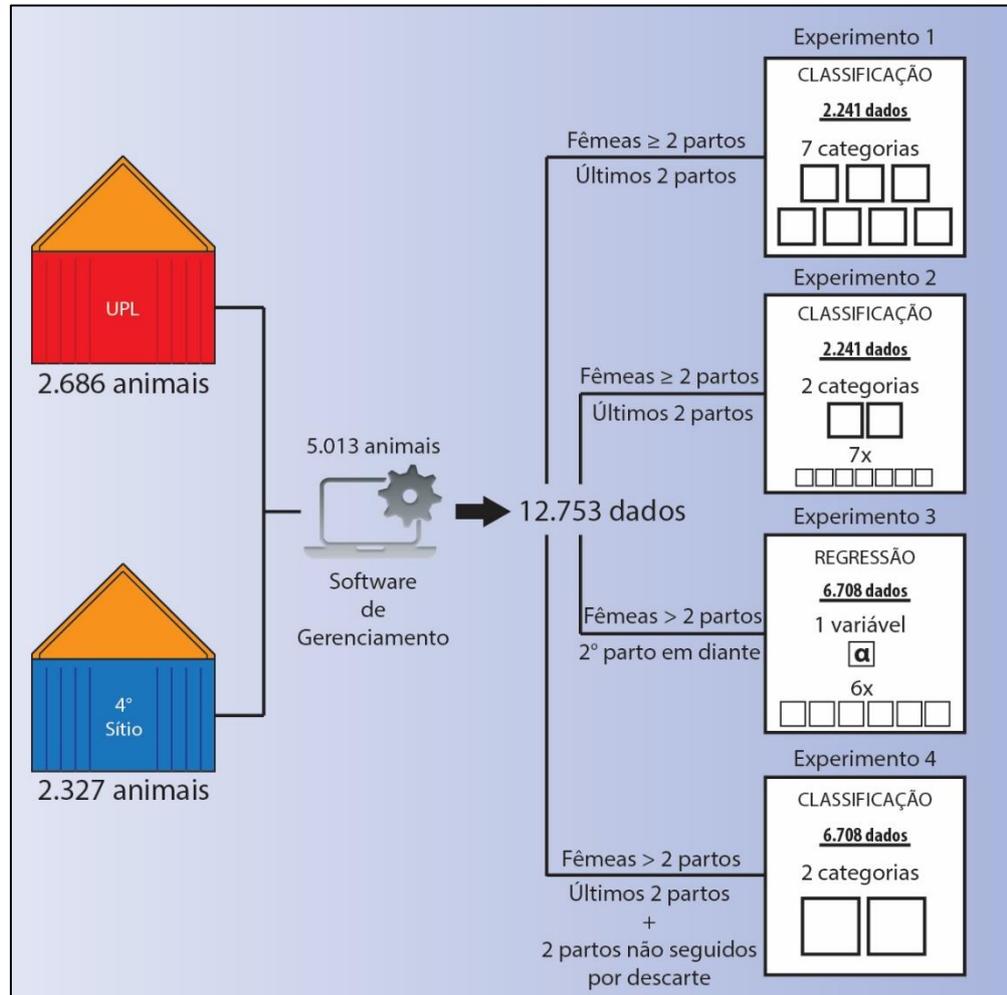


Fonte: o próprio autor.

Após essa filtragem, a quantidade de amostras foi reduzida de 12.753 dados de partos de 5.013 fêmeas suínas para 6.708 dados de dois partos de 1.802 fêmeas suínas.

A ilustração abaixo resume o procedimento de filtragem dos dados para cada experimento:

Figura 4 – Resumo da base de dados, dos critérios utilizados para filtragem dos dados, do tipo de tarefa das redes neurais e da quantidade de valores de saída de cada uma.



Fonte: o próprio autor.

4.4.7 Determinação das melhores arquiteturas para as redes neurais

Para ambas as redes neurais de classificação (experimentos 1, 2 e 4) e regressão (experimento 3), diferentes arquiteturas foram testadas, de modo a encontrar as mais eficazes. As arquiteturas diferiam entre si quanto às funções de ativação utilizadas, à quantidade de camadas, à quantidade de neurônios em cada camada e às funções de otimização. Para ambos os tipos de redes neurais (classificação ou regressão), foram testadas camadas de entrada com e sem a função de ativação *dropout*. Para as redes neurais de classificação, a função de ativação *softmax* foi utilizada na camada de saída. Para as redes neurais de regressão, à camada de saída foi atribuída uma função de ativação linear. As quantidades de camadas testadas foram 1, 3, 5,

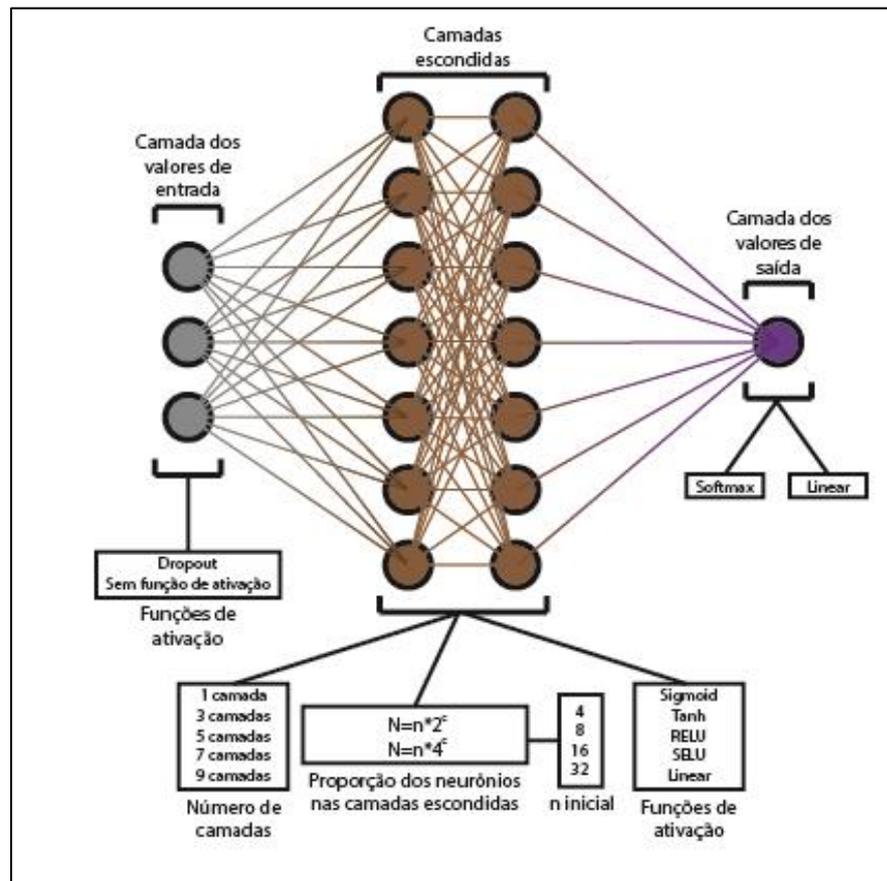
7 e 9. Para o número de neurônios em cada camada, foram definidas duas equações matemáticas que utilizavam um número de neurônios inicial, correspondente ao número de neurônios da primeira camada. As funções eram:

$$N = n * 2^c$$

$$N = n * 4^c$$

Sendo N o número de neurônios de cada camada, c o número da camada (1ª camada = 1) e n o número de neurônios inicial. Os números de neurônios iniciais utilizados foram 4, 8, 16 e 32. As diferentes funções de ativação utilizadas foram *sigmoid*, *tanh*, RELU, SELU e linear. Um resumo dos diferentes parâmetros utilizados pode ser visto na Figura 5.

Figura 5 – Resumo dos parâmetros utilizados nas diferentes arquiteturas de redes neurais artificiais testadas.



Fonte: o próprio autor.

Se todas as combinações possíveis desses diferentes parâmetros fossem utilizadas, o número de arquiteturas a serem testadas ultrapassaria 4.000 para cada um dos experimentos. Por isso, foi decidido modificar um parâmetro por vez e verificar a influência deste na eficácia da rede neural. Primeiramente, testou-se a influência da função de ativação *dropout* na camada de

entrada, utilizando as arquiteturas demonstradas na tabela a seguir. A função de otimização escolhida para todas as arquiteturas testadas foi a *adagrad*, com uma taxa de aprendizado de 0,2. As diferentes arquiteturas foram testadas nos experimentos 1 e 3 utilizando 1000 ciclos de treinamento cada uma. Foi utilizada uma função para que a rede neural parasse o treinamento precocemente, caso a acurácia ou erro médio absoluto não melhorasse após 100 ciclos.

Tabela 4 – Arquiteturas de redes neurais utilizadas para testar a influência de uma função de ativação dropout na camada de entrada para cada um dos experimentos.

Nome	Função de ativação da camada de entrada	Nº Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_drop_1	Dropout (0,5)	9	$N = n * 2^c$	4	RELU	Adagrad
Arq_drop_2	Dropout (0,5)	5	$N = n * 2^c$	16	RELU	Adagrad
Arq_drop_3	Dropout (0,5)	1	$N = n * 2^c$	64	RELU	Adagrad
Arq_drop_4	Dropout (0,5)	9	$N = n * 4^c$	4	RELU	Adagrad
Arq_drop_5	Dropout (0,5)	5	$N = n * 4^c$	16	RELU	Adagrad
Arq_drop_6	Dropout (0,5)	1	$N = n * 4^c$	32	RELU	Adagrad
Arq_drop_7	Dropout (0,5)	9	$N = n * 4^c$	4	Sigmoid	Adagrad
Arq_drop_8	Dropout (0,5)	5	$N = n * 4^c$	16	Sigmoid	Adagrad
Arq_drop_9	Dropout (0,5)	1	$N = n * 4^c$	32	Sigmoid	Adagrad
Arq_nofun_1	Sem função	9	$N = n * 2^c$	4	RELU	Adagrad
Arq_nofun_2	Sem função	5	$N = n * 2^c$	16	RELU	Adagrad
Arq_nofun_3	Sem função	1	$N = n * 2^c$	64	RELU	Adagrad
Arq_nofun_4	Sem função	9	$N = n * 4^c$	4	RELU	Adagrad
Arq_nofun_5	Sem função	5	$N = n * 4^c$	16	RELU	Adagrad
Arq_nofun_6	Sem função	1	$N = n * 4^c$	32	RELU	Adagrad
Arq_nofun_7	Sem função	9	$N = n * 4^c$	4	Sigmoid	Adagrad
Arq_nofun_8	Sem função	5	$N = n * 4^c$	16	Sigmoid	Adagrad
Arq_nofun_9	Sem função	1	$N = n * 4^c$	32	Sigmoid	Adagrad

Fonte: o próprio autor.

Após o treinamento e validação das 18 arquiteturas, uma média foi feita da acurácia das que utilizavam a função *dropout* e das que não utilizavam. Além disso, uma média da quantidade de ciclos de treinamento necessários para atingir a acurácia máxima também foi realizada. Após

concluída essa etapa, o próximo passo foi definir quais funções de ativação ou combinação destas traria os melhores resultados. As arquiteturas testadas são descritas na Tabela 5.

Tabela 5 – Arquiteturas de redes neurais utilizadas para testar a influência de diferentes funções de ativação na eficácia.

Nome	Função de ativação da camada de entrada	Nº Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_sig_1	Dropout (0,5)	1	$N = n * 4^c$	64	Sigmoid	Adagrad
Arq_sig_2	Dropout (0,5)	5	$N = n * 4^c$	16	Sigmoid	Adagrad
Arq_tan_1	Dropout (0,5)	1	$N = n * 4^c$	64	Tanh	Adagrad
Arq_tan_2	Dropout (0,5)	5	$N = n * 4^c$	16	Tanh	Adagrad
Arq_relu_1	Dropout (0,5)	1	$N = n * 4^c$	64	RELU	Adagrad
Arq_relu_2	Dropout (0,5)	5	$N = n * 4^c$	16	RELU	Adagrad
Arq_selu_1	Dropout (0,5)	1	$N = n * 4^c$	64	SELU	Adagrad
Arq_selu_2	Dropout (0,5)	5	$N = n * 4^c$	16	SELU	Adagrad
Arq_lin_1	Dropout (0,5)	1	$N = n * 4^c$	64	Linear	Adagrad
Arq_lin_2	Dropout (0,5)	5	$N = n * 4^c$	16	Linear	Adagrad
Arq_sig_tan_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Sigmoid 2 a 5 - Tanh	Adagrad
Arq_sig_relu_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Sigmoid 2 a 5 - RELU	Adagrad
Arq_sig_selu_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Sigmoid 2 a 5 - SELU	Adagrad
Arq_sig_lin_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Sigmoid 2 a 5 - Linear	Adagrad
Arq_relu_tan_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Relu 2 a 5 - Tanh	Adagrad
Arq_relu_selu_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Relu 2 a 5 - Selu	Adagrad
Arq_relu_lin_1	Dropout (0,5)	5	$N = n * 4^c$	16	1 e 2 – Relu 2 a 5 - Linear	Adagrad

Fonte: o próprio autor.

Definidas as funções de ativações a serem utilizadas na rede neural, o próximo passo consistiu em definir o número de camadas e o número de nós em cada camada. As diferentes arquiteturas testadas são descritas na Tabela 6.

Tabela 6 – Arquiteturas de redes neurais utilizadas para testar a influência de diferentes números de camadas e de nós em cada camada na eficácia.

