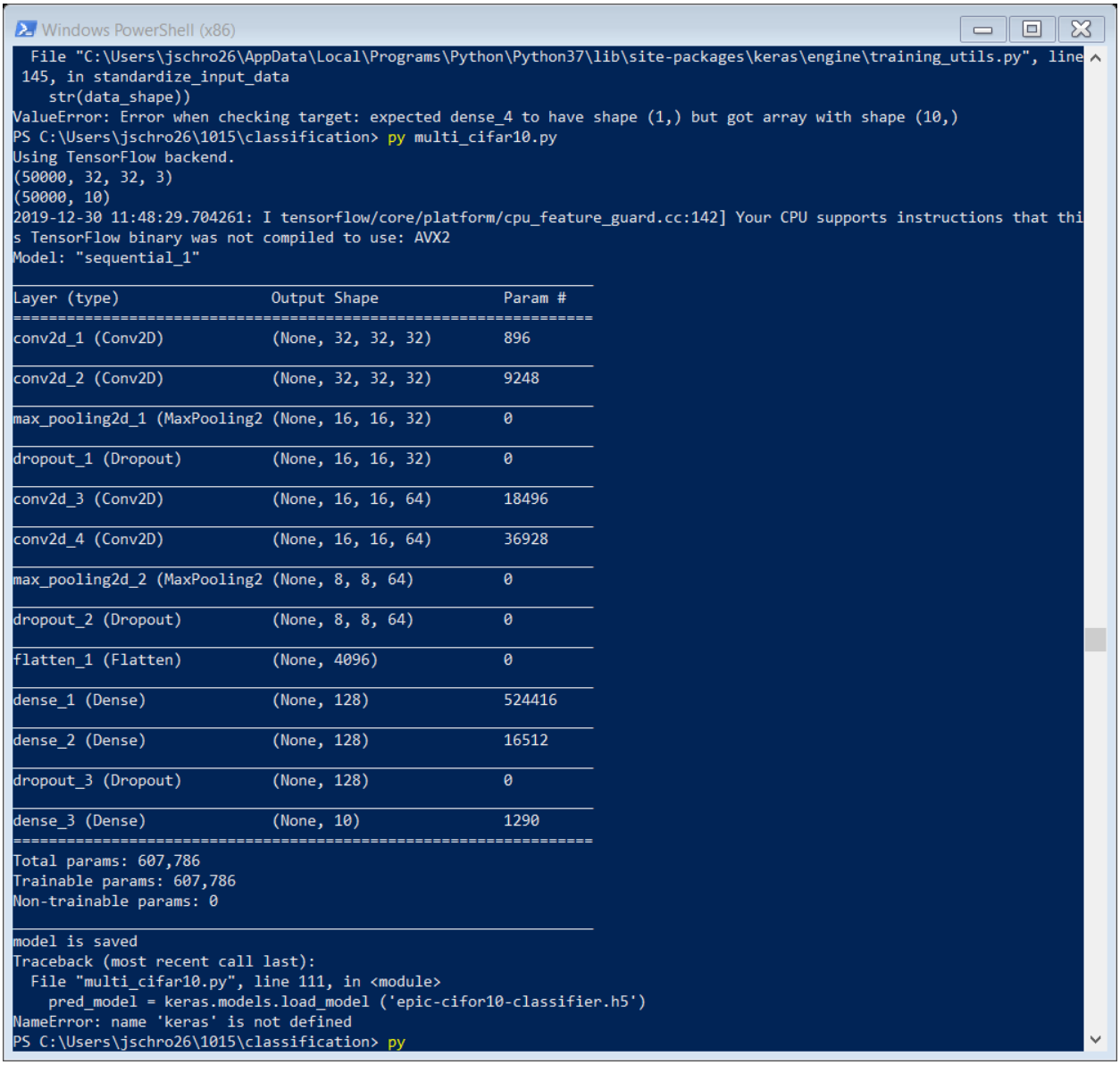
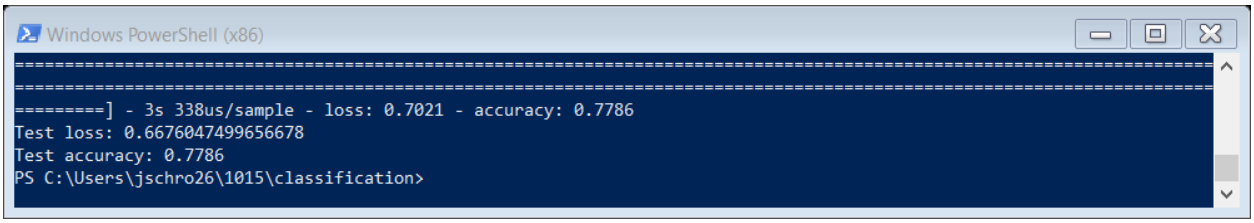
The following screenshots demonstrate the optimization and training process of the NN. 3 iterations of models are presented.

