

Colorização de imagens com Deep Learning

Giovana de Lucca, Elloá B. Guedes

¹Núcleo de Computação
Escola Superior de Tecnologia
Universidade do Estado do Amazonas (UEA)
Manaus – AM – Brasil

gol.eng@uea.edu.br, ebgbcosta@uea.edu.br

1. Introdução

As técnicas de Aprendizado de Máquina (AM) têm sido aplicadas com sucesso em um grande número de problemas reais em diversos domínios. A principal razão deste sucesso é decorrente da natureza inferencial e da boa capacidade de generalização dos métodos e técnicas desta área, cuja ideia central consiste em utilizar algoritmos capazes de aprender padrões por meio de exemplos, baseando-se apenas em dados previamente disponíveis (FACELI et al., 2011; GULLI; PAL, 2017).

A Visão Computacional (VC), por sua vez, é uma área que procura desenvolver métodos capazes de replicar nos computadores as capacidades da visão humana. O reconhecimento de imagens é uma parte integrante do processo de VC e aplica as técnicas de processamento digital de imagens para extrair as características desejadas (KHAN et al., 2018). Antes da difusão das técnicas de AM, o procedimento de extração de características de imagens envolvia um grande esforço devido ao processamento e à aplicação de uma variedade de métodos matemáticos. Em algumas situações, por exemplo, até mesmo análises de especialistas eram utilizadas para descobrir as regras necessárias para o reconhecimento e extração de certos padrões (FACELI et al., 2011).

Devido ao progresso significativo no campo da VC e da tecnologia de sensores visuais, VC e AM desempenharam juntos papéis decisivos no desenvolvimento de uma variedade de aplicações baseadas em processamento digital de imagens (KHAN et al., 2018). Por exemplo, o reconhecimento de dígitos manuscritos proposto pelo *dataset* MNIST (*Modified National Institute of Standards and Technology*) foi uma das aplicações pioneiras que congregou VC e AM, difundida até os dias atuais para fins didáticos de métodos e técnicas deste domínio (MNIST, 2018). Atualmente, em particular, um dos projetos emergentes é o ImageNet, o qual disponibiliza um grande banco de imagens projetado para tarefas de VC e propõe anualmente um desafio chamado *ImageNet Large Scale Visual Recognition Challenge*, que visa elencar os melhores algoritmos em âmbito mundial para detecção de objetos e classificação de imagens em larga escala (IMAGENET, 2018).

Nas últimas décadas, com a crescente complexidade dos problemas a serem tratados computacionalmente e diante do grande volume de dados constantemente gerados por diferentes setores, tornou-se clara a necessidade de ferramentas, algoritmos, métodos e técnicas computacionais mais sofisticados para endereçar estas questões (FACELI et al., 2011). Embora alguns algoritmos de AM já existissem há bastante tempo, a demanda por aplicar automaticamente cálculos matemáticos complexos em uma grande escala de dados tornou-se uma necessidade particularmente mais recente. Embora tenha havido

um aumento do poder de processamento e armazenamento pelos dispositivos computacionais disponíveis, especialmente via Computação em Nuvem, novas estratégias de AM precisavam ser desenvolvidas ou evoluírem a partir de técnicas já existentes. Neste sentido, as técnicas de *Deep Learning* (DL) emergiram, principalmente compreendendo as redes neurais com uma grande quantidade de camadas ocultas e operações convolucionais (KHAN et al., 2018).

Entende-se por DL, um grande conjunto de camadas de redes neurais artificiais cuja principal característica é aprender representações a partir de dados (BUDUMA, 2017; CHOLLET, 2017). As três principais vantagens oferecidas pelo DL são a simplicidade da construção das redes, a escalabilidade, por lidar com um grande volume de dados, e a transferência de conhecimento (do inglês, *Transfer Learning* – TL), pois um modelo treinado para uma determinada tarefa pode ser adaptado para outras tarefas relacionadas (KHAN et al., 2018).

Considerando este cenário e almejando a aplicação destes conceitos de vanguarda, a proposta do presente trabalho de conclusão de curso visa explorar as arquiteturas canônicas de redes neurais convolucionais, utilizando as técnicas de DL e TL, para endereçar o problema da colorização artificial de imagens. Neste problema, uma imagem em tons de cinza, a exemplo de uma imagem histórica, é apresentada à uma rede neural convolucional que, como resposta, propõe uma versão colorida da mesma. Essa colorização deve ser plausível e realista ao ponto de não possuir inconsistência visual quando analisada por pessoas comuns.

Ao longo desta introdução serão mostrados os demais elementos que compõem este trabalho. A Seção 1.1 contempla os objetivos propostos para o desenvolvimento do projeto. Na Seção 1.2 são apresentadas as justificativas que motivam a realização do trabalho em questão. A metodologia adotada é detalhada na Seção 1.3. Por fim, a Seção 1.4 compreende o cronograma das atividades, seguido da Seção 1.5 que dispõe a organização do restante do documento.

1.1. Objetivos

O objetivo geral deste trabalho consiste em explorar estratégias para colorização artificial de imagens utilizando técnicas de *Deep Learning*. Para tanto, faz-se necessário elencar alguns objetivos específicos, descritos a seguir:

1. Consolidar uma base de dados representativa de imagens coloridas para treinamento das redes;
2. Descrever o problema da colorização artificial de imagens segundo uma tarefa de Aprendizado de Máquina;
3. Explorar a utilização das arquiteturas canônicas de redes neurais convolucionais mediante *Transfer Learning* aplicadas ao problema considerado;
4. Propor, treinar e testar diferentes redes neurais convolucionais baseadas nas arquiteturas canônicas elencadas;
5. Analisar os resultados obtidos de maneira quantitativa e qualitativa.

1.2. Justificativa

Imagens em escala de cinzas retêm informações que podem ser importantes em diversos aspectos. A colorização de fotos em arquivos antigos pode agregar algum valor aos seus

respectivos contextos históricos e artísticos. Algum detalhe de uma imagem em tons de cinza talvez possua outra interpretação se esta mesma imagem estivesse colorizada. Seguindo o mesmo raciocínio, a coloração das imagens de câmeras de segurança com baixa resolução pode influenciar nas interpretações das filmagens, permitindo, por exemplo, a identificação mais apropriada de indivíduos presentes nestas imagens.

