

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

Model 1

```
Windows PowerShell (x86)
File "C:\Users\jschro26\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\engine\training_utils.py", line 145, in standardize_input_data
    str(data_shape))
ValueError: Error when checking target: expected dense_4 to have shape (1,) but got array with shape (10,)
PS C:\Users\jschro26\1015\classification> py multi_cifar10.py
Using TensorFlow backend.
(50000, 32, 32, 3)
(50000, 10)
2019-12-30 11:48:29.704261: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this 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
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)          0
dropout_1 (Dropout)          (None, 16, 16, 32)          0
conv2d_3 (Conv2D)            (None, 16, 16, 64)          18496
conv2d_4 (Conv2D)            (None, 16, 16, 64)          36928
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
dropout_2 (Dropout)          (None, 8, 8, 64)          0
flatten_1 (Flatten)          (None, 4096)                0
dense_1 (Dense)              (None, 128)                 524416
dense_2 (Dense)              (None, 128)                 16512
dropout_3 (Dropout)          (None, 128)                 0
dense_3 (Dense)              (None, 10)                 1290
-----
Total params: 607,786
Trainable params: 607,786
Non-trainable params: 0

model is saved
Traceback (most recent call last):
  File "multi_cifar10.py", line 111, in <module>
    pred_model = keras.models.load_model('epic-cifar10-classifier.h5')
NameError: name 'keras' is not defined
PS C:\Users\jschro26\1015\classification> py
```

This screenshot illustrates the 1st model construction.

```
Windows PowerShell (x86)
=====] - 3s 338us/sample - loss: 0.7021 - accuracy: 0.7786
Test loss: 0.6676047499656678
Test accuracy: 0.7786
PS C:\Users\jschro26\1015\classification>
```

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

Model 2

```
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 14:41:14.898128: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this 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
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)          0
dropout_1 (Dropout)          (None, 16, 16, 32)          0
conv2d_3 (Conv2D)            (None, 16, 16, 64)          18496
conv2d_4 (Conv2D)            (None, 16, 16, 64)          36928
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
dropout_2 (Dropout)          (None, 8, 8, 64)          0
flatten_1 (Flatten)          (None, 4096)                0
dense_1 (Dense)              (None, 256)                 1048832
dense_2 (Dense)              (None, 256)                 65792
dropout_3 (Dropout)          (None, 256)                 0
dense_3 (Dense)              (None, 10)                  2570
=====
Total params: 1,182,762
Trainable params: 1,182,762
Non-trainable params: 0
```

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

```
Windows PowerShell (x86)
model is saved
10000/10000 [=====] - 3s 325us/step
Test loss: 0.7045181589603424
Test accuracy: 0.7775999903678894
```

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

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 15:44:47.254127: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this 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
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)          0
dropout_1 (Dropout)          (None, 16, 16, 32)          0
conv2d_3 (Conv2D)            (None, 16, 16, 64)          18496
conv2d_4 (Conv2D)            (None, 16, 16, 64)          36928
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
dropout_2 (Dropout)          (None, 8, 8, 64)          0
flatten_1 (Flatten)          (None, 4096)                0
dense_1 (Dense)              (None, 512)                 2097664
dense_2 (Dense)              (None, 512)                 262656
dropout_3 (Dropout)          (None, 512)                 0
dense_3 (Dense)              (None, 10)                  5130
-----
Total params: 2,431,018
Trainable params: 2,431,018
Non-trainable params: 0

10000/10000 [=====] - 3s 343us/step
Test loss: 0.6363101678371429
Test accuracy: 0.7817000150680542
```

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 this 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 Normalization) (None, 32, 32, 32)          128
max_pooling2d_1 (MaxPooling2D) (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 Normalization) (None, 16, 16, 128)         512
max_pooling2d_2 (MaxPooling2D) (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...

