Relatório EP2 MAC0425

Lucas Eiji Uchiyama 11807470

June 2025

1 Resumo

O seguinte exercício implementa um programa que realiza a classificação de pacientes de COVID-19 de acordo com a fase em que está a doença por meio de uma rede neural em Python utilizando a biblioteca PyTorch, utilizando para isso dados de exames que contenham um teste de imunoglobulina G (IgG).

2 Introdução

Quando ocorre a infecção por COVID-19, o paciente passa por três estágios: na primeiro, já se iniciou a infecção, mas o corpo do paciente ainda não iniciou o processo de produção de anticorpos para reagir à doença. No segundo estágio, são produzidos anticorpos IgM, os primeiros a serem liberados após o contato com um antígeno, e que geralmente ficam permanecem por pouco tempo, na fase aguda da doença. Por fim, no terceiro estágio, quando a infecção já está controlada, são produzidos anticorpos IgG, estes conferindo proteção duradoura contra novas infecções do mesmo antígeno.

Portanto, exames de detectam anticorpos IgM e IgG ajudam a identificar em que estágio da infecção está um paciente. Com a pandemia de COVID-19 entre 2020 e 2022, muitos dados de pacientes puderam ser coletados, o que inclui os dados de exames IgG, o que nos permite criar modelos de inteligência artificial que detectam a presença desses anticorpos sem a necessidade de testes.

O objetivo deste trabalho é realizar essa detecção utilizando uma rede neural que será treinada utilizando dados já existentes de pacientes do Hospital das Clínicas da USP. Após o treinamento, a rede deverá indicar se o paciente possui anticorpos IgG provenientes de uma infecção anterior com COVID-19 com alta probabilidade de acerto.

As próximas seções do relatório apresentam a metodologia de realização do projeto, incluindo tratamento de dados e implementação da rede neural em Python, e os resultados do teste da rede neural com dados de validação. Por fim, concluímos com uma discussão sobre a viabilidade e utilidade de uma arquitetura neural que prevê a fase da infecção.

3 Metodologia

3.1 Tratamento de dados

O primeiro passo a ser realizado foi o tratamento dos dados, de modo a criar um novo documento CSV que contenha todos os dados relevantes e que será utilizado como entrada para a rede neural. O documento esperado é um que, para cada consulta, contenha o sexo e idade do paciente e os testes e seus resultados. Como nem todos os testes são realizados em todas as consultas, para testes que não são realizados no atendimento espera-se valor 0, caso seja um teste numérico, e e valor 0.5, caso seja um teste binário.

A primeira etapa é capturar todos os tipos de testes que foram realizados e, para cada um deles, definir se é um teste que retorna um valor numérico ou binário. Isso não foi uma tarefa simples por razões como a grande quantidade de entradas no arquivo de exames, que não poderia ser mantido inteiro na memória, a não uniformidade dos dados binários, que não possuíam uma forma padrão de se identificar 1/True ou 0/False e a existência de alguns poucos exames que retornam resultados que não são nem numéricos nem binários.

Após isso, é lido o arquivo de exames para se criar um dicionário com todos os atendimentos, possuindo o id do atendimento e os exames realizados, sendo depois removidos os atendimentos que não realizaram o exame 'COVID-19 - PESQUISA DE ANTICORPOS IgG', que é o exame de mede a concentração de anticorpos IgG de COVID-19 no sangue. Em seguida, são acrescentados ao dicionário para cada atendimento os exames que não foram realizados e seu valor é definido como 0 para testes numéricos e 0.5 para testes binários. Depois disso, é acrescentada a coluna de id do paciente, para que seja possível realizar um merge com o arquivo de pacientes, do qual pegaremos o sexo do paciente e a sua idade, sendo o sexo do paciente convertido para um valor binário (0 para M e 1 para F), e a idade deixada como a média das demais caso ela não conste. Por fim, os dados são escritos no arquivo, sendo esses os dados que serão utilizados no treinamento da rede neural.

