

CNN on CIFAR Assignment

1. Please visit this link to access the state-of-art DenseNet code for reference - DenseNet - cifar10 notebook link
2. You need to create a copy of this and "retrain" this model to achieve 90+ test accuracy.
3. You cannot use Dense Layers (also called fully connected layers), or DropOut.
4. You MUST use Image Augmentation Techniques.
5. You cannot use an already trained model as a beginning points, you have to initialize as your own
6. You cannot run the program for more than 300 Epochs, and it should be clear from your log, that you have only used 300 Epochs
7. You cannot use test images for training the model.
8. You cannot change the general architecture of DenseNet (which means you must use Dense Block, Transition and Output blocks as mentioned in the code)
9. You are free to change Convolution types (e.g. from 3x3 normal convolution to Depthwise Separable, etc)
10. You cannot have more than 1 Million parameters in total
11. You are free to move the code from Keras to Tensorflow, Pytorch, MXNET etc.
12. You can use any optimization algorithm you need.
13. You can checkpoint your model and retrain the model from that checkpoint so that no need of training the model from first if you lost at any epoch while training. You can directly load that model and Train from that epoch.

In [1]:

```
# import keras
# from keras.datasets import cifar10
# from keras.models import Model, Sequential
# from keras.layers import Dense, Dropout, Flatten, Input, AveragePooling2D, merge, Activation
# from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
# from keras.layers import Concatenate
# from keras.optimizers import Adam
import numpy as np
np.random.seed(42)
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.regularizers import l1
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, CSVLogger
# this part will prevent tensorflow to allocate all the available GPU Memory
# backend
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
tf.config.list_physical_devices('GPU')
```

Using TensorFlow backend.

```
{name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 11382562692963312915
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 3135687884
locality {
  bus_id: 1
  links {
  }
}
incarnation: 10717626493672370588
physical_device_desc: "device: 0, name: GeForce GTX 1050 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1"
}
```

Out[1]:

```
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

In [2]:

```
# Hyperparameters
num_classes = 10
l = 12
num_filter = 12
compression = 0.5
```

In [3]:

```
# Load CIFAR10 Data
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.cifar10.load_data()
img_height, img_width, channel = X_train.shape[1], X_train.shape[2], X_train.shape[3]

# convert to one hot encoding
y_train = tf.keras.utils.to_categorical(y_train, num_classes)
y_test = tf.keras.utils.to_categorical(y_test, num_classes)
```

In [4]:

```
X_train.shape
```

Out[4]:

```
(50000, 32, 32, 3)
```

In [5]:

```
X_test.shape
```

Out[5]:

```
(10000, 32, 32, 3)
```

In [6]:

```
# Refer: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.htm
1
train_datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

test_datagen = ImageDataGenerator(
    rescale = 1./255)

train_datagen.fit(X_train)
test_datagen.fit(X_test)
```

In [7]:

```
# Dense Block
def denseblock(input, num_filter = 12):
    global compression
    temp = input
    for _ in range(l):
        BatchNorm = layers.BatchNormalization()(temp)
        relu = layers.Activation('relu')(BatchNorm)
        Conv2D_3_3 = layers.Conv2D(int(num_filter*compression), (3,3), use_bias=False, padding='same',
        kernel_initializer=\
```

```

        keras.initializers.glorot_normal(seed=42))(relu)
    concat = layers.Concatenate(axis=-1) ([temp,Conv2D_3_3])
    temp = concat

    return temp

## transition Block
def transition(input, num_filter = 12):
    global compression
    BatchNorm = layers.BatchNormalization() (input)
    relu = layers.Activation('relu') (BatchNorm)
    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (1,1), use_bias=False ,padding='same'
    , \
                                   kernel_initializer= keras.initializers.glorot_normal(seed=42))(relu)
    avg = layers.AveragePooling2D(pool_size=(2,2)) (Conv2D_BottleNeck)
    return avg

#output layer
def output_layer(input):
    global compression
    BatchNorm = layers.BatchNormalization() (input)
    relu = layers.Activation('relu') (BatchNorm)
    AvgPooling = layers.AveragePooling2D(pool_size=(2,2)) (relu)
    output_1 = layers.Conv2D(num_classes, (2,2), kernel_initializer=keras.initializers.glorot_normal(seed=42)) (AvgPooling)
    out_1 = Activation('softmax') (output_1)
    flat = layers.Flatten() (out_1)
    return flat

```

In [8]:

