

# Restauração de Imagens com Redes Neurais

Lucas A. Rodrigues, Renato Candido e Magno T. M. Silva

**Resumo**—A restauração de imagens busca eliminar as distorções causadas no processo de aquisição. Neste artigo, utiliza-se uma rede neuronal MLP (*Multilayer Perceptron*) para restaurar imagens que contêm quatro tons de cinza e que foram degradadas por uma função gaussiana.

**Palavras-Chave**—Processamento de imagens, restauração de imagens, rede neuronal, aprendizado de máquina.

**Abstract**—Image restoration seeks to eliminate distortions caused in the acquisition process. In this paper, we use an MLP (*Multilayer Perceptron*) to restore images containing four gray levels that have been degraded by a Gaussian point spread function.

**Keywords**—Image processing, image restoration, neural network, machine learning.

## I. INTRODUÇÃO E PROBLEMATIZAÇÃO

Durante o processo de aquisição, as imagens sofrem degradações que prejudicam seu uso para o fim desejado [1], [2], [3]. Apesar dos distintos processos de aquisição, é comum modelar a degradação de uma imagem com uma função de espalhamento de ponto (*Point Spread Function* - PSF), em que

$$\mathbf{G} = \mathbf{F} * \mathbf{H}, \quad (1)$$

sendo  $\mathbf{G}$  a matriz que representa a imagem distorcida observada,  $\mathbf{F}$  a matriz que representa a imagem original,  $\mathbf{H}$  a matriz de coeficientes da PSF e  $*$  o operador de convolução em duas dimensões [1], [2].

Cada elemento  $\mathbf{F}(n_1, n_2)$  representa um pixel da imagem original com valor correspondente à cor do pixel numa escala de níveis de cinza. Uma das distorções mais usadas na literatura é a gaussiana de dimensão  $(2N_1 + 1) \times (2N_2 + 1)$  em que

$$\mathbf{H}(n_1, n_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{n_1^2 + n_2^2}{2\sigma^2}\right), \quad (2)$$

sendo  $\sigma$  o desvio padrão,  $n_1 = -N_1, \dots, -1, 0, 1, \dots, N_1$  e  $n_2 = -N_2, \dots, -1, 0, 1, \dots, N_2$ .

## II. A REDE NEURONAL

Das diferentes maneiras de se restaurar imagens, o uso de redes neurais vem demonstrando resultados interessantes [1], [4]. Neste artigo, optou-se pela implementação e treinamento de uma rede neuronal MLP (*Multilayer Perceptron*) para reduzir o efeito da PSF, definida em (2) com  $N_1 = N_2 = 3$  e  $\sigma = 2$ . As imagens utilizadas possuem quatro níveis de cinza e dimensão  $256 \times 256$ . A rede MLP recebe

Lucas A. Rodrigues, Renato Candido e Magno T. M. Silva, Depto. de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, e-mails: lucas.antunes.rodrigues@usp.br, renatocan@lps.usp.br, magno.silva@usp.br. Este trabalho foi financiado pelo CNPq (Bolsa PIBIC e projeto 304715/2017-4) e FAPESP (2017/20378-9).

como entrada os pixels de uma região da imagem degradada transformada em um vetor unidimensional. A rede faz a estimativa do valor original do pixel central dessa região antes da degradação, decidindo entre os quatro possíveis níveis de cinza, conforme esquematizado na Figura 1. Para compor o conjunto de treinamento, a matrizes  $\mathbf{G}$  e  $\mathbf{F}$  são varridas de forma aleatória para obter os vetores de entrada e os valores desejados de saída, respectivamente.

Para a estrutura da rede MLP, decidiu-se utilizar quatro camadas. A camada de entrada com 49 neurônios considera uma região de  $7 \times 7$  pixels, ficando o pixel a ser estimado no centro dessa região ( $n_1 = n_2 = 4$ ). As duas camadas seguintes são compostas por 40 neurônios cada uma. Por fim, na camada de saída, foram considerados 4 neurônios, sendo cada um responsável pela estimativa da probabilidade de um dos 4 níveis de cinza da imagem original. Aquele que resultar no maior valor indica o nível de cinza estimado para a respectiva entrada da rede. Sendo assim, a estrutura da rede corresponde a 49-40-40-4.

