Problem Set 4

Ran Tao

(Dated: March 18, 2021)

Abstract

In this problem set, I used CNN to build an image classifier, that can be used to sort the scans into three categories, namely, the healthy patients, patients with some pre-existing conditions, and patients with various, serious lung conditions that require immediate attention. I tested several models and find the best of them. Since the labels are super imbalanced, I also used data augmentation to balance the dataset before training the models.

I. INTRODUCTION

In this problem set, I was asked by CEO to build an image classifier that can be used to sort the scans into three categories. I was also asked to test my model's performance. In addition, I was asked by my technical manager to balance the labels before training. And I was also asked by the senior developer to give codes that are easy to understand.

II. DATA EXPLORATION

This data set is composed of two parts, images and labels. There are 13260 images in this data set and the shape of each image is $64 \times 64 \times 1$. The labels data has three kinds of labels, 0, 1, and 2. The numbers of each label are: 0: 10506, 1: 2372, 2: 382.

III. DATA PREPROCESSING

First, I used train_test_split methods from sklearn.model_selection to split the data set into three parts, training data, validation data, and test data. The ratio of these three parts are 0.6, 0.2, 0.2.

Secondly, because the labels are super imbalanced, I decided to use data augmentation to balance the training data set. I imported ImageDataGenerator from keras.preprocessing.image, and used this function to create a image generator, which create new images from a given image by rotating, zooming, shearing....Then I used images whose labels are 1 to create many images so that the number of images with label 1 is equal to the number of images with label 0. Then I repeat the same process to create many images with label 2 so that the number of images with label 2 is equal to the number of images with label 0. By doing so, I created a data set with balanced labels.

IV. MODEL SELECTION AND FEATURE IMPORTANCE

Firstly, I used a CNN with only one convolutional layer, the kernel size is 3x3, the strides are 1, the padding are valid, and activation function is relu. Then I used max pooling and add a Dropout to prevent overfitting. Then I used a fully connected neural networks with a

hidden layer and a output layer to output the result of classification. The activation function of these two layers are relu and softmax.

Then I used another CNN with two convolutional layers. The layers are the same as before. After each convolutional layers, I used a max pooling and a dropout layer. Then I used a fully connected neural networks with a hidden layer and a output layer to output the result of classification. The activation function of these two layers are relu and softmax.

I used training data to train these two models, and I used validation data to measure the performance. Then I used early-stopping to decide when to stop. I computed the Training and Validation Accuracy as well as Training and Validation Loss of each epoch and plot them. The first model is Fig.1, the second model is Fig.2. As we can see in the pictures, the epoch of first model should be around 10, and the epoch of the second model should be around 10. And accuracy of the second model is better than the first model.

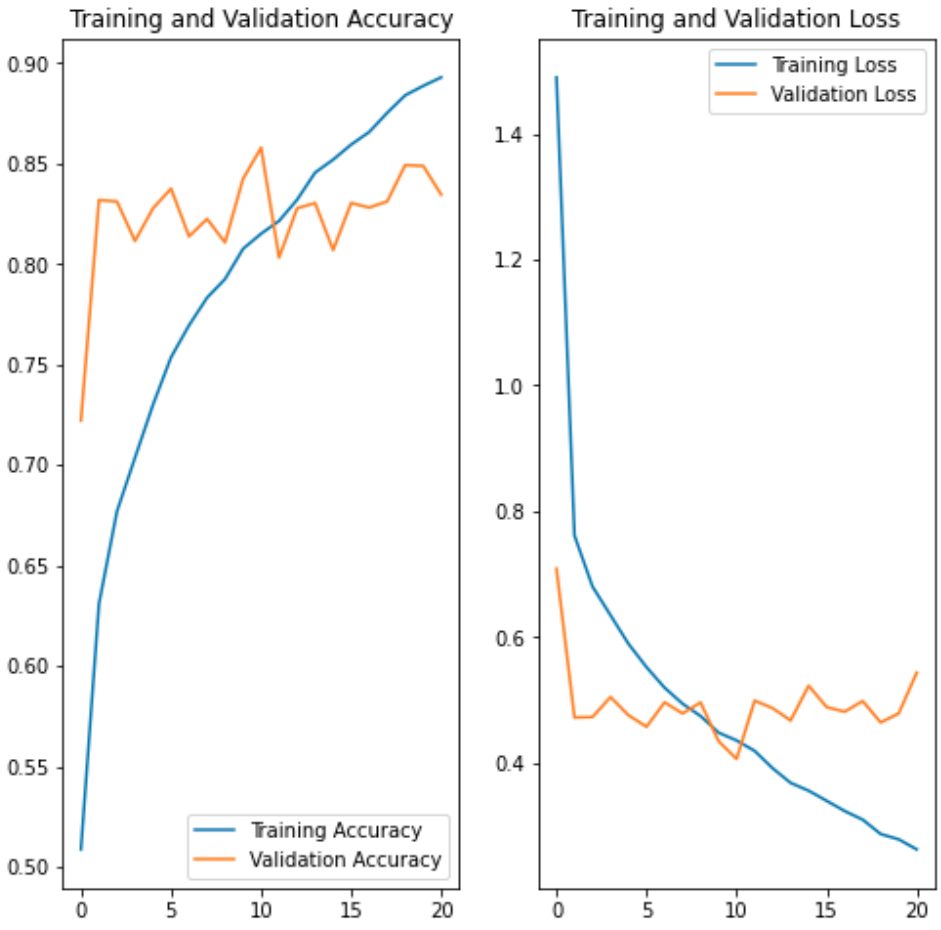

FIG. 1. Training and Validation Accuracy as well as Training and Validation Loss of each epoch