Na área da Saúde, por exemplo, colorizações podem ser ajustadas e modificadas para restaurar algum tipo de perturbação visual de indivíduos, como é o caso do daltonismo. As técnicas de colorização artificial a serem exploradas neste trabalho podem, por exemplo, possibilitar representações visuais mais adequadas para os portadores deste tipo de agravio.

Além do que foi exposto, a proposta considerada neste trabalho incentiva a prática de conceitos, técnicas e tecnologias de uma área emergente da Computação, contribuindo na formação profissional da aluna concluinte. No mais, esta proposta de trabalho de conclusão de curso está alinhada com as atividades desenvolvidas pelo *Laboratório de Sistemas Inteligentes* (LSI), uma iniciativa promovida por docentes do Núcleo de Computação (NUCOMP) da Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (UEA), motivando o desenvolvimento de soluções inovadoras que utilizam técnicas emergentes do Aprendizado de Máquina.

1.3. Metodologia

Para atingir os objetivos propostos no escopo deste trabalho, a condução das atividades obedece à metodologia apresentada a seguir, composta dos seguintes passos:

1. Estudo dos conceitos relacionados à *Machine Learning*, *Deep Learning* e as principais arquiteturas de redes neurais convolucionais;
2. Estudo do ferramental tecnológico para elaboração e execução de projetos de *Deep Learning*, incluindo Python, Keras, Sci-kit Learn, Google Cloud Platform, dentre outros;
3. Elaborar uma base de dados representativa de imagens coloridas e em escalas de cinza para fins de aprendizado dos padrões de coloração pelas redes neurais convolucionais;
4. Elencar um conjunto de arquiteturas canônicas das redes neurais convolucionais aplicáveis ao problema em questão;
5. Propor modificações nas redes neurais identificadas no passo anterior mediante *Transfer Learning*;
6. Treinar as redes modificadas com os exemplos da base de dados;
7. Testar as redes e coletar métricas de desempenho;
8. Analisar os resultados obtidos identificando as redes mais adequadas ao cenário considerado;
9. Escrita da proposta do Trabalho de Conclusão de Curso;
10. Defesa da proposta do Trabalho de Conclusão de Curso;
11. Escrita do Trabalho de Conclusão de Curso;
12. Defesa do Trabalho de Conclusão de Curso.

1.4. Cronograma

Uma visão geral do cronograma de atividades a serem desenvolvidas ao longo deste trabalho pode ser vista na Tabela 1. Essas atividades possuem relação com as atividades listadas na Seção 1.3, a qual detalha a metodologia que este trabalho deverá seguir.

Tabela 1: Cronograma de atividades levando em consideração os onze meses (02/2018 a 12/2018) para realização do presente trabalho de conclusão de curso.

	2018											
	02	03	04	05	06	07	08	09	10	11	12	
Estudo dos conceitos teóricos relacionados à <i>Machine Learning</i>	x	x	x	x								
Estudo do ferramental tecnológico para elaboração do projeto	x	x	x	x								
Consolidação da base de dados			x	x								
Especificação das arquiteturas canônicas de redes neurais convolucionais			x									
Aplicação das técnicas de <i>Transfer Learning</i> nas redes neurais convolucionais identificadas				x								
Treinamento das redes neurais convolucionais com os exemplos da base de dados				x	x	x	x					
Testes das redes e comparação de métricas de desempenho					x	x	x	x				
Análise dos resultados obtidos									x	x	x	
Escrita da proposta do trabalho	x	x	x	x	x							
Defesa da proposta do trabalho					x							
Escrita do trabalho final						x	x	x	x	x	x	
Defesa do trabalho final											x	

1.5. Organização do Documento

Para apresentar a proposta do presente trabalho de conclusão de curso, este documento está organizado como segue. A Seção 2 contempla os fundamentos teóricos necessários para elaboração do projeto, incluindo conceitos de *Machine Learning*, *Deep Learning* e as principais arquiteturas de redes neurais convolucionais. A Seção 3 discorre sobre os trabalhos relacionados. A solução proposta para o tema de colorização de imagens utilizando técnicas de *Deep Learning* é abordada na Seção 4. Por fim, as considerações parciais do trabalho encontram-se na Seção 5.

2. Fundamentação Teórica

Nesta seção serão apresentados os fundamentos teóricos que dão suporte à realização deste trabalho. As redes neurais artificiais, seus principais conceitos, arquiteturas, métodos de aprendizagem e algumas aplicações são discutidas na Seção 2.1. Os conceitos elementares sobre as técnicas de *Machine Learning* conhecidas como *Deep Learning* encontram-se na Seção 2.2. Na Seção 2.3 são descritas as características das redes neurais utilizadas pela técnica de *Deep Learning*, as Redes Neurais Convolucionais. Por fim, as tecnologias utilizadas para a realização deste trabalho são apresentadas na Seção 2.5.

2.1. Redes Neurais Artificiais

As *Redes Neurais Artificiais* (RNAs) são modelos computacionais inspirados na capacidade de processamento de informações do cérebro humano (ROJAS, 1996). De acordo com esta ideia, as RNAs possuem unidades de processamento simples, denominadas

neurônios artificiais, dispostos em camadas interconectadas por ligações associadas a coeficientes numéricos, chamados *pesos* (FACELI et al., 2011). As RNAs são capazes de aprenderem padrões complexos a partir dos dados e prever resultados para exemplos não conhecidos, o que demonstra a sua capacidade de generalização (HAYKIN, 2009).