Model 1

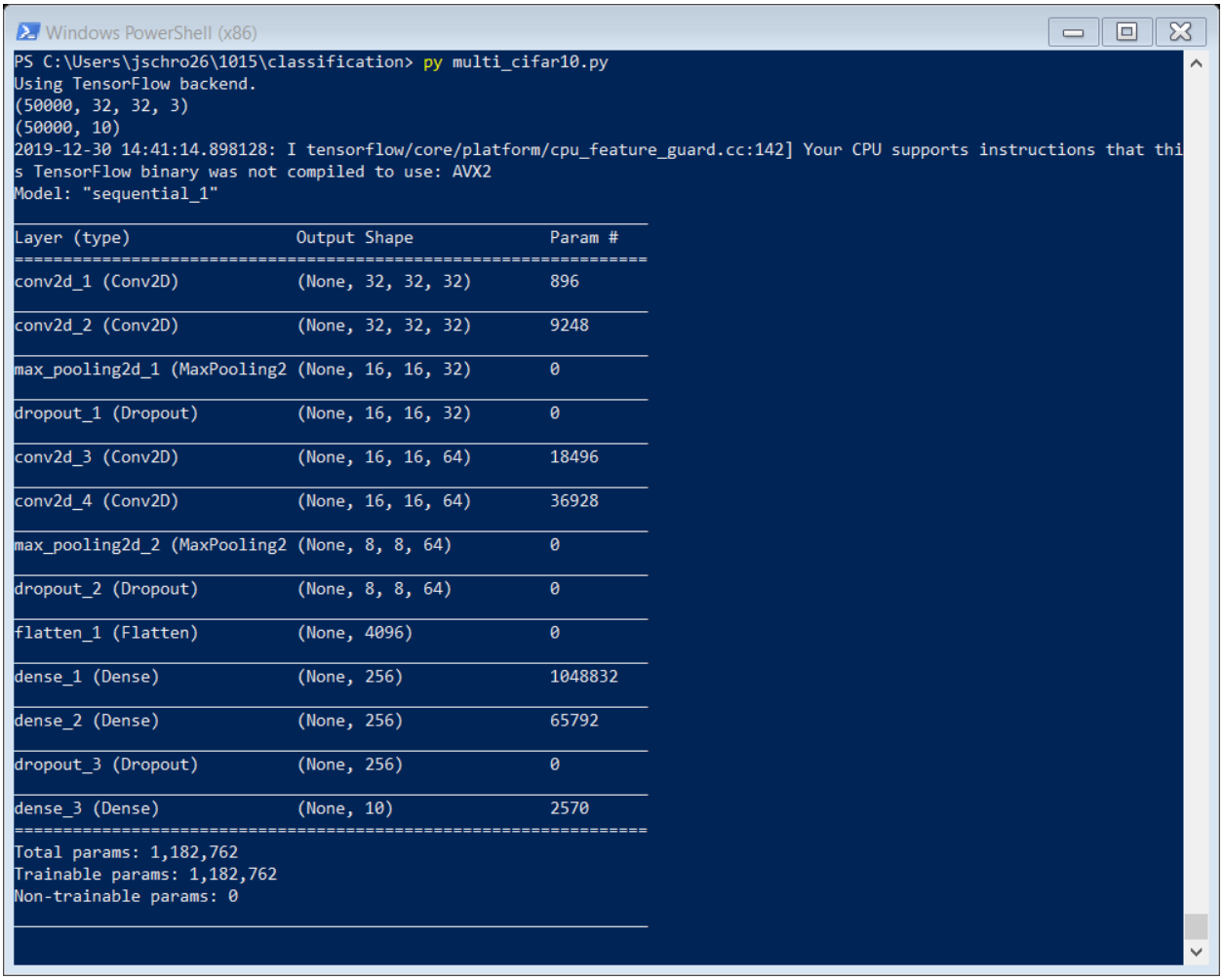


This screenshot illustrates the 1st model construction.

