CS542 Class Challenge Report

shawnzhou10

April 2021

1 Question 1

For both tasks, I decided to reproduce the model provided by the notebook. Using the pre-trained model (vgg16) which is a complete CNN on its own, two dense layers were added in order to match the summary output. At first, for the last layer, I used softmax as the activation method, but it resulted in unchanging accuracy rate. I changed it to sigmoid which solved the problem. The problem could be because of the optimizer in the model compiler function, with from-logits being set to True.

For task 1, I did not change the optimizer because the default settings gave solid results. For the loss function, I used binary crossentropy, since the problem is a binary classification problem.

For task 2, I used categorical crossentropy, because the problem contains multiple dimensions, and is no longer binary. For the optimizer, I tried multiple learning rate with both Adam and SGD, but the results were similar. Adam finishes a bit quicker therefore I decided to go with Adam with a learning rate of 0.005 (though changing learning rate did not affect result much). I ended up obtaining a accuracy rate around 66 percent, though I think this is expected since the four categories overlap a lot more compared to the first task.

2 Architectures

For the other architecture, I tried both ResNet101. During this part of the project, I noticed that if the pre-trained base model is allowed to train in the program, the model achieves higher accuracy with less epochs, though each epoch would take 3-4 times longer.

Using ResNEt101, I tried a few different combinations of optimizer (Adam vs SGD) and learning rate, and ended up using the same setup as the vgg16 model. Some setups take way too long, and others had consistent low val-accuracy.

At the end, the prediction made using ResNet101 as base model is similar to that of vgg16, but the training takes slightly longer to finish.

3 Accuracy Loss

For task 1, I managed to reproduce the accuracy and loss function. Even though there are some fluctuation, the accuracy and val-accuracy stays above 90 percent for most of the epochs. Both the loss and val-loss stay below 0.5 for the most of the epochs, therefore indicates the model provided a solid prediction.

For task 2, the accuracy and loss function both fluctuate a lot more compared to the first task, though they generally retained their shape. val-accuracy seems to fluctuate around 0.7 and val-loss seems to fluctuate around 0.8. Their respective train curves, however, are much smoother.

I think this could mean that the model training is performing its job correctly, though the nature of the dataset (overlapping features) proved accurate prediction to be difficult.

4 tSNE

For task 1, I took inspiration of the scatter plot from piazza discussion. The t-SNE plot shows very clear clustering with distinct differences between COVID-19 dots and normal dots.

For task 2, as expected, the clusters are not as distince. In fact, many dots overlap with each other, especially the ones representing the two types of pneumonia and COVID-19. This further confirmed the difficulty of prediction.