

# Avaliação do Paralelismo em Python para Otimizar uma Abordagem de Identificação de Máscaras Faciais utilizando Redes Neurais Artificiais

Natan Steinbruch<sup>1</sup>, Vinícius L. Santos<sup>1</sup>, Nicholas C. Villela<sup>1</sup>,  
Gabriel R. O. Camargo<sup>1</sup>, Andre R. Xavier<sup>1</sup>, Diego Brandão<sup>1</sup>

<sup>1</sup>Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)  
Rio de Janeiro, RJ – Brasil

diego.brandao@cefet-rj.br

**Resumo.** A pandemia de Covid-19 tem exigido grandes esforços de toda a comunidade científica. No entanto, o controle da pandemia só tem sido possível com o isolamento social e o uso das máscaras faciais. Apesar disso, diariamente são exibidos casos de aglomerações e de pessoas que se recusam a utilizar as máscaras, dificultando o controle da doença. Este artigo apresenta uma abordagem por meio de Redes Neurais Artificiais (RNA) para identificação de pessoas que não estão utilizando máscaras faciais em um banco de dados de imagens. Os resultados obtidos demonstram que apesar de o paralelismo empregado na linguagem Python com a biblioteca Numpy ser pontual, ele impacta positivamente no tempo de execução da abordagem desenvolvida.

**Abstract.** The Covid-19 pandemic has required great efforts from the entire scientific community. However, pandemic control has only been possible with social isolation and the use of facial masks. Despite this, cases of agglomerations and people who refuse to use the masks are shown daily, making it difficult to control the disease. This article presents an approach using Artificial Neural Networks (ANN) to identify people who do not use face masks in an image database. The results obtained demonstrate that although the parallelism used in the Python language with the Numpy library is punctual, it positively impacts the developed approach's execution time.

## 1. Introdução

O avanço da Covid-19 pelo mundo tem exigido esforços cada vez maiores dos cientistas no desenvolvimento de uma vacina. Até o momento as únicas medidas que se mostraram eficazes no controle da doença são baseadas no isolamento social, melhora nos hábitos de higiene e utilização das máscaras faciais. No entanto, inúmeras pessoas tem burlado os controles sanitários não utilizando as máscaras faciais, realizando aglomerações e etc. Verificar se essas medidas tem sido respeitadas por toda a população é uma tarefa complicada e exigiria um esforço muito grande das autoridades.

Neste contexto, métodos de visão computacional podem ser utilizados para auxiliar na identificação automática de aglomerações ou mesmo identificar se um determinado indivíduo está ou não utilizando a máscara dentro do metrô ou de um ônibus. O uso de tais métodos no contexto urbano não é novidade, inúmeras cidades já utilizam sistemas de monitoramento por câmeras [Ullah et al. 2019].

Dentre as técnicas de visão computacional, as Redes Neurais Artificiais (RNA) são uma das técnicas mais utilizadas, pois nos últimos anos elas têm obtido resultados consideráveis para problemas altamente complexos [Szegedy et al. 2016]. Assim, este trabalho propõe avaliar os impactos do paralelismo presente na biblioteca NumPy para implementação de uma RNA utilizada para identificar se uma determinada pessoa está utilizando a máscara facial.

Este trabalho está estruturado conforme descrito a seguir. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta uma pequena introdução sobre RNA *MultiLayer Perceptron* (MLP) e a metodologia desenvolvida. Os resultados obtidos são apresentados na Seção 4. Por fim, a Seção 5 apresenta as considerações finais.

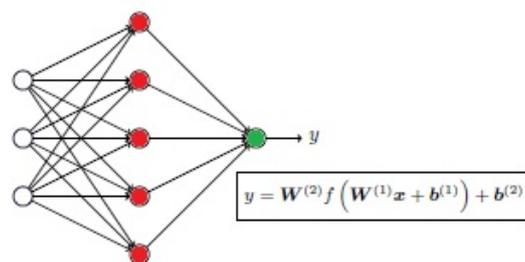
## 2. Trabalhos Relacionados

Com o avanço dos casos de Covid-19 pelo mundo, as contribuições científicas têm sido as mais abrangentes possíveis. Neste contexto, diversas pesquisas demonstraram a efetividade do uso de máscaras faciais para a redução das taxas de transmissão do coronavírus ou até mesmo o grau de severidade da doença [Chughtai et al. 2020]. Contudo, as autoridades sanitárias têm tido cada vez mais dificuldade em fiscalizar o uso das máscaras por parte da população. Visando realizar tal fiscalização de forma automática, diversos pesquisadores têm proposto abordagens utilizando técnicas de visão computacional que identificam se um indivíduo está utilizando a máscara ou não [Ejaz et al. 2019, QIN and LI 2020, Loey et al. 2020]. A técnica de análise de componentes principais (PCA) foi utilizada por [Ejaz et al. 2019]. Já as Redes Neurais Artificiais foram utilizadas por [QIN and LI 2020] e [Loey et al. 2020]. Enquanto em [QIN and LI 2020] utilizaram redes do tipo *Deep Super Resolution Crack Network* (SRCNet), já as redes do tipo *Residual Neural Network* (ResNet50) foram avaliadas em [Loey et al. 2020]. Aqui são utilizadas as redes do tipo MLP, sendo avaliado o desempenho obtido com o uso do paralelismo da linguagem Python por meio da biblioteca NumPy.

