

IA Diagnóstico de Pneumonia a partir de imagens de raio-x do paciente

João Pedro Moro Bolognini
Departamento de Computação - DComp
Universidade Federal de São Carlos - Sorocaba
joaomoro@estudante.ufscar.br

Abstract

Pneumonia é uma infecção pulmonar grave que, se não tratada corretamente, pode ser fatal. O diagnóstico rápido e preciso é essencial para iniciar o tratamento adequado e evitar complicações. Este projeto busca utilizar inteligência artificial para detectar pneumonia a partir de imagens de raio-x, utilizando redes neurais convolucionais para acelerar e melhorar a precisão do diagnóstico. Para a construção do modelo, foi utilizado um conjunto de imagens de raio-x de pacientes saudáveis ou com pneumonia causada por bactérias ou vírus.

O modelo desenvolvido emprega uma arquitetura sequencial com camadas convolucionais, técnicas de normalização e data augmentation para melhorar sua robustez e evitar overfitting. Durante os testes, o modelo alcançou uma precisão de 69,58%, com acurácia de treino e validação acima de 80% a partir da oitava época. No entanto, dificuldades em classificar imagens da classe normal com alta confiança mostram a necessidade de melhoria, como redes convolucionais mais complexas ou técnicas de transfer learning. O objetivo a partir deste relatório é melhorar o modelo, aumentando a eficácia dele.

1. Introdução

A pneumonia é um tipo de infecção pulmonar devido à proliferação de microorganismos infecciosos ao nível dos alvéolos (pequenos sacos de ar). É uma doença que pode ser agressiva e fatal caso não seja tratada corretamente e o quanto antes. Por exemplo, em 2020, segundo o Centro de Controle e Prevenção de Doenças (Center for Disease Control and Prevention - CDC) [5], houve cerca de 47.000 mortes por pneumonia nos Estados Unidos. A doença é a causa de morte mais comum entre as infecções hospitalares e em países que tem uma medicina precária. Também é uma das infecções sérias mais comuns em bebês e crianças. Existem várias causas para a doença, entre elas, as causadas por bactérias e as causadas por vírus [1].

O tratamento da doença pode ser feito por antibióticos,

medicamentos antivirais, antifúngicos ou antiparasitários, e com tratamentos de apoio à respiração. Em alguns casos, é necessário um diagnóstico imediato da doença para iniciar o tratamento indicado. O problema é que em alguns casos este diagnóstico pode ser incorreto ou demorado, seja por problemas humanos, como uma visualização incorreta do diagnóstico ou uma lotação em hospitais, ou por problemas técnicos em aparelhos de diagnóstico, como um mal funcionamento em uma máquina de impressão de raio-x. Por exemplo, em 2017, uma menina de 3 anos morreu de pneumonia após ser diagnosticada incorretamente com virose, tendo visitado o hospital 3 vezes [2]. Diante disso, é necessário melhorar o diagnóstico da doença para evitar mais mortes causadas por ela.

Baseado nisso, podemos utilizar uma Inteligência Artificial a partir de uma captura de raio-x de um paciente para essa detecção, podendo ser muito útil visto que seria uma detecção rápida e praticamente imediata. Assim, a abordagem escolhida para este projeto foi utilizar redes convolucionais para fazer a detecção de pacientes com pneumonia a partir do raio-x, tendo dois tipos de pneumonia no conjunto de dados: bacteriana e viral.

2. Definição do problema

O problema a ser resolvido é detectar pneumonia em pacientes a partir de imagens de raio-x utilizando inteligência artificial para acelerar esse processo, possibilitando um diagnóstico imediato, de modo que o uso dela seja confiável e assertivo.

Para isso, será utilizado o conjunto de dados *Chest X-Ray Images (Pneumonia)*, encontrado no Kaggle [4]. Este dataset possui 3 pastas, que são o conjunto de validação, de treinamento e de teste, e dentro de cada pasta possui imagens de pacientes normais, isto é, saudáveis, e de pacientes com pneumonia, podendo ser causada por bactéria ou por vírus. A partir da implementação de uma rede convolucional e do treinamento e validação, espera-se que o modelo seja capaz, a partir do conjunto de teste, de gerar imagens classificadas como normal ou como pneumonia, informando sua taxa de confiança para a predição, além

de métricas como acurácia, revocação, precisão e f1-score. Também será mostrado um gráfico da curva ROC para ver a eficiência e a perda durante o treinamento e validação do modelo, além da matriz de confusão para explicitar em qual classe o modelo é mais falho.