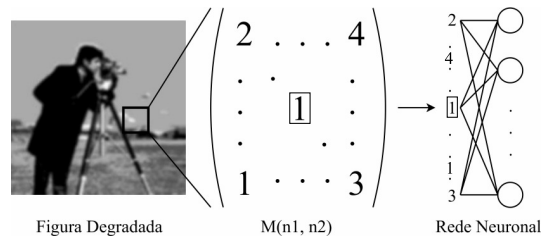


Fig. 1. Esquema ilustrativo da geração da entrada da rede MLP.

Neste trabalho, foram consideradas três tipos de função de ativação: sigmóide, tangente hiperbólica ( $\tanh$ ) e *softmax*, definidas respectivamente como

$$f_s(x_j) = \frac{1}{1 + e^{-x_j}}, f_t(x_j) = \frac{e^{x_j} - e^{-x_j}}{e^{x_j} + e^{-x_j}}, f_m(x_j) = \frac{e^{x_j}}{\sum_{i=1}^N e^{x_i}}.$$

Nessas funções,  $x_j$  representa a entrada da função de ativação do neurônio  $j$ . Em  $f_m(\cdot)$ ,  $x_i$ ,  $i = 1, 2, \dots, j, \dots, N$ , representam as entradas das funções de ativação dos  $N$  neurônios da mesma camada do neurônio  $j$ .

Além disso, para atualização dos pesos da rede foi utilizado o algoritmo *backpropagation* [5]. Utilizando o método do gradiente estocástico, esse algoritmo busca minimizar uma determinada função custo. Neste trabalho, foram consideradas duas funções custo: o erro quadrático [5] e a entropia cruzada [7], dadas respectivamente por

$$\zeta_q(k) = \frac{1}{2} \sum_{j=1}^{N_s} |d_j(k) - y_j(k)|^2 \text{ e } \zeta_t(k) = - \sum_{j=1}^{N_s} d_j(k) \ln(y_j(k)),$$

em que  $N_s$  denota o número de neurônios da camada de saída da rede,  $d_j(k)$  a saída esperada da rede (sinal desejado) e  $y_j(k)$

a saída de cada neurônio com  $j = 1, 2, \dots, N_s$ . Vale ressaltar que a saída da rede foi calculada em todas as iterações  $k$ . No entanto, a atualização dos pesos ocorreu a cada 32 iterações (*batch size* = 32). No treinamento foram consideradas 100 épocas, sendo que em cada uma utilizaram-se 20 imagens degradadas, resultando em  $20 \times 256 \times 256$  pixels. Por fim, foram utilizados no algoritmo *backpropagation* uma taxa de aprendizado reduzida linearmente de  $\eta = 0,1$  a  $\eta = 0,001$  a cada atualização. Para o parâmetro *momentum*, utilizou-se  $\alpha = 0,001$  [5].

### III. RESULTADOS

Inicialmente, a rede MLP foi implementada por completo em MATLAB sem utilizar funções específicas de redes neurais para permitir a compreensão de sua estrutura e de seu algoritmo de treinamento. Em seguida, a rede foi implementada também na *framework Keras* na linguagem *Python*, visando a otimização dos algoritmos de treinamento e testes.

Foram consideradas as configurações da rede MLP especificadas na Tabela I. A Configuração A segue um padrão clássico da literatura [5], enquanto a Configuração B segue resultados recentes da área de *deep neural networks* [7].

TABELA I  
CONFIGURAÇÕES DA REDE MLP USADAS NAS SIMULAÇÕES.

Configuração	A	B
Ativação das camadas ocultas	sigmóide	tanh
Ativação da camada de saída	sigmóide	softmax
Função custo	erro quadrático	entropia cruzada

Depois de treinada com as 20 imagens, a rede foi testada com uma imagem não pertencente ao conjunto de treinamento. Para avaliar a qualidade das imagens restauradas, utilizou-se o índice de similaridade estrutural média (*Mean Structural Similarity* - MSSIM) [6, Eq. (17)].

