

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA / INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM ENGENHARIA COMPUTACIONAL

Gabriel Rezende da Silva

**Classificação de imagens de sobras de dieta de bovinos confinados através de
redes neurais convolucionais**

Juiz de Fora

2022

Gabriel Rezende da Silva

**Classificação de imagens de sobras de dieta de bovinos confinados através de
redes neurais convolucionais**

Monografia apresentado ao Curso de Graduação em Engenharia Computacional da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Computacional.

Orientador: Prof. Dr. Luiz Maurílio da Silva Maciel

Coorientadora: Prof.(a) Dr.(a) Amália Saturnino Chaves

Juiz de Fora

2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Silva, Gabriel da.

Classificação de imagens de sobras de dieta de bovinos confinados através
de redes neurais convolucionais / Gabriel Rezende da Silva. – 2022.

40 f. : il.

Orientador: Luiz Maurílio da Silva Maciel

Coorientadora: Amália Saturnino Chaves

Trabalho de Conclusão de Curso – Universidade Federal de Juiz de Fora,
Faculdade de Engenharia / Instituto de Ciências Exatas. Bacharelado em
Engenharia Computacional, 2022.

1. Nutrição animal. 2. Escore de Cocho. 3. Visão Computacional. 4.
Aprendizado Profundo. 5. Redes neurais convolucionais. I. Maciel, Luiz
Maurílio, orient. II. Chaves, Amália, coorient. III. Título.

Gabriel Rezende da Silva

Classificação de imagens de sobras de dieta de bovinos confinados através de redes neurais convolucionais

Monografia apresentado ao Curso de Graduação em Engenharia Computacional da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Computacional.

Aprovada em 17 de fevereiro de 2022

BANCA EXAMINADORA

Prof. Dr. Luiz Maurílio da Silva Maciel - Orientador
Universidade Federal de Juiz de Fora

Prof.(a) Dr.(a) Amália Saturnino Chaves - Coorientadora
Universidade Federal de Juiz de Fora

Prof. Dr. Marcelo Bernardes Vieira
Universidade Federal de Juiz de Fora

Prof. Dr. Saulo Moraes Villela
Universidade Federal de Juiz de Fora

Aos meus pais, que sempre me apoiaram nas minhas decisões, acreditaram e investiram em mim, e à minha irmã, que sempre me trouxe paz.

AGRADECIMENTOS

Aos meus pais, Lúcia e Jorge, à minha irmã, Patrícia, e à minha namorada, Virginia, por todo o amor. Aos meus amigos, por todo o apoio emocional.

Aos professores orientadores, Luiz Maurílio e Amália, pela oportunidade da iniciação científica, e a todos os professores, por todo o conhecimento compartilhado. Aos servidores, que de alguma forma me ajudaram, em especial Claudio Franco, pela oportunidade do treinamento profissional e pela amizade.

Às minhas amigadas construídas durante a graduação, em especial Pedro Henrique, Davi Rezende, Mathews Edwirds e Brian Maia, por todos os projetos realizados e pelo companheirismo.

“Our intelligence is what makes us human, and AI is an extension of that quality.” – (Yann LeCun)

RESUMO

A nutrição animal é uma das principais áreas responsáveis pela eficiência de produção em sistemas de confinamento. A formulação inconsistente da dieta e o manejo inadequado dos cochos de alimentação pode prejudicar a saúde dos animais e a produção. Durante anos, o manejo dos cochos foi conduzido de forma subjetiva, através de métodos manuais e visuais de classificação. O aperfeiçoamento contínuo de aprendizado profundo e redes neurais artificiais tem contribuído positivamente para problemas em diversos campos com o auxílio de técnicas de visão computacional. O presente trabalho propõe um modelo computacional capaz de analisar diversas imagens de sobras de dietas de bovinos confinados, a fim de classificar essas imagens entre as pontuações estabelecidas pela literatura. Assim, foi desenvolvida uma rede neural convolucional para classificar as imagens de escores de cocho, fornecendo uma acurácia de 89,58%. Além disso, o modelo final atingiu métricas de desempenho computacional favoráveis para uma aplicação de baixo custo e de tempo real para produtores rurais.

Palavras-chave: Nutrição animal. Escore de cocho. Visão Computacional. Aprendizado profundo. Redes neurais convolucionais.

ABSTRACT

Animal nutrition is an important field in livestock production. An inappropriate formulation and bunk management can harm the animals health and the production. For years the feed bunk management was conducted subjectively and inconsistently, through manual and visual methods of classification. The continuous improvement of artificial neural networks and deep learning have contributed positively to problems in several fields with the use of computer vision techniques. The present work proposes a computational model capable of analyzing various images of leftover diets of confined bovines, in order to classify these images between scores established by the literature. Thus, a convolutional neural network was developed to classify bunk score images, achieving 89.58% of accuracy. In addition, the final model achieved computational performance metrics favorable for a low cost and a real time application for farmers.

Keywords: Animal nutrition. Bunk score. Computer Vision. Deep learning. Convolutional neural networks.

LISTA DE ILUSTRAÇÕES

Figura 1 - Esquema de um <i>perceptron</i>	18
Figura 2 - Gráficos de E em função de wx	19
Figura 3 - Exemplos de funções de ativação.	20
Figura 4 - Rede neural <i>feedforward</i> totalmente conectada (<i>fully connected</i>).	21
Figura 5 - Arquitetura da rede convolucional LeNet.	24
Figura 6 - Operações de convolução e <i>pooling</i>	25
Figura 7 - Imagens dos escores.	27
Figura 8 - Modelo de rede neural convolucional proposto.	29
Figura 9 - Geração das imagens dos escores com filtro passa-alta.	30
Figura 10 - Modelo de rede neural convolucional multi-entrada proposto.	31
Figura 11 - Matriz de confusão da melhor validação cruzada do modelo multi-entrada.	35
Figura 12 - Matriz de confusão por recomendação da melhor validação cruzada do modelo multi-entrada.	36

LISTA DE TABELAS

Tabela 1	– Sistema de Escore de Cocho da SDSU.	15
Tabela 2	– Distribuição das imagens do <i>dataset</i> por classe.	28
Tabela 3	– Síntese das variações dos experimentos.	32
Tabela 4	– Média (\bar{X}) e desvio padrão (dp) das acurácias gerais e por classe das 6 validações cruzadas (V.C.). Destacam-se as melhores validações cruzadas de cada modelo e a melhor média entre os modelos.	34
Tabela 5	– Médias (\bar{X}) das acurácias de validação e teste para análise de <i>overfitting</i>	35
Tabela 6	– Média (\bar{X}) e desvio padrão (dp) dos tempos de predição das 6 validações cruzadas, e tamanho dos modelos.	35

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
SDSU	<i>South Dakota State University</i>
MS	Matéria seca
MSE	<i>Mean squared error</i>
ReLU	<i>Rectified Linear Unit</i>
GAN	<i>Generative Adversarial Network</i>
RNN	<i>Recurrent Neural Network</i>
VAE	<i>Variotinal auto-encoder</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	ESCORE DE COCHO	15
2.2	REDES NEURAIAS ARTIFICIAIS E APRENDIZADO DE MÁQUINA .	16
2.2.1	<i>Perceptron</i>	16
2.2.2	Redes neurais multicamadas	19
2.2.3	Redes neurais convolucionais	23
3	ABORDAGEM PROPOSTA	27
3.1	CONJUNTO DE DADOS	27
3.2	MODELO PROPOSTO	28
4	EXPERIMENTOS COMPUTACIONAIS	32
5	CONCLUSÃO	37
	REFERÊNCIAS	38

1 INTRODUÇÃO

O Brasil possui o maior rebanho comercial de gado do mundo, com cerca de 214,7 milhões de animais, sendo o maior exportador de carne e o quinto maior produtor de leite bovino (IBGE, 2019). No entanto, essa superioridade numérica não reflete em eficiência no que tange à produtividade. Com a maior produção mundial, os Estados Unidos possuem cerca de 95 milhões de animais, menos da metade do rebanho brasileiro (FORMIGONI, 2020). A ingerência administrativa de propriedades do setor primário que utilizam registro manuscrito das informações e processos tem sido um dos fatores para inviabilizar o acesso a mercados mais competitivos (FONTOURA et al., 2018). Dessa forma, a coleta e armazenamento de dados é de extrema importância para o gerenciamento de qualquer negócio, uma vez que essa base de dados sustentará todos os sistemas de apoio e tomada de decisão, proporcionando a avaliação de indicadores de desempenho da produção (FONTOURA et al., 2018).

O uso de novas tecnologias nos processos de intensificação da produção faz com que as organizações tornem-se mais competitivas, o que não é diferente no campo (MACHADO; NANTES, 2011). Uma das práticas relacionadas à gestão de tecnologia implementadas na bovinocultura que interfere de forma positiva na produção é a adoção de sistemas confinados (BOM et al., 2020). Esses sistemas permitem a redução da idade e aumento do peso de abate dos animais, gerando melhor rendimento de carcaça e produção de carne de qualidade superior, além do retorno do capital investido em curto prazo de tempo e o descanso de áreas de pastagem durante a seca (SENAR, 2018). O uso do confinamento também evita flutuações da produção entre os períodos de inverno e verão, oferecendo um melhor conforto aos animais e um manejo nutricional especializado.

Pensando nisso, a nutrição animal é uma das principais áreas responsáveis pela eficiência de produção em sistemas confinados. Do ponto de vista econômico, um dos custos de maior influência na propriedade é a dieta, que demanda entre 60% e 80% dos custos variáveis (UYEH et al., 2018). Diferentes situações no manejo alimentar dos rebanhos interferem diretamente no resultado final de desempenho animal, pois a formulação inconsistente da dieta total e o manejo deficiente dos cochos de alimentação podem causar distúrbios digestivos que resultam em padrões erráticos de ingestão, reduzindo o ganho diário dos animais (LOY et al., 2016). Sendo assim, a utilização de meios que permitam avaliar se o manejo alimentar está sendo realizado corretamente é de grande importância (LUZ et al., 2019).

Dentro do manejo alimentar, o escore de cocho é uma ferramenta de observação visual utilizada nos confinamentos comerciais para avaliação das sobras de alimentos nos cochos por meio de notas ou escores (ALMEIDA; CANTRELL, 2001). Mensurar as sobras da dieta no cocho é um processo de grande importância e fornece informações para que

se avalie a necessidade de ajustes na formulação e fornecimento da dieta, evitando o desperdício de alimento e influenciando na produtividade do rebanho (LUZ et al., 2019). No entanto, essa avaliação é subjetiva, e a sua precisão depende diretamente da experiência da pessoa que realiza a análise (LUZ et al., 2019). Buscando alternativas para automatizar esse processo, técnicas computacionais podem ser empregadas a fim de padronizar as avaliações visuais de escore de cocho e contribuir para o ajuste adequado de dietas em confinamentos de bovinos.