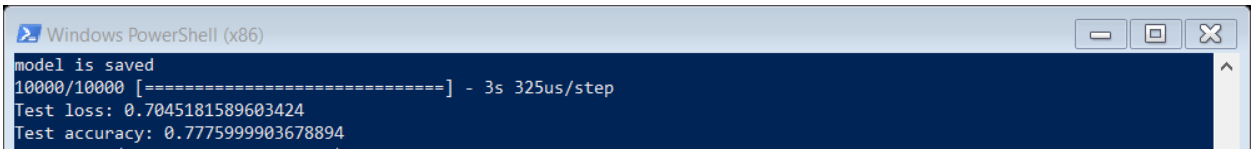


This screenshot illustrates the amount of loss and initial accuracy of the model.

Model 2

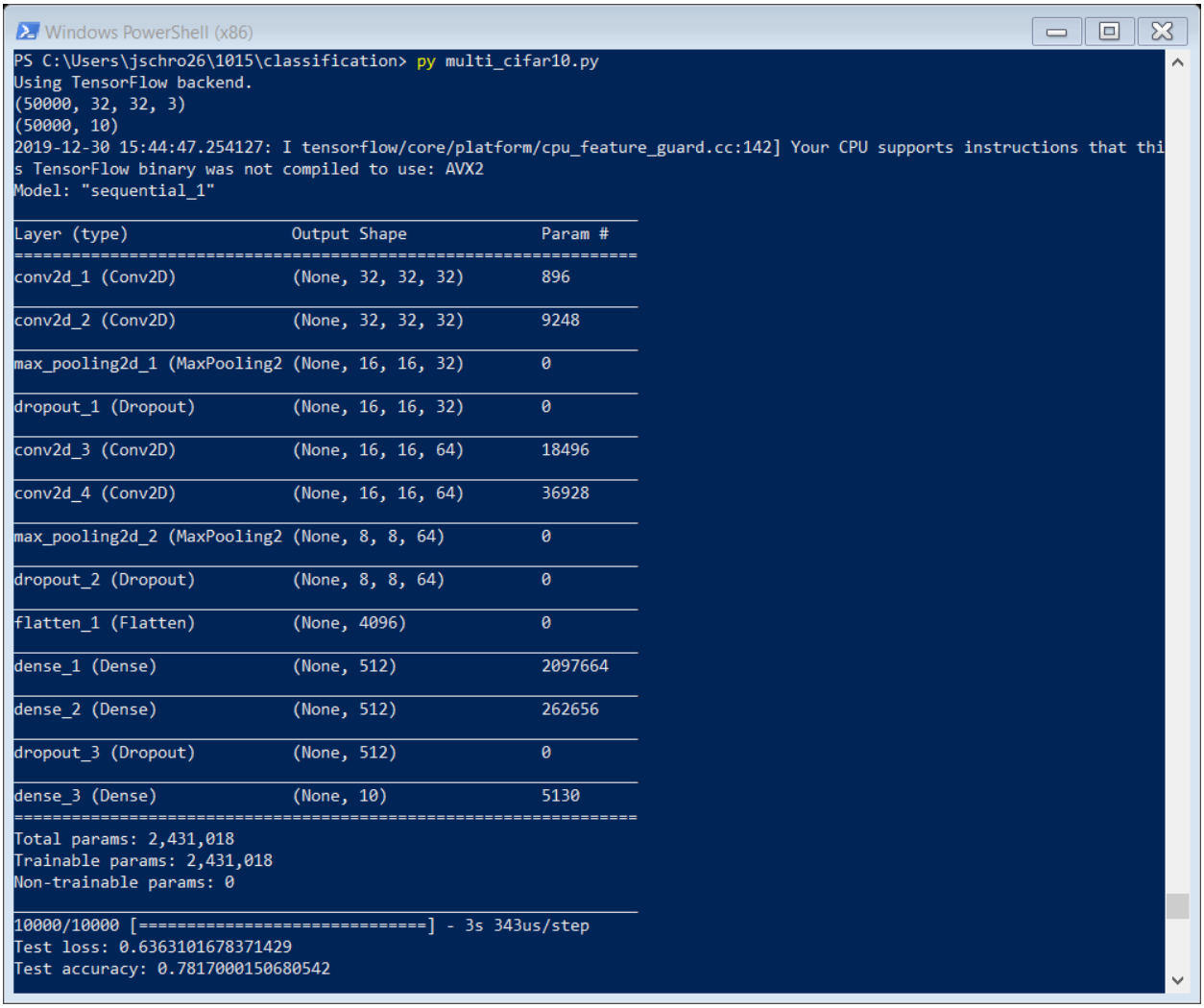