Com base nas métricas e na visualização das imagens, espera-se um bom funcionamento do modelo. Quanto maior a eficácia do modelo, melhor seu desempenho. Mas também será observada as imagens de teste com sua classificação predita e sua taxa de confiança, para ver o quão discrepantes estão as classificações incorretas. Assim, caso uma imagem de classe normal seja classificada como pneumonia com uma taxa de confiança alta, indicará que o modelo está com um problema na classificação, e deve ser corrigido (Figure 1).



Figure 1. Exemplo de classificação incorreta - Pode-se ver que a classificação incorreta possui uma confiança alta, o que deve ser evitado

3. Abordagem técnica

Para resolver o problema de classificar da melhor maneira possíveis as imagens de raio-x dos pacientes com base nas classes normal e pneumonia, é necessário que o modelo seja capaz de aprender as características relevantes, para generalizar novos exemplos e evitar problemas como overfitting. A abordagem utilizada no projeto utilizada uma rede convolucional, o modelo *Sequential()* do *tensorflow*.

Inicialmente é feito o carregamento e pré-processamento das imagens. Elas são separadas em pastas, importadas para o *Google Drive*, contendo os conjuntos de treinamento, validação e teste são organizadas e lidas utilizando a classe *ImageDataGenerator*, da biblioteca *tensorflow*. Essa classe tem a função de carregar as imagens e realizar transformações que sejam capaz de melhorar a performance e a robustez do modelo. Durante o treinamento, as imagens são normalizadas e há um aumento de dados (*data augmentation*), isto é, há um aumento artificial no tamanho e na diversidade das imagens de treinamento para o modelo usar

durante o treinamento. Há uma divisão de pixel por 255, ajustando todos os valores em uma faixa dentro do intervalo $[0,1]$, assim, garantindo que os cálculos da rede neural sejam rápidos, estáveis e eficientes.

Além disso, o *data augmentation* adiciona variedade ao conjunto de treinamento através da aplicação de transformações aleatórias como rotação, translação, zoom e espelhamento horizontal. Essas alterações permitem que o modelo seja exposto a uma gama maior de variações nas imagens, simulando cenários reais que podem ocorrer fora do ambiente controlado dos dados de treinamento e reduzindo as chances de que o modelo memorize padrões específicos demais, o que causa *overfitting*. Vale lembrar que no conjunto de validação e teste apenas a normalização é aplicada, garantindo que essas imagens permaneçam fiéis às condições reais em que o modelo será utilizado. O parâmetro *shuffle* também é aplicado de forma estratégica: as imagens de treinamento são embaralhadas para evitar que sequências de dados afetem o aprendizado, enquanto os conjuntos de validação e teste permanecem inalterados para uma avaliação precisa.

A rede neural foi projetada com foco na simplicidade e eficiência. Ela começa com uma camada de entrada que espera imagens no tamanho 150×150 com três canais de cor (RGB) e passa por três camadas convolucionais. Cada uma dessas camadas aumenta gradativamente o número de filtros (32, 64 e 128) e utiliza a função de ativação ReLU, função não linear que limita a entrada a zero, retornando 0 para valores negativos e a própria entrada para valores positivos, para introduzir não linearidade no modelo, essencial para identificar padrões complexos nas imagens. Após cada camada convolucional, é aplicada uma operação chamada *MaxPooling*, que reduz a quantidade de dados e ajuda o modelo a focar nas características mais importantes da imagem.

Além disso, para evitar o overfitting, foram adicionadas camadas de dropout que desligam aleatoriamente 25% ou 50% da rede durante o treinamento. Isso força o modelo a se tornar mais robusto e menos dependente de padrões específicos. Após a extração de recursos com as camadas convolucionais e as operações de *MaxPooling*, adicionamos uma camada de Flatten, responsável por transformar as informações espaciais da imagem em uma estrutura unidimensional para que possam ser processadas. Em seguida, temos duas camadas densas: uma com 128 neurônios e ativação ReLU, responsável por aprender padrões mais abstratos, e outra com apenas um neurônio e ativação sigmoid, que produz a classificação final entre as classes normal e pneumonia.

O modelo foi compilado utilizando a função de perda *binary_crossentropy*, uma abordagem adequada para problemas de classificação binária. O otimizador escolhido foi o Adam, que ajusta dinamicamente a taxa de apren-

dizado para garantir que o treinamento seja rápido e eficiente. Além disso, as métricas utilizadas incluem a precisão, a acurácia, e a *AUC*, gráfico para avaliar o desempenho do modelo, que ajuda a entender como o modelo se comporta mesmo quando os dados estão desbalanceados.