Métodos não invasivos são úteis para coleta de dados na produção animal, pois não interferem diretamente no organismo que detém esta informação. Assim, o bem-estar dos animais é mantido e é possível otimizar a tomada de decisões relativas à utilização dos recursos naturais que compõem o processo de produção animal (LEÃO et al., 2015). Através de uma rede de computadores portáteis e um sistema de balança digital, Dyck et al. (2007) desenvolveram um software capaz de registrar o histórico de consumo de ração e prever correções com base em regras de gestão de alimentação em um sistema de confinamento de bovinos. O desempenho do software foi avaliado pela comparação das suas previsões com as feitas pelos funcionários da propriedade, obtendo um ajuste satisfatório da sobra de ração dentro de limites inferiores e superiores aceitáveis.

Técnicas de processamento de imagens (GONZALEZ; WOODS, 2018; HOSHIRO; GOLDSCHMIDT, 2008) e visão computacional (SZELISKI, 2010) têm sido aplicadas com grande sucesso em problemas das mais diversas áreas. Bloch, Levit e Halachmi (2019) apresentaram uma técnica para medir o consumo individual de ração de vacas leiteiras através de fotogrametria. O sistema proposto consiste no posicionamento de várias câmeras de diferentes resoluções em torno da pilha de ração antes e depois da alimentação do animal. A massa da ração é estimada através do seu volume, que pode ser calculado com o método de fotogrametria, utilizando o software *PhotoModeler Scanner*. A estimativa do volume leva de 20 a 30 minutos. Apesar da redução dos custos e do tempo gasto para estimar a ingestão individual de ração pelas vacas, a técnica é difícil de ser implementada e os melhores resultados foram obtidos sob condições laboratoriais controladas.

Em especial, nos últimos anos, tem-se obtido destaque em visão computacional termos como, aprendizado de máquina (*machine learning*) e aprendizado profundo (*deep learning*) (LECUN; BENGIO; HINTON, 2015), que estão associados à solução de problemas de aprendizado a partir de bases de imagens e vídeos. Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs) são modelos matemáticos e computacionais que têm sido amplamente utilizados em diversas aplicações, tais como diagnóstico médico (LI et al., 2014), análise meteorológica (ELHOSEINY; HUANG; ELGAMMAL, 2015) e sensoriamento remoto (MAGGIORI et al., 2016). Alguns trabalhos têm aplicado CNNs para resolver problemas em sistemas de produção animal. Yao et al. (2019) aplicaram CNNs no reconhecimento facial de vacas que combina detecção facial e identificação individual. Bezen, Edan e Halachmi (2020) apresentaram um trabalho baseado em modelos de CNNs

para identificar vacas e monitorar a ingestão individual de ração, utilizando uma câmera RGB-D de baixo custo para medir o tamanho das pilhas de ração. O consumo é obtido pela diferença entre os dados obtidos antes e depois do animal atingir a faixa de alimentação. Para isso, foi necessário treinar um modelo de CNN que calcula o peso da refeição e faz previsões com base nos resultados. É importante notar que foi preciso um sistema de aquisição específico para câmeras RGB-D.

Nesse contexto, o presente trabalho propõe uma arquitetura de Redes Neurais Convolucionais para o aprendizado supervisionado e predição do escore de cocho a partir de uma base de imagens pré-rotuladas. Espera-se que a arquitetura possa ser incorporada em uma aplicação para *smarthphones*, permitindo a utilização prática de uma ferramenta capaz de padronizar o processo de classificação do escore de cocho, além de garantir uma solução de baixo de custo. Assim, pode-se contribuir para tornar a gestão alimentar do rebanho de bovinos mais eficiente, minimizando as perdas e otimizando os lucros nas propriedades do sistema de produção pecuário.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver um modelo de Redes Neurais Convolucionais a fim de classificar o escore de cocho a partir de uma base de imagens. Por conseguinte, os objetivos específicos são:

- estudar modelos de *deep learning*, com ênfase em Redes Neurais Convolucionais;
- organizar uma base de dados de imagens de acordo com as classificações dos escores de cocho;
- avaliar a acurácia do modelo a partir de conjuntos de imagens de teste;
- avaliar métricas de desempenho que visam a incorporação do modelo desenvolvido em uma aplicação de tempo real.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ESCORE DE COCHO

A determinação do escore de cocho é realizada por um sistema de classificação estabelecido pela Universidade do Estado da Dakota do Sul (*South Dakota State University* - SDSU). A técnica consiste na avaliação diária do cocho, antes do primeiro trato, preferencialmente pela mesma pessoa capacitada para a tarefa. As sobras da alimentação dos animais são classificadas em 6 (seis) escores (Tabela 1). No escore 0 (zero), o cocho não apresenta sobras (cocho limpo), enquanto que no escore 1/2 (meio) há em torno de 5% de sobras, estando o alimento disperso. Em ambos, recomenda-se um ajuste de aumento de 1kg de matéria seca (MS) ou 5% da dieta ofertada no dia anterior. No escore 1 (um) é possível perceber uma fina camada de sobras no cocho (< 5cm), sendo recomendado a manutenção da quantidade de alimento ofertado no último trato. Os escores 2 (dois) e 3 (três) são caracterizados por camadas médias (5 a 8cm) e grossas (> 8cm), respectivamente. Por fim, no escore 4 (quatro) o cocho está intacto. O ajuste nos últimos três escores indicam uma diminuição de 1kg de MS ou 5% da dieta ofertada no dia anterior. É importante observar que nessa metodologia não há grandes variações nas recomendações de correção da oferta de alimentação entre diferentes escores, pois uma ingestão exagerada de ração pelo animal provoca distúrbios metabólicos, principalmente acidose clínica, que pode levar à morte do animal (PRITCHARD; BRUNS, 2003). O escore 1 é o escore adequado, pois indica que os animais estão se alimentando de acordo com suas necessidades (LUZ et al., 2019).

Tabela 1 – Sistema de Escore de Cocho da SDSU.

Escore	Descrição	Recomendação
0	Não resta ração no cocho.	Aumento
1/2	Alimentação restante dispersa. A maior parte do fundo do cocho está exposta.	Aumento
1	Camada fina e uniforme de ração. Por volta de um carço de milho de profundidade.	Manutenção
2	25 a 50% de alimentação restante.	Diminuição
3	Mais de 50% de alimentação restante. Topo da ração está perturbada.	Diminuição
4	Alimentação praticamente intocada. Topo da ração ainda perceptível.	Diminuição

Fonte – Adaptado de Rusche (2020).

Diversos fatores podem afetar os escores, incluindo comportamento animal, condições climáticas e instalações. O comportamento ingestivo dos animais serve de parâmetro para avaliar a eficiência do manejo alimentar. Segundo Horton (1990), no momento que o vagão tratador chega ao lote, o ideal seria que: 25% dos animais estivessem na linha de

cocho esperando; 50% estivessem se dirigindo para o cocho; e 25% estivessem se levantando, deitados ou mesmo ruminando. Comportamentos diferentes, podem sugerir excesso ou falta de comida no cocho e manejo incorreto no escore de cocho.

Alterações na temperatura e na precipitação podem levar a variações na ingestão de ração. O aumento do calor pode diminuir o apetite dos animais, e maior precipitação pode tanto afetar a palatabilidade do alimento, quanto restringir a movimentação dos animais por efeitos da lama nos lotes, dificultando o acesso à alimentação. No entanto, os animais podem aumentar o consumo ao perceberem a chegada de uma tempestade (LOY et al., 2016).

Dentro das instalações, o objetivo é que a lotação dos currais e espaçamento do cocho seja suficiente para evitar competição por comida entre os animais, permitindo que todos os animais possam se alimentar simultaneamente. A recomendação é que o cocho deverá apresentar no mínimo 0,7m de comprimento para cada animal (CARDOSO, 1996) e quanto à área, é sugerido em média de 15 a 30m² por animal (NUNES, 1983).

Dessa forma, é possível inferir que as pontuações indesejadas no sistema de escore não são causadas, necessariamente, pela reposição equivocada de ração (PRITCHARD, 1993). Todas essas condições, atreladas às informações de consumo e a correta análise dos escores dos cochinhos, são de fundamental importância para o controle da variação de ingestão diária de dieta dos animais.

2.2 REDES NEURAIS ARTIFICIAIS E APRENDIZADO DE MÁQUINA

Na atualidade, modelos de aprendizado de máquina baseados em redes neurais artificiais se consolidaram em problemas de reconhecimento de padrões e classificação de imagens. No entanto, o interesse em se desenvolver sistemas computacionais que mimetizam a organização funcional de neurônios biológicos data de décadas atrás, com uma sólida fundamentação matemática. Nesta seção, serão discutidas as principais estruturas das redes neurais artificiais e, em particular, como as redes neurais convolucionais são capazes de aprender informações visuais. Os conceitos e notação utilizados neste trabalho foram baseados no livro de Gonzalez e Woods (2018).

2.2.1 *Perceptron*

Desenvolvido originalmente por Rosenblatt (1958), inspirado no trabalho de McCulloch e Pitts (1943), a unidade *perceptron* é um modelo matemático que aprende um limite linear entre duas classes de padrões linearmente separáveis. Uma classe de padrões pode ser interpretada como uma nuvem de pontos em um espaço euclidiano n -dimensional. Em duas dimensões, o limite linear entre duas classes de padrões é a equação reduzida da reta $y = ax + b$. Ao se variar o coeficiente linear b , a reta se desloca em relação à origem.

Não sendo multiplicado por nenhuma coordenada, esse coeficiente é conhecido como o viés (*bias*).