```

num_filter = 37

input = layers.Input(shape=(img_height, img_width, channel))
First_Conv2D = layers.Conv2D(num_filter, (3,3), use_bias=False ,padding='same') (input)

First_Block = denseblock(First_Conv2D, num_filter)
First_Transition = transition(First_Block, num_filter)

Second_Block = denseblock(First_Transition, num_filter)
Second_Transition = transition(Second_Block, num_filter)

Third_Block = denseblock(Second_Transition, num_filter)
Third_Transition = transition(Third_Block, num_filter)

Last_Block = denseblock(Third_Transition, num_filter)
output = output_layer(Last_Block)

```

In [9]:

```

#https://arxiv.org/pdf/1608.06993.pdf
# from IPython.display import IFrame, YouTubeVideo
# YouTubeVideo(id='-W6y8xnd--U', width=600)

```

In [10]:

```

model = Model(inputs=[input], outputs=[output])
model.summary()

```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 32, 32, 3)]	0	
conv2d (Conv2D)	(None, 32, 32, 37)	999	input_1[0][0]
batch_normalization (BatchNorma	(None, 32, 32, 37)	148	conv2d[0][0]
activation (Activation)	(None, 32, 32, 37)	0	batch_normalization[0][0]

conv2d_1 (Conv2D)	(None, 32, 32, 18)	5994	activation[0][0]
concatenate (Concatenate)	(None, 32, 32, 55)	0	conv2d[0][0] conv2d_1[0][0]
batch_normalization_1 (BatchNor	(None, 32, 32, 55)	220	concatenate[0][0]
activation_1 (Activation)	(None, 32, 32, 55)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 32, 32, 18)	8910	activation_1[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 73)	0	concatenate[0][0] conv2d_2[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 73)	292	concatenate_1[0][0]
activation_2 (Activation)	(None, 32, 32, 73)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 32, 32, 18)	11826	activation_2[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 91)	0	concatenate_1[0][0] conv2d_3[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 91)	364	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 91)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 18)	14742	activation_3[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 109)	0	concatenate_2[0][0] conv2d_4[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 109)	436	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 109)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 18)	17658	activation_4[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 127)	0	concatenate_3[0][0] conv2d_5[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 127)	508	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 127)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 18)	20574	activation_5[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 145)	0	concatenate_4[0][0] conv2d_6[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 145)	580	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 145)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 18)	23490	activation_6[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 163)	0	concatenate_5[0][0] conv2d_7[0][0]
batch_normalization_7 (BatchNor	(None, 32, 32, 163)	652	concatenate_6[0][0]
activation_7 (Activation)	(None, 32, 32, 163)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 18)	26406	activation_7[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 181)	0	concatenate_6[0][0] conv2d_8[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 181)	724	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 181)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 18)	29322	activation_8[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 199)	0	concatenate_7[0][0] conv2d_9[0][0]

batch_normalization_9	(BatchNor	(None, 32, 32, 199)	796	concatenate_8[0][0]
activation_9	(Activation)	(None, 32, 32, 199)	0	batch_normalization_9[0][0]
conv2d_10	(Conv2D)	(None, 32, 32, 18)	32238	activation_9[0][0]
concatenate_9	(Concatenate)	(None, 32, 32, 217)	0	concatenate_8[0][0] conv2d_10[0][0]
batch_normalization_10	(BatchNo	(None, 32, 32, 217)	868	concatenate_9[0][0]
activation_10	(Activation)	(None, 32, 32, 217)	0	batch_normalization_10[0][0]
conv2d_11	(Conv2D)	(None, 32, 32, 18)	35154	activation_10[0][0]
concatenate_10	(Concatenate)	(None, 32, 32, 235)	0	concatenate_9[0][0] conv2d_11[0][0]
