

# Colorização de imagens com Deep Learning

Giovana de Lucca, Elloá B. Guedes

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

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

## 1. Introdução

### 1.1. Objetivos

O objetivo geral deste trabalho consiste em propor meios para colorização artificial de imagens utilizando técnicas de *Deep Learning*. Para tanto, fez-se necessário elencar alguns objetivos específicos descritos na subseção a seguir.

#### 1.1.1. Objetivos Específicos

1. Consolidar uma base de dados;
2. Explorar a utilização de arquiteturas canônicas de redes neurais convolucionais mediante *Transfer Learning* aplicadas ao problema de colorização;
3. Analisar os resultados obtidos de maneira quantitativa e qualitativa.

### 1.2. Justificativa

### 1.3. Metodologia

Para atingir os objetivos propostos no escopo deste trabalho, a condução das atividades seguiu a metodologia apresentada nesta seção, 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 com *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

A tabela a seguir contempla o cronograma das 12 atividades descritas na seção 1.3 distribuídas pelos meses de fevereiro à dezembro.

Atividade	02	03	04	05	06	07	08	09	10	11	12
1	x	x	x	x							
2	x	x	x	x							
3			x	x							
4			x								
5				x							
6				x	x	x	x				
7					x	x	x	x			
8									x	x	x
9	x	x	x	x	x						
10					x						
11						x	x	x	x	x	x
12											x

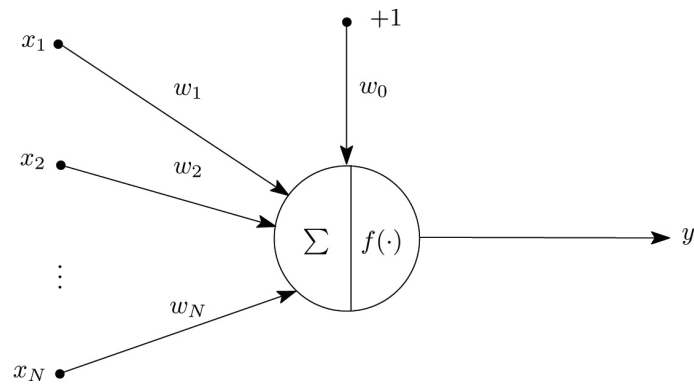
## 2. Fundamentação Teórica

### 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).

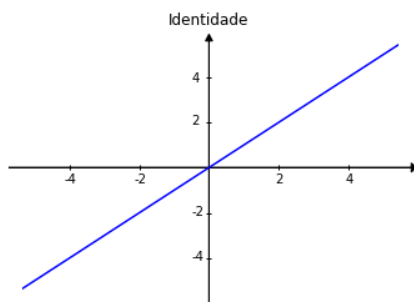
O neurônio artificial é a unidade fundamental na construção de RNAs, tendo sido inspirado no seu análogo biológico. Segundo Rosenblatt, existe um conjunto de  $m$  entradas, equivalentes aos dendritos de um neurônio biológico, por onde os sinais são introduzidos (ROSENBLAT, 1961). Associa-se um peso 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).

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

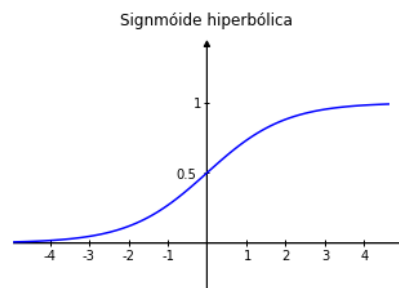


**Figura 1: Neurônio artificial: a combinação linear das entradas  $x$  ponderadas pelos pesos  $w$  é transformada pela função de ativação  $f$  na saída  $y$ .**

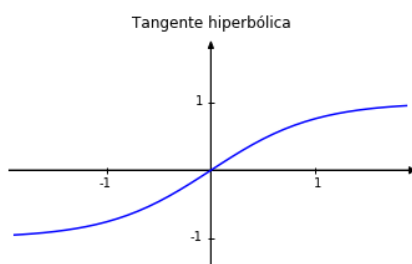
A função  $f$  é chamada de *função de ativação* e fornece a resposta de um neurônio para uma dada entrada. Esta função precisa ser monotônica e contínua, podendo comumente ser as funções identidade, sigmóide, tangente hiperbólica, ou a retificada linear (ReLU) (DELICATO; PIRES; SILVEIRA, 2017). Estas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