De maneira geral, padrões possuem dimensões muito superiores a duas. Em n dimensões, as coordenadas de um ponto são representados por x_1, x_2, \dots, x_n , e os coeficientes do limite entre duas classes utilizam a notação $w_1, w_2, \dots, w_n, w_{n+1}$, em que w_{n+1} é o *bias*. Assim, a fronteira em n dimensões seria um hiperplano, cuja equação é:

$$w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1} = 0,$$

e em forma de somatório:

$$\sum_{i=1}^n w_i x_i + w_{n+1} = 0. \quad (2.1)$$

Na forma vetorial:

$$\mathbf{w}^T \mathbf{x} + w_{n+1} = 0,$$

em que \mathbf{w} e \mathbf{x} são vetores colunas n -dimensionais. No contexto do problema de separação de duas classes C_1 e C_2 , dado um vetor de padrões \mathbf{x} , determina-se um vetor de pesos \mathbf{w} com a propriedade:

$$\mathbf{w}^T \mathbf{x} + w_{n+1} \begin{cases} > 0, & \text{se } \mathbf{x} \in C_1 \\ < 0, & \text{se } \mathbf{x} \in C_2. \end{cases} \quad (2.2)$$

Para problemas com $n > 3$, encontrar um hiperplano por inspeção se torna inviável. O *perceptron* é um algoritmo iterativo que se propõe a isso. Inicialmente, se estabelece um vetor de pesos e *bias* arbitrariamente. Após um número finito de iterações, é garantida a convergência para uma solução, caso ela exista. Seja $\alpha > 0$ um incremento de correção, também denominado de taxa de aprendizagem (*learning rate*). Considere $\mathbf{w}(1)$ um vetor com valores arbitrários, e $w_{n+1}(1)$ uma constante arbitrária. Então, para um vetor de padrões, $\mathbf{x}(k)$, no passo k :

1. Se $\mathbf{x}(k) \in C_1$ e a Equação (2.2) não retorna um valor positivo, aplica-se a correção:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \alpha \mathbf{x}(k), \\ w_{n+1}(k+1) &= w_{n+1}(k) + \alpha. \end{aligned}$$

2. Da mesma forma, se $\mathbf{x}(k) \in C_2$ e a Equação (2.2) não retorna um valor negativo, aplica-se a correção:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \alpha \mathbf{x}(k), \\ w_{n+1}(k+1) &= w_{n+1}(k) - \alpha. \end{aligned}$$

3. Caso contrário, quando a Equação (2.2) retorna o valor esperado, a correção não é aplicada:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k), \\ w_{n+1}(k+1) &= w_{n+1}(k). \end{aligned}$$

A Figura 1 demonstra o esquema de um *perceptron*. Ao receber um padrão de entrada, o *perceptron* calcula a soma de produtos dos pesos por essas entradas e adiciona o *bias*. O resultado desse processo é limiarizado por uma função de ativação (*activation function*), levando valores positivos a $+1$ e valores negativos a -1 . Se a saída for $+1$, o padrão pertence à classe C_1 . Caso contrário, se a saída for -1 , o padrão pertence à classe C_2 . Entretanto, na prática, classes de padrões raramente são linearmente separáveis. Dessa forma, uma abordagem relevante para o entendimento de redes neurais se baseia em encontrar o vetor de pesos \mathbf{w} que minimiza o erro quadrático médio (MSE - *mean squared error*) entre a saída esperada e a saída estimada em cada etapa do treinamento. Essa função é da forma:

$$E(\mathbf{w}) = \frac{1}{2}(r - \mathbf{w}^T \mathbf{x})^2, \quad (2.3)$$

em que r é a saída esperada para o padrão \mathbf{x} . A minimização do erro E é feita por um algoritmo iterativo de gradiente descendente (ROBBINS; MONRO, 1951):

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha \left[\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \right]_{\mathbf{w}=\mathbf{w}(k)}. \quad (2.4)$$

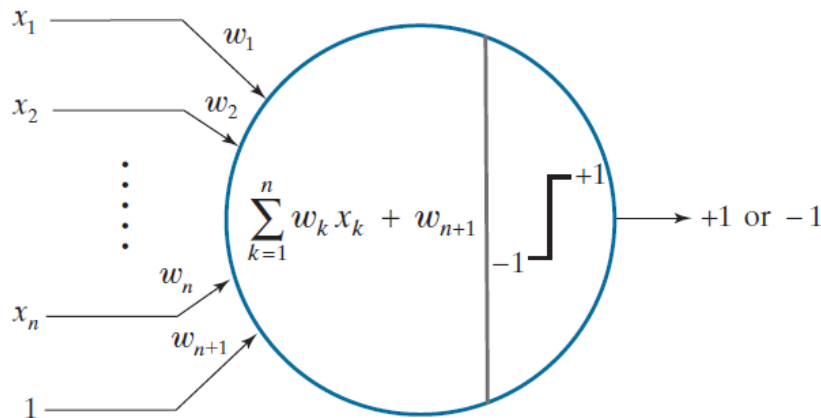


Figura 1: Esquema de um *perceptron*.

Fonte – Gonzalez e Woods (2018).

A Figura 2 mostra um exemplo do valor do erro E em função do produto dos valores escalares w e x , em que a saída esperada $r = 1$. O objetivo é aproximar $E(w)$ de um mínimo. Isso implica, ao longo de k iterações, que a derivada parcial de $E(w)$ em relação a w , na Equação (2.4), tenda a zero. Ao final desse processo, espera-se que a solução estimada convirja para a solução esperada. O valor da taxa de aprendizagem determina a fração do gradiente utilizada na correção do peso. Quanto menor o valor de α , menor é a correção de w . A Figura 2a ilustra essa situação, em que o movimento em direção à minimização do erro é lento. Por outro lado, quanto maior o valor de α , maior é a correção de w . Isso pode causar grandes oscilações no movimento de convergência, ou mesmo invertê-lo em direção ao aumento do erro (Figura 2b).

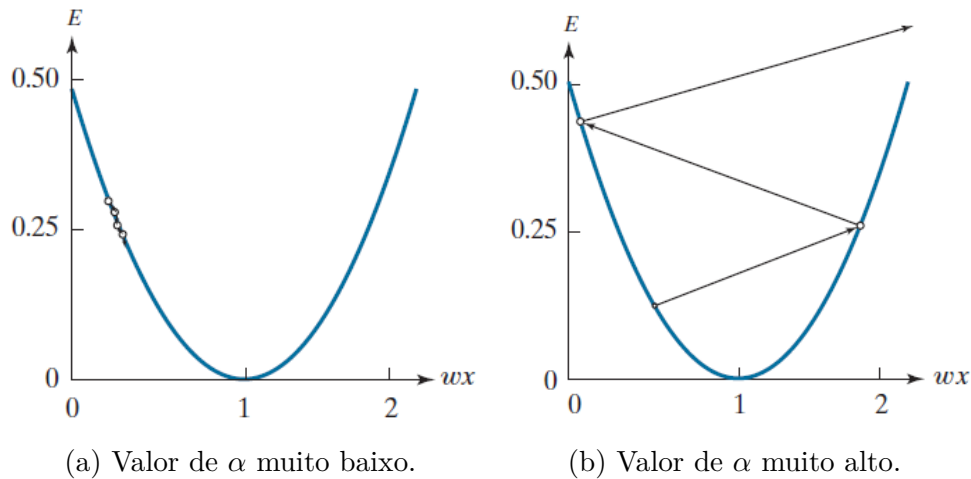


Figura 2: Gráficos de E em função de wx , para $r = 1$. Dependendo do valor da taxa de aprendizagem α , as iterações do algoritmo de gradiente descendente podem levar a mínimos locais ou à divergência.

Fonte – Gonzalez e Woods (2018).

Existem limitações em problemas de decisão de fronteiras lineares. Por exemplo, algumas funções lógicas linearmente separáveis, como os operadores AND, OR e NOT, podem ser implementadas por uma única unidade *perceptron*. No entanto, o operador booleano XOR, por ser não linear, precisa de mais de um *perceptron*. Problemas do mundo real são mais complexos, como os de reconhecimento de padrões e classificação de imagens em visão computacional. Esses problemas requerem arquiteturas mais avançadas, tanto em relação a modificações no *perceptron*, quanto na forma como os neurônios se conectam em estruturas multicamadas.

2.2.2 Redes neurais multicamadas

As redes neurais são estruturadas em camadas a partir de unidades computacionais semelhantes ao *perceptron*, chamadas de neurônios artificiais. O que difere um *perceptron* de um neurônio artificial é a presença de uma função de ativação mais suave, que normaliza o resultado da computação em um intervalo maior de valores, não apenas em um único valor negativo ou positivo. Como em uma rede a saída de um neurônio afeta todas as unidades subsequentes, pequenas alterações nos pesos e/ou *bias* refletem em uma pequena alteração equivalente na saída da rede. Isso não é possível em uma hipotética rede de *perceptrons*, como o da Figura 1. Suponha que a computação antes da função de limiarização resulte em um valor infinitesimalmente maior ou menor que zero. Ao ser limiarizado por uma função de ativação mais rígida para $+1$ ou -1 , esse valor causaria uma grande oscilação na saída, afetando a estabilidade da rede.

A Figura 3 mostra alguns exemplos de funções de ativação, denotadas por h , em que z , representado no eixo horizontal, é o resultado da computação de um neurônio artificial. Assim, o valor de ativação $a = h(z)$, representado no eixo vertical, é a saída desse neurônio.

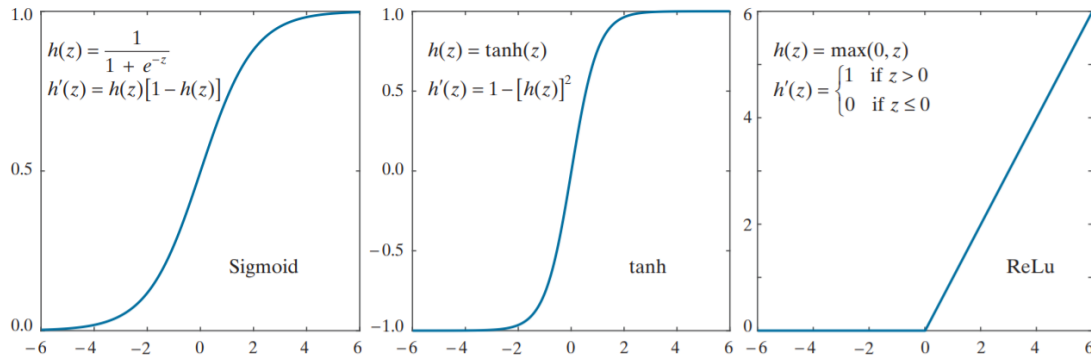


Figura 3: Exemplos de funções de ativação. Da esquerda para a direita: *Sigmoid*, Tangente Hiperbólica e *Rectifier linear unit* (ReLU).

Fonte – Gonzalez e Woods (2018).