Nome	Função de ativação da camada de entrada	Nº Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_num_1	Dropout (0,5)	1	$N = n * 4^c$	4	RELU	Adagrad
Arq_num_2	Dropout (0,5)	1	$N = n * 4^c$	8	RELU	Adagrad
Arq_num_3	Dropout (0,5)	1	$N = n * 4^c$	16	RELU	Adagrad
Arq_num_4	Dropout (0,5)	1	$N = n * 4^c$	32	RELU	Adagrad
Arq_num_5	Dropout (0,5)	3	$N = n * 4^c$	4	RELU	Adagrad
Arq_num_6	Dropout (0,5)	3	$N = n * 4^c$	8	RELU	Adagrad
Arq_num_7	Dropout (0,5)	3	$N = n * 4^c$	16	RELU	Adagrad
Arq_num_8	Dropout (0,5)	3	$N = n * 4^c$	32	RELU	Adagrad
Arq_num_9	Dropout (0,5)	5	$N = n * 4^c$	4	RELU	Adagrad
Arq_num_10	Dropout (0,5)	5	$N = n * 4^c$	8	RELU	Adagrad
Arq_num_11	Dropout (0,5)	5	$N = n * 4^c$	16	RELU	Adagrad
Arq_num_12	Dropout (0,5)	5	$N = n * 4^c$	32	RELU	Adagrad
Arq_num_13	Dropout (0,5)	7	$N = n * 4^c$	4	RELU	Adagrad
Arq_num_14	Dropout (0,5)	7	$N = n * 4^c$	8	RELU	Adagrad
Arq_num_15	Dropout (0,5)	7	$N = n * 4^c$	16	RELU	Adagrad
Arq_num_16	Dropout (0,5)	7	$N = n * 4^c$	32	RELU	Adagrad
Arq_num_17	Dropout (0,5)	9	$N = n * 4^c$	4	RELU	Adagrad
Arq_num_18	Dropout (0,5)	9	$N = n * 4^c$	8	RELU	Adagrad
Arq_num_19	Dropout (0,5)	9	$N = n * 4^c$	16	RELU	Adagrad
Arq_num_20	Dropout (0,5)	9	$N = n * 4^c$	32	RELU	Adagrad

Fonte: o próprio autor.

Por último, diferentes funções de otimização foram testadas. As arquiteturas utilizadas para avaliar a eficácia das redes neurais são demonstradas na Tabela 7.

Tabela 7 – Arquiteturas de redes neurais utilizadas para testar a influência de diferentes números de camadas e de nódulos em cada camada na eficácia.

Nome	Função de ativação da camada de entrada	Nº Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_otim_1	Dropout (0,5)	3	$N = n * 4^c$	8	RELU, SELU, RELU	Adagrad
Arq_otim_2	Dropout (0,5)	3	$N = n * 4^c$	8	RELU, SELU, RELU	Adam
Arq_otim_3	Dropout (0,5)	3	$N = n * 4^c$	8	RELU, SELU, RELU	RMSprop
Arq_otim_4	Dropout (0,5)	3	$N = n * 4^c$	8	RELU, SELU, RELU	SGD

Fonte: o próprio autor.

As arquiteturas escolhidas para os experimentos são compostas pelos parâmetros que apresentaram os melhores resultados de acurácia ou menor erro médio absoluto. A acurácia é definida como a proporção das amostras que foram classificadas corretamente pela rede neural, quando essa foi testada na base de dados de validação. O erro médio absoluto corresponde à média dos erros absolutos das amostras, que é definido pela distância da previsão gerada pela rede neural ao valor real da amostra.

4.4.8 Arquitetura das redes neurais para os experimentos 1, 2 e 4

Para os experimentos que possuíam uma tarefa de classificação (experimentos 1, 2 e 4), as seguintes arquiteturas foram utilizadas:

Tabela 8 – Arquiteturas de redes neurais utilizadas nos experimentos 1, 2 e 4. (continua)

Nome	Função de ativação da camada de entrada	Nº Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_class_1	Dropout (0,5)	5	$N = n * 4^c$	32	Tanh	Adagrad
Arq_class_2	Dropout (0,5)	5	$N = n * 4^c$	32	1 e 2 – Sigmoid 2 a 5 - Tanh	Adagrad
Arq_class_3	Dropout (0,5)	5	$N = n * 4^c$	32	1 e 2 – Sigmoid 2 a 5 – Linear	Adagrad
Arq_class_4	Dropout (0,5)	5	$N = n * 4^c$	32	1 e 2 – RELU 2 a 5 - Tanh	Adagrad
Arq_class_5	Dropout (0,5)	5	$N = n * 4^c$	32	1 e 2 – RELU 2 a 5 - SELU	Adagrad

(conclusão)

Nome	Função de ativação da camada de entrada	N° Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_class_6	Dropout (0,5)	5	$N = n * 4^c$	32	Linear	Adagrad

Fonte: o próprio autor.

4.4.9 Arquitetura das redes neurais para o experimento 3

Para o experimento 3, que possuía uma tarefa de regressão, as seguintes arquiteturas foram utilizadas:

Tabela 9 – Arquiteturas de redes neurais utilizadas no experimento 3.

Nome	Função de ativação da camada de entrada	N° Camadas	Equação	n	Funções de ativação	Função de otimização
Arq_reg_1	Sem função	5	$N = n * 4^c$	4	Sigmoid	Adagrad
Arq_reg_2	Sem função	5	$N = n * 4^c$	4	Tanh	Adagrad
Arq_reg_3	Sem função	5	$N = n * 4^c$	4	1 e 2 – Sigmoid 2 a 5 – Tanh	Adagrad
Arq_reg_4	Sem função	5	$N = n * 4^c$	4	1 e 2 – Sigmoid 2 a 5 – SELU	Adagrad
Arq_reg_5	Sem função	5	$N = n * 4^c$	4	1 e 2 – RELU 2 a 5 – Tanh	Adagrad

Fonte: o próprio autor.

4.4.10 Software e hardware utilizados para a implementação das redes neurais

As redes neurais foram implementadas utilizando a linguagem de programação Python 3.7.3 com a biblioteca de aprendizagem profunda Keras v2.2.4 (CHOLLET, 2015), utilizando o módulo TensorFlow v1.14.0 (MARTÍN, 2016) como *back-end*. O computador utilizado possui um processador AMD Ryzen 3 2200 G de 3.50 GHz de frequência, 8 gigabytes de memória RAM DDR4, placa de vídeo GeForce GTX 650 Ti com 1 gigabyte de VRAM GDDR5 e sistema operacional Windows 10 Pro de 64 bits. Para visualização da performance das redes neurais, foi utilizado o API do software Weights & Biases (WEIGHTS & BIASES, 2019).

4.5 RESULTADOS

4.5.1 Resultados das arquiteturas testadas para determinar o efeito de uma função *dropout* na camada de entrada das redes neurais

Os resultados das arquiteturas utilizadas para testarem o efeito da utilização da função *dropout* na camada de entrada de redes neurais do experimento 1 são mostrados abaixo na Tabela 10.

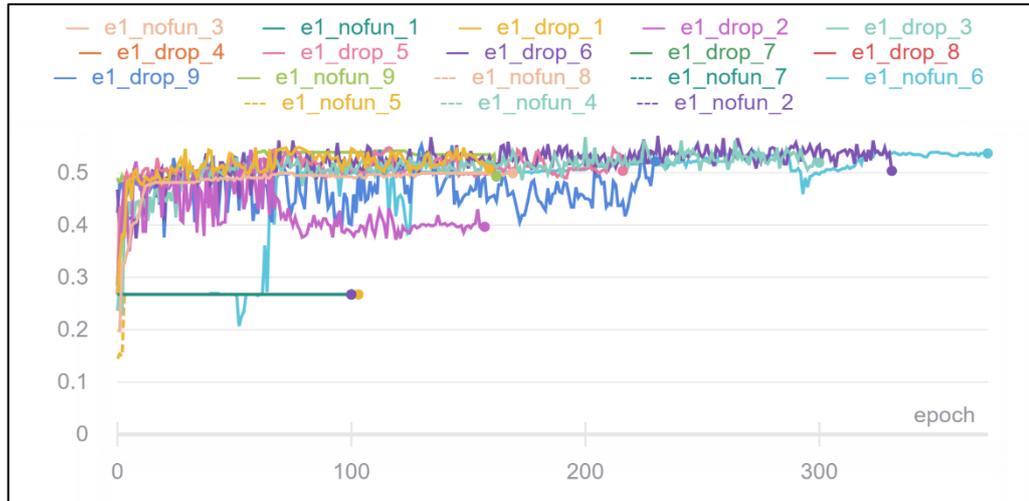
Tabela 10 – Acurácia das arquiteturas utilizadas para testar o efeito da utilização de função *dropout* na camada de entrada para o experimento 1.

Nome	Melhor acurácia	Épocas	Tempo de treinamento
Arq_drop_1	0,5479	159	39 segundos
Arq_drop_2	0,5434	157	34 segundos
Arq_drop_3	0,5679	300	53 segundos
Arq_drop_4	0,2673	100	68 segundos
Arq_drop_5	0,5479	216	43 segundos
Arq_drop_6	0,5679	331	52 segundos
Arq_drop_7	0,2673	100	61 segundos
Arq_drop_8	0,2673	100	26 segundos
Arq_drop_9	0,5523	230	38 segundos
Arq_nofun_1	0,2673	100	32 segundos
Arq_nofun_2	0,2673	100	18 segundos
Arq_nofun_3	0,4989	169	35 segundos
Arq_nofun_4	0,2673	100	65 segundos
Arq_nofun_5	0,2673	100	24 segundos
Arq_nofun_6	0,539	372	48 segundos
Arq_nofun_7	0,2673	100	63 segundos
Arq_nofun_8	0,2673	100	25 segundos
Arq_nofun_9	0,5412	162	25 segundos

Fonte: o próprio autor.

Oito das dezoito redes neurais testadas apresentaram um comportamento anormal, ao alcançar uma acurácia de 0,2673 e não apresentar incrementos a esta durante os ciclos de treinamento subsequentes. Isso pôde ser constatado não só pela análise dos registros de cada ciclo de treinamento, mas também pela análise do Gráfico 6, que mostra a acurácia por época de cada arquitetura.

Gráfico 6 – Acurácia por época de cada arquitetura testada



Fonte: o próprio autor.

A Tabela 11 mostra as médias dos parâmetros de arquiteturas testadas no experimento 1 que apresentaram comportamento considerado normal durante os ciclos de treinamento.

Tabela 11 – Média de parâmetros de arquiteturas sem comportamento anormal, utilizadas para testar o efeito da utilização de função *dropout* na camada de entrada para o experimento 1.

Função dropout na camada de entrada	Média das acurácias	Média das épocas	Médio do tempo de treinamento
Com função dropout	0,55455	232,16	43,16
Sem função	0,52636	234,33	36

Fonte: o próprio autor.

Arquiteturas que possuíam a função *dropout* na camada de entrada apresentaram melhor acurácia do que as que não possuíam nenhuma função de ativação nas redes neurais testadas para o experimento 1. É notável também que, mesmo apresentando menos épocas, as redes neurais com a função *dropout* deixam a desejar no quesito tempo de treinamento.

Os resultados das arquiteturas utilizadas para testarem o efeito da utilização da função *dropout* na camada de entrada de redes neurais do experimento 3 são mostrados abaixo na Tabela 12.

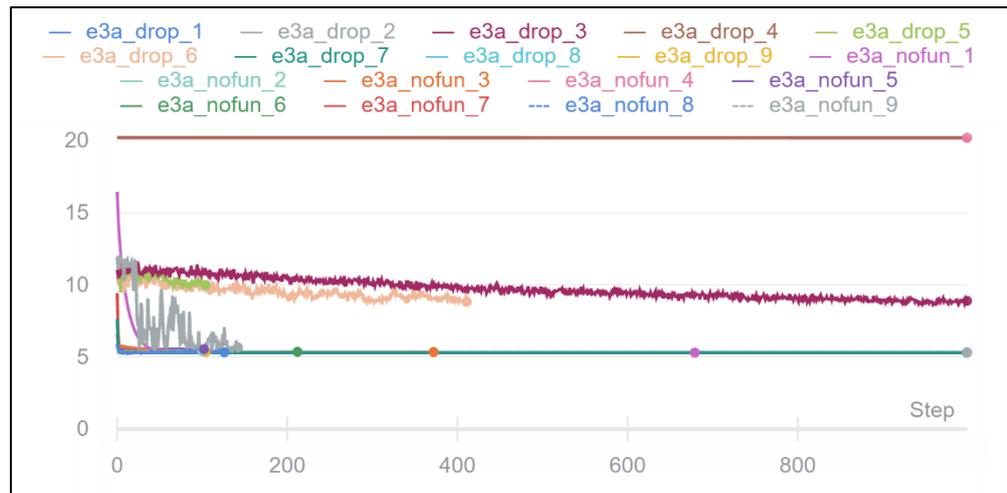
Tabela 12 – Erro médio absoluto (MAE) das arquiteturas utilizadas para testar o efeito da utilização de função *dropout* na camada de entrada para o experimento 3.

Nome	Melhor MAE	Épocas	Tempo de treinamento
Arq_drop_1	5,321	126	160 segundos
Arq_drop_2	5,662	142	156 segundos
Arq_drop_3	8,887	999	831 segundos
Arq_drop_4	20,202	999	1532 segundos
Arq_drop_5	9,985	104	122 segundos
Arq_drop_6	8,823	411	332 segundos
Arq_drop_7	5,299	999	1482 segundos
Arq_drop_8	5,302	999	1022 segundos
Arq_drop_9	5,336	104	94 segundos
Arq_nofun_1	5,298	679	816 segundos
Arq_nofun_2	5,395	100	111 segundos
Arq_nofun_3	5,337	372	296 segundos
Arq_nofun_4	20,179	999	1528 segundos
Arq_nofun_5	5,533	102	110 segundos
Arq_nofun_6	5,354	212	171 segundos
Arq_nofun_7	5,298	999	1440 segundos
Arq_nofun_8	5,303	999	983 segundos
Arq_nofun_9	5,306	999	774 segundos

Fonte: o próprio autor.