This illustrates changes including the addition of 20 epochs and 256 nodes in the 3rd layer of dense neurons.



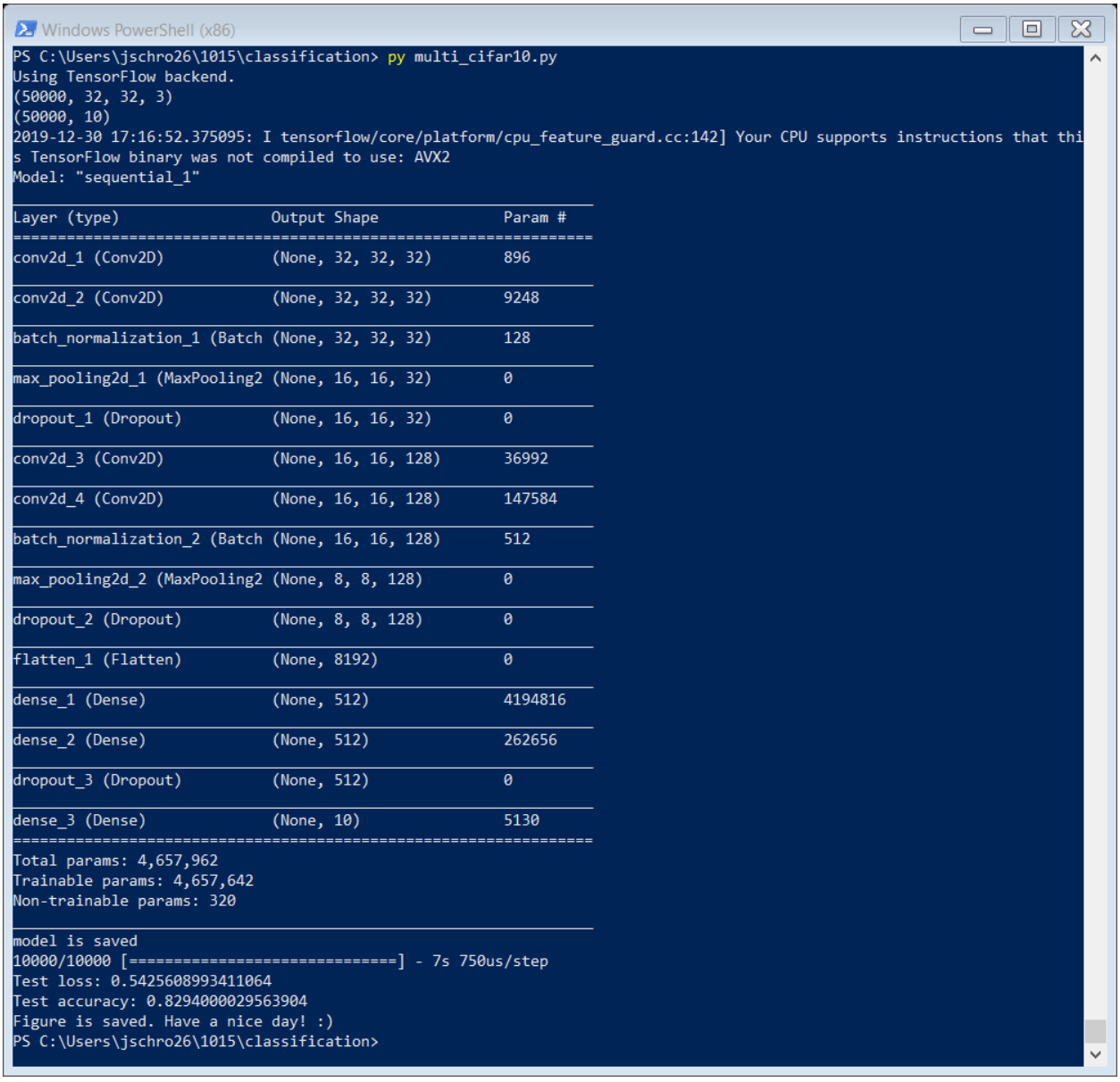
These changes actually reduced accuracy and increased loss in the model.