O MSSIM mede a similaridade entre duas imagens e funciona como uma medida de qualidade de uma imagem em relação a outra, considerando características do sistema visual humano. Essa medida assume valor no intervalo  $[0, 1]$ , sendo igual a um quando as duas imagens são iguais. No cálculo do MSSIM, foi usado o código em MATLAB disponível em <https://ece.uwaterloo.ca/~z70wang/research/ssim/> com  $K = [0.01 \quad 0.03]$  e  $\text{window} = \text{fspecial}('gaussian', 11, 1.5)$ , cuja versão para *Python* está disponível em <http://helderc.github.io/>. A escolha dessa medida se deve ao fato de que métricas baseadas no erro quadrático médio não levam em conta o sistema visual humano e podem apresentar valores baixos para imagens com qualidade ruim segundo a percepção visual [6], [8].

A imagem original, a imagem degradada e as imagens restauradas com a rede MLP seguindo as Configurações A e B estão mostradas na Figura 2. Pode-se verificar que a imagem restaurada pela MLP com a Configuração B é superior que a restaurada pela MLP com a Configuração A. Os valores de MSSIM corroboram com essa observação, já que o MSSIM entre a imagem original e a degradada é de 0,77, enquanto esse valor entre a imagem original e a restaurada com a Configuração A é 0,83 e com a Configuração B é 0,89.

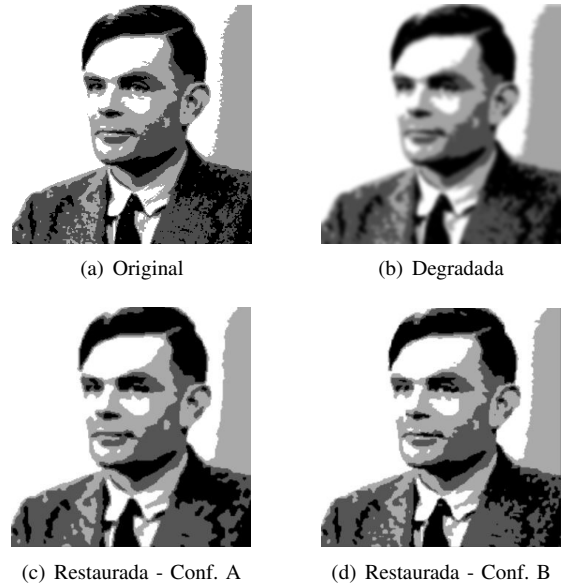


Fig. 2. (a) Imagem original; (b) Imagem degradada com PSF gaussiana ( $N_1 = N_2 = 3, \sigma = 2$ ) - MSSIM= 0,77; (c) Imagem restaurada pela MLP com a Configuração A - MSSIM= 0,83; e (d) Imagem restaurada pela MLP com a Configuração B - MSSIM= 0,89.

### IV. CONCLUSÕES

Neste artigo, uma rede MLP foi utilizada para restaurar uma imagem degradada por uma PSF gaussiana. Foram consideradas duas configurações distintas para a rede: uma clássica baseada na minimização do erro quadrático e função de ativação sigmóide e outra baseada em avanços recentes da área de *deep learning*. Esses avanços incluem a minimização da entropia cruzada e a utilização da função de ativação tangente hiperbólica para os neurônios das camadas ocultas e *softmax* para os neurônios de saída. Por meio de simulações, obteve-se uma imagem de melhor qualidade considerando a configuração que leva em conta os avanços recentes. Na continuidade do trabalho, pretende-se usar redes neurais convolucionais para buscar resultados ainda melhores.

### REFERÊNCIAS

- [1] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks - a review", *Pattern recognition*, vol. 35, no. 10, pp. 2279-2301, 2002.
- [2] J. S. Lim, *Two-dimensional signal and image processing*, Prentice Hall, Upper Saddle River, NJ, USA, 1990.
- [3] R. C. Gonzales, R. E. Woods, *Processamento digital de imagens*, Prentice Hall, 3.a edição, São Paulo, 2010.
- [4] A. Lucas, M. Iliadis, R. Molina, A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods" *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20-36, Jan 2018.
- [5] S. S. Haykin, *Neural networks and learning machines*, vol. 3, Pearson Upper Saddle River, NJ, USA, 2009.
- [6] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, 2004.
- [7] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- [8] Z. Wang, A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures" *IEEE Signal Processing Magazine*, vol. 26, pp.98-117, Jan. 2009.