

Metodologia

Com o objetivo de conseguir uma classificação satisfatória e com o máximo de acurácia possível, foi definida uma metodologia que utiliza uma rede neural convolucional (RNC) 1D com 12 camadas (4 camadas convolucionais, 4 camadas de *pooling*, uma camada de flatten e 3 camadas totalmente conectadas). A linguagem de programação utilizada para a codificação foi o Python, utilizando a biblioteca Keras. A base de dados possui dimensão de 1000x1025, com 17 classes. A primeira etapa foi carregar os dados e seus respectivos labels. Como foi utilizado uma RNC, é necessário que a matriz dos dados possua 3 dimensões. Portanto, após os dados terem sido carregados, foi adicionado mais uma dimensão ao conjunto de dados. Como o problema em questão é multiclasse, quando trabalhamos com uma RNC é necessário realizar um pré-processamento no conjunto de labels, no qual o mesmo passou de um shape de (1000x1) para (1000,17), ou seja, uma máscara binária foi criada para representar cada classe de cada amostra. Os dados foram aleatoriamente divididos nos conjuntos de treino e teste (70/30), não sendo realizado nenhum tipo de data augmentation ou geração de novos dados, pelo fato do conjunto de dados estar desbalanceado, isso poderia ser necessário. Para chegar aos resultados que serão mostrados em seguida, todos os dados reais da base foram utilizados, resultando em um conjunto de treinamento de 700 amostras (700x1025x1) e teste com 300 amostras (300x1025x1). A taxa de aprendizado é otimizada, diminuindo no decorrer das épocas. A quantidade de épocas foi 150, com um tamanho de lote de 32. Foi realizado apenas uma simulação com 150 épocas. Foi utilizada a função de ativação Softmax na camada de saída e a função de loss foi a categorical crossentropy, juntamente com o otimizador Adam.

Resultados

A acurácia do modelo de RNC para a etapa de treinamento foi 100% e para a etapa de teste foi de 87%. O fato de a base de dados estar desbalanceada contribuiu para a discrepância entre a acurácia do treino e do teste (e claramente para o aprendizado). A partir dos resultados de precision, recall e f1-score (disponível no notebook), é possível visualizar que há poucos exemplos em algumas classes, o que impacta diretamente no cálculo da taxa de acerto. A Figura 1 demonstra o desempenho do modelo durante a etapa de treinamento (curva azul) e teste (curva laranja).

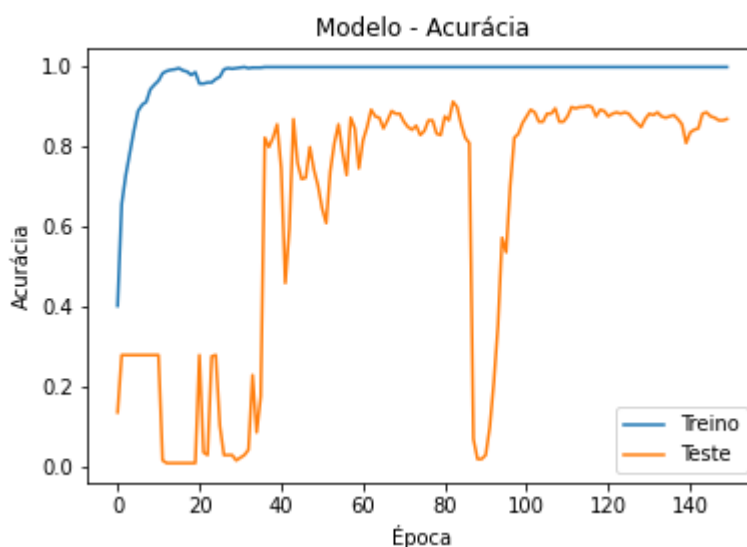


Figura 1 – Desempenho do modelo nas etapas de treinamento e teste.

A Figura 2 mostra a matriz de confusão, na qual mostra o desempenho do modelo na etapa de teste. As classes que possuem os maiores acertos são aquelas nas quais possuem mais imagens no dataset inteiro.

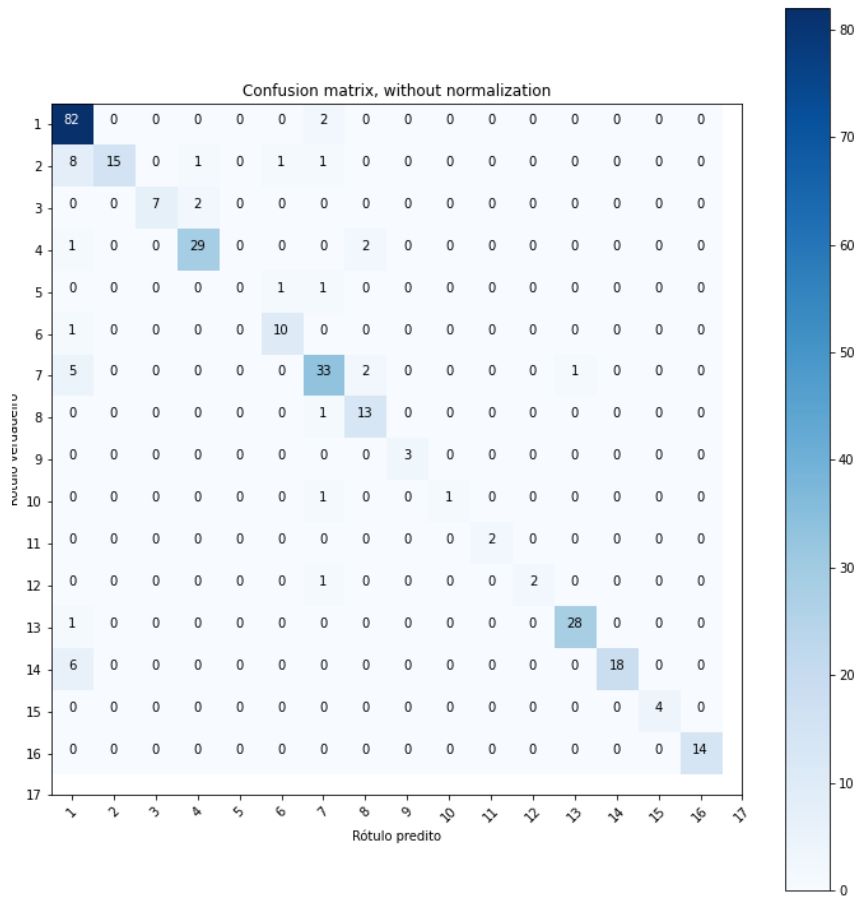


Figura 2 – Matriz de confusão.

Possíveis ajustes

Ainda seria necessário realizar o balanceamento da base de dados, a fim de tornar o modelo o mais generalista possível.