A Figura 4 destaca o esquema de um neurônio artificial em um modelo genérico de uma rede neural multicamadas. A rede é composta por L camadas, onde cada camada ℓ , em que $\ell = 1, 2, \dots, L$, contém n_ℓ neurônios. Para $\ell = 1$, a camada de entrada (*input layer*) é composta de nós que representam os componentes de um vetor de padrões \mathbf{x} . Esses componentes são as próprias saídas da camada de entrada:

$$a_j(1) = x_j, \quad j = 1, 2, \dots, n_1,$$

em que n_1 é a dimensionalidade do vetor de padrões. Entre $1 < \ell < L$, nas chamadas camadas ocultas (*hidden layers*), cada nó é um neurônio artificial, cuja saída (valor de ativação) está conectada à entrada de todos os nós da camada subsequente, formando uma rede totalmente conectada (*fully connected*). Quando não há *loops* na rede, ela é chamada de rede *feedforward*. A computação das entradas e dos pesos em um rede *feedforward* utiliza uma notação semelhante à Equação (2.1), porém mais complexa, da forma:

$$z_i(\ell) = \sum_{j=1}^{n_{\ell-1}} w_{ij}(\ell) a_j(\ell - 1) + b_i(\ell). \quad (2.5)$$

O peso $w_{ij}(\ell)$ associa a saída do neurônio j da camada $\ell - 1$ à entrada do neurônio i da camada ℓ . Da mesma forma, $b_i(\ell)$ é o valor do *bias* associado ao i -ésimo neurônio da camada ℓ , e $a_j(\ell - 1)$ o valor de ativação do j -ésimo neurônio da camada anterior. A saída de um neurônio i em uma camada ℓ é dado por:

$$a_i(\ell) = h(z_i(\ell)), \quad i = 1, 2, \dots, n_\ell.$$

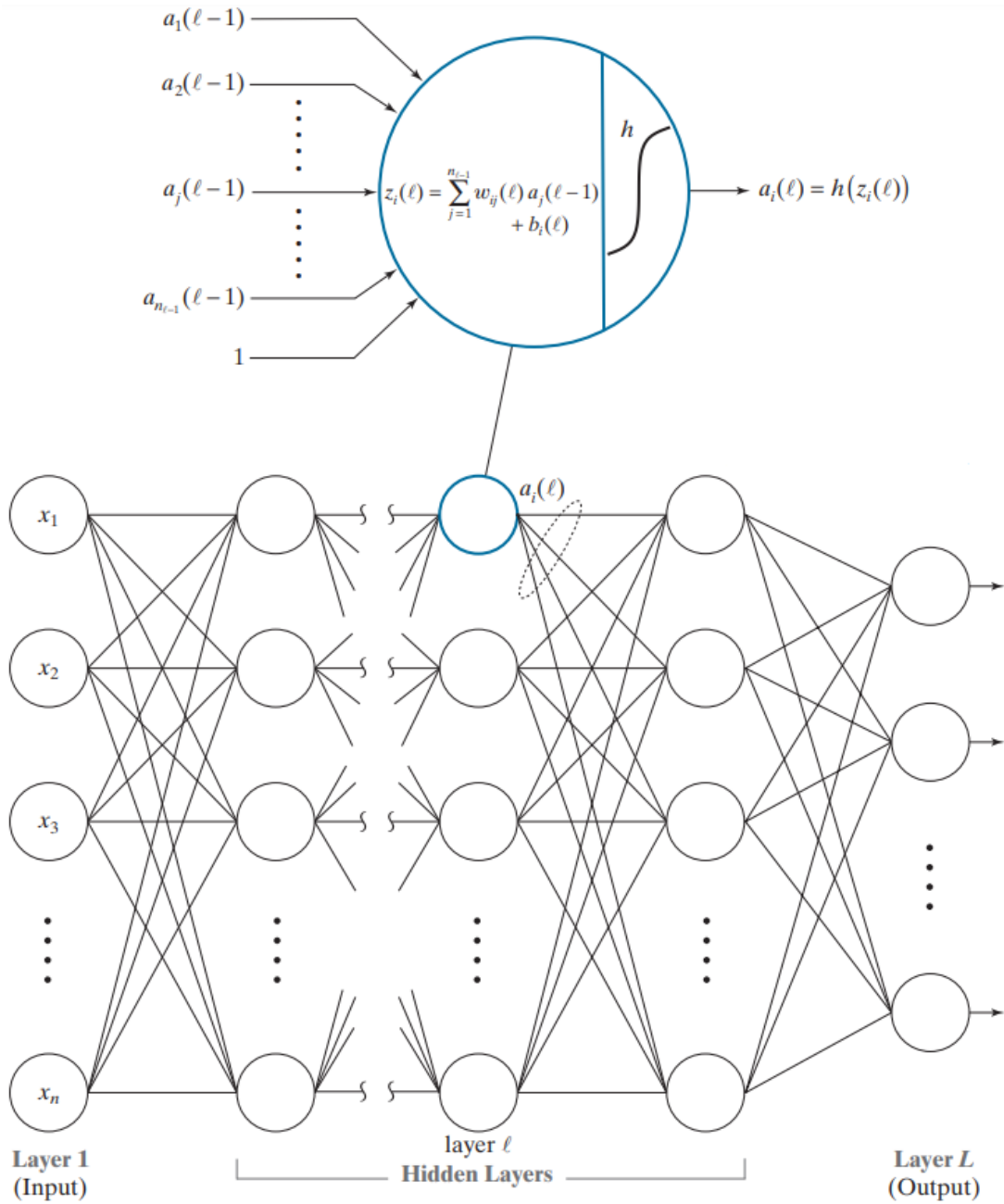


Figura 4: Modelo genérico de uma rede neural *feedforward* totalmente conectada (*fully connected*). Destaca-se na parte superior o esquema de um neurônio artificial.

Fonte – Adaptado de Gonzalez e Woods (2018).

Por fim, a camada de saída (*output layer*) possui n_L neurônios, que classificam um padrão desconhecido em uma das n_L classes. Uma função de ativação muito utilizada na camada de saída é a *softmax*. Essa função leva a soma de todos valores de ativação a 1, resultando em uma interpretação probabilística das saídas da rede. Assim, o valor de saída de um neurônio i na camada L da rede é:

$$a_i(L) = h(z_i(L)), \quad i = 1, 2, \dots, n_L.$$

A cada etapa de treinamento em uma rede *feedforward*, são realizados dois tipos de movimentos: a passagem para a frente (*forward pass*) e a passagem para trás (*backward pass*). No *forward pass*, propaga-se os padrões através da rede a fim de classificá-los e determinar o erro entre as saídas esperadas e as saídas estimadas. No *backward pass*, ocorre a retropropagação (*backpropagation*) dos erros para atualizar os pesos e *bias* através da rede. Esses dois movimentos consecutivos de ida e volta se repetem até que se atinja o estado de convergência, no qual o erro não pode mais ser reduzido.

A abordagem do *backpropagation* consiste em encontrar os parâmetros da rede que minimizem uma função de erro. Reescrevendo a Equação (2.3) do erro quadrático médio, o erro em relação a um vetor de padrões \mathbf{x} é definido como o somatório dos erros de cada neurônio individualmente:

$$E = \frac{1}{2} \sum_{j=1}^{n_L} (r_j - a_j(L))^2,$$

em que r_j é saída esperada e $a_j(L)$ é saída estimada de um neurônio j da camada de saída.

O erro E varia em relação aos pesos na rede da seguinte forma:

$$\delta_j(\ell) = \frac{\partial E}{\partial z_j(\ell)},$$

em que a expressão para $z_j(\ell)$ contém os pesos de entrada da rede para o nó j na camada ℓ (Eq. (2.5)), exceto para a camada de entrada. Aplicando a regra da cadeia, é possível expressar $\delta_j(\ell)$ em termos de $\delta_j(\ell + 1)$:

$$\begin{aligned} \delta_j(\ell) &= \frac{\partial E}{\partial z_j(\ell)} = \sum_i \frac{\partial E}{\partial z_i(\ell + 1)} \frac{\partial z_i(\ell + 1)}{\partial a_j(\ell)} \frac{\partial a_j(\ell)}{\partial z_j(\ell)} \\ &= \sum_i \delta_i(\ell + 1) \frac{\partial z_i(\ell + 1)}{\partial a_j(\ell)} h'(z_j(\ell)) \\ &= h'(z_j(\ell)) \sum_i w_{ij}(\ell + 1) \delta_i(\ell + 1), \end{aligned}$$

para $\ell = L - 1, L - 2, \dots, 2$. Uma vez que não é possível computar os gradientes dos pesos nos neurônios das camadas ocultas, esse desenvolvimento estabelece como determinar $\delta_i(\ell)$ a partir de $\delta_i(\ell + 1)$. Assim, partindo do erro na camada de saída, é possível calcular a variação em relação às camadas ocultas em cadeia.

O principal objetivo do *backpropagation* é atualizar os pesos e o *bias*. Aplicando novamente a regra da cadeia, é possível obter uma expressão para $\partial E / \partial w_{ij}(\ell)$ e $\partial E / \partial b_i(\ell)$, em termos de $\delta_j(\ell)$. Utilizando o gradiente descendente (Eq. (2.4)), a atualização dos pesos é dada da seguinte forma:

$$\begin{aligned} w_{ij}(\ell) &= w_{ij}(\ell) - \alpha \frac{\partial E}{\partial w_{ij}(\ell)} \\ &= w_{ij}(\ell) - \alpha \frac{\partial E}{\partial z_i(\ell)} \frac{\partial z_i(\ell)}{\partial w_{ij}(\ell)} \\ &= w_{ij}(\ell) - \alpha \delta_i(\ell) a_j(\ell - 1). \end{aligned}$$

De forma similar, atualiza-se o *bias*:

$$\begin{aligned} b_i(\ell) &= b_i(\ell) - \alpha \frac{\partial E}{\partial b_i(\ell)} \\ &= b_i(\ell) - \alpha \frac{\partial E}{\partial z_i(\ell)} \frac{\partial z_i(\ell)}{\partial b_i(\ell)} \\ &= b_i(\ell) - \alpha \delta_i(\ell). \end{aligned}$$

2.2.3 Redes neurais convolucionais

Para que uma imagem possa ser inserida em uma rede neural como descrita na subseção anterior, os padrões precisam ser projetados em um vetor. Porém, esse tipo de representação não preserva as relações espaciais entre os píxeis. Além disso, a premissa de uma rede neural é o aprendizado realizado diretamente dos dados de treinamento. LeCun et al. (1989) desenvolveram uma rede, posteriormente chamada de LeNet, capaz de aceitar imagens como entrada e processá-las na forma de tensores. Cada entrada de uma camada recebe um único valor, determinado pelo produto de convolução (por isso o nome rede neural convolucional) sobre a vizinhança de cada píxel de entrada da camada anterior. Assim, redes neurais convolucionais não são totalmente conectadas. Entretanto, os cálculos para a formação da soma de produtos, adição do *bias*, função de ativação, *forward pass* e *backward pass* em uma rede convolucional são semelhantes aos apresentados na seção anterior.