O neurônio artificial é a unidade fundamental na construção de RNAs, tendo sido inspirado no seu análogo biológico. Segundo Rosenblatt, existe um conjunto de m entradas, equivalentes aos dendritos de um neurônio biológico, por onde os sinais são introduzidos (ROSENBLAT, 1961). Associa-se um peso a cada entrada, representando a relevância referente a uma conexão sináptica. Há também o peso w_0 , um termo de polarização criado com a intenção de estabelecer um limiar de ativação para cada neurônio. Este peso corresponde à entrada *bias*, cujo valor é sempre unitário. Pode-se então definir um vetor de entradas $X = [+1, x_1, x_2, \dots, x_m]$ e um vetor de pesos $W = [w_0, w_1, \dots, w_m]$. As entradas e pesos são combinados por meio de uma função $\phi : \mathbb{R}^{m+1} \rightarrow \mathbb{R}$, que é geralmente a soma ponderada das entradas e pesos, conforme Equação 1. Este modelo de neurônio encontra-se ilustrado na Figura 1 (LIMA, 2016).

$$\phi(X, W) = \sum_{i=0}^m x_i \cdot w_i. \quad (1)$$

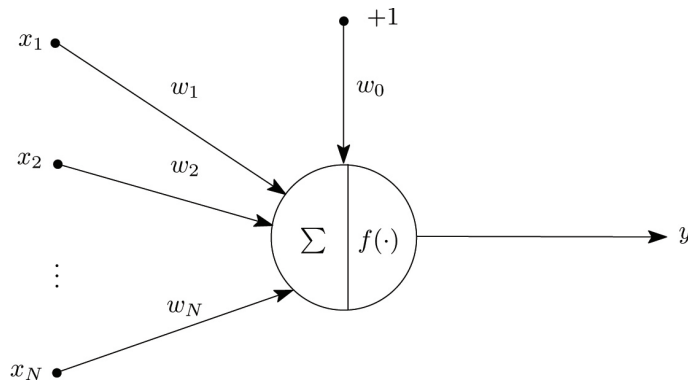


Figura 1: Neurônio artificial: a combinação linear das entradas x ponderadas pelos pesos w é transformada pela função de ativação f na saída y .

A função f é chamada de *função de ativação* e fornece a resposta de um neurônio para uma dada entrada. Esta função precisa ser monotônica e contínua, podendo comumente ser as funções identidade, sigmóide, tangente hiperbólica, ou a retificada linear (ReLU) (DELICATO; PIRES; SILVEIRA, 2017). Estas funções encontram-se representadas na Figura 2.

Neurônios artificiais individuais têm uma capacidade computacional limitada, independentemente da função de ativação escolhida, pois resolvem apenas problemas linearmente separáveis. No entanto, um conjunto de neurônios artificiais conectados na forma de uma rede – *rede neural artificial* – adquirem a capacidade de resolver problemas de elevada complexidade (BRAGA; CARVALHO; LUDERMIR, 2007). A alternativa mais utilizada para resolver estes problemas é distribuir os neurônios em uma ou mais camadas conhecidas como camadas ocultas (FACELI et al., 2011). Segundo Cybenko, uma rede com uma camada oculta pode implementar qualquer função contínua e uma rede com

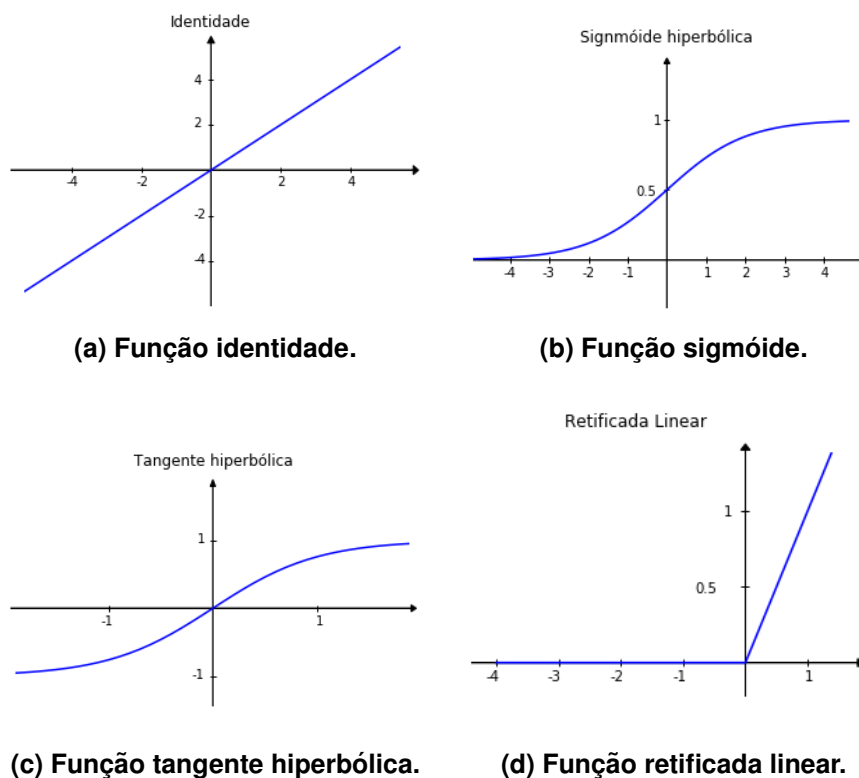


Figura 2: Exemplos de diferentes funções de ativação.

duas camadas ocultas permite a aproximação de qualquer função (CYBENKO, 1989).

As RNAs do tipo Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*) apresentam uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Uma das principais características de uma rede MLP é o seu alto grau de conectividade entre os neurônios, cuja intensidade está associada aos pesos da rede.

Cada neurônio em uma rede MLP atua ponderando as entradas recebidas dos neurônios de uma camada anterior a ele conectados, produzindo como saída um valor, resultante de sua função de ativação, que é propagado às camadas seguintes da rede neural. Conforme exemplificado na Figura 3, a combinação das atuações individuais desempenhadas por cada neurônio da rede que define a atuação associada à RNA como um todo (BRAGA; CARVALHO; LUDERMIR, 2007; FACELI et al., 2011; HAYKIN, 2009).

Uma importante característica das RNAs é a sua capacidade de aprender por meio de exemplos. O processo de *aprendizado* de uma rede neural consiste em sucessivos ajustes de pesos associados aos seus neurônios, de modo a aprimorar seu desempenho de acordo com um critério pré-estabelecido. Tais ajustes são realizados por algoritmos de treinamento formados por um conjunto de regras bem definidas que especificam quando e como deve ser alterado o valor de cada peso. Diversos algoritmos de aprendizado foram propostos, dentre os quais se destacam aqueles que seguem o paradigma de *aprendizado supervisionado* (FACELI et al., 2011; LIMA, 2016).