Da mesma forma que as arquiteturas utilizadas no experimento 1, algumas do experimento 3 apresentaram comportamento anormal, obtendo incrementos extremamente pequenos a cada ciclo de treinamento, impedindo que o treinamento parasse precocemente, porém não apresentando melhora significativa, como demonstrado no Gráfico 7. As arquiteturas das redes neurais que apresentaram tal comportamento não foram incluídas nas médias mostradas na Tabela 13.

Gráfico 7 – Erro médio absoluto (MAE) por época de cada arquitetura testada



Fonte: o próprio autor.

Tabela 13 – Média de parâmetros de arquiteturas sem comportamento anormal, utilizadas para testar o efeito da utilização de função *dropout* na camada de entrada para o experimento 3.

Função dropout na camada de entrada	Média dos erros médios absolutos	Média das épocas	Médio do tempo de treinamento
Com função dropout	5,96	561,5	624,16
Sem função	5,35	557,75	587,62

Fonte: o próprio autor.

Arquiteturas que não possuíam nenhuma função na camada de entrada apresentaram menor erro médio absoluto do que as que possuíam uma função *dropout* nas redes neurais testadas para o experimento 3. Da mesma forma que as arquiteturas testadas no experimento 1, os ciclos de treinamento das redes neurais com presença de função *dropout* apresentam maior duração.

4.5.2 Resultados das arquiteturas testadas para determinar o efeito das funções de ativação das redes neurais

Os resultados das arquiteturas utilizadas para testarem o efeito da utilização de diferentes funções de ativação nas camadas escondidas de redes neurais do experimento 1 são mostrados na Tabela 14.

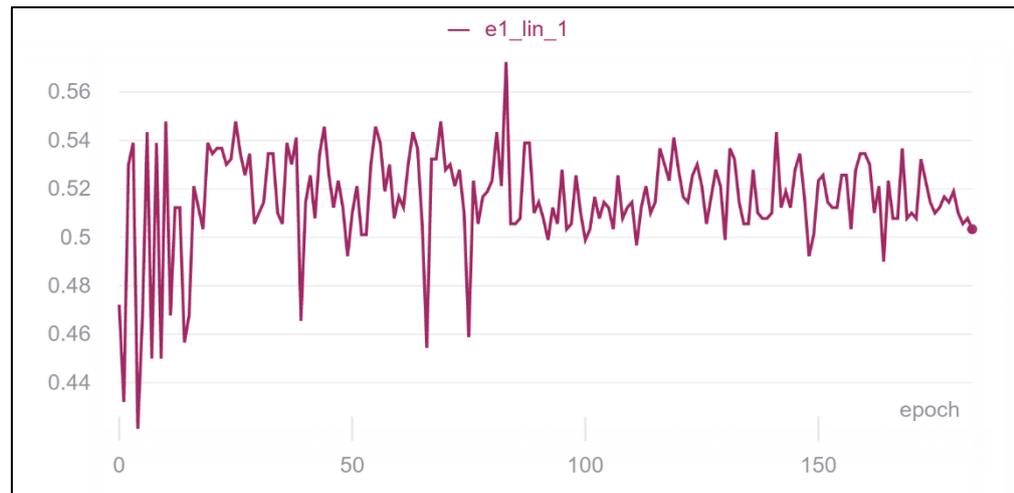
Tabela 14 – Acurácia das arquiteturas utilizadas para testar o efeito da utilização de diferentes funções de ativações para o experimento 1.

Nome	Melhor acurácia	Épocas	Tempo de treinamento
Arq_sig_1	0,5145	146	73 segundos
Arq_sig_2	0,2673	100	72 segundos
Arq_tan_1	0,5323	179	88 segundos
Arq_tan_2	0,5479	187	117 segundos
Arq_relu_1	0,4699	174	96 segundos
Arq_relu_2	0,5434	130	94 segundos
Arq_selu_1	0,51	135	78 segundos
Arq_selu_2	0,2673	100	72 segundos
Arq_lin_1	0,5724	183	99 segundos
Arq_lin_2	0,5345	108	74 segundos
Arq_sig_tan_1	0,5479	154	103 segundos
Arq_sig_relu_1	0,5457	237	140 segundos
Arq_sig_selu_1	0,5457	193	120 segundos
Arq_sig_lin_1	0,5479	122	88 segundos
Arq_relu_tan_1	0,5479	226	137 segundos
Arq_relu_selu_1	0,5479	126	87 segundos

Fonte: o próprio autor.

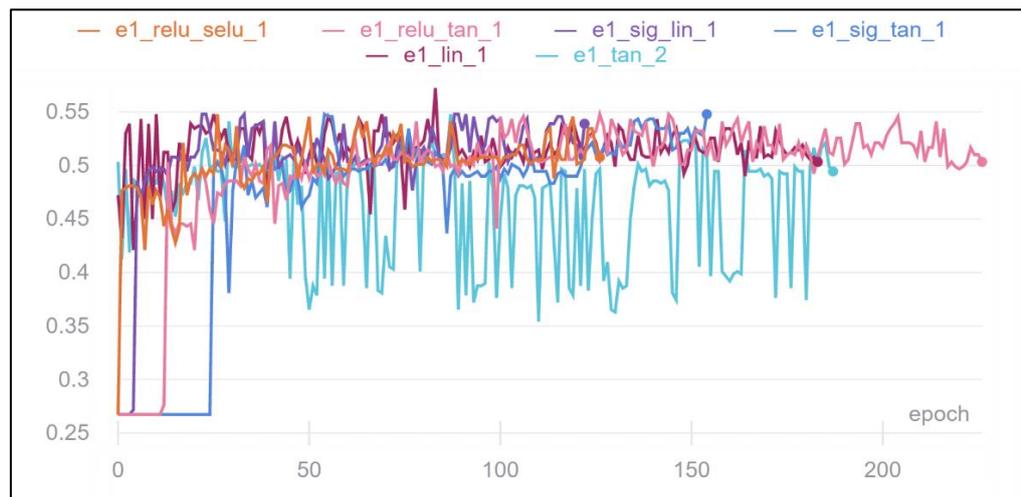
A arquitetura denominada Arq_lin_1, que utiliza apenas funções de ativação lineares nas camadas escondidas, atingiu a maior acurácia (0,5724), porém apresentou alta variação ao longo do treinamento, como é possível notar em seu gráfico de acurácia por época (Gráfico 8). As arquiteturas Arq_tan_2, Arq_sig_tan_1, Arq_sig_lin_1, Arq_relu_tan_1 e Arq_relu_selu_1 apresentaram a segunda maior acurácia (0,5479) e, portanto, as suas funções de ativação foram escolhidas para o treinamento do experimento 1. O Gráfico 9 mostra a acurácia por época das arquiteturas escolhidas.

Gráfico 8 – Acurácia por época da arquitetura Arq_lin_1.



Fonte: o próprio autor.

Gráfico 9 – Acurácia por época de arquiteturas das funções de ativação escolhidas para experimento 1.



Fonte: o próprio autor.

Os resultados das arquiteturas utilizadas para testarem o efeito da utilização de diferentes funções de ativação nas camadas escondidas de redes neurais do experimento 3 são mostrados abaixo na Tabela 15.

Tabela 15 – Acurácia das arquiteturas utilizadas para testar o efeito da utilização de diferentes funções de ativações para o experimento 3.

Nome	Melhor MAE	Épocas	Tempo de treinamento
Arq_sig_1	5,319	386	318 segundos
Arq_sig_2	5,306	353	388 segundos
Arq_tan_1	5,356	103	96 segundos
Arq_tan_2	5,307	561	592 segundos
Arq_relu_1	8,121	406	345 segundos
Arq_relu_2	5,872	352	385 segundos
Arq_selu_1	5,496	284	258 segundos
Arq_selu_2	5,349	629	695 segundos
Arq_lin_1	5,346	133	123 segundos
Arq_lin_2	5,371	704	754 segundos
Arq_sig_tan_1	5,307	539	602 segundos
Arq_sig_relu_1	5,361	189	203 segundos
Arq_sig_selu_1	5,308	246	270 segundos
Arq_sig_lin_1	5,402	155	181 segundos
Arq_relu_tan_1	5,307	526	598 segundos
Arq_relu_selu_1	5,466	468	517 segundos

Fonte: o próprio autor.

A arquitetura denominada Arq_sig_2 apresentou os melhores resultados, com um erro médio absoluto de 5,306. Devido às arquiteturas Arq_tan_2, Arq_sig_tan_1, Arq_sig_selu_1 e Arq_relu_tan_1 apresentarem um erro médio absoluto muito próximo, suas funções de ativação também foram incluídas nas redes neurais utilizados no treinamento para o experimento 3.

4.5.3 Resultados das arquiteturas testadas para determinar o número de camadas e de nódulos nas camadas escondidas

Os resultados das arquiteturas utilizadas para testarem o efeito de diferentes números de camadas e de nódulos nas camadas escondidas de redes neurais do experimento 1 são mostrados abaixo na Tabela 16.

Tabela 16 – Acurácia das arquiteturas utilizadas para testar determinar o número de camadas e de nódulos nas camadas escondidas para o experimento 1. (continua)

Nome	Melhor acurácia	Épocas	Tempo de treinamento
Arq_num_1	0,2673	100	61 segundos
Arq_num_2	0,5367	191	112 segundos

(conclusão)

Nome	Melhor acurácia	Épocas	Tempo de treinamento
Arq_num_3	0,4209	145	94 segundos
Arq_num_4	0,4543	281	147 segundos
Arq_num_5	0,2673	100	73 segundos
Arq_num_6	0,5479	113	83 segundos
Arq_num_7	0,5479	207	129 segundos
Arq_num_8	0,5412	169	116 segundos
Arq_num_9	0,5479	155	112 segundos
Arq_num_10	0,5457	228	163 segundos
Arq_num_11	0,4076	112	94 segundos
Arq_num_12	0,5523	283	189 segundos
Arq_num_13	0,2673	100	97 segundos
Arq_num_14	0,2673	100	104 segundos
Arq_num_15	0,2673	100	113 segundos
Arq_num_16	0,2673	101	152 segundos
Arq_num_17	0,2673	100	118 segundos
Arq_num_18	0,2673	100	151 segundos
Arq_num_19	0,2673	101	275 segundos
Arq_num_20	0,2673	101	890 segundos

Fonte: o próprio autor.

Diversas arquiteturas apresentaram o comportamento anormal descrito no item 4.5.1. A arquitetura Arq_num_12 se destacou com uma acurácia bastante superior às outras (0,5523) e os números de camadas e nós desta foram utilizados para as redes neurais do experimento 1. Essa mesma arquitetura também apresentou o maior número de ciclos de treinamento, indicando um aumento constante em sua acurácia e um potencial para melhorá-la, além de, apesar disso, não apresentar o maior tempo total de treinamento.

Para o teste com o experimento 3, os resultados das arquiteturas são demonstrados na Tabela abaixo.

Tabela 17 – Erro médio absoluto (MAE) das arquiteturas utilizadas para testar determinar o número de camadas e de nós nas camadas escondidas para o experimento 3. (continua)

Nome	Melhor MAE	Épocas	Tempo de treinamento
Arq_num_1	7,996	102	110 segundos
Arq_num_2	9,208	100	105 segundos
Arq_num_3	9,562	100	101 segundos
Arq_num_4	9,503	125	131 segundos
Arq_num_5	5,39	119	137 segundos
Arq_num_6	5,748	263	290 segundos
Arq_num_7	8,457	225	256 segundos
Arq_num_8	11,38	139	165 segundos

(conclusão)

Nome	Melhor MAE	Épocas	Tempo de treinamento
Arq_num_9	5,298	179	228 segundos
Arq_num_10	5,401	114	149 segundos
Arq_num_11	11,076	100	137 segundos
Arq_num_12	12,103	100	165 segundos
Arq_num_13	5,336	129	184 segundos
Arq_num_14	5,605	307	491 segundos
Arq_num_15	9,055	549	709 segundos
Arq_num_16	5,397	401	520 segundos
Arq_num_17	19,947	999	1850 segundos
Arq_num_18	20,202	999	2849 segundos
Arq_num_19	20,566	252	1454 segundos
Arq_num_20	224430,266	537	10976 segundos

Fonte: o próprio autor.

A arquitetura que apresentou menor erro médio absoluto (5,298) e foi, portanto, a escolhida para as redes neurais utilizadas no experimento 3 foi a Arq_num_9.

4.5.4 Resultados das arquiteturas testadas para determinar a função de otimização utilizada

As acurácias de diferentes redes neurais utilizando quatro tipos de funções de otimização para a base de dados do experimento 1 são demonstradas na Tabela 18.

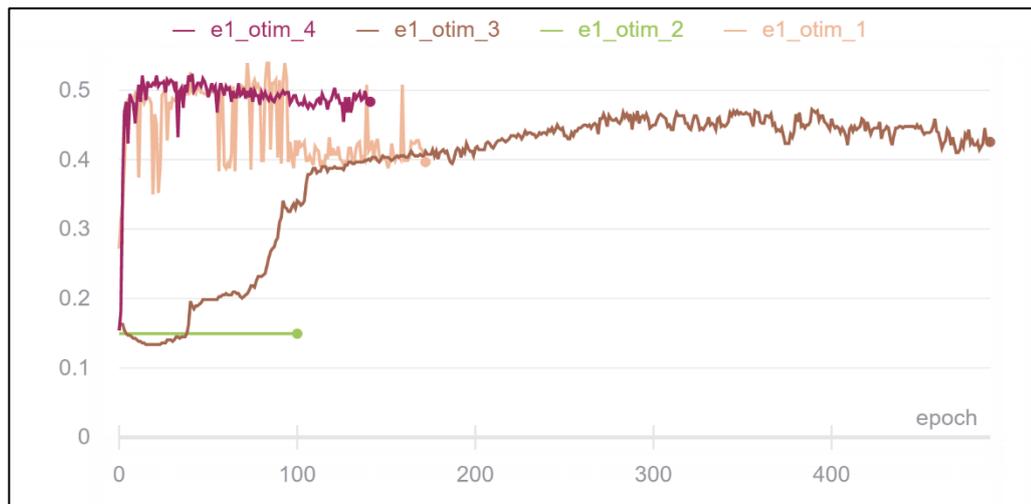
Tabela 18 – Acurácia das arquiteturas utilizadas para testar o efeito da utilização de diferentes funções de ativações para o experimento 1.