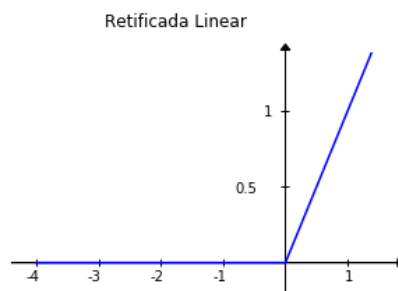
**(a) Função identidade.**



**(b) Função sigmóide.**



**(c) Função tangente hiperbólica.**



**(d) Função retificada linear.**

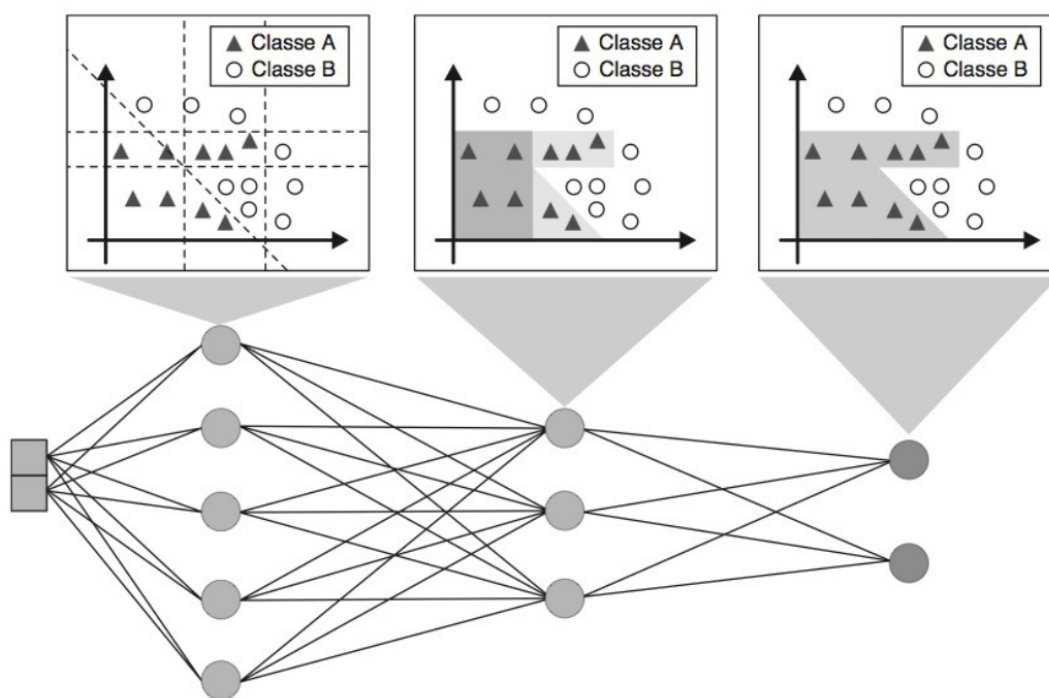
**Figura 2: Exemplos de diferentes funções de ativação.**

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 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).



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

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 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 *back-propagation*, 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 predição 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**

O aprendizado profundo ou *deep learning* é um modelo matemático de aprendizagem de máquina que utiliza RNAs para aprender representações a partir de dados (CHOLLET, 2017; BUDUMA, 2017). Informalmente, a palavra *profundo* se refere à grande quantidade de camadas e neurônios presentes nas arquiteturas das redes (GULLI; PAL, 2017). Através de análise de padrões, os sistemas baseados em técnicas de *deep learning* são capazes de reconhecer, traduzir, sintetizar e até prever sinais das mais diferentes naturezas (DELICATO; PIRES; SILVEIRA, 2017).

### **2.2.1. 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).

A camada convolucional é a camada mais importante de uma CNN e utiliza uma operação matemática linear chamada *convolução* (GOODFELLOW; BENGIO; COURVILLE, 2016). O processo de convolução é aplicado em um conjunto de *filtros* e uma dada entrada para gerar uma saída conhecida como *mapa de características*. Cada filtro consiste em uma matriz de números discretos que representam os pesos da CNN (KHAN et al., 2018).

A camada convolucional recebe um volume de entrada de largura  $w_{in}$ , altura  $h_{in}$  e profundidade  $d_{in}$  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. Cada filtro 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 consecutivas do filtro no volume de entrada. A saída da camada de convolução é um volume de largura  $w_{out}$  calculado conforme a Equação 2, altura  $h_{out}$  conforme Equação 3 e profundidade  $d_{out}$  igual a  $k$  (BUDUMA, 2017).

$$w_{out} = \frac{w_{in} - e + 2p}{s} + 1 \quad (2)$$

$$h_{out} = \frac{h_{in} - e + 2p}{s} + 1 \quad (3)$$

### 2.2.2. Arquiteturas canônicas de Redes Neurais Convolucionais

**LeNet** Yann le Cun desenvolveu, em 1990, um conjunto de redes neurais convolucionais denominado LeNet. Esta arquitetura é composta por duas camadas convolucionais alternadas com camadas de *pooling*. Além destas, as últimas camadas da rede são camadas completamente conectadas baseadas nas tradicionais camadas ocultas da rede MLP (GULLI; PAL, 2017).

**AlexNet** Em 2012 uma outra arquitetura foi proposta por Alex Krizhevsky, conhecida como AlexNet. A arquitetura 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 redes neurais convolucionais 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 quatro primeiras camadas da arquitetura AlexNet são camadas de convolução e *pooling* de forma similar à LeNet, porém, seguem-se mais três camadas convolucionais. As três últimas camadas são completamente conectadas, mas além destas existem camadas *dropout* que ajudam a reduzir *overfitting*, o que resulta em uma melhor generalização (KHAN et al., 2018).

**VGGNet** A arquitetura VGGNet é uma das arquiteturas mais populares desde sua criação em 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. 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 camadas completamente conectadas para classificação. A utilização dos *kernels* serve para reduzir o número de parâmetros e aumentar a eficiência do treinamento. Além das camadas de convolução, *pooling* e das camadas conectadas, esta arquitetura também possui as camadas *dropout* (KHAN et al., 2018).

**GoogLeNet** A arquitetura GoogLeNet

### 2.2.3. *Transfer Learning*

## 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: [s.n.], 2008.
- CARVALHO, C. E. S. de. *Reconhecimento de Caracteres Manuscritos Utilizando Redes Neurais Artificiais*. Brasília, Distrito Federal: [s.n.], 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: \_\_\_\_\_. 1. ed. Porto Alegre, Rio Grande do Sul: SBC, 2017. (1, v. 1), 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.

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. [S.l.]: 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.

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.

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.

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: [s.n.], 1999.