O aprendizado supervisionado ajusta os pesos aplicando um conjunto de exemplos de treinamento rotulados. Cada exemplo consiste em um sinal de entrada associado à sua

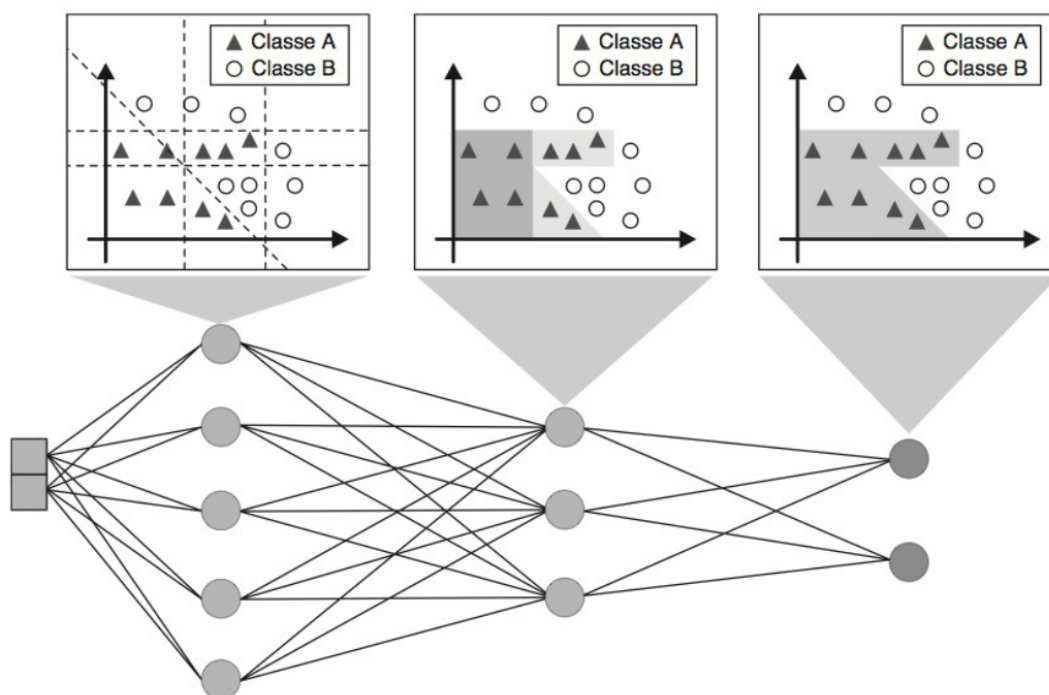


Figura 3: Papel desempenhado pelos neurônios das diferentes camadas de uma rede MLP. Fonte: (FACELI et al., 2011).

resposta alvo desejada. A cada padrão de entrada submetido à rede, compara-se a resposta desejada com a resposta calculada, ajustando-se os pesos das conexões para minimizar o erro (HAYKIN, 2009).

O algoritmo mais utilizado para o treinamento de redes MLP é o algoritmo *backpropagation*, também chamado de retropropagação do erro ou ainda regra delta generalizada. Este algoritmo respeita o aprendizado supervisionado em que os pesos são modificados e ajustados para reduzir a distância entre a resposta desejada e a resposta produzida pela rede (HAYKIN, 2009). O treinamento é constituído da iteração de duas fases, uma fase para frente (*forward*) e uma fase para trás (*backwards*) (FACELI et al., 2011). A fase *forward*, que compreende o fluxo da informação a partir da entrada até a saída da rede, é utilizada para produzir uma saída para um dado sinal de entrada. A fase *backwards*, com fluxo da informação da saída da rede em direção à entrada, utiliza a diferença entre as saídas desejada e produzida para atualizar os pesos das conexões entre os neurônios e assim minimizar o erro (BRAGA; CARVALHO; LUDERMIR, 2007). Os ciclos de apresentação dos dados de treinamento e eventuais ajustes de pesos no *backpropagation* são iterados até que seja atingido um critério de parada como, por exemplo, um número máximo de ciclos ou uma taxa máxima de erro (FACELI et al., 2011).

As RNAs são modelos computacionais com ampla aplicação na resolução de problemas de previsão. Algumas aplicações utilizam RNAs para previsão de condições climáticas, como em (SOUSA et al., 2017) e (ARAUJO et al., 2017), em que se deseja prever precipitações de chuva de determinados locais. Diversas outras aplicações de RNAs dizem respeito à classificação de padrões. Dentre as aplicações na área de classificação financeira, uma das mais bem sucedidas é a análise de crédito (BRAGA;

CARVALHO; LUDERMIR, 2007). Além disso, as RNAs também são muito utilizadas para diagnósticos médicos, como em (SILVA et al., 2016) e (PEREIRA et al., 2016). Outras aplicações tratam de reconhecimento de caracteres (CARVALHO, 2006), robótica (TINOS, 1999), jogos (CAEXETA, 2008), comunicação (SILVA et al., 2017), dentre outros.

2.2. Deep Learning

Deep Learning (DL), ou Aprendizado Profundo, é uma subárea do AM e consiste em uma abordagem que utiliza uma grande quantidade de camadas de RNAs para aprender representações de um vasto volume dados (CHOLLET, 2017; KHAN et al., 2018). Informalmente, a palavra *profundo* se refere ao volume de camadas e neurônios presentes nas arquiteturas das redes (GULLI; PAL, 2017). Através de análise de padrões, os sistemas baseados em técnicas de DL são capazes de reconhecer, traduzir, sintetizar e até prever sinais das mais diferentes naturezas (DELICATO; PIRES; SILVEIRA, 2017). Para tarefas voltadas para a área de Visão Computacional, os métodos utilizados pelas técnicas de DL envolvem as redes neurais chamadas Redes Neurais Convolucionais (KHAN et al., 2018).