batch_normalization_11	(BatchNo	(None, 32, 32, 235)	940	concatenate_10[0][0]
activation_11	(Activation)	(None, 32, 32, 235)	0	batch_normalization_11[0][0]
conv2d_12	(Conv2D)	(None, 32, 32, 18)	38070	activation_11[0][0]
concatenate_11	(Concatenate)	(None, 32, 32, 253)	0	concatenate_10[0][0] conv2d_12[0][0]
batch_normalization_12	(BatchNo	(None, 32, 32, 253)	1012	concatenate_11[0][0]
activation_12	(Activation)	(None, 32, 32, 253)	0	batch_normalization_12[0][0]
conv2d_13	(Conv2D)	(None, 32, 32, 18)	4554	activation_12[0][0]
average_pooling2d	(AveragePooli	(None, 16, 16, 18)	0	conv2d_13[0][0]
batch_normalization_13	(BatchNo	(None, 16, 16, 18)	72	average_pooling2d[0][0]
activation_13	(Activation)	(None, 16, 16, 18)	0	batch_normalization_13[0][0]
conv2d_14	(Conv2D)	(None, 16, 16, 18)	2916	activation_13[0][0]
concatenate_12	(Concatenate)	(None, 16, 16, 36)	0	average_pooling2d[0][0] conv2d_14[0][0]
batch_normalization_14	(BatchNo	(None, 16, 16, 36)	144	concatenate_12[0][0]
activation_14	(Activation)	(None, 16, 16, 36)	0	batch_normalization_14[0][0]
conv2d_15	(Conv2D)	(None, 16, 16, 18)	5832	activation_14[0][0]
concatenate_13	(Concatenate)	(None, 16, 16, 54)	0	concatenate_12[0][0] conv2d_15[0][0]
batch_normalization_15	(BatchNo	(None, 16, 16, 54)	216	concatenate_13[0][0]
activation_15	(Activation)	(None, 16, 16, 54)	0	batch_normalization_15[0][0]
conv2d_16	(Conv2D)	(None, 16, 16, 18)	8748	activation_15[0][0]
concatenate_14	(Concatenate)	(None, 16, 16, 72)	0	concatenate_13[0][0] conv2d_16[0][0]
batch_normalization_16	(BatchNo	(None, 16, 16, 72)	288	concatenate_14[0][0]
activation_16	(Activation)	(None, 16, 16, 72)	0	batch_normalization_16[0][0]
conv2d_17	(Conv2D)	(None, 16, 16, 18)	11664	activation_16[0][0]
concatenate_15	(Concatenate)	(None, 16, 16, 90)	0	concatenate_14[0][0] conv2d_17[0][0]
batch_normalization_17	(BatchNo	(None, 16, 16, 90)	360	concatenate_15[0][0]
activation_17	(Activation)	(None, 16, 16, 90)	0	batch_normalization_17[0][0]
conv2d_18	(Conv2D)	(None, 16, 16, 18)	14580	activation_17[0][0]

concatenate_16 (Concatenate)	(None, 16, 16, 108)	0	concatenate_15[0][0] conv2d_18[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 108)	432	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 108)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 18)	17496	activation_18[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 126)	0	concatenate_16[0][0] conv2d_19[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 126)	504	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 126)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 18)	20412	activation_19[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 144)	0	concatenate_17[0][0] conv2d_20[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 144)	576	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 144)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 18)	23328	activation_20[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 162)	0	concatenate_18[0][0] conv2d_21[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 162)	648	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 162)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 18)	26244	activation_21[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 180)	0	concatenate_19[0][0] conv2d_22[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 180)	720	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 180)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 18)	29160	activation_22[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 198)	0	concatenate_20[0][0] conv2d_23[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 198)	792	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 198)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 18)	32076	activation_23[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 216)	0	concatenate_21[0][0] conv2d_24[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 216)	864	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 216)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 18)	34992	activation_24[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 234)	0	concatenate_22[0][0] conv2d_25[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 234)	936	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 234)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 18)	4212	activation_25[0][0]