Como o resultado do exame 'COVID-19 - PESQUISA DE ANTICORPOS IgG' é numérico, mas gostaríamos de prever a presença ou não desses anticorpos, gostaríamos de alterar seus valores para 0 ou 1, indicando ausência ou presença, considerando um limiar que separa ambos os resultados. Considerando o teste mais comum, Enzyme-Linked Immunosorbent Assay (ELISA), que possui um limiar entre 0.9 e 1.1, podemos considerar um IgG significativo como acima de 1, e que será usando como limiar na conversão de numérico para binário.

3.2 Construção da rede neural

Antes do treinamento, os dados do arquivo, já tratados, são carregados no arquivo de treinamento e são separados em X, uma matriz com todos os dados utilizados para predizer o valor da pesquisa de anticorpos IgG, e Y, o vetor com os resultados do teste. Depois, os dados de X são normalizados, ou seja, subtraídos pela média e divididos pelo desvio padrão, para que todos possam

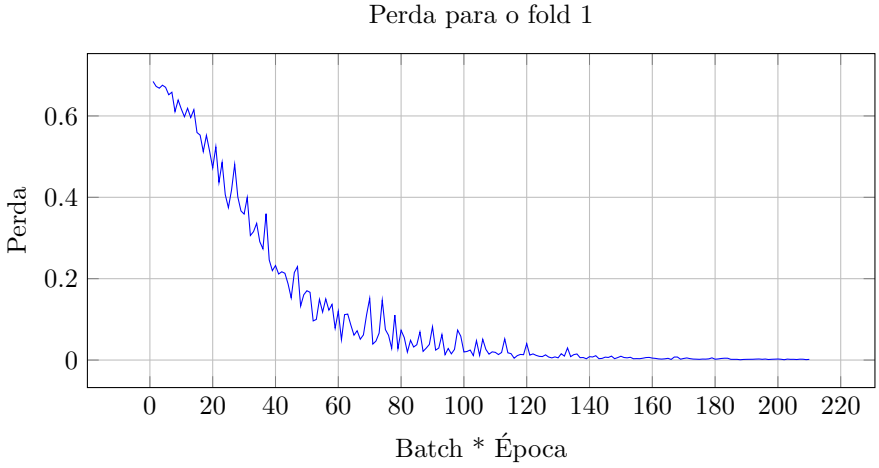
ser manipulados igualmente. Como forma de melhorar a acurácia do modelo, os testes são reduzidos para os 50 mais relevantes, removendo assim variáveis que atrapalhem o aprendizado. O scaler e selector são salvos para poderem ser usados durante a validação.

A função de perda utilizada foi a Binary Cross-Entropy com entrada normalizada, ou seja, com média 0 e desvio padrão 1, podendo assim tratar entradas binárias e numéricas da mesma forma e retornar ao final uma saída entre 0 e 1 que representa a probabilidade de o paciente ter uma concentração de IgG significativa para considerar a situação controlada. O estágio da doença no paciente será dada com base nesse resultado: Se a saída for maior que 0.5, podemos considerar que o paciente está no estágio final da doença, em que os níveis de IgG aumentam, conferindo imunidade ao paciente. Se a saída for menor que 0.5, o paciente estará no estágio intermediário, em que as concentrações de IgM aumentam mas as de IgG ainda estão em seu nível normal.

A rede neural conta com cinco camadas: uma camada que recebe os dados de entrada, três camadas ocultas com a função ReLU como função de ativação e uma camada de saída função sigmoide como função de ativação, uma vez que esperamos uma saída binária. Em relação ao número de nós das camadas ocultas, elas possuem 128, 64 e 128 nós respectivamente. Foram testadas redes com mais nós, mas elas não resultavam em aumento de acurácia. O algoritmo utilizado na atualização é o batch gradient descent com tamanho do batch 64, um valor comum em treinamentos, utilizando o otimizador Adaptative Moment Estimation (Adam), e o treinamento é feito com um processo de k-fold, com k=5, utilizando assim para cada etapa 80% dos dados para treino e 20% para validação. Analisando os dados para a quantidade de épocas, estabeleci 30 épocas como suficiente, uma vez que após esse ponto não havia quase nenhuma mudança na perda.