Por fim, durante o treinamento, utilizamos o callback *ReduceLROnPlateau*. Ele monitora a perda de validação e ajusta automaticamente a taxa de aprendizado caso o desempenho da rede pare de melhorar por três épocas consecutivas. O treinamento é realizado através do método *fit*, definindo os passos por época e o número de épocas (neste caso, 10), com validação ocorrendo ao fim de cada época para garantir que o modelo está evoluindo conforme o esperado. Após o treinamento, o modelo será testado em imagens completamente novas, que nunca foram usadas no treinamento, para avaliar seu desempenho real e sua capacidade de generalizar para novos dados.

4. Resultados intermediários / preliminares

Nesse momento, com tudo o que foi descrito até aqui, o modelo está com um desempenho mediano. No conjunto de teste, até o momento deste relatório, ele está com uma precisão de 69,58%. Ao considerar que por motivos de limites computacionais impostos pelo *Google Colab* [3], que é a plataforma onde está sendo desenvolvido esse projeto, e por usar uma rede convolucional um pouco simples, os resultados estão indo bem. Vale considerar também que o foco desse projeto é melhorar ao máximo a eficácia. Como a eficácia varia muito entre as 10 épocas de treinamento, essa eficácia mediana está relacionada às épocas em que, de alguma forma, há um problema com a quantidade de imagens, e elas são afetadas, tendo uma eficácia normalmente de 50%; por outro lado, há algumas épocas que batem a casa dos 90%, demonstrando que o modelo consegue performar muito bem.

Apesar desse problema, a acurácia de treino e validação, analisando o gráfico da *curva AUC*, a partir da época 8, possui acurácia acima de 80%, como podemos ver no gráfico abaixo.

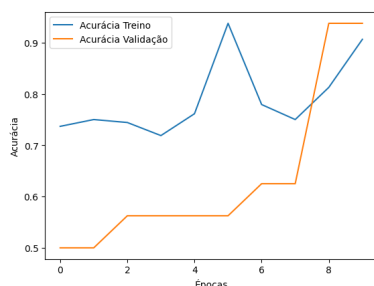


Figure 2. Gráfico da curva AOC

Também é visto que o modelo está cometendo o erro

de classificar erroneamente imagens com alto grau de confiança, principalmente quando a classe verdadeira é saudável. Com base nisso, será necessário fazer adaptações com o foco de melhorar o desempenho nesses casos, visto que é o que está derrubando a eficácia do modelo. O objetivo do projeto é corrigir isso a partir de uma construção de uma rede neural mais complexa ou por meio de *transfer learning*, mas o que for decidido será descrito por completo em relatórios futuros. Ao analisar a matriz de confusão, fica evidente que o modelo está com dificuldades em classificar imagens da classe normal, visto que classifica com precisão imagens da classe pneumonia, como pode ser visto na imagem a seguir.

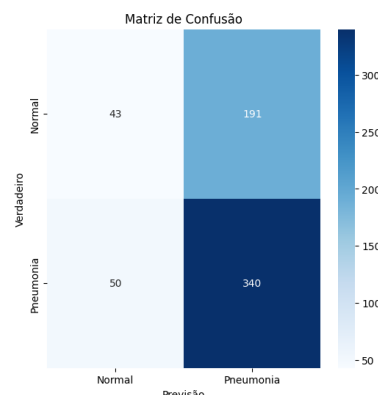


Figure 3. Matriz de confusão do modelo

4.1. Considerações finais

Com base em tudo que foi descrito até aqui, fica evidente a necessidade de melhorar o modelo a fim de aumentar a precisão na classificação de imagens da classe normal, e esse é o objetivo da próxima versão do projeto. Essa melhora é inicialmente projetada melhorando o treinamento do modelo e aumentando as camadas da rede neural, mas não é descartado utilizar de *transfer learning* para melhorar o desempenho.

References

- [1] Dr. António Costa. Pneumonia, 2022. Accessed: 2024-12-04. [1](#)
- [2] Por G1 Vale do Paraíba e Região. Menina de três anos morre de pneumonia após diagnóstico de virose, diz família, 2017. Accessed: 2024-12-04. [1](#)
- [3] Google. Google colab. Accessed: 2024-11-01. [3](#)
- [4] Paul Mooney. Chest x-ray images (pneumonia), 2017. Accessed: 2024-11-01. [1](#)
- [5] Sanjay Sethi. Considerações gerais sobre pneumonia, 2024. Accessed: 2024-12-04. [1](#)