average_pooling2d_1 (AveragePoo	(None, 8, 8, 18)	0	conv2d_26[0][0]
batch_normalization_26 (BatchNo	(None, 8, 8, 18)	72	average_pooling2d_1[0][0]

activation_26 (Activation)	(None, 8, 8, 18)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 8, 8, 18)	2916	activation_26[0][0]
concatenate_24 (Concatenate)	(None, 8, 8, 36)	0	average_pooling2d_1[0][0] conv2d_27[0][0]
batch_normalization_27 (BatchNo	(None, 8, 8, 36)	144	concatenate_24[0][0]
activation_27 (Activation)	(None, 8, 8, 36)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 8, 8, 18)	5832	activation_27[0][0]
concatenate_25 (Concatenate)	(None, 8, 8, 54)	0	concatenate_24[0][0] conv2d_28[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 54)	216	concatenate_25[0][0]
activation_28 (Activation)	(None, 8, 8, 54)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 18)	8748	activation_28[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 72)	0	concatenate_25[0][0] conv2d_29[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 72)	288	concatenate_26[0][0]
activation_29 (Activation)	(None, 8, 8, 72)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 18)	11664	activation_29[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 90)	0	concatenate_26[0][0] conv2d_30[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 90)	360	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 90)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 18)	14580	activation_30[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 108)	0	concatenate_27[0][0] conv2d_31[0][0]
batch_normalization_31 (BatchNo	(None, 8, 8, 108)	432	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 108)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 18)	17496	activation_31[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 126)	0	concatenate_28[0][0] conv2d_32[0][0]
batch_normalization_32 (BatchNo	(None, 8, 8, 126)	504	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 126)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 18)	20412	activation_32[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 144)	0	concatenate_29[0][0] conv2d_33[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 144)	576	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 144)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 18)	23328	activation_33[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 162)	0	concatenate_30[0][0] conv2d_34[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 162)	648	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 162)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 18)	26244	activation_34[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 180)	0	concatenate_31[0][0]

			conv2d_35[0][0]
batch_normalization_35	(BatchNo (None, 8, 8, 180)	720	concatenate_32[0][0]
activation_35	(Activation) (None, 8, 8, 180)	0	batch_normalization_35[0][0]
conv2d_36	(Conv2D) (None, 8, 8, 18)	29160	activation_35[0][0]
concatenate_33	(Concatenate) (None, 8, 8, 198)	0	concatenate_32[0][0] conv2d_36[0][0]
batch_normalization_36	(BatchNo (None, 8, 8, 198)	792	concatenate_33[0][0]
activation_36	(Activation) (None, 8, 8, 198)	0	batch_normalization_36[0][0]
conv2d_37	(Conv2D) (None, 8, 8, 18)	32076	activation_36[0][0]
concatenate_34	(Concatenate) (None, 8, 8, 216)	0	concatenate_33[0][0] conv2d_37[0][0]
batch_normalization_37	(BatchNo (None, 8, 8, 216)	864	concatenate_34[0][0]
activation_37	(Activation) (None, 8, 8, 216)	0	batch_normalization_37[0][0]
conv2d_38	(Conv2D) (None, 8, 8, 18)	34992	activation_37[0][0]
concatenate_35	(Concatenate) (None, 8, 8, 234)	0	concatenate_34[0][0] conv2d_38[0][0]
batch_normalization_38	(BatchNo (None, 8, 8, 234)	936	concatenate_35[0][0]
activation_38	(Activation) (None, 8, 8, 234)	0	batch_normalization_38[0][0]
conv2d_39	(Conv2D) (None, 8, 8, 18)	4212	activation_38[0][0]
average_pooling2d_2	(AveragePoo (None, 4, 4, 18)	0	conv2d_39[0][0]