As técnicas de DL têm sido aplicadas com êxito em diversos problemas de diferentes naturezas (DELICATO; PIRES; SILVEIRA, 2017). Essas técnicas de vanguarda e a crescente disponibilidade de grandes conjuntos de dados de imagens médicas estão produzindo notáveis avanços na compreensão automatizada deste tipo de dado. Essa perspectiva propõe os chamados diagnósticos médicos auxiliados por computador, do inglês *Computer-Aided Diagnosis* (CAD), os quais podem ser utilizados, por exemplo, para detecção automatizada de doenças e para segmentação de órgãos e lesões (LU et al., 2017). Considerando ainda o contexto de análise de imagens, o projeto ImageNet disponibiliza gratuitamente um banco de imagens para aplicações deste domínio e propõe, desde 2010, um desafio anual intitulado *ImageNet Large Scale Visual Recognition Challenge*, que visa elencar os melhores algoritmos em âmbito mundial para detecção de objetos e classificação de imagens em larga escala (IMAGENET, 2018).

2.3. Redes Neurais Convolucionais

As *Redes Neurais Convolucionais*, do inglês *Convolutional Neural Networks* (CNNs), são modelos de redes neurais especializados em processamento de dados compostos pela união de vários segmentos elementares denominados camadas (GOODFELLOW; BENGIO; COURVILLE, 2016). Cada camada possui uma finalidade específica e implementa uma determinada funcionalidade básica, como convolução, normalização, *pooling*, etc (KHAN et al., 2018).

2.3.1. Convolução

A *convolução* é uma operação linear que calcula a soma dos produtos de toda a extensão de duas entradas em função de um determinado deslocamento. Essa operação é a propriedade fundamental da camada convolucional, a principal camada de uma CNN (GOODFELLOW; BENGIO; COURVILLE, 2016). O principal objetivo da operação de convolução nas CNN é a extração das características de uma determinada entrada (SEWAK; KARIM; PUJARI, 2018).

O processo de convolução utilizado nas CNNs é aplicado em um conjunto de *filtros* e uma dada entrada para gerar uma saída conhecida como *mapa de características*. A camada convolucional recebe um volume de entrada de n dimensões e pode possuir um preenchimento p de zeros (*zero-padding*), aplicado ao redor da entrada. Essa entrada é processada por k filtros que representam os pesos e as conexões da CNN (KHAN et al., 2018). Cada filtro consiste em uma matriz de números discretos e possui uma extensão espacial e , que é igual ao valor da altura e da largura do filtro, e um *stride* s , que é a distância entre as aplicações de convolução consecutivas do filtro no volume de entrada (BUDUMA, 2017).

A Figura 4 exemplifica o processo de convolução aplicado em uma entrada de tamanho 5×5 e *zero-padding* $p = 1$. O filtro utilizado no exemplo possui extensão espacial $e = 2$ e *stride* $s = 2$ com inicialização de pesos aleatória. É possível observar que o mapa de características é uma saída de tamanho 3×3 em que cada um dos seus componentes é a soma das multiplicações dos elementos do filtro com os elementos de um segmento da entrada (KHAN et al., 2018).

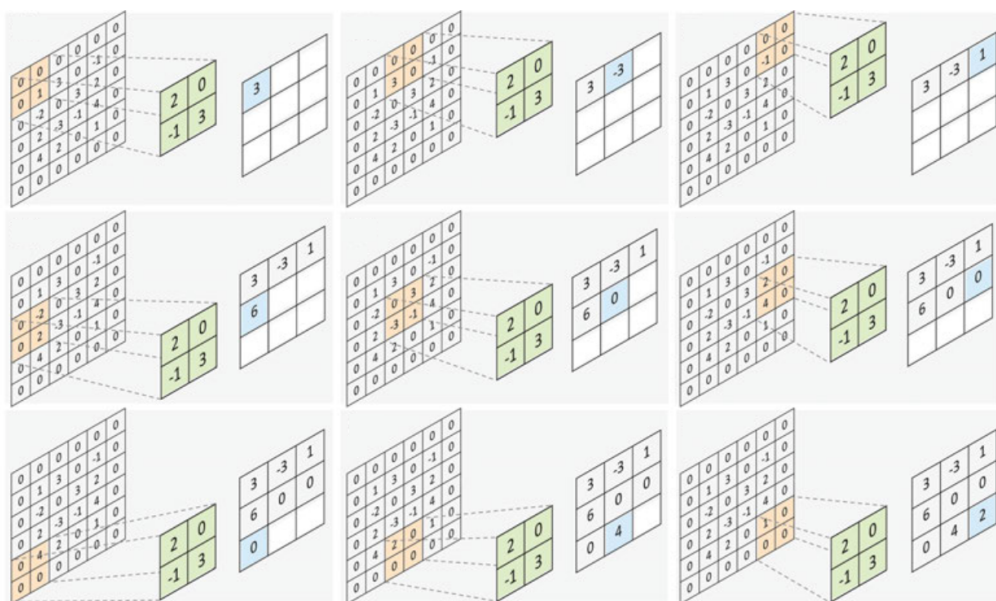


Figura 4: Exemplo de um processo de convolução aplicado em uma entrada de tamanho 5×5 e um filtro de tamanho 2×2 . Fonte: (KHAN et al., 2018).

2.3.2. Camadas de uma CNN

Para reduzir significativamente a dimensão dos mapas de características, eventualmente, após a camada convolucional, é utilizada uma camada chamada *pooling*. A camada de *pooling* divide o mapa de características de entrada em blocos de tamanhos iguais e processa cada bloco para criar um mapa de características condensado. O processamento dos blocos é definido por uma função de *pool* que pode ser, por exemplo, a função máxima. Assim como na camada de convolução, na camada de *pooling* também é preciso especificar o tamanho da região de *pooling* e o *stride* s da operação, conforme exemplificado

na Figura 5 em que a região possui tamanho 2×2 e o *stride* $s = 1$ (BUDUMA, 2017; KHAN et al., 2018).