Model 3



This model reflects an increase in modes in the 3rd layer to 512 and this increased accuracy to 78% and reduced loss to 63%

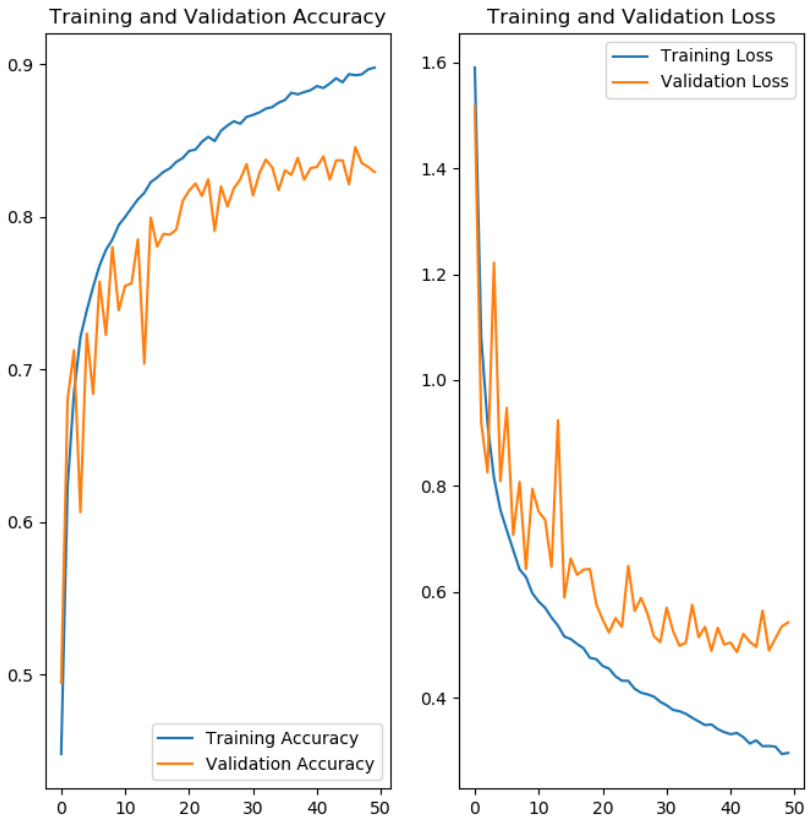
Model 3



```
Windows PowerShell (x86)
PS C:\Users\jschro26\1015\classification> py multi_cifar10.py
Using TensorFlow backend.
(50000, 32, 32, 3)
(50000, 10)
2019-12-30 17:16:52.375095: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that thi
s TensorFlow binary was not compiled to use: AVX2
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 128)       36992
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 128)       147584
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               4194816
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                5130
=================================================================
Total params: 4,657,962
Trainable params: 4,657,642
Non-trainable params: 320
_________________________________________________________________
model is saved
10000/10000 [==============================] - 7s 750us/step
Test loss: 0.5425608993411064
Test accuracy: 0.8294000029563904
Figure is saved. Have a nice day! :)
PS C:\Users\jschro26\1015\classification>
```

Final Run: Increased nodes of 2nd layer to 128 and added Batch size 64 and Batch Normalization in an attempt to improve accuracy.

Plot follows…


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Training and Validation Loss
Training Loss
Validation Loss