A Figura 5 mostra a arquitetura e elementos básicos de uma rede neural convolucional, mais especificamente a LeNet. Dado um píxel na posição (x, y) de uma imagem de entrada (*input image*), a convolução espacial é determinada pela soma de produtos na vizinhança desse píxel. Os pesos do núcleo (*kernel*) utilizado na convolução são parâmetros aprendidos pela rede. Ao adicionar um *bias* e passar por uma função de ativação, o resultado é um único valor escalar (ponto A), que servirá de entrada na localização correspondente (x, y) da próxima camada. A vizinhança de um píxel é chamada de campo receptivo (*receptive field*). A operação é realizada sobre toda a imagem, movendo-se esse campo por incrementos espaciais, chamados de *strides*. No final desse processo, um conjunto de valores é armazenado na camada subsequente, formando novamente um tensor, denominado mapa de características (*feature map*). A motivação dessa terminologia reside no papel desempenhado pela convolução, que é extrair características como pontos, curvas e bordas. Pode-se ter inúmeros *feature maps*, cada um com o seu próprio filtro com diferentes pesos do *kernel* e *bias*. Coletivamente, esses mapas são referidos como camada convolucional (*convolutional layer*). Como exemplo, na arquitetura da Figura 5, a primeira camada convolucional possui 3 *feature maps*. Supondo um filtro com $M \times N$, o número de parâmetros aprendidos nessa camada seria de $(M \times N + 1) \times 3$.

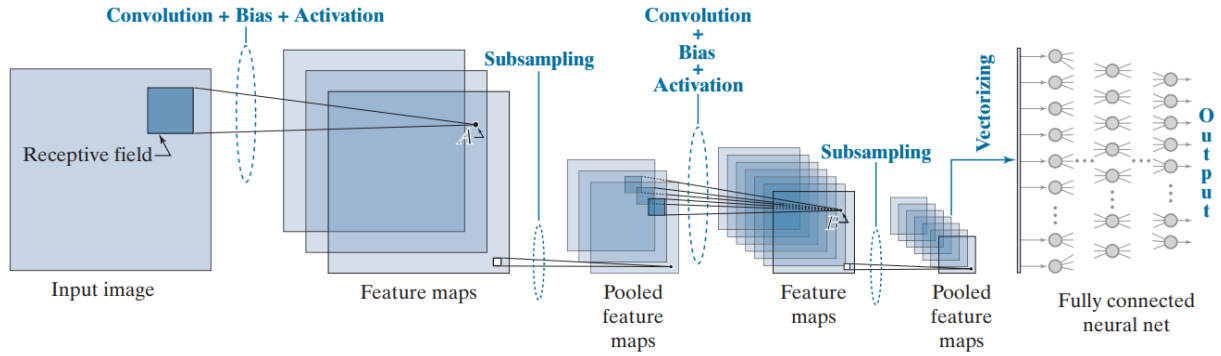


Figura 5: Arquitetura e elementos básicos da rede convolucional LeNet.

Fonte – Gonzalez e Woods (2018).

O próximo processo é a subamostragem (*subsampling*), inspirado por um modelo do córtex visual de mamíferos proposto por Hubel e Wiesel (1962). Esse modelo sugere que a extração de características (*feature extraction*) é realizada por células simples, enquanto que células complexas agrupam essas características em estruturas mais significativas. O *pooling* é uma forma de modelar esse agrupamento, além de reduzir a dimensionalidade do volume de dados. Subdividindo o mapa de características em pequenas regiões adjacentes (normalmente 2×2), há três métodos comuns de *pooling* que substituem essas regiões por um único valor: através da média dos valores da vizinhança (*average pooling*), do máximo valor da vizinhança (*max pooling*) e da raiz quadrada da soma dos valores da vizinhança ao quadrado (*L_2 pooling*). Assim, o conjunto desses valores formam um mapa de características agrupado (*pooled feature map*) que, coletivamente, são referidos como camada de *pooling* (*pooling layer*).

As operações de convolução e *pooling* são ilustradas na Figura 6. À medida que uma imagem se propaga pela rede neural convolucional, ela sofre uma série de reduções dimensionais. A redução causada pela operação de convolução depende do tamanho do campo receptivo utilizado e do *stride*. Uma forma de evitar essa redução consiste em adicionar uma quantidade (chamada de *padding*) de píxeis (normalmente fixados em zero) ao redor do limite da imagem. A fórmula para se obter a dimensão (*size*) do *feature map* ao final do processo convolutivo é:

$$feature\ map\ size = \frac{input\ image\ size + 2 * padding - receptive\ field\ size}{stride} + 1,$$

que, no exemplo da Figura 6, tem-se:

$$\frac{28 + 2 * 0 - 5}{1} + 1 = 24.$$

Ao contrário da convolução, a operação de *pooling* reduz intencionalmente o tamanho da imagem, e de forma mais significativa, a fim de agrupar as características mais relevantes. A vizinhança de *pooling* mínima de 2×2 reduz as dimensões do mapa de características pela metade.

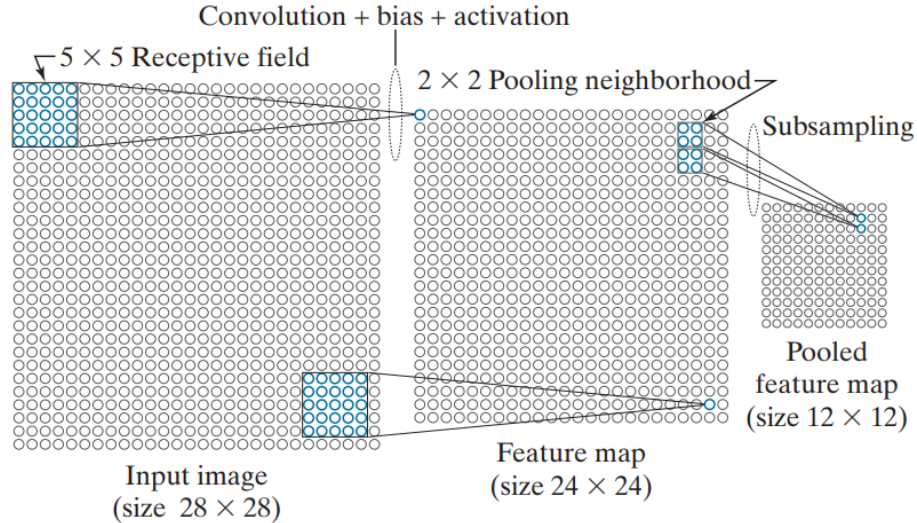


Figura 6: Operações de convolução e *pooling*. O tamanho dos campos receptivos e o pooling afetam o tamanho da imagem de entrada e do mapa de características.

Fonte – Adaptado de Gonzalez e Woods (2018).

Dada uma camada convolucional ℓ , a convolução a partir de uma camada anterior $\ell - 1$ com múltiplos canais é realizada da seguinte forma: para cada localidade em um mapa de características da camada ℓ – representada pelo ponto B na arquitetura da Figura 5 – é realizado o cálculo do produto de convolução sucessivamente para todos os canais da camada $\ell - 1$. Para cada canal utiliza-se um *kernel* com pesos diferentes. Como a convolução é um processo linear, os valores calculados para cada canal são somados. O resultando é um único valor, que é adicionado a um *bias* e limiarizado por uma função de ativação, gerando o valor do ponto B. Os mapas de características da última camada convolucional são vetorizados, servindo como entrada para uma rede neural totalmente conectada.

Como mostrado na Equação (2.5), o cálculo realizado por um neurônio artificial em uma determinada camada é uma soma de produtos entre os pesos da camada atual e as saídas da camada anterior. O cálculo realizado em uma CNN é a convolução, que é uma soma de produtos (em 2-D, no caso de imagens) entre os pesos w de um *kernel* e os elementos a_{xy} correspondes do mapa de características sobrepostos pelo *kernel* (*receptive field*), dada por:

$$\begin{aligned}
 w \star a_{xy} &= \sum_{n=1}^N \sum_{m=1}^M w_{n,m} a_{x-n,y-m} \\
 &= w_{1,1} a_{x-1,y-1} + w_{1,2} a_{x-1,y-2} + \dots + w_{N,M} a_{x-N,y-M},
 \end{aligned} \tag{2.6}$$

em que n e m percorrem as dimensões N e M do *kernel*, respectivamente. Considerando

que o somatório contém $N \times M$ parcelas:

$$\begin{aligned} w \star a_{xy} &= w_1 a_1 + w_2 a_2 + \cdots + w_{N \times M} a_{N \times M} \\ &= \sum_{i=1}^{N \times M} w_i a_i, \end{aligned}$$

e adicionando um *bias*, o resultado é indicado por z na forma da Equação (2.5):

$$z = w \star a_{xy} + b.$$

Assim, o resultado pode ser expresso de forma análoga ao realizado por um neurônio artificial em um rede neural totalmente conectada. Ao passar por uma função de ativação, tem-se $a = h(z)$, que é o valor computado para qualquer localidade em um mapa de características, como o ponto A na Figura 5. O valor do ponto B dessa mesma figura é dado pela soma de três equações de convolução, uma vez que há três mapas de características na camada anterior:

$$\begin{aligned} w_{n,m}^{(1)} \star a_{xy}^{(1)} + w_{n,m}^{(2)} \star a_{xy}^{(2)} + w_{n,m}^{(3)} \star a_{xy}^{(3)} &= \sum_{n=1}^N \sum_{m=1}^M w_{n,m}^{(1)} a_{x-n,y-m}^{(1)} + \\ &\quad \sum_{n=1}^N \sum_{m=1}^M w_{n,m}^{(2)} a_{x-n,y-m}^{(2)} + \sum_{n=1}^N \sum_{m=1}^M w_{n,m}^{(3)} a_{x-n,y-m}^{(3)}. \end{aligned}$$

Como os *kernels* possuem as mesmas dimensões e se movem em sincronia, os valores de n , m , x e y são os mesmos nos três somatórios.

Os valores de a_{xy} podem tanto representar os píxeis na camada de entrada, quanto os mapas de características agrupados nas camadas posteriores. Portanto, as equações não se diferenciam pelo que a variável a_{xy} representa. Por exemplo, caso as imagens possuam mais de um canal, como em uma imagem RGB, as equações do ponto A seriam da mesma forma do ponto B. No presente trabalho, o banco de imagens é composto por imagens RGB, e a arquitetura da rede neural convolucional desenvolvida é mais profunda do que a representada na Figura 5, possuindo mais camadas e um número maior de *feature maps* por camada. Como não existe um algoritmo que determina os parâmetros ótimos a serem usados em uma rede neural, suas especificações exigem experiência e experimentação.

3 ABORDAGEM PROPOSTA

Neste capítulo, são detalhadas as características do modelo de redes neurais convolucionais desenvolvido, bem como o conjunto de dados (*dataset*) de imagens construído de acordo com as classificações dos escores de cocho.