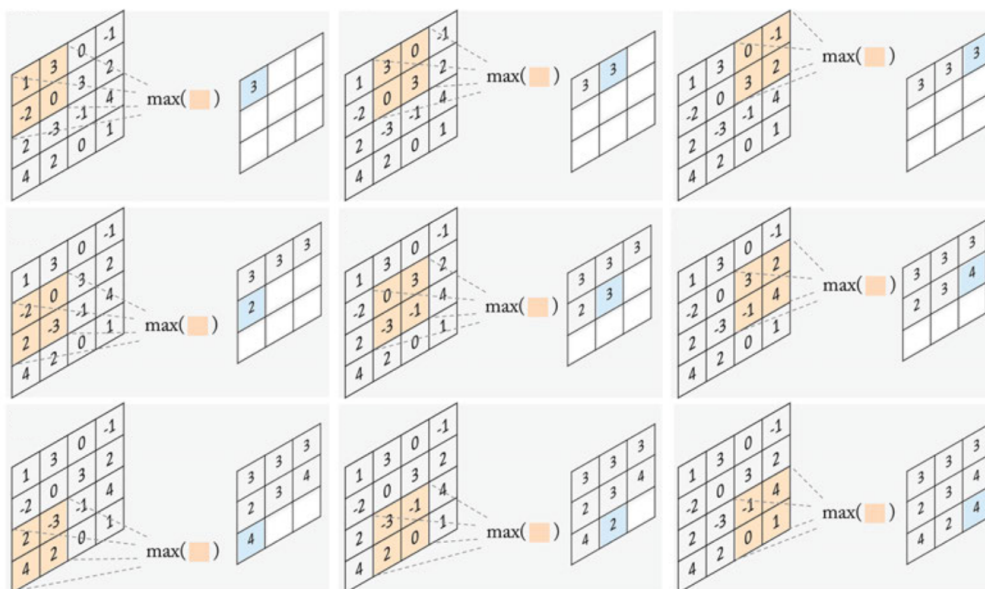


Figura 5: Exemplo de um processo de *pooling* aplicado em uma entrada de tamanho 5×5 e uma região *pooling* de tamanho 2×2 . Fonte: (KHAN et al., 2018).

As últimas camadas de uma CNN normalmente são as *Fully Connected Layers* (FCL), ou camadas completamente conectadas, as quais correspondem essencialmente às camadas de convolução que utilizam filtros de tamanho 1×1 em que cada elemento é densamente conectado à todos os elementos da camada anterior (KHAN et al., 2018). Nesta última camada, adota-se tipicamente o uso da função *Softmax*, também chamada de função exponencialmente normalizada, a qual escala o vetor de saída de forma que seus elementos pertençam a m intervalo entre 0 e 1 e que quando somados sejam igual a 1 (DELICATO; PIRES; SILVEIRA, 2017). O processo de aprendizagem utilizado pelas FCL é baseado nos conceitos das RNAs do tipo MLP e no algoritmo de treinamento *backpropagation* (GULLI; PAL, 2017; KHAN et al., 2018).

Durante o processo de treinamento, as CNNs utilizam algumas técnicas para tentar evitar possíveis problemas, como *overfitting*. Qualquer modificação feita em um algoritmo de AM com o objetivo de reduzir os erros de generalização é chamada de regularização (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma das abordagens mais populares utilizadas pelas CNNs para regularização é a técnica de *Dropout*, a qual utiliza uma probabilidade p de ativação em cada neurônio durante o treinamento. Essa abordagem elimina temporariamente alguns neurônios e suas respectivas ligações de acordo com a probabilidade a qual normalmente é $p = 0.5$. A camada *Dropout* de ativação de um CNN aumenta de forma significativa o desempenho da rede para prever dados não vistos na fase de teste (KHAN et al., 2018).

2.3.3. Arquiteturas canônicas de Redes Neurais Convolucionais

Ao longo dos últimos anos, o desafio proposto pelo projeto ImageNet *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) proporcionou o surgimento de algumas arquiteturas de CNNs que se tornaram bastante populares e são bastante utilizadas para resolver problemas dos mais diferentes domínios. As composições de algumas destas arquiteturas são detalhadas a seguir:

LeNet Yann le Cun desenvolveu, em 1990, uma das primeiras arquiteturas utilizadas para o reconhecimento de dígitos manuscritos, a LeNet. Vencedora do ILSVRC 2010, esta arquitetura é composta por três camadas convolucionais alternadas com camadas de *pooling* seguidas de duas FCLs conforme representado na Figura 6 (SEWAK; KARIM; PUJARI, 2018; KHAN et al., 2018).

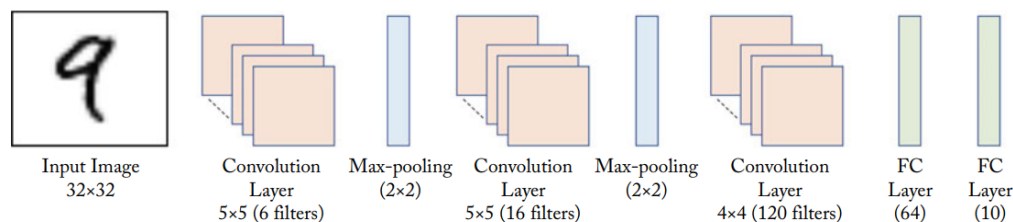


Figura 6: Representação da arquitetura de CNN LeNet. Fonte: (KHAN et al., 2018).

AlexNet Em 2012 a vencedora do ILSVRC foi a arquitetura proposta por Alex Krizhevsky, conhecida como AlexNet. A AlexNet é mais profunda e uma versão muito mais ampla da arquitetura LeNet (JOSHI et al., 2017). A principal diferença entre a AlexNet e as CNNs predecessoras é a sua maior profundidade que lida muito bem com sua grande quantidade de parâmetros, além da utilização de artifícios como *dropout* e *data augmentation*. As cinco primeiras camadas da arquitetura AlexNet são camadas de convolução e *pooling* alternadas de forma similar à LeNet, porém, seguem-se mais duas camadas uma convolucional e uma *depooling*. As três últimas camadas são FCL, mas além destas existem camadas *dropout* que ajudam a reduzir *overfitting*, o que resulta em uma melhor generalização (KHAN et al., 2018). A AlexNet encontra-se representada na Figura 7.

