1. Lowering the epochs from 200 to 65

```
Epoch 191/200
1562/1562 [==============================] - 40s - loss: 1.3214 - acc: 0.5763 - val_loss: 1.4629 - val_acc: 0.4990
Epoch 192/200
1562/1562 [==============================] - 40s - loss: 1.3366 - acc: 0.5734 - val_loss: 1.2753 - val_acc: 0.6168
Epoch 193/200
1562/1562 [==============================] - 39s - loss: 1.3132 - acc: 0.5790 - val_loss: 1.2697 - val_acc: 0.5770
Epoch 194/200
1562/1562 [==============================] - 37s - loss: 1.3468 - acc: 0.5673 - val_loss: 1.3911 - val_acc: 0.4845
Epoch 195/200
1562/1562 [==============================] - 38s - loss: 1.3384 - acc: 0.5742 - val_loss: 1.2960 - val_acc: 0.5715
Epoch 196/200
1562/1562 [==============================] - 42s - loss: 1.3458 - acc: 0.5710 - val_loss: 1.3576 - val_acc: 0.5482
Epoch 197/200
1562/1562 [==============================] - 38s - loss: 1.3579 - acc: 0.5698 - val_loss: 1.1148 - val_acc: 0.6615
Epoch 198/200
1562/1562 [==============================] - 37s - loss: 1.3487 - acc: 0.5674 - val_loss: 1.6051 - val_acc: 0.4042
Epoch 199/200
1562/1562 [==============================] - 37s - loss: 1.3468 - acc: 0.5676 - val_loss: 1.2821 - val_acc: 0.5386
Epoch 200/200
1562/1562 [==============================] - 37s - loss: 1.3416 - acc: 0.5682 - val_loss: 1.2945 - val_acc: 0.5395
```

As shown above I noticed when the initial code was running 200 epochs, I've noticed that the accuracy was decreasing after around 100 epoch which resulted in an accuracy around 0.57, therefore I decreased the epochs to 65 to increase the accuracy. This is because these epochs are overfitting the training data. As a result, the data accuracy I received was 0.77

2. Lowering the dropout rate.

The dropout rate means the rate as which how many neurons are not working in the neural network thus decreasing the dropout rate will increase the number of neurons. Also, even though some dropout rate does increase in accuracy, 0.5 is high percentage which might lead to under learning since it might make the model overfitting. Therefore, I lowered the dropout rate to 0.2. The resulting accuracy is around 0.8 therefore increasing by 0.3

3. Enabling no data augmentation

To study the process of the data augmentation and how it effects the accuracy of the whole neural network, the data augmentation was taken off. Data augmentation is then turned off by enabling as false. As a result of this, the remaining accuracy was decreased by 0.03 which was the data accuracy the first time.

4. Increasing the bach size

When the batch size increased, there is a higher learning time. However, it will need more epochs to reach the limit before it overfits. Even before it overfits, it remained the same accuracy

Conclusion:

As a result of this experiment, it shows data augmentation is needed to increase accuracy. One the other hand the number of epochs and dropout rate needs to have the right value or the accuracy might be not as accurate. For example, a larger epoch will overfit the training data but a small epoch will prevent it from learning its full potential.