batch_normalization_39	(BatchNo (None, 4, 4, 18)	72	average_pooling2d_2[0][0]
activation_39	(Activation) (None, 4, 4, 18)	0	batch_normalization_39[0][0]
conv2d_40	(Conv2D) (None, 4, 4, 18)	2916	activation_39[0][0]
concatenate_36	(Concatenate) (None, 4, 4, 36)	0	average_pooling2d_2[0][0] conv2d_40[0][0]
batch_normalization_40	(BatchNo (None, 4, 4, 36)	144	concatenate_36[0][0]
activation_40	(Activation) (None, 4, 4, 36)	0	batch_normalization_40[0][0]
conv2d_41	(Conv2D) (None, 4, 4, 18)	5832	activation_40[0][0]
concatenate_37	(Concatenate) (None, 4, 4, 54)	0	concatenate_36[0][0] conv2d_41[0][0]
batch_normalization_41	(BatchNo (None, 4, 4, 54)	216	concatenate_37[0][0]
activation_41	(Activation) (None, 4, 4, 54)	0	batch_normalization_41[0][0]
conv2d_42	(Conv2D) (None, 4, 4, 18)	8748	activation_41[0][0]
concatenate_38	(Concatenate) (None, 4, 4, 72)	0	concatenate_37[0][0] conv2d_42[0][0]
batch_normalization_42	(BatchNo (None, 4, 4, 72)	288	concatenate_38[0][0]
activation_42	(Activation) (None, 4, 4, 72)	0	batch_normalization_42[0][0]
conv2d_43	(Conv2D) (None, 4, 4, 18)	11664	activation_42[0][0]
concatenate_39	(Concatenate) (None, 4, 4, 90)	0	concatenate_38[0][0] conv2d_43[0][0]
batch_normalization_43	(BatchNo (None, 4, 4, 90)	360	concatenate_39[0][0]
activation_43	(Activation) (None, 4, 4, 90)	0	batch_normalization_43[0][0]

conv2d_44 (Conv2D)	(None, 4, 4, 18)	14580	activation_43[0][0]
concatenate_40 (Concatenate)	(None, 4, 4, 108)	0	concatenate_39[0][0] conv2d_44[0][0]
batch_normalization_44 (BatchNo	(None, 4, 4, 108)	432	concatenate_40[0][0]
activation_44 (Activation)	(None, 4, 4, 108)	0	batch_normalization_44[0][0]
conv2d_45 (Conv2D)	(None, 4, 4, 18)	17496	activation_44[0][0]
concatenate_41 (Concatenate)	(None, 4, 4, 126)	0	concatenate_40[0][0] conv2d_45[0][0]
batch_normalization_45 (BatchNo	(None, 4, 4, 126)	504	concatenate_41[0][0]
activation_45 (Activation)	(None, 4, 4, 126)	0	batch_normalization_45[0][0]
conv2d_46 (Conv2D)	(None, 4, 4, 18)	20412	activation_45[0][0]
concatenate_42 (Concatenate)	(None, 4, 4, 144)	0	concatenate_41[0][0] conv2d_46[0][0]
batch_normalization_46 (BatchNo	(None, 4, 4, 144)	576	concatenate_42[0][0]
activation_46 (Activation)	(None, 4, 4, 144)	0	batch_normalization_46[0][0]
conv2d_47 (Conv2D)	(None, 4, 4, 18)	23328	activation_46[0][0]
concatenate_43 (Concatenate)	(None, 4, 4, 162)	0	concatenate_42[0][0] conv2d_47[0][0]
batch_normalization_47 (BatchNo	(None, 4, 4, 162)	648	concatenate_43[0][0]
activation_47 (Activation)	(None, 4, 4, 162)	0	batch_normalization_47[0][0]
conv2d_48 (Conv2D)	(None, 4, 4, 18)	26244	activation_47[0][0]
concatenate_44 (Concatenate)	(None, 4, 4, 180)	0	concatenate_43[0][0] conv2d_48[0][0]
batch_normalization_48 (BatchNo	(None, 4, 4, 180)	720	concatenate_44[0][0]
activation_48 (Activation)	(None, 4, 4, 180)	0	batch_normalization_48[0][0]
conv2d_49 (Conv2D)	(None, 4, 4, 18)	29160	activation_48[0][0]
concatenate_45 (Concatenate)	(None, 4, 4, 198)	0	concatenate_44[0][0] conv2d_49[0][0]
batch_normalization_49 (BatchNo	(None, 4, 4, 198)	792	concatenate_45[0][0]
activation_49 (Activation)	(None, 4, 4, 198)	0	batch_normalization_49[0][0]
conv2d_50 (Conv2D)	(None, 4, 4, 18)	32076	activation_49[0][0]
concatenate_46 (Concatenate)	(None, 4, 4, 216)	0	concatenate_45[0][0] conv2d_50[0][0]
batch_normalization_50 (BatchNo	(None, 4, 4, 216)	864	concatenate_46[0][0]
activation_50 (Activation)	(None, 4, 4, 216)	0	batch_normalization_50[0][0]
conv2d_51 (Conv2D)	(None, 4, 4, 18)	34992	activation_50[0][0]
concatenate_47 (Concatenate)	(None, 4, 4, 234)	0	concatenate_46[0][0] conv2d_51[0][0]
batch_normalization_51 (BatchNo	(None, 4, 4, 234)	936	concatenate_47[0][0]
activation_51 (Activation)	(None, 4, 4, 234)	0	batch_normalization_51[0][0]
average_pooling2d_3 (AveragePoo	(None, 2, 2, 234)	0	activation_51[0][0]
conv2d_52 (Conv2D)	(None, 1, 1, 10)	9370	average_pooling2d_3[0][0]
activation_52 (Activation)	(None, 1, 1, 10)	0	conv2d_52[0][0]

flatten (Flatten)	(None, 10)	0	activation_52[0][0]
-------------------	------------	---	---------------------

Total params: 997,271
Trainable params: 983,673
Non-trainable params: 13,598

In [11]:

```
# determine Loss function and Optimizer
sgd = SGD(learning_rate=0.1, momentum=0.9)
model.compile(loss='categorical_crossentropy',
              optimizer=sgd,
              metrics=['accuracy'])
```

In [24]:

```
csv_logger = CSVLogger('training6.csv', append=True)
```

In [25]:

```
early_stop = tf.keras.callbacks.EarlyStopping(
    monitor='val_accuracy', patience=3, verbose=0, mode='auto', restore_best_weights=False
)
```

In [26]:

```
model.fit_generator(train_datagen.flow(X_train, y_train, batch_size=32),
                    epochs=50,
                    steps_per_epoch = X_train.shape[0]/32,
                    verbose=1,
                    validation_data=test_datagen.flow(X_test, y_test, batch_size=32),
                    callbacks=[csv_logger])
```

```
W0321 13:13:39.592768 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']
W0321 13:13:39.609715 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']
```

Train for 1562.5 steps, validate for 313 steps

Epoch 1/50

1563/1562 [=====] - 223s 142ms/step - loss: 1.2339 - accuracy: 0.5585 - val_loss: 1.3928 - val_accuracy: 0.5294

Epoch 2/50

1563/1562 [=====] - 223s 143ms/step - loss: 1.1902 - accuracy: 0.5762 - val_loss: 1.0998 - val_accuracy: 0.6190

Epoch 3/50

1563/1562 [=====] - 223s 143ms/step - loss: 1.1482 - accuracy: 0.5916 - val_loss: 1.0794 - val_accuracy: 0.6226

Epoch 4/50

1563/1562 [=====] - 224s 143ms/step - loss: 1.1028 - accuracy: 0.6080 - val_loss: 1.1026 - val_accuracy: 0.6247

Epoch 5/50

1563/1562 [=====] - 221s 142ms/step - loss: 1.0636 - accuracy: 0.6242 - val_loss: 1.0651 - val_accuracy: 0.6269

Epoch 6/50

1563/1562 [=====] - 221s 142ms/step - loss: 1.0279 - accuracy: 0.6339 - val_loss: 1.0576 - val_accuracy: 0.6418

Epoch 7/50

1563/1562 [=====] - 221s 142ms/step - loss: 1.0009 - accuracy: 0.6462 - val_loss: 0.9596 - val_accuracy: 0.6705

Epoch 8/50

1563/1562 [=====] - 221s 142ms/step - loss: 0.9659 - accuracy: 0.6592 - val_loss: 0.9075 - val_accuracy: 0.6842

Epoch 9/50

1563/1562 [=====] - 221s 142ms/step - loss: 0.9342 - accuracy: 0.6714 - val_lo
ss: 0.9834 - val_accuracy: 0.6693
Epoch 10/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.9095 - accuracy: 0.6818 - val_lo
ss: 1.1130 - val_accuracy: 0.6370
Epoch 11/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.8812 - accuracy: 0.6890 - val_lo
ss: 0.9751 - val_accuracy: 0.6926
Epoch 12/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.8577 - accuracy: 0.7004 - val_lo
ss: 0.9365 - val_accuracy: 0.6848
Epoch 13/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.8390 - accuracy: 0.7076 - val_lo
ss: 0.8133 - val_accuracy: 0.7308
Epoch 14/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.8104 - accuracy: 0.7178 - val_lo
ss: 0.7772 - val_accuracy: 0.7426
Epoch 15/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7859 - accuracy: 0.7281 - val_lo
ss: 0.8761 - val_accuracy: 0.7176
Epoch 16/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7695 - accuracy: 0.7333 - val_lo
ss: 0.7139 - val_accuracy: 0.7589
Epoch 17/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7505 - accuracy: 0.7392 - val_lo
ss: 0.7250 - val_accuracy: 0.7568
Epoch 18/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7388 - accuracy: 0.7457 - val_lo
ss: 0.7973 - val_accuracy: 0.7427
Epoch 19/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7169 - accuracy: 0.7508 - val_lo
ss: 0.7122 - val_accuracy: 0.7549
Epoch 20/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.7081 - accuracy: 0.7551 - val_lo
ss: 0.8314 - val_accuracy: 0.7446
Epoch 21/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6883 - accuracy: 0.7622 - val_lo
ss: 0.6679 - val_accuracy: 0.7758
Epoch 22/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.6769 - accuracy: 0.7668 - val_lo
ss: 0.7407 - val_accuracy: 0.7593
Epoch 23/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6631 - accuracy: 0.7706 - val_lo
ss: 0.7185 - val_accuracy: 0.7643
Epoch 24/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6561 - accuracy: 0.7734 - val_lo
ss: 0.7943 - val_accuracy: 0.7447
Epoch 25/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6312 - accuracy: 0.7821 - val_lo
ss: 0.7531 - val_accuracy: 0.7533
Epoch 26/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6273 - accuracy: 0.7841 - val_lo
ss: 0.6028 - val_accuracy: 0.7945
Epoch 27/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.6140 - accuracy: 0.7868 - val_lo
ss: 0.7844 - val_accuracy: 0.7593
Epoch 28/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5999 - accuracy: 0.7914 - val_lo
ss: 0.7311 - val_accuracy: 0.7705
Epoch 29/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5935 - accuracy: 0.7942 - val_lo
ss: 0.5647 - val_accuracy: 0.8081
Epoch 30/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5832 - accuracy: 0.7987 - val_lo
ss: 0.9300 - val_accuracy: 0.7285
Epoch 31/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.5696 - accuracy: 0.8012 - val_lo
ss: 0.5801 - val_accuracy: 0.8047
Epoch 32/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5581 - accuracy: 0.8074 - val_lo
ss: 0.7149 - val_accuracy: 0.7788
Epoch 33/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.5520 - accuracy: 0.8084 - val_lo
ss: 0.5928 - val_accuracy: 0.8071
Epoch 34/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.5424 - accuracy: 0.8128 - val_lo
ss: 0.6979 - val accuracy: 0.7879