Nome	Melhor acurácia	Épocas	Tempo de treinamento
Arq_otim_1	0,539	172	134 segundos
Arq_otim_2	0,1492	100	100 segundos
Arq_otim_3	0,4722	489	327 segundos
Arq_otim_4	0,5234	141	117 segundos

Fonte: o próprio autor.

A arquitetura que apresentou maior acurácia foi a Arq_otim_1, cuja função de ativação era a adagrad, com uma taxa de aprendizado de 0,2. A rede neural com a função de otimização RMSprop apresentou quase o triplo de ciclos de treinamento que as outras redes neurais. A acurácia por época das diferentes arquiteturas testadas é demonstrada no Gráfico 10.

Gráfico 10 – Acurácia por época de arquiteturas com diferentes funções de otimização, utilizadas no experimento 1.



Fonte: o próprio autor.

Os erros médios absolutos (MAE) das arquiteturas para a base de dados do experimento 3 são demonstrados na Tabela 19.

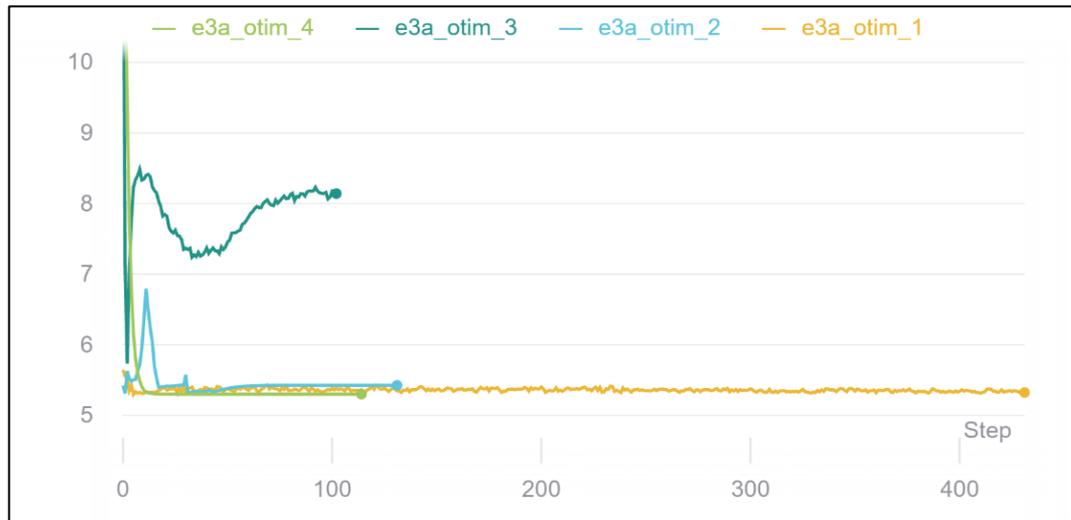
Tabela 19 – Acurácia das arquiteturas utilizadas para testar o efeito da utilização de diferentes funções de ativações para o experimento 3.

Nome	Melhor MAE	Épocas	Tempo de treinamento
Arq_otim_1	5,323	431	575 segundos
Arq_otim_2	5,426	131	193 segundos
Arq_otim_3	8,141	102	176 segundos
Arq_otim_4	5,3	114	144 segundos

Fonte: o próprio autor.

A arquitetura que apresentou maior acurácia foi a Arq_otim_4, cuja função de ativação era a SGD, com uma taxa de aprendizado de 0,001. A acurácia por época das diferentes arquiteturas testadas é demonstrada no Gráfico 11.

Gráfico 11 – Erro médio absoluto (MAE) por época de arquiteturas com diferentes funções de otimização, utilizadas no experimento 3.



Fonte: o próprio autor.

4.5.5 Resultados do experimento 1

As acurácias das redes neurais utilizadas no experimento 1 são descritas na tabela a seguir.

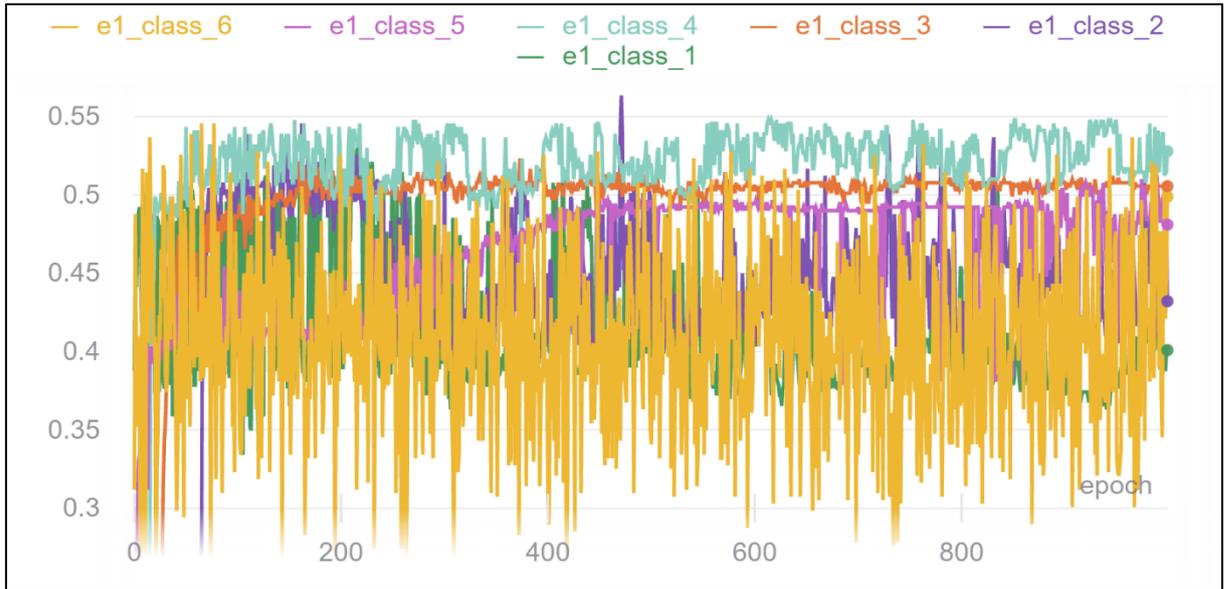
Tabela 20 – Acurácia das redes neurais utilizadas na base de dados do experimento 1.

Nome	Melhor acurácia	Tempo de treinamento
Arq_class_1	0,5301	331 segundos
Arq_class_2	0,5635	314 segundos
Arq_class_3	0,5234	329 segundos
Arq_class_4	0,5479	337 segundos
Arq_class_5	0,5122	326 segundos
Arq_class_6	0,5457	357 segundos

Fonte: o próprio autor.

A melhor acurácia alcançada pelas seis redes neurais testadas foi de 56,35%, indicando uma baixa correlação entre as variáveis de entrada e os grupos de motivos de descarte ou inabilidade das redes neurais de encontrar os padrões presentes nos dados da base de treinamento. A acurácia por época das redes é mostrada no Gráfico 12.

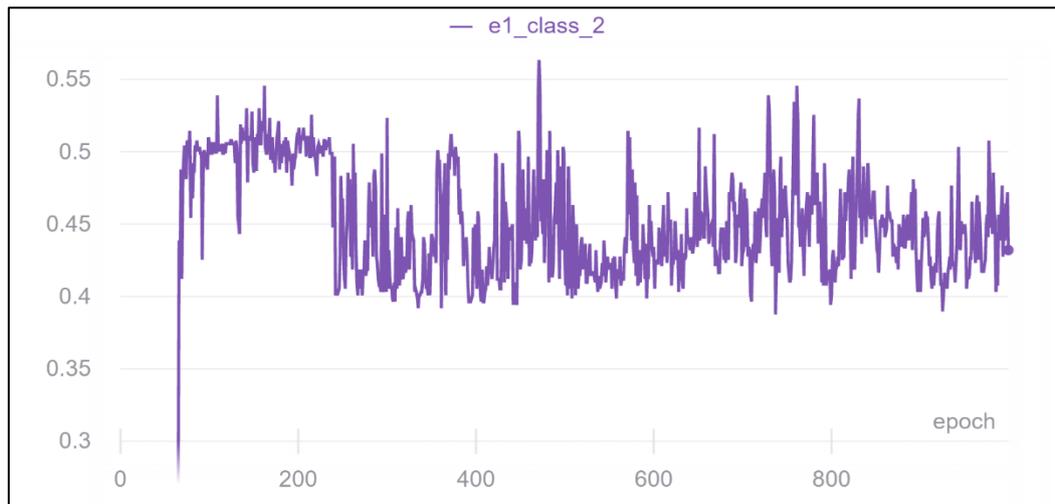
Gráfico 12 – Acurácia por época das redes neurais utilizadas no experimento 1.



Fonte: o próprio autor.

É possível notar que a rede neural e1_class_6, cuja arquitetura corresponde à Arq_class_6 da Tabela 6, que possui apenas funções lineares nas suas camadas escondidas, apresenta a maior variação de acurácia ao longo dos ciclos de treinamento. A acurácia por ciclo de treinamento da melhor rede neural (Arq_class_2) é demonstrada no Gráfico 13.

Gráfico 13 – Acurácia por época da rede neural Arq_class_2.



Fonte: o próprio autor.

4.5.6 Resultados do experimento 2

As acurácias das redes neurais utilizadas no experimento 2 são descritas na Tabela 23 e as acurácias por época dessas redes são evidenciadas nos Gráficos 14, 15, 16, 17, 18, 19, 20 e 21.

Tabela 23 – Acurácia das redes neurais utilizadas na base de dados do experimento 2. (continua)

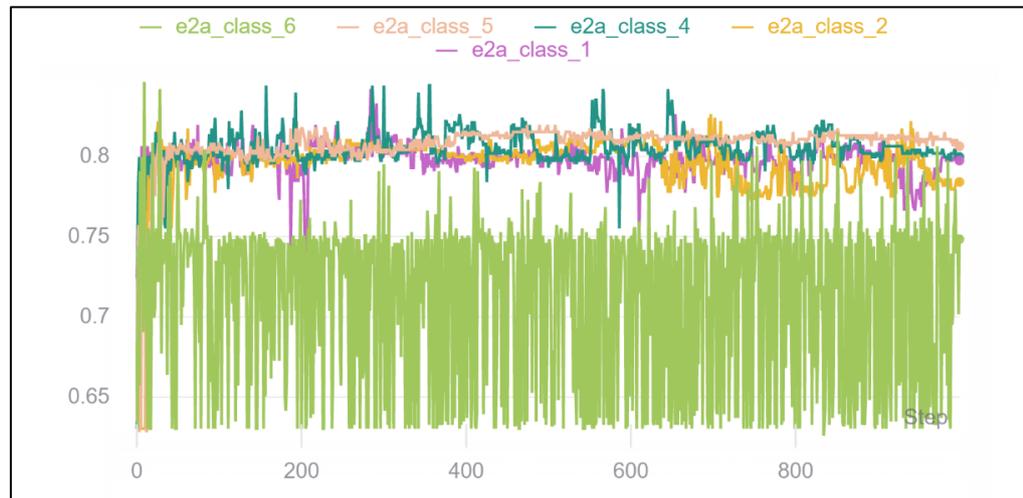
Categoria avaliada	Arquitetura	Melhor acurácia	Tempo de treinamento
C1	Arq_class_1	0,8419	305 segundos
	Arq_class_2	0,8263	323 segundos
	Arq_class_3	0,8218	313 segundos
	Arq_class_4	0,8441	317 segundos
	Arq_class_5	0,8196	345 segundos
	Arq_class_6	0,8463	330 segundos
C2	Arq_class_1	0,9978	326 segundos
	Arq_class_2	0,9978	316 segundos
	Arq_class_3	0,9978	345 segundos
	Arq_class_4	0,9978	302 segundos
	Arq_class_5	0,9978	318 segundos
	Arq_class_6	0,9978	333 segundos
C3	Arq_class_1	0,9488	323 segundos
	Arq_class_2	0,9465	311 segundos
	Arq_class_3	0,9465	319 segundos
	Arq_class_4	0,9465	328 segundos
	Arq_class_5	0,9465	338 segundos
	Arq_class_6	0,9465	341 segundos
C4	Arq_class_1	0,8486	377 segundos
	Arq_class_2	0,8486	367 segundos
	Arq_class_3	0,8486	353 segundos
	Arq_class_4	0,8486	335 segundos
	Arq_class_5	0,8486	337 segundos
	Arq_class_6	0,8486	340 segundos
C5	Arq_class_1	0,8129	368 segundos
	Arq_class_2	0,7862	328 segundos
	Arq_class_3	0,8107	314 segundos
	Arq_class_4	0,7906	348 segundos
	Arq_class_5	0,735	342 segundos
	Arq_class_6	0,7751	307 segundos
C6	Arq_class_1	0,8129	335 segundos
	Arq_class_2	0,8508	335 segundos
	Arq_class_3	0,8508	339 segundos
	Arq_class_4	0,8508	334 segundos
	Arq_class_5	0,8552	313 segundos
	Arq_class_6	0,8575	312 segundos

(conclusão)

Categoria avaliada	Arquitetura	Melhor acurácia	Tempo de treinamento
C7	Arq_class_1	0,9933	356 segundos
	Arq_class_2	0,9933	374 segundos
	Arq_class_3	0,9933	337 segundos
	Arq_class_4	0,9933	360 segundos
	Arq_class_5	0,9933	369 segundos
	Arq_class_6	0,9933	352 segundos

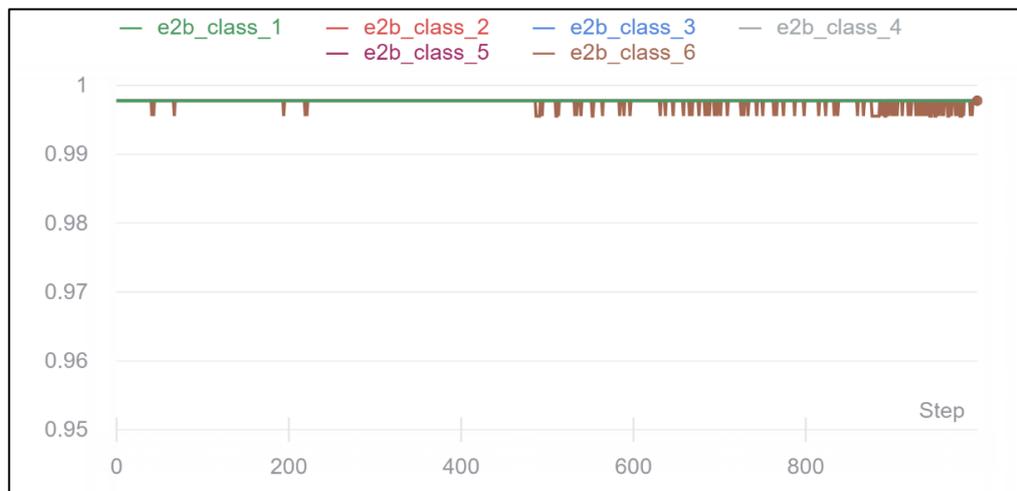
Fonte: o próprio autor.

