Colorização de imagens com Deep Learning

Giovana de Lucca, Elloá B. Guedes

[1]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

1.2. Justificativa

1.3. Metodologia

1.4. Cronograma

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 $X = [+1, x_1, x_2, \ldots, x_m]$ e um vetor de pesos $W = [w_0, w_1, \ldots, w_m]$. 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).

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

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) . Estas funções encontram-se representadas na Figura 2.

Citar JAI

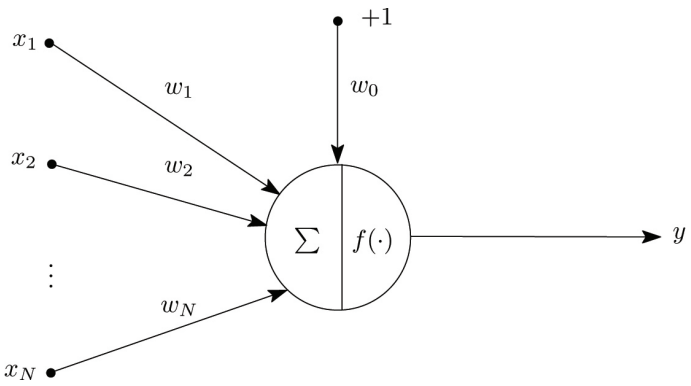
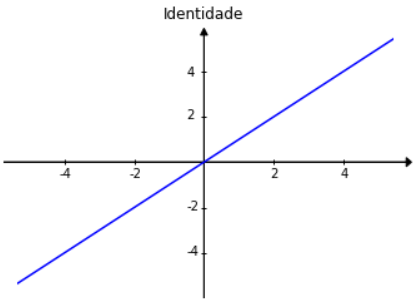
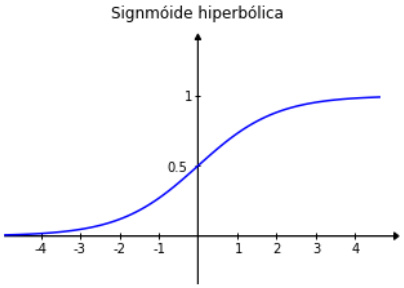


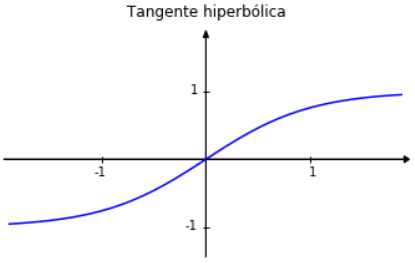
Figura 1: Neurônio artificial: a combinação linear das entradas *x* ponderadas pelos pesos *w* é transformada pela função de ativação *f* na saída *y*.



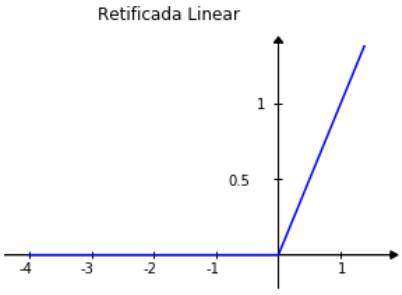
(a) Função identidade.



(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).

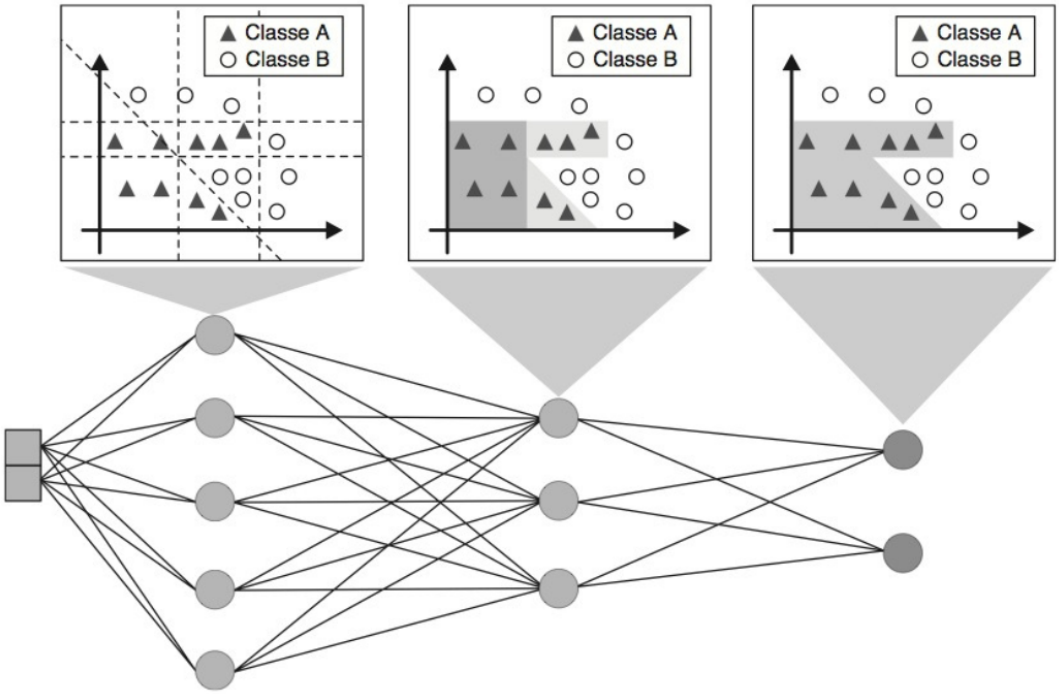


Figura 3: Papel desempenhado pelos neurônios das diferentes camadas de uma rede MLP. Fonte: (FACELI et al., 2011).

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 um modelo de computação com ampla aplicação na resolução de problemas de previsão. As áreas de Biologia [2], Comunicação [3], Jogos [4] e Robótica [5], por exemplo, possuem diversos re

Tentar ser mais geral, mas sem spoilers

As RNAs têm sido utilizadas para aplicações em diversas áreas como Biologia [2], Comunicação [3], Jogos [4] e Robótica [5]. Muitos estudos utilizam as RNAs em problemas de classificação de dados, como em [6], ou para previsão de informações como em [7]. No processamento de imagens, as RNAs atuam principalmente em conjunto com as técnicas de aprendizado profundo que têm sido aplicadas com êxito em diversos problemas como em [8].

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

Teste . resto do parágrafo ajdhdkahakjh

3. Trabalhos Relacionados

4. Solução Proposta

4.1. Visão Geral da solução proposta

4.2. Análise Comparativa

5. Considerações Parciais

Referências

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.

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. v. 1, cap. Deep Learning - Teoria e Prática.

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

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.

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.

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.