3.1 CONJUNTO DE DADOS

Para treinar e avaliar o modelo de rede neural convolucional proposto, foi construído um *dataset* de imagens de escores de cocho. Essas imagens foram obtidas por *smartphones* de diversos modelos e resoluções de câmera em 8 propriedades rurais distintas no Estado de Minas Gerais, a fim de garantir uma variabilidade satisfatória de formas e materiais utilizados nas estruturas dos cochos, além de tipos diferentes de dietas. Os dispositivos foram posicionados acima do cocho, com algumas variações de ângulos e a uma distância que permitiu que todo o interior do cocho fosse capturado na imagem. As imagens foram capturadas por colaboradores da área de Medicina Veterinária. A Figura 7 apresenta uma imagem de cocho para cada escore. O *dataset* possui um total de 1001 imagens, cuja distribuição é descrita na Tabela 2.

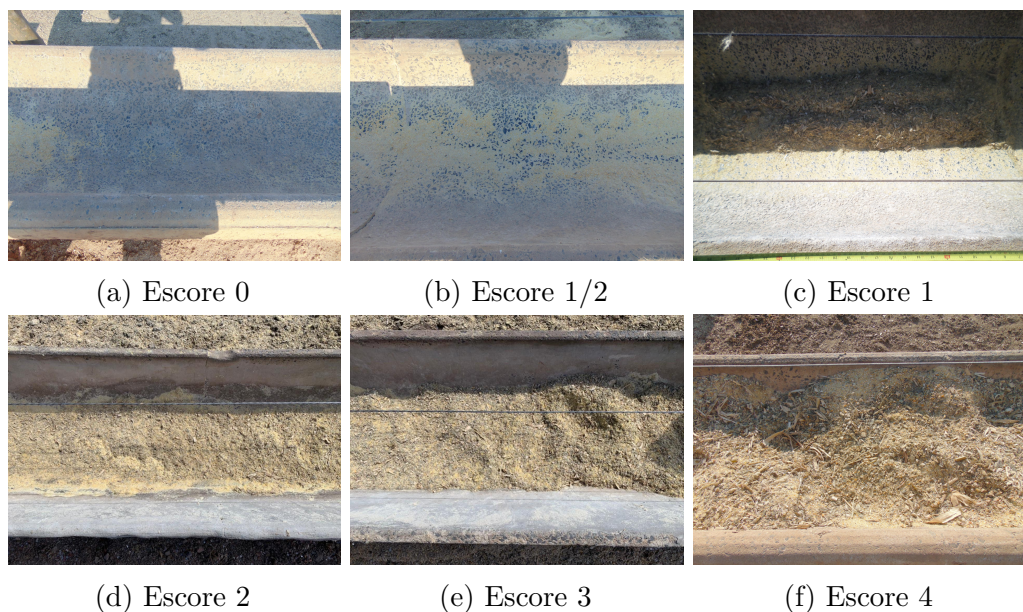


Figura 7: Imagens dos escores.

Fonte – Elaborado pelo autor (2021).

Tabela 2 – Distribuição das imagens do *dataset* por classe.

Classe	Número de imagens	Proporção
Escore 0	83	8,29%
Escore 1/2	137	13,69%
Escore 1	193	19,28%
Escore 2	112	11,19%
Escore 3	164	16,38%
Escore 4	312	31,17%

Fonte – Elaborado pelo autor (2022).

Considerando que o problema de classificação apresenta 6 classes, o ideal seria cada classe ter uma proporção em torno de 16,67%. É possível inferir que há 3 classes sub-representadas (escores 0, 1/2 e 2), duas sobre-representadas (escores 1 e 4), e uma próxima do número de amostras ideal (escore 3). Uma característica importante deste problema é a recomendação quanto o ajuste ou não da dieta, de acordo com a Tabela 1. Para os escores 0 e 1/2, são recomendados o ajuste de aumento, para o escore 1 é recomendado a manutenção, e para os escores 2, 3 e 4 são recomendados o ajuste de diminuição. Levando em conta essas distribuições, o escore 1 é uma classe de interesse, pois apenas essa classe é responsável por um tipo de recomendação.

3.2 MODELO PROPOSTO

Para a classificação das imagens de escore de cocho foi proposta uma rede neural convolucional, cuja arquitetura é apresentada na Figura 8 (ilustração elaborada utilizando a ferramenta PlotNeuralNet¹). A imagem de entrada possui dimensões de 188×188 píxeis no espaço de cor RGB. A rede é composta por 4 camadas convolucionais de 128, 128, 256 e 512 *feature maps*, respectivamente, com tamanho do *kernel* de 5×5 e função de ativação ReLU. Após cada camada convolucional, há uma camada de *max pooling* 2×2 . Ao final do bloco convolucional, são gerados 32768 parâmetros ($8 \times 8 \times 512$), que alimentam a rede totalmente conectada. Essa rede é formada por duas camadas totalmente conectadas com 128 e 64 unidades e função de ativação ReLU. Ambas as camadas são sucedidas por uma camada de *dropout* de 0,2. *Dropout* é uma técnica que zera aleatoriamente os pesos de uma porcentagem de neurônios a cada iteração. Uma iteração é um movimento de *forward pass* e *backward pass* de um único lote (*batch*) de amostras. O objetivo é evitar que ocorra uma adaptação exagerada da rede ao conjunto de treinamento. Essa superadaptação (*overfitting*) levaria a um desempenho menor de amostras não utilizadas no treinamento. Finalmente, uma última camada totalmente conectada é utilizada para classificação, possuindo 6 unidades, uma para cada classe.

¹ <https://github.com/HarisIqbal88/PlotNeuralNet>

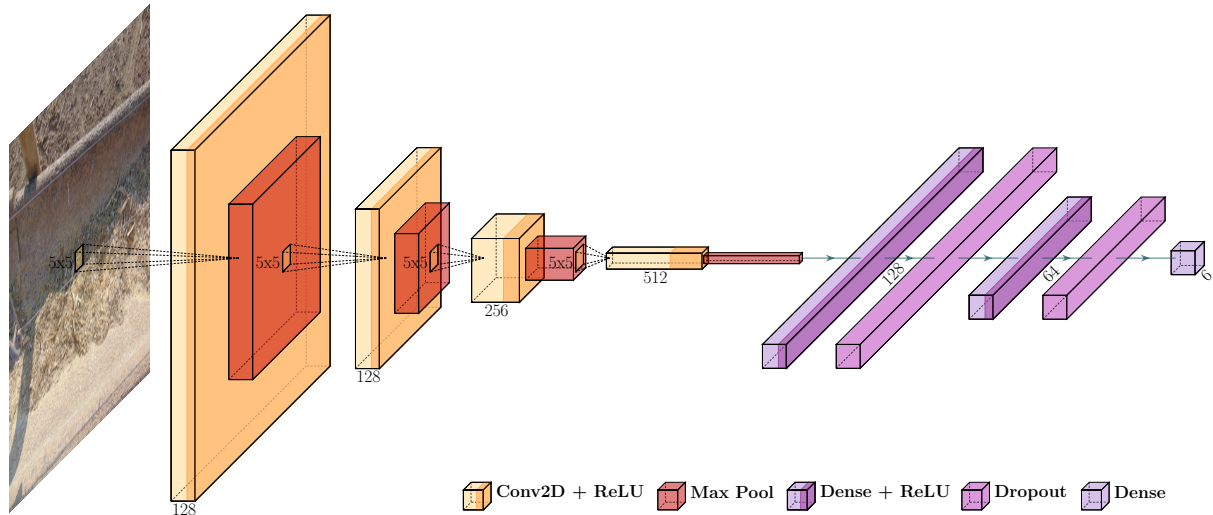


Figura 8: Modelo de rede neural convolucional proposto. A imagem de entrada possui dimensões de 188×188 pixels no espaço de cor RGB.

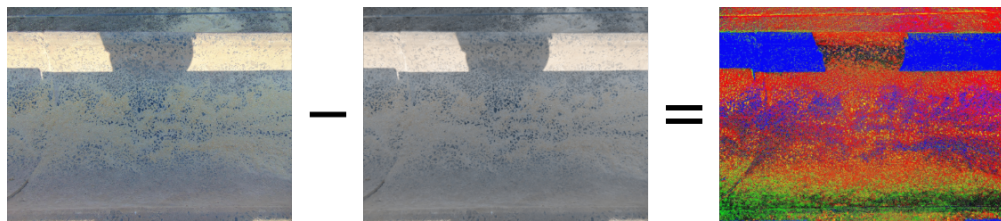
Fonte – Elaborado pelo autor (2022).

De acordo com a Tabela 1, os escores são definidos pela quantidade de alimentação restante no cocho. Portanto, o objeto a ser reconhecido e classificado pela rede neural é a dieta contida no cocho. Pela Figura 7, observa-se que a ração é composta de características de textura, por vezes semelhante ao material do cocho e/ou à terra capturada como ruído. Com base nessa evidência, foi aplicada uma filtragem nas imagens do *dataset*, a fim de agudizar altas frequências, onde são armazenadas informações como a de textura. A estratégia utilizada consiste em subtrair os pixels das imagens originais pelos pixels das imagens com um filtro passa-baixa, no caso, o filtro gaussiano com desvio padrão $\sigma = 1$. As imagens geradas são apresentadas na Figura 9, que para efeito de comparação, são as mesmas apresentadas na Figura 7. Após essa etapa, foi proposta uma arquitetura multi-entrada, como mostra a Figura 10, de modo a processar o par de imagens, original e com altas frequências realçadas. As redes neurais convolucionais e totalmente conectadas são idênticas, e a concatenação ocorre na camada de classificação. Por limitações do poder computacional disponível, as dimensões das imagens foram reduzidas para 156×156 pixels.

Figura 9: Geração das imagens dos escores com filtro passa-alta. Os píxeis das imagens originais foram subtraídos das imagens com filtro gaussiano ($\sigma = 1$).



(a) Escore 0



(b) Escore 1/2



(c) Escore 1



(d) Escore 2



(e) Escore 3



(f) Escore 4

Fonte – Elaborado pelo autor (2022).