Gráfico 14 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos reprodutivos (C1) ou não (C2 a C7).



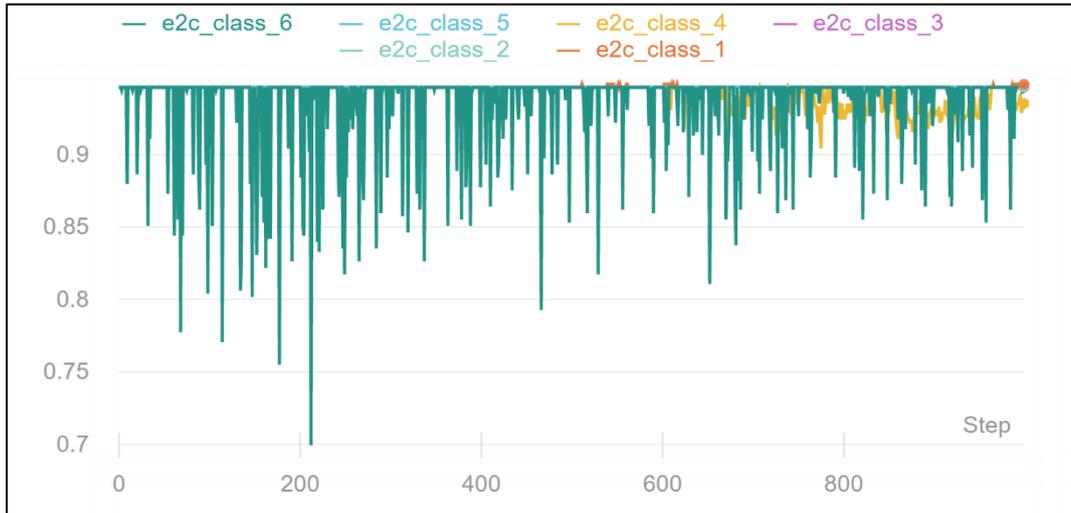
Fonte: o próprio autor.

Gráfico 15 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C2 ou não (C1 e C3 a C7).



Fonte: o próprio autor.

Gráfico 16 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C3 ou não (C1 a C2 e C4 a C7).



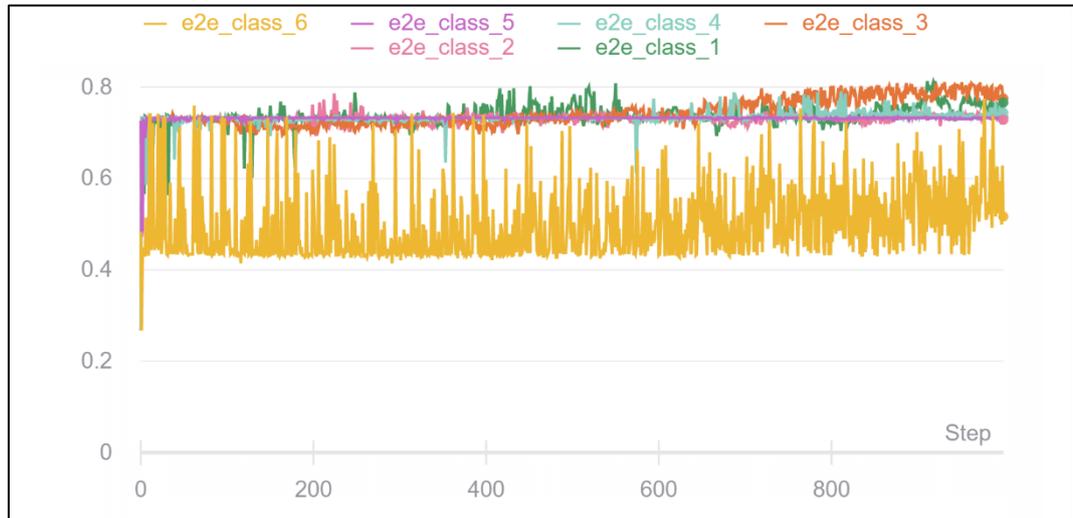
Fonte: o próprio autor.

Gráfico 17 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C4 ou não (C1 a C3 e C5 a C7).



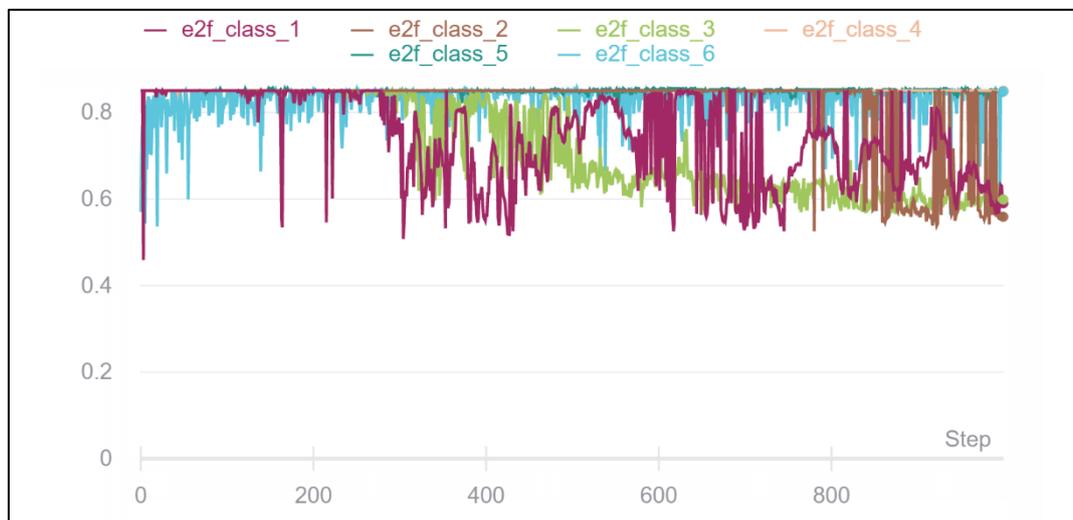
Fonte: o próprio autor.

Gráfico 18 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C5 ou não (C1 a C4 e C6 a C7).



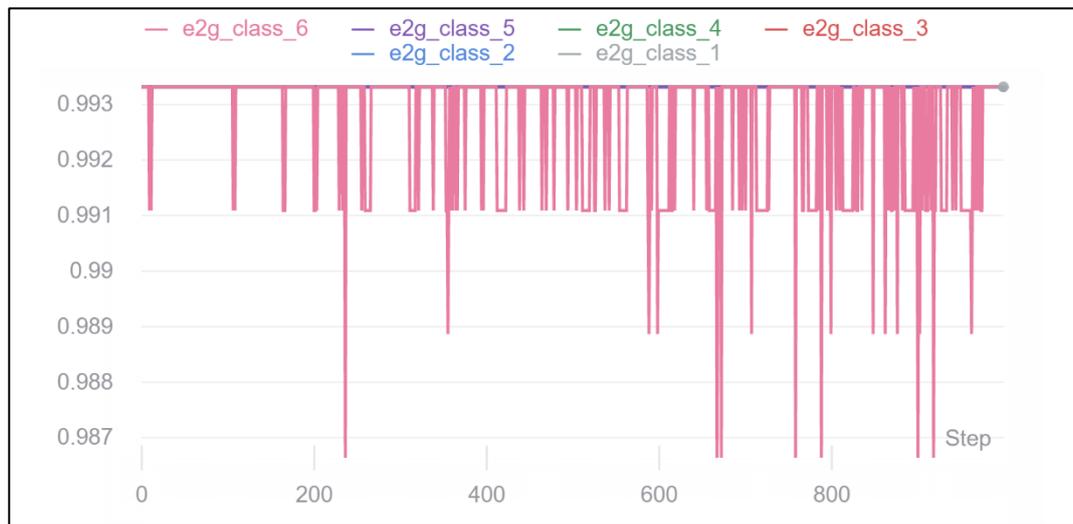
Fonte: o próprio autor.

Gráfico 19 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C6 ou não (C1 a C5 e C7).



Fonte: o próprio autor.

Gráfico 20 – Acurácia por época das redes neurais utilizadas no experimento 2, para classificação de motivos da categoria C7 ou não (C1 e C3 a C7).



Fonte: o próprio autor.

É possível notar que todas as redes neurais cujas arquiteturas correspondiam à Arq_class_6 da Tabela 8 apresentam grande variação de acurácia por época, independente da categoria a ser classificada. O mesmo foi observado nos experimentos 1 e 4.

As redes neurais designadas para a classificação das categorias 7, 4, 3 e 2 não apresentaram grandes diferenças entre as arquiteturas testadas, no que diz respeito à acurácia máxima alcançada na base de dados de validação. Essas mesmas redes neurais parecem ter apresentado um limite máximo de acurácia, que foi atingido nas épocas iniciais e permaneceu igual até o final dos 1000 ciclos de treinamento. A falta de inclinação nas curvas de todos os gráficos indica que a grande quantidade de épocas utilizada não era necessária e que a acurácia não aumentou significativamente após a primeira centena de ciclos de treinamento.

4.5.7 Resultados do experimento 3

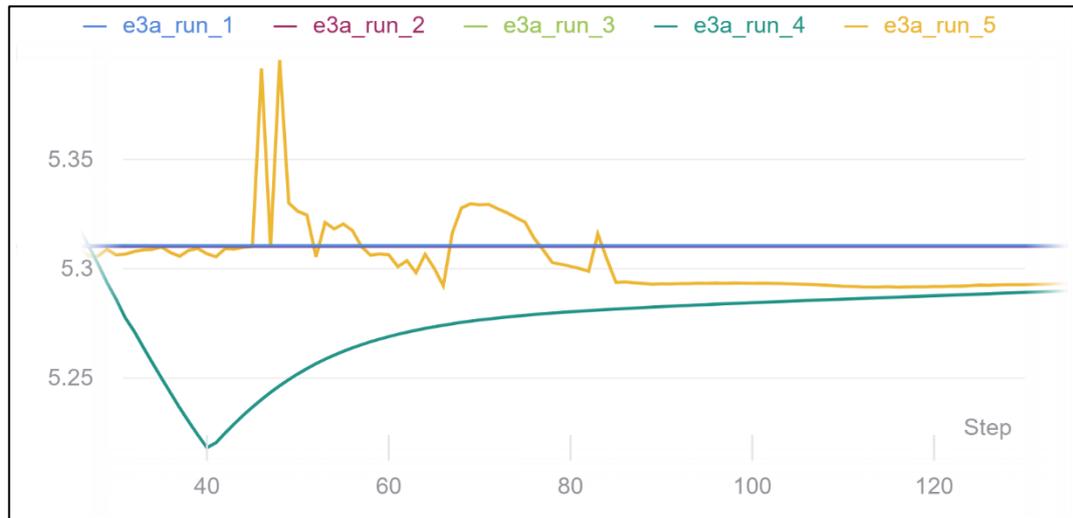
Os erros médios absolutos das redes neurais utilizadas no experimento 3 e os mínimos e máximos das variáveis estimadas são descritos na Tabela 24 e MAE por época dessas redes são visíveis nos Gráficos 17, 18, 19, 20, 21 e 22.

Tabela 24 – Erro médio absoluto (MAE) das redes neurais utilizadas na base de dados do experimento 3.

