

CS440 Class Challenge Report

WEIQJI

Section 1

Task1

In the beginning of my convolutional neural network, I implemented a VGG16 model with pre-trained weights from the Keras application library. It takes as input the images fed into the model, which have dimensions of 224x224 and 3 color channels. From this, the VGG16 model returns batches with dimension of 7x7 and a depth of 512. Next, I've added a Flatten layer which takes the output from VGG16 and flattens all of the characteristics into one dimension of $7 \times 7 \times 512 = 25088$ units. Then, a dense layer with a RELU non-linear activation function is used which takes the flattened layer as input to reduce its output to 256 dimensionalities. Lastly, another dense layer with a sigmoid activation is used to reduce the final output to 1 dimensionality, which is useful for binary classification tasks. I did not implement a dropout layer although I recognize its benefits of potentially reducing the running time per epoch.

Task2

The architecture I have implemented for task 2 is similar to that of task one – it implements a pre-trained Keras model followed by a flatten layer and two dense layers. In model 1 of task2, it also uses VGG16 and the only difference with this model from task 1 is the second dense layer. The second dense layer takes as input the dense layer from before (which still has 256 dimensionalities) and uses a softmax activation function to reduce the number of dimensionalities to 4 in the final output, which represent the 4 categories we are trying to differentiate from in this task. Model 2 is similar to that of model 1 as just described, except for that it uses a different pretrained architecture, which is the InceptionV3 from Keras' application library. Unlike VGG16, the InceptionV3 model outputs batches with dimension of 5x5 and a depth of 2048. The flatten layer following this takes the output from InceptionV3 and returns one dimension of $5 \times 5 \times 2048 = 51200$ units.

Section 2

Task1

I implemented the Adam optimizer with binary_crossentropy as the loss function to compile the model. I did not use any extra regularization techniques.

Task2

For the two models, I implemented the same process for compiling and training. I used the Adam optimizer again because there seems to be negligible difference between the different optimizers. However, I used categorical_crossentropy as the loss function in this task because it is more suiting for categorical results.

Section 3

The VGG16 model was significantly slower to compute than InceptionV3, taking 2-3 seconds per step in training compared to less than one second per step. In total, the VGG16 model took an average of around 1 minute per epoch whereas the InceptionV3 model took around 20 seconds. The VGG16 model is also much larger in file size than InceptionV3, although the Inception model had more depth. The VGG16 model can also compute for many more parameters than Inception can, although Inception much faster in computation.

Section 4

Of the two architectures I have implemented in task 2, VGG16 was the one that produced more accurate results. While the InceptionV3 model also had very high training accuracy at around 0.8 as the training iterated over the epochs, it was not able to produce good testing accuracy which stayed the same in the 0.7 range and did not significantly increase as training went on. Moreover, the InceptionV3 was able to reduce training losses down to around 0.3 and had a lower loss than the VGG16 model towards the end. This also reflects on its inaccuracy as the validation loss appear very scattered throughout training and does not seem to have a decreasing trend. On the other hand, the VGG16 model produced correlations as accuracy increased and validation decreased throughout the training.

Section 5

Task1

The t-SNE visualization for covid and normal lungs has two distinct regions with points scattered in those regions. However, I can see that there is clearly a bit of overlap where a few normal lungs are grouped in the covid scatter that is closest to the normal lung scatter. This tells me that some of the covid lungs are not that much different than normal lungs since a few of the normal lungs can be passed as the covid lungs according to what this plot is telling me about my model.

Task2

The visualization for the four categories of lungs is mostly distinct in this scatter plot. You can clearly tell that normal lungs are differentiated from the rest, being in its own cluster in the corner with once again some dots in the covid cluster and only a few in the other cluster. Covid lungs have its own cluster as well, which is separated from the rest while being closer to the pneumonia lungs. The two types of pneumonia lungs – bacterial and viral are hard to differentiate because they show up in the same region on the scatter plot and their clusters overlap with each other. However, it is good that pneumonia in general is differentiated from the rest.