## 3. Metodologia

Uma Rede Neural Artificial (RNA) *MultiLayer Perceptron* (MLP) é uma técnica de aprendizado de máquina bioinspirada utilizada em diversas áreas da ciência [Haddad et al. 2018]. Ela tem a capacidade de modelar relações não-lineares complexas entre dados que ela seja apresentada. Sua arquitetura pode incluir uma ou muitas camadas escondidas, uma camada de entrada e uma camada de saída.

A Figura 1 ilustra uma RNA de 3 camadas genérica. Cada uma dessas camadas pode ser formada por inúmeras unidades menores, chamadas de neurônios. Estas unidades estão conectados entre si por uma sinapse, que tem um certo grau de importância ( $W$ ). O estado de cada neurônio é avaliado matematicamente. Caso o valor obtido por essa avaliação seja maior do que um certo limiar, esse valor é utilizado como saída deste neurônio. Este processo se repete por todas as camadas da rede até que seja obtido um resultado de saída ( $y$ ). Caso este resultado não esteja dentro de uma precisão desejada, os valores  $W$  (também chamados de pesos) são ajustados, de acordo com um algoritmo de minimização. Neste trabalho optou-se pelo algoritmo de minimização dos Gradientes Estocásticos, devido sua simplicidade de implementação, além de convergência mais rápida quando comparado ao Método dos Gradientes Descendentes.



**Figura 1. Arquitetura de uma RNA MLP. Adaptado de [Haddad et al. 2018].**

Foram implementadas duas versões da RNA MLP descrita. A primeira versão desenvolvida realizava todas as operações matemáticas sem empregar qualquer otimização ou paralelismo. Na segunda versão foi utilizada a biblioteca NumPy, esta biblioteca permite que o Python realize uma vetorização do código, permitindo a utilização de *threads* para realizar a divisão de carga do trabalho. Por meio da NumPy, o Python faz uso de uma biblioteca especializada em cálculos matriciais, denominada *LAPACK* [Oliphant 2006].

#### 4. Experimentos Computacionais

Neste projeto foi utilizado a linguagem Python versão 3.7 no Anaconda Spyder em uma máquina Core i7 8a. geração (3.9GHz) com 8 núcleos, 16GB de RAM, placa de vídeo NVIDIA GEFORCE 1050TI com 4Gb. O código desenvolvido encontra-se disponível em: <https://github.com/NatanStein/IA-do-Ramo>.

A base de dados utilizada é composta por dados disponíveis no site Kaggle (<https://www.kaggle.com/dhruvmak/face-mask-detection>) e no Github (<https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset>).

##### 4.1. Resultados de Qualidade da Classificação

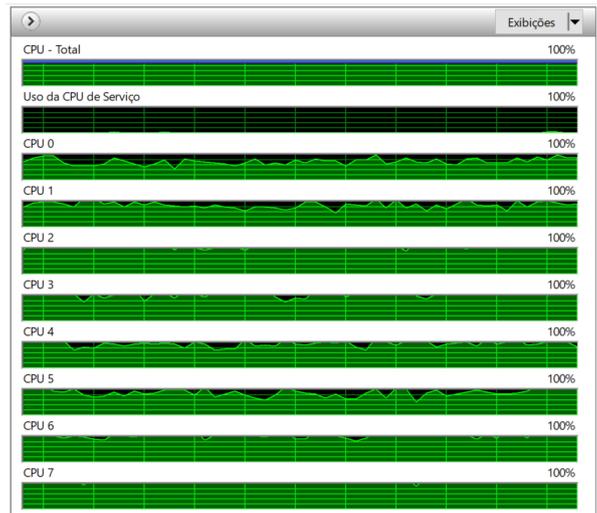
A rede neural utilizada neste trabalho é formada por uma camada de entrada com 784 neurônios. Três camadas intermediárias, sendo a primeira com 53 neurônios, a segunda com 36 neurônios e a terceira com 25 neurônios. Por fim, a camada de saída consta com 1 único neurônio, constituindo-se como um classificador binário que diz se um indivíduo está ou não de máscara em uma determinada imagem.

O desempenho da RNA em relação a acurácia também foi avaliado, sendo obtido um valor de 86% para essa métrica quando usado a regularização *L1* em cada camada.