Variável estimada	Intervalo mín e max	Arquitetura	Melhor MAE	Tempo de treinamento
Duração da lactação	1 – 72	Arq_reg_1	5,311	963 segundos
		Arq_reg_2	5,31	900 segundos
		Arq_reg_3	5,31	908 segundos
		Arq_reg_4	5,218	905 segundos
		Arq_reg_5	5,292	883 segundos
Duração da gestação	12 - 148	Arq_reg_1	14,778	939 segundos
		Arq_reg_2	14,778	936 segundos
		Arq_reg_3	14,778	948 segundos
		Arq_reg_4	14,374	1048 segundos
		Arq_reg_5	14,778	966 segundos
Intervalo desmame cio	0 - 108	Arq_reg_1	2,117	950 segundos
		Arq_reg_2	2,374	848 segundos
		Arq_reg_3	2,335	799 segundos
		Arq_reg_4	2,382	846 segundos
		Arq_reg_5	2,354	836 segundos
Número de nascidos totais	0 – 25	Arq_reg_1	2,986	833 segundos
		Arq_reg_2	2,984	812 segundos
		Arq_reg_3	2,916	842 segundos
		Arq_reg_4	3,008	880 segundos
		Arq_reg_5	2,984	891 segundos
Número de nascidos vivos	0 - 23	Arq_reg_1	2,892	881 segundos
		Arq_reg_2	2,892	892 segundos
		Arq_reg_3	2,811	868 segundos
		Arq_reg_4	2,857	903 segundos
		Arq_reg_5	2,795	873 segundos
Número de desmamados	0 - 27	Arq_reg_1	1,966	900 segundos
		Arq_reg_2	1,966	906 segundos
		Arq_reg_3	1,966	931 segundos
		Arq_reg_4	1,954	948 segundos
		Arq_reg_5	1,66	935 segundos

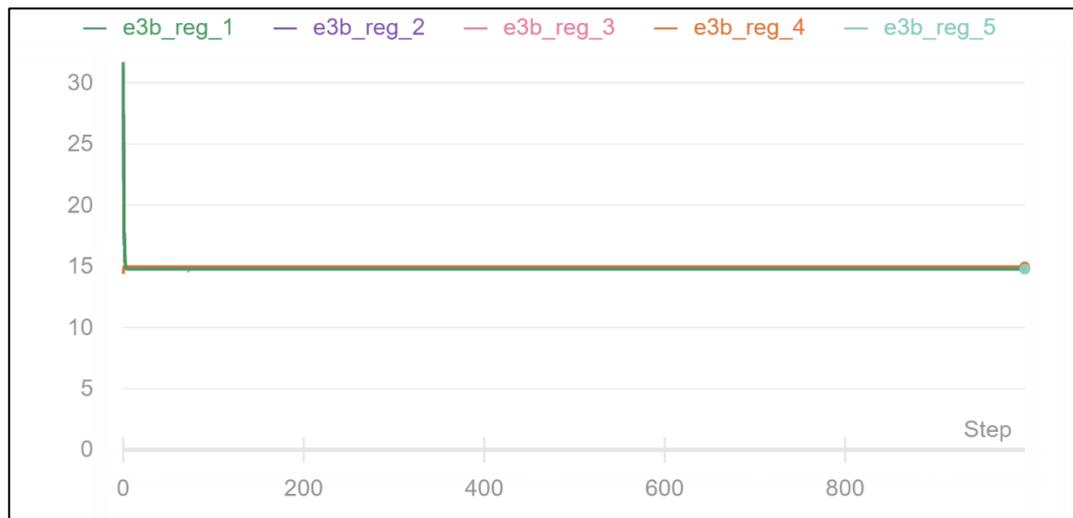
Fonte: o próprio autor.

Gráfico 21 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável duração da lactação.



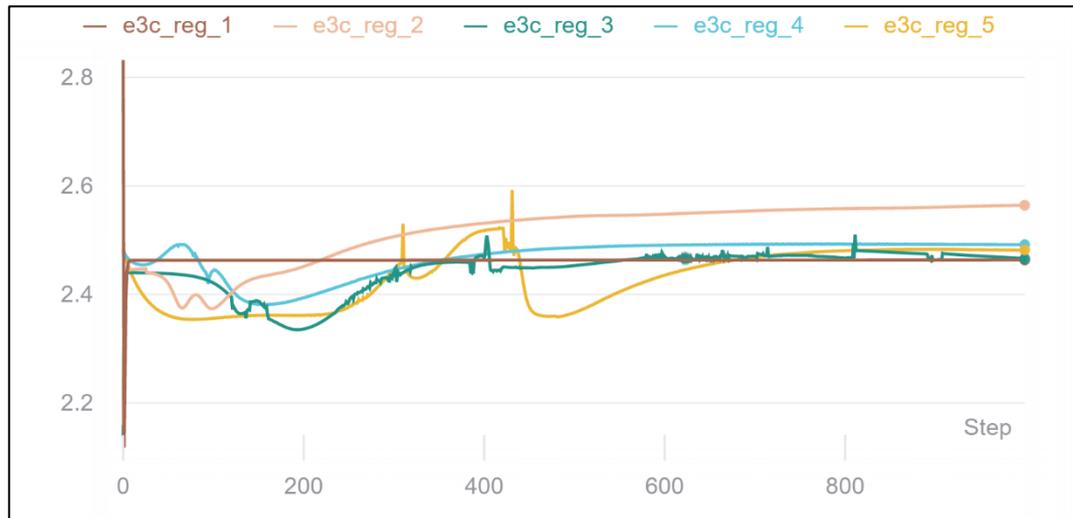
Fonte: o próprio autor.

Gráfico 22 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável duração da gestação.



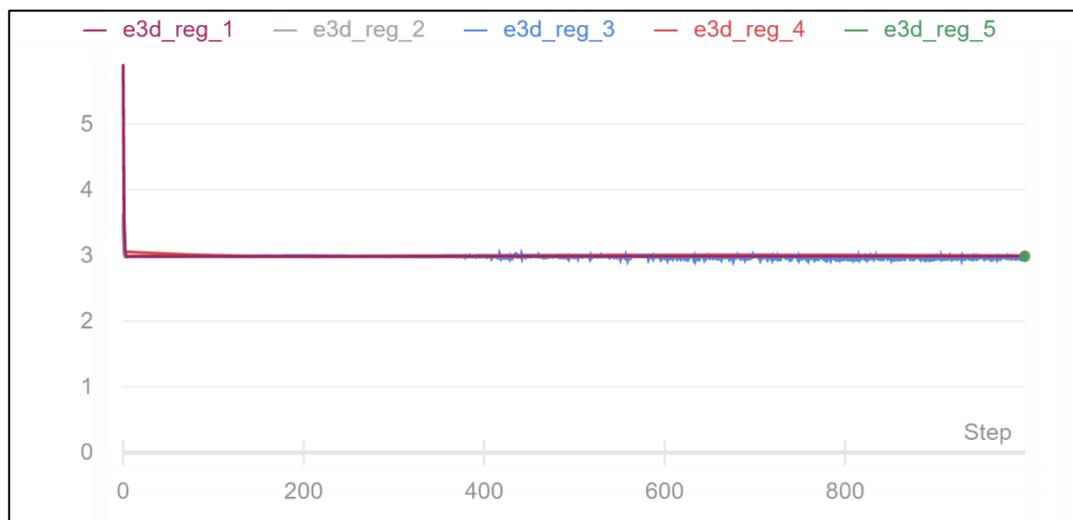
Fonte: o próprio autor.

Gráfico 23 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável intervalo desmame cio.



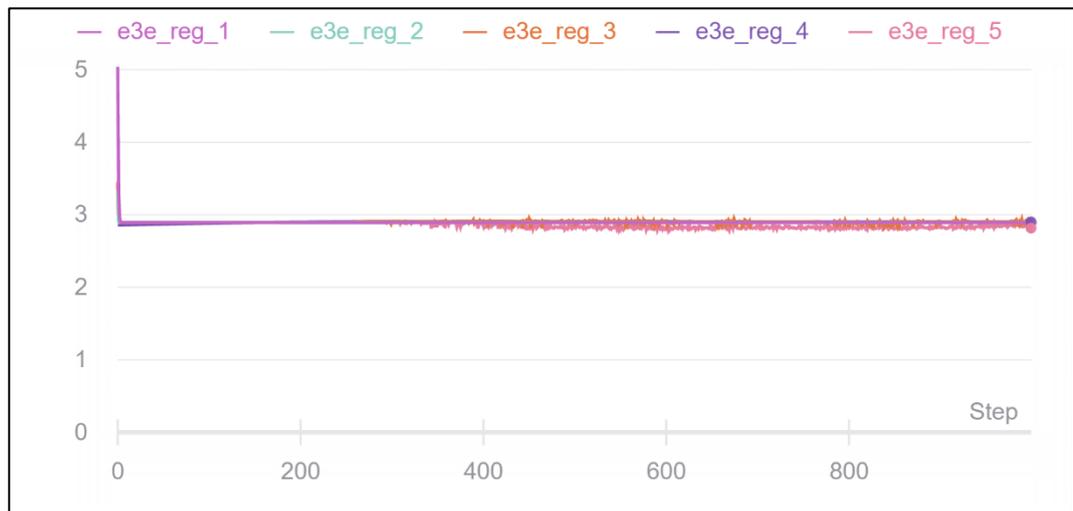
Fonte: o próprio autor.

Gráfico 24 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável “número de nascidos totais”.



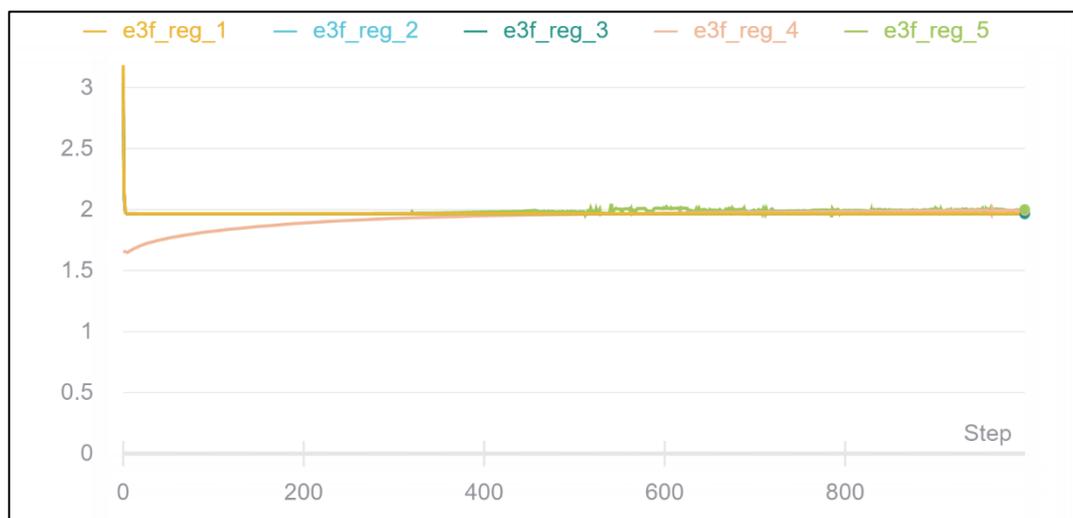
Fonte: o próprio autor.

Gráfico 25 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável “número de nascidos vivos”.



Fonte: o próprio autor.

Gráfico 26 – MAE por época das redes neurais utilizadas no experimento 3, para estimativa da variável “número de nascidos totais”.



Fonte: o próprio autor.

Algo evidente em todas as variações de redes neurais é que as diferentes arquiteturas testadas tiveram resultados muito semelhantes e que, em vários casos, o comportamento anormal relatado nos experimentos 1 e 2 voltou a se repetir neste. Nenhuma arquitetura obteve o melhor desempenho em todos os testes realizados. O erro médio absoluto em todas as redes neurais, principalmente as que avaliaram a duração de gestação e duração de lactação, é alto demais para que as redes neurais sejam consideradas aplicáveis a situações do mundo real.

4.5.8 Resultados do experimento 4

As acurácias das redes neurais utilizadas no experimento 4 são descritas na tabela a seguir.

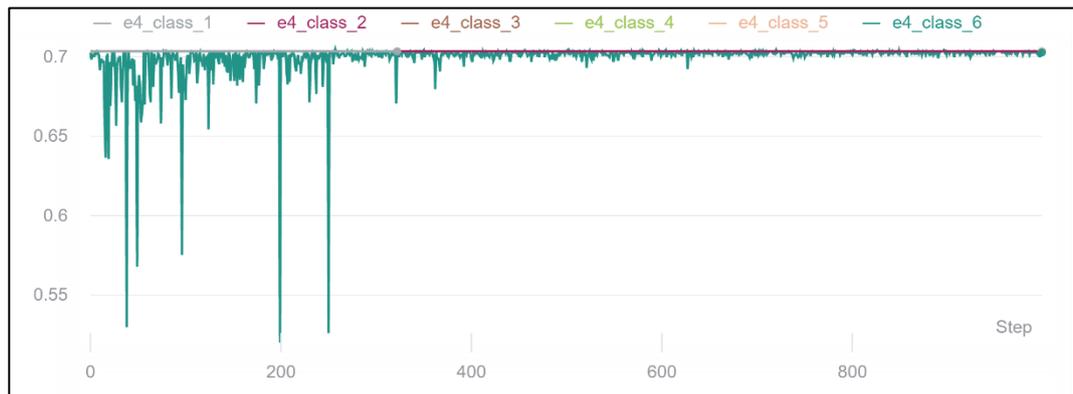
Tabela 25 – Acurácia das redes neurais utilizadas na base de dados do experimento 4.

Nome	Melhor acurácia	Tempo de treinamento
Arq_class_1	0,7034	978 segundos
Arq_class_2	0,7034	995 segundos
Arq_class_3	0,7034	997 segundos
Arq_class_4	0,7034	1034 segundos
Arq_class_5	0,7034	1005 segundos
Arq_class_6	0,7042	1009 segundos

Fonte: o próprio autor.

A melhor acurácia alcançada pelas seis redes neurais testadas foi de 70,42%, com uma rede neural cuja arquitetura correspondia à Arq_class_6. A rede neural com melhor acurácia ficou apenas 0,08% acima das outras 5 redes, indicando que o baixo desempenho observado está muito provavelmente relacionado à base de dados utilizada. A acurácia por época das redes é mostrada no gráfico abaixo.

Gráfico 27 – Acurácia por época das redes neurais utilizadas no experimento 4.



Fonte: o próprio autor.

Com a exclusão da Arq_class_6, todas as arquiteturas apresentaram pouca variação na acurácia, ao longo dos 1.000 ciclos de treinamento.