4 Resultados

Durante o treino da rede neural, foi monitorada a perda do modelo, que representa o aprendizado da rede neural com o passar do tempo. Obteve-se o seguinte gráfico de perda para o fold 1:



Como é possível notar, o aprendizado ocorre rapidamente, se tornando praticamente 0 ao final do treinamento, na 30ª época.

Após o treino da rede neural, também verificou-se os resultados através da matriz de confusão de cada um dos 5 folds, obtendo as seguintes matrizes:

Fold 1:

-	P	N
T	44	2
F	10	55

Fold 2:

-	P	N
T	38	6
F	7	60

Fold 3:

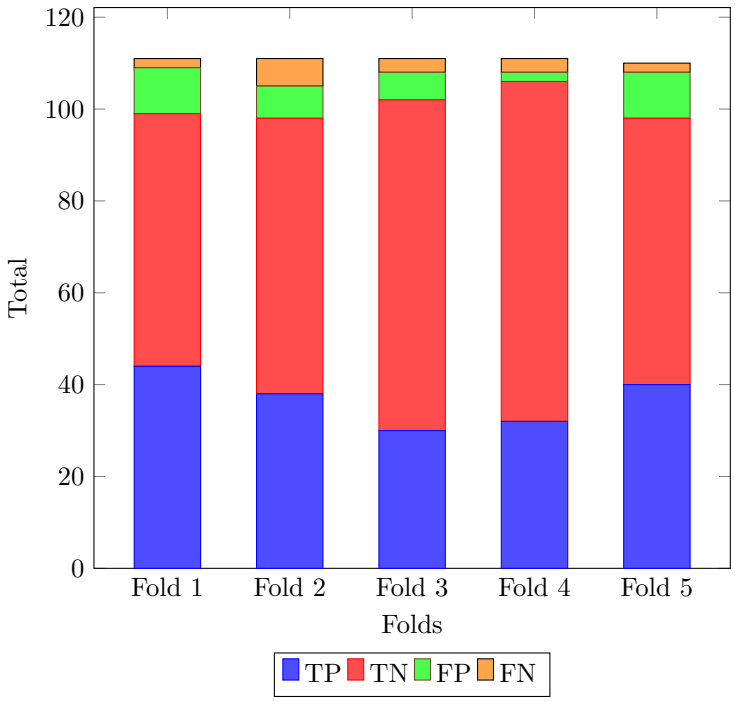
-	P	N
T	30	3
F	6	72

Fold 4:

-	P	N
T	32	3
F	2	74

Fold 5:

-	P	N
T	40	2
F	10	58



E, com base nelas, calculou-se as medidas básicas para cada fold e a média:

Fold 1:

Acurácia	89,2%
Precisão	81,5%
Cobertura	95,6%
Medida-F	87,9%

Fold 2:

Acurácia	88,3%
Precisão	84,4%
Cobertura	86,4%
Medida-F	85,4%

Fold 3:

Acurácia	91,9%
Precisão	83,3%
Cobertura	90,1%
Medida-F	86,6%

Fold 4:

Acurácia	95,5%
Precisão	94,1%
Cobertura	92,7%
Medida-F	93,4%

Fold 5:

Acurácia	89,1%
Precisão	80%
Cobertura	95,2%
Medida-F	86,9%

Média:

Acurácia	90,8%
Precisão	84,7%
Cobertura	92%
Medida-F	88%

5 Discussão

Uma vez que se conseguiu uma acurácia considerável nos testes (90,8%), pode-se dizer que esse tipo de arquitetura neural pode definitivamente ser usada na predição dos níveis de IgG de um paciente e, portanto, de seu estágio atual na infecção.

6 Bibliografia