##### 4.2. Resultados de Desempenho

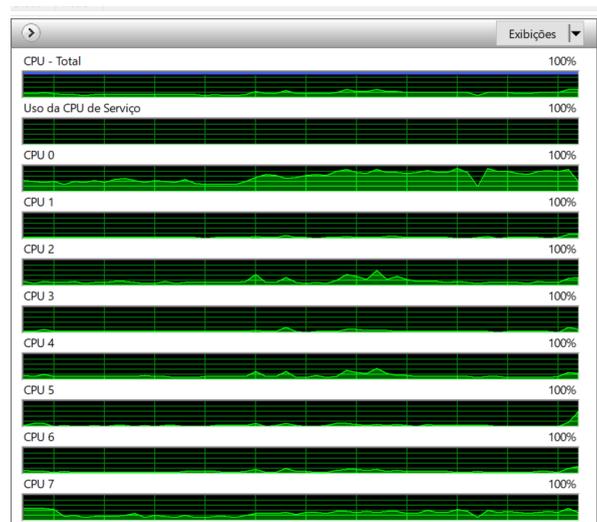
Em relação ao desempenho durante a fase de treinamento avaliamos o uso dos múltiplos *cores* do computador. A Figura 2 ilustra que todos os *cores* são utilizados quase que no máximo de processamento quando a biblioteca Numpy é empregada. Já a Figura 3 mostra que sem esta biblioteca o processamento fica concentrado em um único processador levando mais tempo para execução de todo o processo de treinamento. O processo de escalonamento de tarefas pelo sistema operacional não foi avaliado neste trabalho.

A Figura 4 apresenta uma comparação dos tempos de execução da versão sequencial com a versão paralela, o ganho de desempenho obtido com o uso da biblioteca NumPy é perceptível, chegando no caso do treinamento com 10.000 imagens e 1000 épocas a um



**Figura 2. Uso dos múltiplos cores durante o processo de treinamento com Numpy**

Speedup de 3,92. O mais interessante deste resultado é que o paralelismo empregado foi pontual, especificamente na função *dot* responsável pelo produto matricial.



**Figura 3. Uso dos múltiplos cores durante o processo de treinamento sem Numpy**

## 5. Considerações Finais

Este trabalho apresenta a implementação em Python de uma Rede Neural Artificial do tipo MLP para a identificação de pessoas utilizando máscaras faciais. A implementação apresentada é otimizada com a utilização da biblioteca NumPy que fornece um paralelismo transparente ao usuário. Os resultados obtidos demonstram que o uso de tal biblioteca fornece uma redução do tempo de execução do programa. A próxima etapa desta pesquisa consiste no desenvolvimento de uma versão que realize a identificação do uso de máscaras faciais em tempo real e que possa ser integrado com câmeras de segurança, além de uma avaliação experimental mais profunda com testes estatísticos e etc. Espera-se que essa

	Numpy	Sem Numpy
<b>Rede Congelada:</b>		
5000 Imagens	5,8579 Segundos	7,1959 Segundos
10000 Imagens	6,9312 Segundos	12,1276 Segundos
<b>Processamento do DataSet:</b>		
5000 Imagens	45,0023 Segundos	48,9001 Segundos
10000 Imagens	60,2175 Segundos	70,9981 Segundos
<b>Rede Neural 1000 Epocas:</b>		
10000 Imagens	41,3901 Segundos	162,4209 Segundos

**Figura 4. Comparação de Duração de Processamento da Rede Neural.**

abordagem possa ser utilizada no auxílio as autoridades sanitárias em identificar de forma automática indivíduos que contrariam as normas de saúde e não utilizam as máscaras, colocando em risco toda a população.

### Agradecimentos

Os autores agradecem ao Ramo Estudantil IEEE CEFET/RJ e ao Conselho Nacional de Pesquisa (CNPq) pelo apoio ao desenvolvimento desta pesquisa.

### Referências

- Chughtai, A. A., Seale, H., and Macintyre, C. R. (2020). Effectiveness of cloth masks for protection against severe acute respiratory syndrome coronavirus 2. *Emerging infectious diseases*, 26(10).
- Ejaz, M. S., Islam, M. R., Sifatullah, M., and Sarker, A. (2019). Implementation of principal component analysis on masked and non-masked face recognition. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–5. IEEE.
- Haddad, D. B., de Assis, L. S., Tarrataca, L., Gomes, A. d. S., Ceddia, M. B., Oliveira, R. F., Junior, J. R. d. P., and Brandão, D. N. (2018). Brazilian soil bulk density prediction based on a committee of neural regressors. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Loey, M., Manogaran, G., Taha, M. H. N., and Khalifa, N. E. M. (2020). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167:108288.
- Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.
- QIN, B. and LI, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Ullah, F. U. M., Ullah, A., Muhammad, K., Haq, I. U., and Baik, S. W. (2019). Violence detection using spatiotemporal features with 3d convolutional neural network. *Sensors*, 19(11):2472.