4.6 DISCUSSÃO

As acurácias alcançadas pelas redes neurais do experimento 1 foram relativamente baixas, se comparadas com trabalhos presentes na literatura (BRETHOUR, 1994; HEALD et al., 2000; CRANER; ROUSH, 1999; SANZOGNI; KERR, 2001; YU; LEUNG; BIENFANG, 2005; GANDHI; RAJA; RUHIL; KUMAR, 2009; GORGULU, 2012; NJUBI; WAKHUNGU; BADAMANA, 2009; OSORIO et al., 2009; SALAWU et al., 2014; SAVEGNAGO et al., 2011; SEDGHI et al., 2012). Uma possível explicação para tal fato pode ser o baixo número de dados disponíveis para treinamento de determinadas categorias de motivos, como a 4, 3, 7 e 2, que incluíam 285, 133, 18 e 6 fêmeas, respectivamente, na base de dados original. Esse número de dados ficou ainda menor após o processo de filtragem de dados. A incapacidade de classificar corretamente essas matrizes pela falta de dados de treinamento pode ter resultado na baixa acurácia observada. Ainda, a classificação de diversos motivos de descarte em distintas categorias foi realizada utilizando o conhecimento disponível na literatura sobre as patologias e alterações fisiológicas citadas, porém é possível que alguns motivos não tenham relações com os outros incluídos na mesma categoria, prejudicando, então, a acurácia das predições.

Outra possível razão para uma baixa acurácia é o fato de que algumas fêmeas incluídas na base de dados foram descartadas por motivos que dependiam de critérios dos seus produtores e que não possuíam nenhuma relação com variáveis biológicas, mas sim com os protocolos adotados em suas granjas. Para que isso não seja um problema, uma alternativa é incluir um número relativo à granja nas variáveis de entrada, de modo que a rede neural aprenda os critérios adotados pelos produtores delas.

Em uma situação real, um usuário da rede neural alimentaria somente com dados provenientes de sua granja, não tendo que lidar com o problema descrito acima. Eventualmente, a rede neural aprenderia os critérios do produtor e usaria eles para determinar o descarte das matrizes, porém um problema surgiria se esses critérios mudassem ao longo do tempo, o que geralmente ocorre em granjas que investem no melhoramento genético de suas matrizes. Uma alternativa para isso seria utilizar redes neurais de regressão para prever os índices produtivos em si, para apenas após isso definir, segundo critérios que o produtor desejar, o descarte ou não de uma matriz. O experimento 3 foi planejado com esse intuito, porém não alcançou resultados bons o suficiente para ser aplicável em seu estado atual.

Algumas redes neurais do experimento 3 retornavam valores fixos independente das variáveis de entrada da base de dados de validação, provavelmente pois a rede neural aprendeu que esse número fixo apresentava menores erros médios absolutos do que configurações de

pesos que tentassem estimar corretamente os valores de saída. Muitas variáveis de entrada apresentavam valores que variavam muito pouco entre si, isso talvez tenha diminuído a eficácia da rede neural e favorecido a situação descrita acima.

Ao eliminar a necessidade de classificação entre sete motivos específicos, o experimento 2 apresentou acurácias satisfatórias e suas redes neurais poderiam, então, ser utilizadas em uma situação real para prever o motivo de descarte de matrizes suínas. Deve-se observar, no entanto, que as categorias de motivos 4,3, 7 e 2 apresentam um número de dados extremamente baixo e que, mesmo que fossem erroneamente classificadas como falsos negativos, as acurácias da base de dados de validação ainda assim seriam altas. Em futuros trabalhos, o número de dados nas duas categorias testadas em cada experimentos deverão ser iguais, de modo a eliminar essa complicação. Essa análise levanta um questionamento sobre o uso de dados provenientes de uma granja para alimentar automaticamente uma rede neural. É possível perceber que, se a base de dados não for cuidadosamente manipulada e preparada para o treinamento, é possível que dados adicionais tenham efeitos danosos à eficácia da rede neural.

Diversas redes neurais, durante o experimento para determinar a melhor arquitetura, apresentaram um comportamento que pode ser classificado como o problema de dissipação do gradiente, onde as taxas de incremento à acurácia ou decréscimo do erro médio absoluto se tornam cada vez menores a cada ciclo de treinamento. (DATA SCIENCE ACADEMY, 2019). Esse tipo de problema é comum quando as funções de ativação *sigmoid* e *tanh* são utilizadas, mas foi observado em arquiteturas que utilizavam outras funções de ativação também (AGGARWAL, 2018). Ainda, a função de ativação parece ter tido pouca influência no aparecimento dessa complicação, sendo que o principal responsável parece ser o número de camadas e nódulos. Para ambos os experimentos de classificação e regressão, as arquiteturas com cinco camadas apresentaram o melhor desempenho e o motivo para isso não parece claro (VASILEV et al., 2019).

As funções de otimização testadas durante o experimento para determinar a melhor arquitetura tiveram seus padrões definidos no começo e estes não foram modificados. É sabido que a modificação dos parâmetros das funções de otimização pode ter efeitos bastantes benéficos à eficácia das redes neurais (AGGARWAL, 2018). Em futuros trabalhos, modificações nesses parâmetros devem ser feitas e sua influência nas acurácias e erros médios absolutos devem ser medidas.

Todas as camadas das diferentes arquiteturas testadas possuíam um número de nódulos que era algum múltiplo de dois, devido a recentes relatos na literatura que indicam que esta configuração gera melhores resultados, independente do número de camadas (LIN;

TEGMARK; ROLNICK, 2017). Apesar disso, em futuros trabalhos deve ser feito um esforço para testar diferentes quantidades de nódulos que não sigam esse padrão, tanto para testar a hipótese proposta no parágrafo anterior, quanto para descobrir novas configurações que apresentem resultados melhores.

Do contrário do que se esperaria, segundo a literatura atual que trata de funções de ativação, as arquiteturas lineares apresentaram melhor acurácia e menor erro médio absoluto do que as outras funções (GOYAL; PANDEY; JAIN, 2018). Apesar disso, essas mesmas arquiteturas apresentaram também maior variabilidade desses parâmetros ao longo dos ciclos de treinamento, indicando que talvez seja necessário diminuir a taxa de aprendizado das funções de otimização quando essas funções de ativação são utilizadas.

Sabe-se que o aumento do número de variáveis de entrada dificilmente tem um efeito negativo na acurácia das redes neurais, então se utilizadas bases de dados de granjas que coletam mais informações sobre suas matrizes, talvez a acurácia de classificação alcançada pelas mesmas redes neurais do presente trabalho possa ser maior (LACROIX; SALEHI; YANG, 1997; PARK; CHEN; NGUYEN, 1998; KOMINAKIS; ABAS; MALTARIS, 2002).

Apesar de tudo isso, o crescimento de seres vivos é um processo complexo e de dinâmica não linear, não sendo facilmente modelado por algoritmos de computadores. A não-linearidade desses processos dificulta o processo de aprendizagem e talvez o número de etapas biológicas pode estar associado ao número de nódulos e camadas das redes (MILOSEVIC, 2019).

Comparado com análises de regressão e suas variações, redes neurais artificiais não requerem nenhum conhecimento do problema que tenta resolver. Além disso, os ajustes nas relações e padrões dos dados é feito de forma automática e de modo a alcançar o melhor resultado final possível, sem a necessidade de constante supervisão do usuário. Por isso, redes neurais são uma ferramenta mais adequada para estimativa e previsão de dados do que métodos de análise lineares (NJUBI; WAKHUNGU; BADAMANA, 2010).

4.5 CONCLUSÃO

As diferentes arquiteturas de redes neurais analisadas e utilizadas no presente trabalho foram capazes de, ao serem aplicadas a um banco de dados de matrizes suínas provenientes de granjas de quarto sítio e unidades produtoras de leitões, prever não só a ocorrência de descarte no parto seguinte dessas fêmeas, como também o provável motivo desses descartes, com acurácias que variaram de 81% a 99%. Futuros trabalhos devem analisar a influência da

modificação de parâmetros das funções de otimização e determinar métodos para solucionar problemas relacionados a diferentes números de amostras para cada categoria nas bases de dados.

5. CONCLUSÕES

As redes neurais apresentam um ótimo desempenho na tarefa de encontrar padrões não-lineares e realizar previsões baseadas em bases de dados que tenham um número adequado de amostras e sejam previamente organizadas de maneira adequada. A quantidade de trabalhos relacionadas à inteligência artificial, voltados para a ciência e produção animal, aumentou consideravelmente nos últimos anos, demonstrando quão promissora pode ser essa ferramenta de análise.

Há uma evidente barreira a ser quebrada para a utilização dessa tecnologia e ela está relacionada com a dificuldade de acesso e domínio de conhecimentos da área de computação e matemática, mas diversos trabalhos com autores de diferentes áreas acadêmicas mostram que a sua utilização nas mais diversas áreas das ciências veterinárias é possível.

A utilização de redes neurais para a previsão da ocorrência e do tipo de motivo de descarte de matrizes suínas provenientes de granjas de quarto sítio e unidades produtoras de leitões é possível. A utilização de diferentes parâmetros e arquiteturas de redes neurais devem ser testadas em trabalhos futuros para melhorar a eficiência dos modelos.

REFERÊNCIAS

ACHARYA, U.; RAJENDRA, O.H.; SHU, L. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. **Computers in Biology and Medicine**, v. 100, p. 270-278, 2018.

AGGARWAL, C. C. Neural Networks and Deep Learning. 1. ed. Springer International Publishing, 2018.

ANTHIMOPOULOS, M.; CHRISTODOULIDIS, S.; EBNER, L.; CHRISSTE, A.; MOUGIAKAKOU, S. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1207-1216, 2016.

BENYUS, J. Biomimicry: Innovation Inspired by Nature. 1. ed. Nova Iorque: HarperCollins Publishers Inc., 1997.

B. LACEY, T. K. HAMRITAR. MCCLENDON. Feasibility of Using Neural Networks for Real-Time Prediction of Poultry Deep Body Temperature Responses to Stressful Changes in Ambient Temperature. **Applied Engineering in Agriculture**, v. 16, n. 3, p. 303-308, 2000.

BONIECKI, P.; DACH, J.; MUELLER, W.; KOSZELA, K.; PRZYBYL, J.; PILARSKI, K.; OLSZEWSKI, T. Neural prediction of heat loss in the pig manure composting process. **Applied Thermal Engineering**, v. 58, n. 1-2, p. 650-655, 2013.

BRETHOUR, J. R. Estimating marbling score in live cattle from ultrasound images using pattern recognition and neural network procedures1. **Journal of Animal Science**, v. 72, n. 6, p. 1425-1432, 1994.

BRIEF HISTORY OF NEURAL NETWORKS. Medium. Disponível em: <<https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec>>. Acesso em: 11 dez. 2019.

CHEDAD, A.; MOSHOU, D; AERTS, J.M. AP—Animal Production Technology. **Journal of Agricultural Engineering Research**, v. 79, n. 4, p. 449-457, 2001.

- CHERRY, R. H. History of Sericulture. **Bulletin of the Entomological Society of America**, v. 33, n. 2, p. 83-85, 1987.
- CIREŞAN, D.; MEIER, U.; JONATHAN, M. Multi-column deep neural network for traffic sign classification. **Neural Networks**, v. 32, p. 333-338, 2012.
- CORKERY, G.P.; GONZALES-BARRONF, U. A.; BUTLER F.; DONNELL K. M. A Preliminary Investigation on Face Recognition as a Biometric Identifier of Sheep. **Transactions of the ASABE**, v. 50, n. 1, p. 313-320, 2007.
- CRAIG, C. L.; BERNARD, G. D. Insect Attraction to Ultraviolet-Reflecting Spider Webs and Web Decorations. **Ecology**, v. 71, n. 2, p. 616-623, 1990.
- CRAVENER, T.R.; ROUSH, W. Improving neural network prediction of amino acid levels in feed ingredients. **Poultry Science**, v. 78, n. 7, p. 983-991, 1999.
- DATA SCIENCE ACADEMY. Deep Learning Book, 2019. Disponível em: <<http://www.deeplearningbook.com.br/>>. Acesso em: 10 Outubro. 2019.
- DYBOWSKI, R.V. Artificial neural networks in pathology and medical laboratories. **The Lancet**, v. 346, n. 8984, p. 1203-1207, 1995.
- DYBOWSKI, R.V. Clinical Applications of Artificial Neural Networks. Leiden: Cambridge University Press, 2007.
- EISNER, T.; NOWICKI, S. Spider Web Protection Through Visual Advertisement: Role of the Stabilimentum. **Science**, v. 219, n. 4581, p. 185-187, 1983.
- EL-HENAWY, I.; BAKRY, H.; HADAD, H. Cattle Identification Using Segmentation-based Fractal Texture Analysis and Artificial Neural Networks. **Int. J. of Electronics and Information Engineering**, vol.4, n.2, p.82-93, 2016

FARIDI, A.; MOTTAGHITALAB, M.; DARMANI-KUHI, H. Predicting carcass energy content and composition in broilers using the group method of data handling-type neural networks. **The Journal of Agricultural Science**, v. 149, n. 2, p. 249-254, 2010.

FEITOSA, F.L.F. Semiologia veterinária. 3. ed. [s.l.]: Grupo Gen - Editora Roca Ltda., 2014.

GANDHI, R. S.; RAJA, T. V.; RUHIL, A. P. Artificial Neural Network versus Multiple Regression Analysis for Prediction of Lifetime Milk Production in Sahiwal Cattle. **Journal of Applied Animal Research**, v. 38, n. 2, p. 233-237, 2010.

GHOTBALDINI, H.; MOHAMMADABADI, M.; NEZAMABADI-POUR, H. Predicting breeding value of body weight at 6-month age using Artificial Neural Networks in Kermani sheep breed. **Acta Scientiarum. Animal Sciences**, v. 41, n. 1, p. 45282, 2019.

GORGULU, O. Prediction of 305-day milk yield in Brown Swiss cattle using artificial neural networks. **South African Journal of Animal Science**, v. 42, n. 3, 2012.

GOYAL, P.; PANDEY, S. K. Deep learning for natural language processing. Berkeley, CA: Apress, 2018.