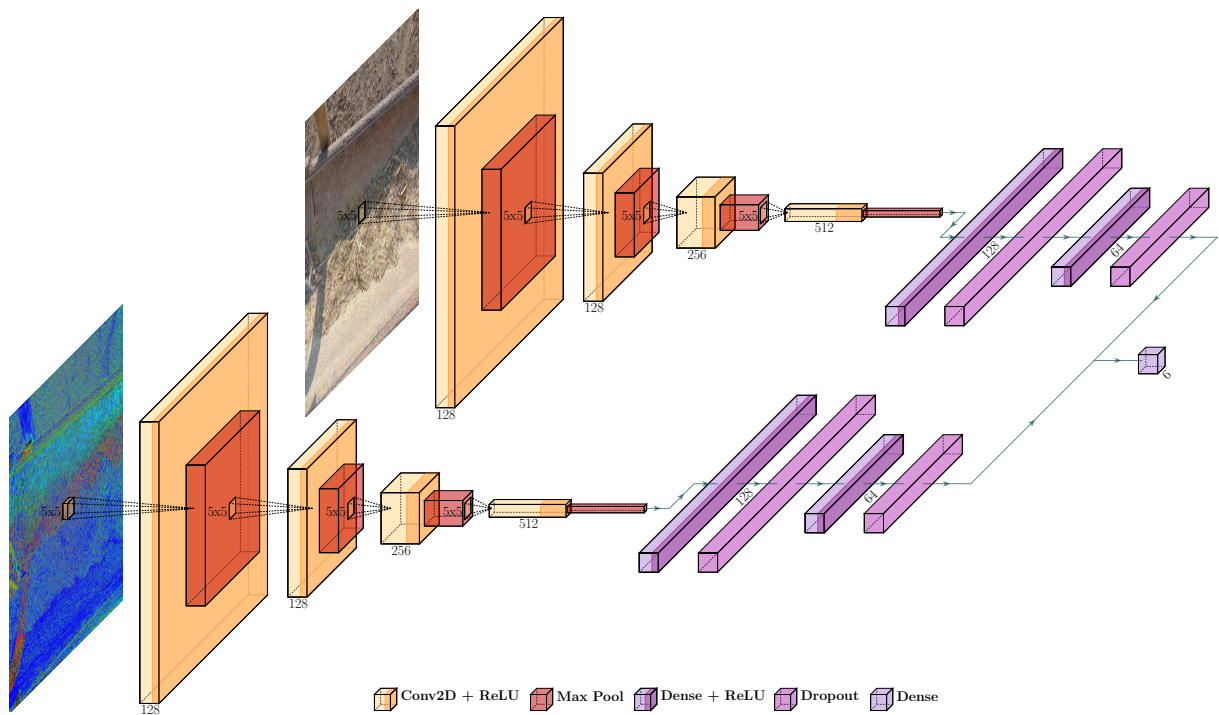


Figura 10: Modelo de rede neural convolucional multi-entrada proposto. As imagens de entrada possuem dimensões de 156×156 píxeis.

Fonte – Elaborado pelo autor (2022).

4 EXPERIMENTOS COMPUTACIONAIS

Os experimentos computacionais documentados neste capítulo visam comparar variações do modelo desenvolvido, a fim de levantar discussões acerca dos resultados obtidos. Para avaliar o desempenho dos modelos, foram realizadas 6 validações cruzadas *3-fold*, com cerca de 2/3 das amostras para o conjunto de treinamento e 1/3 para o conjunto de teste, respeitando a proporção de imagens de cada classe. Do conjunto de treinamento, 1/5 foi utilizado para validação. Dessa forma, o modelo performou em 18 distribuições diferentes de dados.

Foram aplicadas algumas técnicas de aumento de dados (*data augmentation*), a fim de evitar o *overfitting* e simular possíveis condições de campo. Essas técnicas consistem em aumentar a diversidade do conjunto de treinamento, aplicando transformações nessas amostras. Neste trabalho, essas transformações foram aplicadas durante o treinamento – portanto, sem a criação de novas amostras (*oversampling*) – incluindo inversão horizontal com uma probabilidade de 50%, rotação por um ângulo aleatório no intervalo $\pm\pi/4$, *zoom out* por um fator de 25% e ajuste de contraste por um fator aleatório entre 0,75 e 1,25.

Inicialmente, foram realizados experimentos com o objetivo de compreender os impactos do modelo apresentado na Figura 8 para dois tipos de entrada: imagem original (M1) e imagem com filtro passa-alta (M2). Em seguida, o modelo multi-entrada ilustrado na Figura 10 foi avaliado para ambas as as entradas (M3). A Tabela 3 sintetiza as informações de composição dos modelos. As amostras foram divididas em lotes de tamanho (*batch size*) 8. Todos os modelos foram treinados por 300 épocas, com taxa de aprendizagem 0,001, função de perda entropia cruzada categórica esparsa e otimizador Adam (KINGMA; BA, 2014) com taxa de decaimento exponencial para estimação do primeiro momento, β_1 , fixado em 0,98, e com constante de estabilidade numérica, ϵ , fixado em 0,1. Esses parâmetros foram obtidos empiricamente. Os experimentos foram executados em uma arquitetura computacional com Processador Intel Core i7-8550U 1,80 GHz, 8 GB de RAM e GPU NVIDIA GeForce MX150 com 2 GB RAM. Os modelos foram desenvolvidos na linguagem Python 3.8.8 utilizando a biblioteca TensorFlow 2.3.

Tabela 3 – Síntese das variações dos experimentos.

Nome do Modelo	Descrição
M1	Modelo alimentado com as imagens originais.
M2	Modelo alimentado com as imagens com filtro passa-alta.
M3	Modelo multi-entrada com as imagens originais e as imagens com filtro passa-alta.

Fonte – Elaborado pelo autor (2022).

Os resultados obtidos para as variações do modelo desenvolvido são apresentados na Tabela 4. A métrica utilizada na avaliação foi a acurácia, que considera o número de amostras preditas corretamente sobre o número total de amostras. Foram extraídas a média (\bar{X}) e o desvio padrão (dp) das 6 validações cruzadas (V.C.). Considerando a acurácia geral média das validações cruzadas, é possível concluir que o modelo multi-entrada obteve o melhor desempenho (89,58%), superando o modelo com imagens originais em 3,97%, além de ter o menor desvio padrão entre os modelos. Analisando as classes separadamente, o modelo M3 melhorou 5 das 6 classes do problema (escores 0, 1/2, 1, 4 e, principalmente, 3), estando todas próximas da acurácia geral, com exceção do escore 4. O modelo M1 performou melhor em relação ao modelo M2, e para o escore 2 atingiu a maior acurácia dos três modelos. A Tabela 5 demonstra que as acurácias de validação não estão exageradamente acima das acurácias de teste, o que evidencia a não ocorrência de *overfitting*.

A Tabela 6 contém outras duas métricas de interesse neste trabalho: o tempo de predição por amostra e o tamanho do modelo final, ambas essenciais para se determinar a capacidade de incorporação dos modelos em uma aplicação de tempo real. Observa-se que a complexidade da rede impacta diretamente essas métricas. O modelo M3 possui um tempo de predição por amostra cerca de duas vezes maior que os tempos medidos dos modelos M1 e M2. O tamanho do modelo final do modelo M3 também é relativamente superior aos outros dois modelos. Apesar do aumento considerável do tempo de predição e do tamanho, o modelo multi-entrada ainda possui resultados que o credencia para ser incorporado em uma aplicação.

A Figura 11 apresenta a matriz de confusão para a melhor validação cruzada do modelo multi-entrada. Nota-se um viés para o escore 4, que com exceção do escore 0, é a maior confusão das demais classes e a maior acurácia entre os escores. Um possível fator é a proporção desse escore, que corresponde a 31,17% do *dataset*. Vale destacar que, apesar da menor proporção de amostras, o escore 0 obteve o segundo melhor resultado e acima da acurácia geral. Dado que essa é uma classe que se caracteriza por não apresentar sobras de dieta no cocho, acaba se destacando das outras e sendo mais confundida entre escores próximos com menos sobras (1/2 e 1).

Conforme discutido no capítulo anterior, há três grupos em relação à recomendação do ajuste da dieta: escores 0 e 1/2 (aumento da dieta), escore 1 (manutenção da dieta), e escores 2, 3 e 4 (diminuição da dieta). Portanto, as confusões menos desejáveis são entre escores de grupos diferentes, e não entre escores do mesmo grupo. A fim de melhor visualizar essas características, e portanto não houve um novo treinamento considerando as três recomendações, a Figura 12 apresenta os resultados da matriz de confusão da Figura 11 agrupados por recomendação. É possível perceber que as recomendações com o maior número de classes envolvidas possuem as maiores acurácias. Isso pode influenciar na proporção ideal de amostras das classes do problema, uma vez que a recomendação

de manutenção da dieta contém apenas o escore 1. Um resultado significativo é a baixa confusão entre recomendações opostas (aumento e diminuição), o que no sentido da aplicação, seria a confusão mais indesejada.

Tabela 4 – Média (\bar{X}) e desvio padrão (dp) das acurácias gerais e por classe das 6 validações cruzadas (V.C.). Destacam-se as melhores validações cruzadas de cada modelo e a melhor média entre os modelos.

Modelo	V.C.	Acurácia (%)	Acurácia por Escore (%)					
			0	½	1	2	3	4
M1	1	83.02	85.54	83.21	78.76	76.79	74.39	91.67
	2	85.91	84.34	86.13	85.49	86.61	78.66	90.06
	3	86.81	83.13	84.67	82.90	91.07	79.27	93.59
	4	85.11	84.34	80.29	83.42	86.61	76.83	92.31
	5	86.31	87.95	89.05	79.79	87.50	81.71	90.71
	6	86.51	79.52	83.94	85.49	89.29	82.93	91.03
	X	85.61	84.14	84.55	82.64	86.31	78.96	91.56
	dp	1.28	2.55	2.68	2.59	4.54	2.85	1.15
M2	1	75.92	79.52	70.80	76.17	75.89	63.41	83.65
	2	78.62	80.72	71.53	77.20	80.36	64.02	89.10
	3	79.12	79.52	72.99	76.68	78.57	66.46	90.06
	4	77.52	77.11	72.99	80.31	75.89	65.24	84.94
	5	76.42	77.11	72.26	77.72	70.54	60.37	87.82
	6	77.92	75.90	72.26	81.35	78.57	70.73	82.37
	X	77.59	78.31	72.14	78.24	76.64	65.04	86.32
	dp	1.13	1.70	0.78	1.92	3.15	3.16	2.85
M3	1	89.81	84.34	86.13	88.08	87.50	84.15	97.76
	2	88.31	85.54	93.43	83.94	81.25	85.98	93.27
	3	89.21	83.13	89.78	88.08	83.04	86.59	94.87
	4	89.51	87.95	83.94	91.19	88.39	84.76	94.23
	5	90.11	86.75	83.94	88.60	85.71	87.80	97.44
	6	90.51	90.36	89.78	88.08	86.61	86.59	95.83
	X	89.58	86.35	87.83	88.00	85.42	85.98	95.57
	dp	0.70	2.38	3.47	2.12	2.51	1.22	1.63

Fonte – Elaborado pelo autor (2022).

Tabela 5 – Médias (\bar{X}) das acurácias de validação e teste para análise de *overfitting*.

Modelo		Acurácia de Validação (%)	Acurácia de Teste (%)
M1	\bar{X}	89.16	85.61
M2	\bar{X}	82.55	77.59
M3	\bar{X}	92.94	89.58

Fonte – Elaborado pelo autor (2022).