VGGNet A arquitetura VGGNet é uma das arquiteturas mais populares desde sua criação em 2014, apesar de não ter sido a vencedora do ILSVRC 2014. A razão de sua popularidade se dá pela simplicidade do modelo e pelo uso de pequenos “*kernels*” de convolução que lidam muito bem com redes profundas por reduzirem o número de parâmetros o que aumenta a eficiência do treinamento. A arquitetura VGGNet usa estritamente *kernels* de convolução de dimensão 3 x 3 combinados com camadas de *pooling* para extração de características e um conjunto de três FCLs. Além das camadas de convolução,

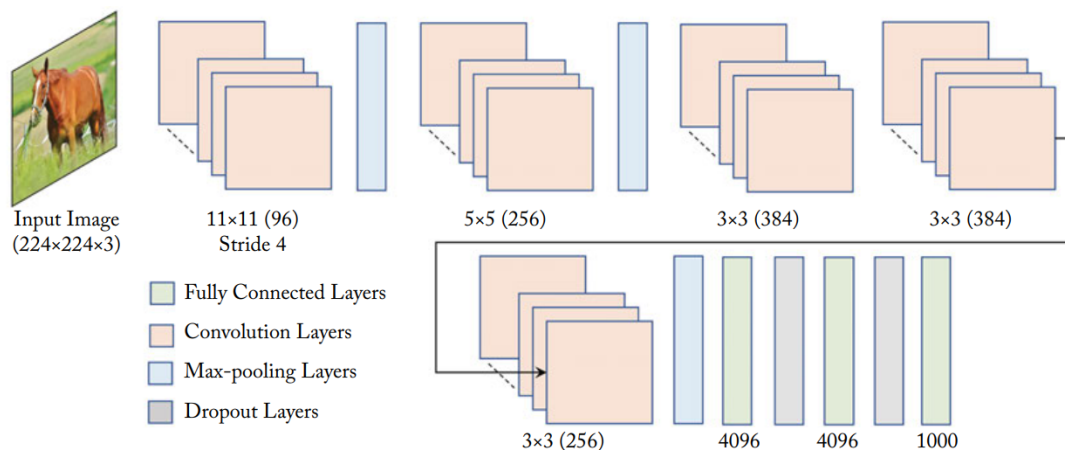


Figura 7: Representação da arquitetura de CNN AlexNet. Fonte: (KHAN et al., 2018).

pooling e das camadas conectadas, esta arquitetura também possui as camadas *dropout* como pode ser observado na Figura 8 (KHAN et al., 2018).

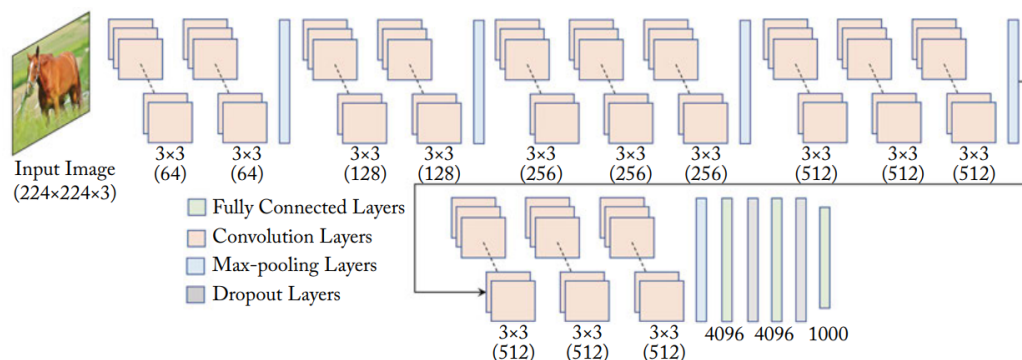


Figura 8: Representação da arquitetura de CNN VGGNet. Fonte: (KHAN et al., 2018).

GoogLeNet Desenvolvida pela empresa Google e vencedora do ILSVRC 2014, a arquitetura GoogLeNet possui 22 camadas baseadas em um módulo elementar chamado *Inception Module*. O processamento desses módulos ocorre de forma paralela diferentemente do processamento sequencial das arquiteturas discutidas anteriormente. A ideia central da arquitetura GoogLeNet é paralelizar os módulos e combinar as características da saída sem se preocupar com as funções individuais de cada camada. No entanto, essa abordagem resulta em um mapa de características com muitos elementos, mas para contornar este problema, após a execução do primeiro módulo, a rede realiza uma redução de dimensionalidade utilizando uma FCL antes de continuar o processo de treinamento (KHAN et al., 2018). A representação da arquitetura GoogLeNet encontra-se na Figura 9.

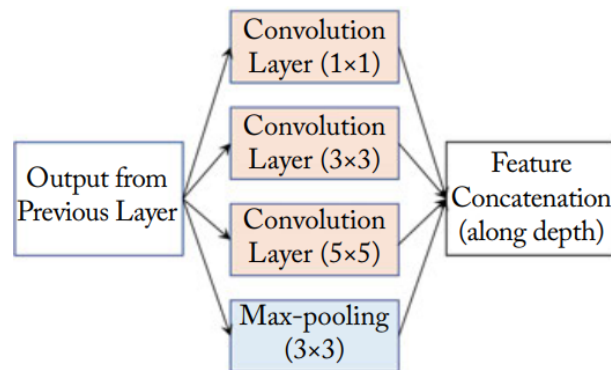


Figura 9: Representação da arquitetura de CNN GoogLeNet. Fonte: (KHAN et al., 2018).

2.4. *Transfer Learning*

Transfer Learning (TL), ou Transferência de Conhecimento, é uma poderosa técnica de DL a qual possui diversas aplicações em diferentes domínios (GULLI; PAL, 2017). Ao invés de estruturar uma arquitetura de uma CNN e treiná-la por completo, esta técnica permite reutilizar uma rede pré-treinada e adaptá-la a um novo conjunto de dados (SEWAK; KARIM; PUJARI, 2018). Modelos que foram pré-treinados utilizando um vasto e genérico conjunto de dados conseguem capturar características universais, como por exemplo curvas e arestas, em suas primeiras camadas (ZACCONE; KARIM, 2018).