HAN, J.; CHEWNING, S. K.; WURTZ, K.E.; SIEGFORD, J.; STEIBEL, J.P. 65 Using 3D images and deep learning to predict feeder occupancy in grow-finish pigs. **Journal of Animal Science**, v. 97, n. Supplement_2, p. 33-33, 2019.

HANSEN, C. Activation Functions Explained - GELU, SELU, ELU, ReLU and more. Machine Learning From Scratch. Disponível em: <https://mlfromscratch.com/activation-functions-explained/>. Acesso em: 2 jan. 2020.

HEALD, C.W.; KIM, T.; SISCHO, W.M. *et al.* A Computerized Mastitis Decision Aid Using Farm-Based Records: An Artificial Neural Network Approach. **Journal of Dairy Science**, v. 83, n. 4, p. 711-720, 2000.

HOCHREITER, S.; SEPPSCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735-1780, 1997.

JANSSENS, O.; SLAVKOVIKJ, V.; VERVISH, B. *et al.* Convolutional Neural Network Based Fault Detection for Rotating Machinery. **Journal of Sound and Vibration**, v. 377, p. 331-345, 2016.

KOMINAKIS, A.P.; ABAS, Z.; MALTARIS, I. *et al.* A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep. **Computers and Electronics in Agriculture**, v.35, p.35-48, 2002.

KONTORAVDIS, D.; LIKAS, A.; KRAKITSOS, P. Cytological Diagnosis Based on Fuzzy Neural Networks. **Journal of Intelligent Systems**, v. 8, n. 1-2, 1998.

KRAUSE, D. O. Diet influences the ecology of lactic acid bacteria and Escherichia coli along the digestive tract of cattle: neural networks and 16S rDNA. **Microbiology**, v. 149, n. 1, p. 57-65, 2003.

KUAN, C. Y.; TSAI, Y. C.; HSU, J. T. *et al.* An Imaging System Based on Deep Learning for Monitoring the Feeding Behavior of Dairy Cows. 2019 Boston, Massachusetts July 7- July 10, 2019, 2019.

LACROIX, R.; SALEHI, F.; YANG, X.Z. *et al.* Effects of data preprocessing on the performance of artificial neural networks for dairy yield prediction and cow culling classification. **Transactions of the ASAE**, v.40, n.3, p.839-846, 1997.

LAMPERT, C. H. Kernel Methods in Computer Vision. **Foundations and Trends® in Computer Graphics and Vision**, v. 4, n. 3, p. 193-285, 2007.

LECUN, Y.; BOSER, B.; DENKER, J. S. *et al.* Backpropagation Applied to Handwritten Zip Code Recognition. **Neural Computation**, v. 1, n. 4, p. 541-551, 1989.

LECUN, Y.; BOTTOU, L.; BENGIO, Y. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998.

LI, Z.; SHEN, S.; GE, C. *et al.* Cow Individual Identification Based on Convolutional Neural Network. **Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence - ACAI 2018**, 2018.

LIU, Y.; ZENG, Q.; SHAOCHENG, T. *et al.* Adaptive Neural Network Control for Active Suspension Systems With Time-Varying Vertical Displacement and Speed Constraints. **IEEE Transactions on Industrial Electronics**, v. 66, n. 12, p. 9458-9466, 2019.

LIU, Y.; TANG, J.; RONGHUA, W. *et al.* Artificial lotus leaf structures from assembling carbon nanotubes and their applications in hydrophobic textiles. **J. Mater. Chem.**, v. 17, n. 11, p. 1071-1078, 2007.

LIN, H.W.; TEGMARK, M.; ROLNICK, D. Why Does Deep and Cheap Learning Work So Well?. **Journal of Statistical Physics**, v. 168, n. 6, p. 1223-1247, 2017.

LUCIA, T.; DIAL, G. D.; MARSH, W. E. Lifetime reproductive performance in female pigs having distinct reasons for removal. **Livestock Production Science**, v. 63, n. 3, p. 213-222, 2000.

MANN, E. E.; MANNA, D.; MICHAEL R. *et al.* Surface micropattern limits bacterial contamination. **Antimicrobial Resistance and Infection Control**, v. 3, n. 1, p. 28, 2014.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biology**, v. 52, n. 1-2, p. 99-115, 1990.

MCCULLOCH, W. S.; PITTS, W. The Statistical Organization of Nervous Activity. **Biometrics**, v. 4, n. 2, p. 91, 1948.

MCVAY, G. Biomimicry, a New Method for Materials Synthesis. **Materials and Processing Report**, v. 5, n. 3, p. 1-3, 1990.

MWANGA, G.; LOCKWOOD, S.; MUJIBI, D. F. N. *et al.* Machine learning models for predicting the use of different animal breeding services in smallholder dairy farms in Sub-Saharan Africa. **Tropical Animal Health and Production**, 2019.

OSORIO, M.T.; ZUMALACÁRREGUI, J.M.; ALAIZ-RODRÍGUEZ, R. *et al.*

Differentiation of perirenal and omental fat quality of suckling lambs according to the rearing system from Fourier transforms mid-infrared spectra using partial least squares and artificial neural networks analysis. **Meat Science**, v. 83, n. 1, p. 140-147, 2009.

OSOWSKI, S.; TRAN, H.L. ECG beat recognition using fuzzy hybrid neural network. **IEEE Transactions on Biomedical Engineering**, v. 48, n. 11, p. 1265-1271, 2001.

NÄÄS, I. DE A.; CAMPOS, L. S. L.; BARACHO, M. DOS S. *et al.* Uso de redes neurais artificiais na identificação de vocalização de suínos. **Engenharia Agrícola**, v. 28, n. 2, p. 204-216, 2008.

NEURAL NETWORKS - HISTORY. Cs.stanford.edu. Disponível em:
<<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html>>. Acesso em: 11 dez. 2019.

PANDORFI, H.; SILVA, I. J.; OLIVEIRA, S.; HAUSEN, V. C. R. *et al.* Uso de redes neurais artificiais para predição de índices zootécnicos nas fases de gestação e maternidade na suinocultura. **Revista Brasileira de Zootecnia**, v. 40, n. 3, p. 676-681, 2011.

PARK, B., CHEN, Y.R.N.; GUYEN, M. Multi-spectral Image Analysis using Neural Network Algorithm for Inspection of Poultry Carcasses. **Journal of Agricultural Engineering Research**, v. 69, n. 4, p. 351-363, 1998.

PASTELL, M.E.; KUJALA, M. A Probabilistic Neural Network Model for Lameness Detection. **Journal of Dairy Science**, v. 90, n. 5, p. 2283-2292, 2007.

PEROL, T.; GHARBI, M.; DENOLLE, M. Convolutional neural network for earthquake detection and location. **Science Advances**, v. 4, n. 2, p. e1700578, 2018.

PIERCE, J.R *et al.* Language and Machines — Computers in Translation and Linguistics. ALPAC report, **National Academy of Sciences, National Research Council**, Washington, DC, 1966.

PIGCHAMP. Benchmarking PigCHAMP 2009: year end summary. Disponível em: <<https://www.pigchamp.com/Portals/0/Documents/Benchmarking%20Summaries/USA%202009.pdf>>. Acesso em: 10 Setembro. 2019.

PITTS, W. The linear theory of neuron networks: The dynamic problem. **The Bulletin of Mathematical Biophysics**, v. 5, n. 1, p. 23-31, 1943.

PLÀ-ARAGONÉS, L. M. Handbook of Operations Research in Agriculture and the Agri-Food Industry. New York, NY: Springer New York, 2015.

ROUSH, W.B.; CRAVENER, T.; KIRBY, Y. *et al.* Probabilistic neural network prediction of ascites in broilers based on minimally invasive physiological factors. **Poultry Science**, v. 76, n. 11, p. 1513-1516, 1997.

ROUSH, W. B.; KIRBY, Y. K.; CRAVENER, T. L. *et al.* Artificial Neural Network Prediction of Ascites in Broilers. **Poultry Science**, v. 75, n. 12, p. 1479-1487, 1996.

ROUSH, W. B.; WIDEMAN, R. F.; CAHANER, A. *et al.* Minimal Number of Chicken Daily Growth Velocities for Artificial Neural Network Detection of Pulmonary Hypertension Syndrome (PHS). **Poultry Science**, v. 80, n. 3, p. 254-259, 2001.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386-408, 1958.

SALAWU, E. O.; ABDULRAHEEM, M.; SHOYOMBO, A. *et al.* Using Artificial Neural Network to Predict Body Weights of Rabbits. **Open Journal of Animal Sciences**, v. 04, n. 04, p. 182-186, 2014.

SALLE, F. DE O.; BORGES F. B.F.; ROCHA, A. C. G. P. *et al.* Utilização de inteligência artificial (redes neurais artificiais) para a classificação do comportamento bioquímico de amostras de *Escherichia coli* isoladas de frangos de corte. **Acta Scientiae Veterinariae**, v. 38, n. 1, p. 59, 2018.

SANTOS, A. M. DOS.; SEIXAS, J. M. DE P.; PEREIRA, B. B.; MEDRONHO, R.A. Usando redes neurais artificiais e regressão logística na predição da Hepatite A. **Revista Brasileira de Epidemiologia**, v. 8, n. 2, p. 117-126, 2005.

SANZOGNI, L. D. Milk production estimates using feed forward artificial neural networks. **Computers and Electronics in Agriculture**, v. 32, n. 1, p. 21-30, 2001.

SAVEGNAGO, R. P.; NUNES, B. N.; CAETANO, S. L. *et al.* Comparison of logistic and neural network models to fit to the egg production curve of White Leghorn hens. **Poultry Science**, v. 90, n. 3, p. 705-711, 2011.

SAXENA, P. Y. Application of Artificial Neural Network (ANN) for Animal Diet Formulation Modeling. **Procedia Computer Science**, v. 152, p. 261-266, 2019.

SHAHINFAR, S.; MEHRABANI-YEGANEH, H.; CARO, L. *et al.* Prediction of Breeding Values for Dairy Cattle Using Artificial Neural Networks and Neuro-Fuzzy Systems. **Computational and Mathematical Methods in Medicine**, v. 2012, p. 1-9, 2012.

SHEIKHOLESAMI, M.; GERDROODBARY, M.; BARZEGARMORADI, R. *et al.* Application of Neural Network for estimation of heat transfer treatment of Al₂O₃-H₂O nanofluid through a channel. **Computer Methods in Applied Mechanics and Engineering**, v. 344, p. 1-12, 2019.

SOHN, J.H.; SMITH, R. E. *et al.* Quantification of Odours from Piggery Effluent Ponds using an Electronic Nose and an Artificial Neural Network. **Biosystems Engineering**, v. 86, n. 4, p. 399-410, 2003.

SORNAM, M.; PRABHAKARAN, M. Logit-Based Artificial Bee Colony Optimization (LB-ABC) Approach for Dental Caries Classification Using a Back Propagation Neural Network. **Integrated Intelligent Computing, Communication and Security**, p. 79-91, 2018.

TORRES, M. MARTINEZ, C.H.; AMADOR, F. Approximating the sheep milk production curve through the use of artificial neural networks and genetic algorithms. **Computers & Operations Research**, v. 32, n. 10, p. 2653-2670, 2005.

VASILEV, I.; SLATER, D.S; GIANMARIO *et al.* Python Deep Learning. Birmingham: Packt Publishing Ltd, 2019.

VENTURA, R.V.; SILVA, M.A.; MEDEIROS, T.H. *et al.* Uso de redes neurais artificiais na predição de valores genéticos para peso aos 205 dias em bovinos da raça Tabapuã. **Arquivo Brasileiro de Medicina Veterinária e Zootecnia**, v. 64, n. 2, p. 411-418, 2012.

WACHINGER, C.; REUTER, M.; KLEIN, T. DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. **NeuroImage**, v. 170, p. 434-445, 2018.

WANG, K.; JIANG, T.; LIANG, M. *et al.* Discriminative Analysis of Early Alzheimer's Disease Based on Two Intrinsically Anti-correlated Networks with Resting-State fMRI. **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006**, p. 340-347, 2006.

WANG, E.; SAMARASINGHE, S. On-line detection of mastitis in dairy herds using artificial neural networks. Researcharchive.lincoln.ac.nz. Disponível em: <<https://researcharchive.lincoln.ac.nz/handle/10182/5444>>. Acesso em: 17 nov. 2019.

WANG, Y.; YANG, W.; WINTER, P. *et al.* Walk-through weighing of pigs using machine vision and an artificial neural network. **Biosystems Engineering**, v. 100, n. 1, p. 117-125, 2008.

WEIGHTS & BIASES. Docs.wandb.com. Disponível em: <<https://docs.wandb.com>>. Acesso em: 11 dez. 2019.

WERBOS, P. J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D, Harvard University, 1974.

WIDROW, B.; HUFF, W. E. Adaptive Switching Circuits. California: Stanford University, 2019.

XIN, H. Assessing Swine Thermal Comfort by Image Analysis of Postural Behaviors. **Journal of Animal Science**, v. 77, n. suppl_2, p. 1, 1999.

YADAV, N.; YADAV, A.; KUMAR, M.. An introduction to neural network methods for differential equations. Dordrecht: Springer, 2015.

YANG, X. Z.; LACROIX, R.; WADE, K. M. Investigation into the production and conformation traits associated with clinical mastitis using artificial neural networks. **Canadian Journal of Animal Science**, v. 80, n. 3, p. 415-426, 2000.

YAO, L.; HU, Z.; LIU, C. *et al.* Cow face detection and recognition based on automatic feature extraction algorithm. Proceedings of the ACM Turing Celebration Conference - China on - ACM TURC '19, 2019.

YU, R.; LEUNG, P.; BIENFANG, P. Predicting shrimp growth: Artificial neural network versus nonlinear regression models. **Aquacultural Engineering**, v. 34, n. 1, p. 26-32, 2006.

ZHANG, L.; YANG, F.Z.; ZHANG, Y. D. *et al.* Road crack detection using deep convolutional neural network. **2016 IEEE International Conference on Image Processing (ICIP)**, 2016.