Tabela 6 – Média (\bar{X}) e desvio padrão (dp) dos tempos de predição das 6 validações cruzadas, e tamanho dos modelos.

Modelo		Tempo de predição (ms)	Tamanho (MB)
M1	\bar{X}	11,56	97,8
	dp	0,78	
M2	\bar{X}	11,37	97,8
	dp	0,48	
M3	\bar{X}	21,06	158
	dp	2,61	

Fonte – Elaborado pelo autor (2022).

		Predição					
		Escore 0	Escore 1/2	Escore 1	Escore 2	Escore 3	Escore 4
Classe	Escore 0	90.36% 75	3.61% 3	3.61% 3	0.00% 0	0.00% 0	2.41% 2
	Escore 1/2	2.19% 3	89.78% 123	3.65% 5	0.00% 0	0.00% 0	4.38% 6
	Escore 1	1.04% 2	3.63% 7	88.08% 170	1.04% 2	1.55% 3	4.66% 9
	Escore 2	0.00% 0	0.89% 1	0.89% 1	86.61% 97	1.79% 2	9.82% 11
	Escore 3	0.61% 1	0.00% 0	3.66% 6	3.05% 5	86.59% 142	6.10% 10
	Escore 4	0.64% 2	1.28% 4	1.60% 5	0.64% 2	0.00% 0	95.83% 299

Figura 11: Matriz de confusão da melhor validação cruzada do modelo multi-entrada.

Fonte – Elaborado pelo autor (2022).

		Predição		
		Aumento	Manutenção	Diminuição
Recomendação	Aumento	92.73% 204	3.64% 8	3.64% 8
	Manutenção	4.66% 9	88.08% 170	7.25% 14
	Diminuição	1.36% 8	2.04% 12	96.60% 568

Figura 12: Matriz de confusão por recomendação. Os escores apresentados na Figura 11 foram agrupados de acordo com suas recomendações quanto ao ajuste de dieta, não havendo um novo treinamento.

Fonte – Elaborado pelo autor (2022).

5 CONCLUSÃO

Uma gestão objetiva das sobras da dieta bovina é essencial para garantir a eficiência das fazendas. Com base nessa necessidade, foi desenvolvido um modelo de Rede Neural Convolutacional capaz de classificar imagens de escores de cocho. Os resultados dos experimentos demonstraram que o modelo desenvolvido alcança uma acurácia satisfatória em todas as classes de imagens e recomendações quanto à reposição da dieta. O modelo apresenta tempo de predição e tamanho que o credencia para aplicações de tempo real e em diversas plataformas. Além disso, os resultados não indicaram *overfitting*, apresentando um potencial significativo de generalização. Por não existir uma base de dados de imagens específica na literatura, as imagens utilizadas no treinamento do método proposto compõem um *dataset* construído especialmente para este projeto, podendo ser uma contribuição adicional.

O desempenho do modelo multi-entrada pode atestar a evidência de que as características visuais relevantes da dieta estão armazenadas na textura, que por sua vez se manifesta nas altas frequências. Apesar do tipo de material do cocho e/ou do terreno capturados como ruído poderem apresentar texturas semelhantes, o modelo multi-entrada, que processa o par de imagens, original e com filtro passa-alta, incrementou as acurácias da maioria das classes do problema em comparação com os modelos que recebem como entrada apenas um dos dois tipos de imagens.

Uma vez que o *dataset* está sendo construído especificamente para o problema proposto, continuar o trabalho de captura de mais amostras é fundamental, procurando sempre balancear as imagens entre os escores e/ou recomendações, e presando pela variabilidade de formas e materiais utilizados nas estruturas dos cochos, além de tipos diferentes de dietas. Com o aumento da variabilidade e do número de amostras, pode ser necessário explorar outras arquiteturas de redes neurais, como Redes Adversárias Generativas (*Generative Adversarial Network* - GAN), Redes Neurais Recorrentes (*Recurrent Neural Network* - RNN), Autoencoder Variacional (*Variotinal auto-encoder* - VAE), entre outras. Além disso, a recomendação quanto ao ajuste da dieta também pode ser levada em consideração, avaliando-se a possibilidade de um treinamento com três classes: aumento, manutenção e diminuição da oferta de dieta. Dado que o modelo multi-entrada demonstrou resultados satisfatórios, um ponto a ser explorado é a estratégia de filtragem, como alterar o desvio padrão do filtro gaussiano, aplicar o filtro laplaciano – e, portanto, sem a necessidade da operação de subtração dos píxeis das imagens originais – e utilizar imagens em escala de cinza. Por fim, como um dos objetivos do presente trabalho é fornecer um modelo que possa ser incorporado em um aplicação de tempo real, seria possível verificar a capacidade de generalização do modelo em situações de campo.

REFERÊNCIAS

- ALMEIDA, A.; CANTRELL, D. Novas tecnologias para confinamento no brasil: leitura de cochos. In: ABB. *Congresso Brasileiro de Buiatria*. Campo Grande, 2001.
- BEZEN, R.; EDAN, Y.; HALACHMI, I. Computer vision system for measuring individual cow feed intake using rgb-d camera and deep learning algorithms. *Computers and Electronics in Agriculture*, Elsevier, v. 172, p. 105345, 2020.
- BLOCH, V.; LEVIT, H.; HALACHMI, I. Assessing the potential of photogrammetry to monitor feed intake of dairy cows. *Journal of Dairy Research*, Cambridge University Press, v. 86, n. 1, p. 34–39, 2019.
- BOM, R. et al. Estimativa do crescimento de bovinos de corte em confinamento utilizando modelos não lineares. *Zootecnia: Nutrição e Produção Animal*, Editora Científica Digital, p. 124–133, 2020.
- CARDOSO, E. G. Engorda de bovinos em confinamento (aspectos gerais). *Embrapa Gado de Corte-Docmentos (INFOTECA-E)*, Campo Grande, MS: EMBRAPA-CNPGC, 1996., 1996.
- DYCK, N. et al. Feed prediction and management software for beef feedlots. *Canadian Biosystems Engineering*, NOT KNOWN, v. 49, p. 7, 2007.
- ELHOSEINY, M.; HUANG, S.; ELGAMMAL, A. Weather classification with deep convolutional neural networks. In: IEEE. *2015 IEEE International Conference on Image Processing (ICIP)*. Quebec, 2015. p. 3349–3353.
- FONTOURA, D. C. et al. Um modelo de aplicativo para dispositivos móveis com reconhecimento de voz para coleta de dados de pesagem de bovinos em apoio ao processo de tomada de decisão de produtores rurais. In: SADIO. *X Congresso de AgroInformática (CAI)-JAIIO 47*. Caba, 2018. p. 266–279.
- FORMIGONI, I. *Maiores rebanhos e produtores de carne bovina previstos para 2021*. 2020. Acesso em: 26 de julho de 2021. Disponível em: <<https://www.farmnews.com.br/mercado/maiores-rebanhos-e-produtores-de-carne-bovina-2/>>.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. Harlow: Pearson Education Limited, 2018.
- HORTON, J. M. Bunk management, feed delivery and water trough management. *Cattle Feeding: A Guide to Management*. RC Albin and GB Thompson, ed. Trafton Printing Inc., Amarillo, TX, p. 137, 1990.
- HOSHIRO, S.; GOLDSCHMIDT, R. Processamento de imagens. *Rio de Janeiro: Seção de Engenharia de Computação*, p. 1, 2008.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, Wiley Online Library, v. 160, n. 1, p. 106–154, 1962.
- IBGE. Produção da pecuária municipal. v. 47, p. 1–8, 2019.

- KINGMA, D. P.; BA, J. Adam: a method for stochastic optimization (2014). *arXiv preprint arXiv:1412.6980*, v. 22, 2014.
- LEÃO, J. M. et al. Uso da termografia infravermelha na pecuária de precisão. *Embrapa Gado de Leite-Artigo em periódico indexado (ALICE)*, Cadernos Técnicos de Veterinária e Zootecnia, n. 79, p. 97-109, 2015., 2015.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, MIT Press, v. 1, n. 4, p. 541–551, 1989.
- LI, Q. et al. Medical image classification with convolutional neural network. In: IEEE. *2014 13th international conference on control automation robotics & vision (ICARCV)*. Singapore, 2014. p. 844–848.
- LOY, D. D. et al. Iowa beef center. *Iowa State University Animal Industry Report*, Iowa State University Digital Press, v. 13, n. 1, 2016.
- LUZ, G. B. et al. Exigências nutricionais, cálculos de dieta e mensuração de sobras no manejo nutricional de vacas leiteiras. *Pesquisa Agropecuária Gaúcha*, v. 25, n. 1/2, p. 16–31, 2019.
- MACHADO, J. G. d. C. F.; NANTES, J. F. D. Adoção da tecnologia da informação em organizações rurais: o caso da pecuária de corte. *Gestão & Produção*, SciELO Brasil, v. 18, p. 555–570, 2011.
- MAGGIORI, E. et al. Fully convolutional neural networks for remote sensing image classification. In: IEEE. *2016 IEEE international geoscience and remote sensing symposium (IGARSS)*. Beijing, 2016. p. 5071–5074.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- NUNES, S. G. Curral para bovinos de corte "módulo 500". *Embrapa Gado de Corte-Circular Técnica (INFOTECA-E)*, Campo Grande, MS: EMBRAPA-CNPGC, 1983., 1983.
- PRITCHARD, R.; BRUNS, K. Controlling variation in feed intake through bunk management. *Journal of Animal Science*, Oxford University Press, v. 81, n. 14_suppl_2, p. E133–E138, 2003.
- PRITCHARD, R. H. Bunk management. In: LAND O'LAKES INC. *Land O'Lakes Beef Seminar*. Cedar Rapids, 1993. p. 4–15.
- ROBBINS, H.; MONRO, S. A stochastic approximation method. *The annals of mathematical statistics*, JSTOR, p. 400–407, 1951.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- RUSCHE, W. *Feed Bunk Management*. 2020. Acesso em: 29 de outubro de 2021. Disponível em: <<https://extension.sdstate.edu/feed-bunk-management>>.

SENAR. Bovinocultura: manejo e alimentação de bovinos de corte em confinamento. *Coleção SENAR*, v. 232, p. 56, 2018.

SZELISKI, R. *Computer vision: algorithms and applications*. London: Springer Science & Business Media, 2010.

UYEH, D. D. et al. Interactive livestock feed ration optimization using evolutionary algorithms. *Computers and Electronics in Agriculture*, Elsevier, v. 155, p. 1–11, 2018.

YAO, L. et al. Cow face detection and recognition based on automatic feature extraction algorithm. In: ACM. *Turing Celebration Conference*. Chengdu, 2019. p. 1–5.