As técnicas de TL podem ser utilizadas de diferentes maneiras, baseando-se nas arquiteturas das CNNs. Existem alguns modelos disponíveis para aplicações que foram pré-treinados utilizando as principais arquiteturas canônicas de CNN e aprenderam as características de grandes conjuntos de dados bastante conhecidos, como o ImageNet e o Places205 (IMAGENET, 2018; PLACES205, 2018). Para diferentes tarefas, esses modelos podem ser alterados modificando a camada de saída e fazendo um retreinamento nas últimas camadas das redes para se obter o aprendizado desejado (KHAN et al., 2018).

2.5. Tecnologias Utilizadas

3. Trabalhos Relacionados

4. Solução Proposta

4.1. Visão Geral da solução proposta

4.2. Análise Comparativa

5. Considerações Parciais

Referências

ARAÚJO, N. et al. Previsão do volume mensal de precipitações em manaus, amazonas com redes neurais aritificiais. In: *Anais do IV Encontro Regional de Informática da Região Norte I*. Manaus, Amazonas: ERIN, 2017. p. 108–116.

BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. 2. ed. Rio de Janeiro, Rio de Janeiro: LTC, 2007.

BUDUMA, N. *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. 1. ed. Sebastopol, California: O'Reilly Media, 2017.

CAEXETA, G. S. *VisionDraughts - Um Sistema de Aprendizagem de Jogos de Damas em Redes Neurais, Diferenças Temporais, Algoritmos Eficientes de Busca em Árvore e Informações Perfeitas Contidas em Bases de Dados*. Uberlândia, Minas Gerais: Universidade Federal de Uberlândia, 2008.

CARVALHO, C. E. S. de. *Reconhecimento de Caracteres Manuscritos Utilizando Redes Neurais Artificiais*. Brasília, Distrito Federal: Centro Universitário de Brasília, 2006.

CHOLLET, F. *Deep Learning with Python*. 1. ed. Shelter Island, New York: Manning, 2017.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, v. 2, p. 303–314, December 1989.

DELICATO, F.; PIRES, P.; SILVEIRA, I. Jornada de Atualização em Informática 2017. In: CSBC. *Congresso da Sociedade Brasileira de Computação*. Porto Alegre, Rio Grande do Sul: SBC, 2017. cap. 6, p. 212–260.

FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. 1. ed. Rio de Janeiro, Rio de Janeiro: LTC, 2011.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. 1. ed. Cambridge, Massachusetts: The MIT Press, 2016. v. 1. (Adaptive computation and machine learning series, v. 1).

GULLI, A.; PAL, S. *Deep Learning with Keras*. 1. ed. Birmingham, West Midlands: Packt, 2017.

HAYKIN, S. S. *Neural Networks and Learning Machines*. 3. ed. Upper Saddle River, New Jersey: Perason, 2009.

IMAGENET. *ImageNet*. 2018. Disponível em: <http://www.image-net.org/>.

JOSHI et al. *Information and Communication Technology for Intelligent Systems (ICTIS 2017) Volume 2*. 1. ed. [S.l.]: Springer, 2017. v. 2. (Smart innovation systems and technologies 84, v. 2).

KHAN, S. et al. *A Guide to Convolutional Neural Networks for Computer Vision*. 1. ed. Australia: Morgan and Claypool, 2018. v. 1. (Synthesis Lectures on Computer Vision, v. 1).

LIMA, P. M. de. *Redes Neurais para Predição de Séries Temporais de Precipitação em Manaus, Amazonas*. 2016. Trabalho de Conclusão de Curso da Universidade do Estado do Amazonas. Universidade do Estado do Amazonas.

LU, L. et al. *Deep Learning and Convolutional Neural Networks for Medical Image Computing*. Cham, Switzerland: Springer, 2017. (Advances in Computer Vision and Pattern Recognition).

MNIST. *THE MNIST DATABASE of handwritten digits*. 2018. Disponível em: <http://yann.lecun.com/exdb/mnist/>.

PEREIRA, R. et al. Abordagem deep learning para classificação de lesões mamárias. In: *Anais do XXXVI Congresso da Sociedade Brasileira de Computação*. Porto Alegre, Rio Grande do Sul: CSBC, 2016. p. 2597–2600.

PLACES205. *Places: the scene recognition database*. 2018. Disponível em: <http://places.csail.mit.edu/>.

ROJAS, R. *Neural Networks: A Systematic Introduction*. 1. ed. Heidelberg, Berlin: Springer, 1996.

ROSENBLAT, F. *Principles of Neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington, DC, 1961. 626 p.

SEWAK, M.; KARIM, R.; PUJARI, P. *Practical Convolutional Neural Networks*. 1. ed. Birmingham, West Midlands: Packt, 2018.

SILVA, A. G. et al. Classificação de expressões faciais negativas na língua brasileira de sinais. In: *Anais do IV Encontro Regional de Informática da Região Norte I*. Manaus, Amazonas: ERIN, 2017. p. 80–88.

SILVA, G. da et al. Classification of malignancy of lung nodules in ct images using convolutional neural network. In: *Anais do XXXVI Congresso da Sociedade Brasileira de Computação*. Porto Alegre, Rio Grande do Sul: CSBC, 2016. p. 2481–2489.

SOUSA, R. et al. Redes neurais artificiais aplicadas à previsão antecipada de precipitações na região central de manaus. In: *Anais do IV Encontro Regional de Informática da Região Norte I*. Manaus, Amazonas: ERIN, 2017. p. 89–97.

TINOS, R. *Detecção e Diagnóstico de Falhas em Robôs Manipuladores via Redes Neurais Artificiais*. São Carlos, São Paulo: Universidade de São Paulo, 1999.

ZACCONE, G.; KARIM, M. R. *Deep Learning with TensorFlow*. Birmingham, West Midlands: Packt, 2018.