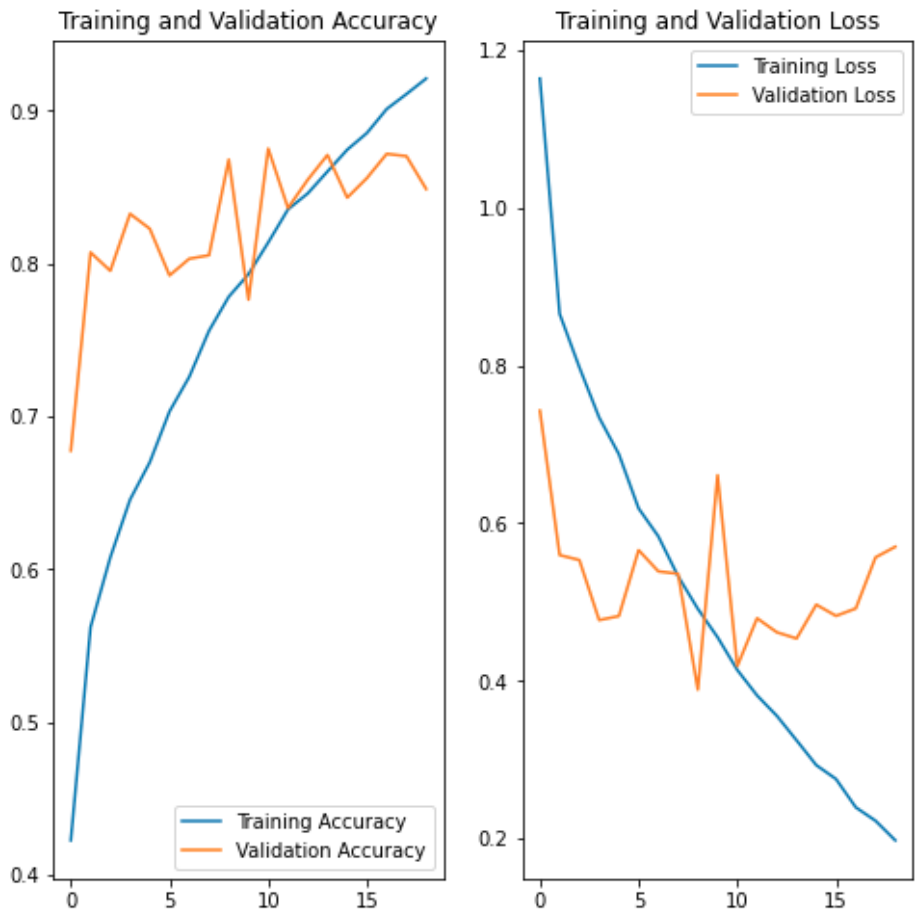

FIG. 2. Training and Validation Accuracy as well as Training and Validation Loss of each epoch

V. MODEL EVALUATION

Finally, I computed the confusion matrix of these two models. See Fig.3 and Fig.4. I also computed the precision score, recall score and F1 score. The F1 score of three classes are changing from 0.801 to 0.905, from 0.471 to 0.565, and from 0.038 to 0.063. Therefore, the second model is better than the first one in classification of all of these three classes.

VI. CONCLUSIONS

In this problem set, I first use data augmentation to balance the data set, and then I built the image classifier using two CNN models and used dropout to avoid overfitting. Then I used F1 score to compared these two models and find the second one is better.

```
[[1523  483  105]
 [ 136  304   15]
 [  34   48    4]]
              precision    recall  f1-score   support

           0      0.900     0.721     0.801      2111
           1      0.364     0.668     0.471       455
           2      0.032     0.047     0.038        86

    accuracy                          0.690      2652
   macro avg      0.432     0.479     0.437      2652
weighted avg      0.780     0.690     0.719      2652
```

FIG. 3. Confusion matrix of first model

```
[[1949   69   93]
 [ 200  218   37]
 [  49   30    7]]
              precision    recall  f1-score   support

           0      0.887     0.923     0.905      2111
           1      0.688     0.479     0.565       455
           2      0.051     0.081     0.063        86

    accuracy                          0.820      2652
   macro avg      0.542     0.495     0.511      2652
weighted avg      0.825     0.820     0.819      2652
```

FIG. 4. Confusion matrix of second model

DATA AVAILABILITY

Data is available at https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps4/data.

CODE AVAILABILITY

Code is available at https://github.com/usc-dsci552-32415D-spring2021/problem-set-04-rantao-usc/blob/main/Untitled.ipynb.