```

Epoch 35/50
1563/1562 [=====] - 224s 143ms/step - loss: 0.5411 - accuracy: 0.8122 - val_loss: 0.5513 - val_accuracy: 0.8158
Epoch 36/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.5250 - accuracy: 0.8183 - val_loss: 0.5941 - val_accuracy: 0.8064
Epoch 37/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5167 - accuracy: 0.8216 - val_loss: 0.7703 - val_accuracy: 0.7635
Epoch 38/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.5151 - accuracy: 0.8215 - val_loss: 0.5298 - val_accuracy: 0.8195
Epoch 39/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.5072 - accuracy: 0.8253 - val_loss: 0.6229 - val_accuracy: 0.7958
Epoch 40/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.5005 - accuracy: 0.8263 - val_loss: 0.5387 - val_accuracy: 0.8229
Epoch 41/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.4938 - accuracy: 0.8285 - val_loss: 0.4456 - val_accuracy: 0.8493
Epoch 42/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.4848 - accuracy: 0.8328 - val_loss: 0.4997 - val_accuracy: 0.8373
Epoch 43/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.4817 - accuracy: 0.8322 - val_loss: 0.6552 - val_accuracy: 0.7951
Epoch 44/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.4672 - accuracy: 0.8378 - val_loss: 0.4859 - val_accuracy: 0.8408
Epoch 45/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.4649 - accuracy: 0.8392 - val_loss: 0.5344 - val_accuracy: 0.8315
Epoch 46/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.4561 - accuracy: 0.8418 - val_loss: 0.5993 - val_accuracy: 0.8064
Epoch 47/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.4516 - accuracy: 0.8438 - val_loss: 0.5377 - val_accuracy: 0.8313
Epoch 48/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.4480 - accuracy: 0.8450 - val_loss: 0.5339 - val_accuracy: 0.8337
Epoch 49/50
1563/1562 [=====] - 223s 143ms/step - loss: 0.4409 - accuracy: 0.8468 - val_loss: 0.5507 - val_accuracy: 0.8249
Epoch 50/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.4415 - accuracy: 0.8463 - val_loss: 0.5689 - val_accuracy: 0.8188

```

Out[26]:

```
<tensorflow.python.keras.callbacks.History at 0x18181c99160>
```

Epoch: 50 done

In [27]:

```

model.fit_generator(train_datagen.flow(X_train, y_train, batch_size=32),
                    epochs=50,
                    steps_per_epoch = X_train.shape[0]/32,
                    verbose=1,
                    validation_data=test_datagen.flow(X_test, y_test, batch_size=32),
                    callbacks=[csv_logger])

```

```

W0321 16:23:05.149328 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']
W0321 16:23:05.166312 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']

```

Train for 1562.5 steps, validate for 313 steps

Epoch 1/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.4463 - accuracy: 0.8456 - val_loss: 0.5936 - val_accuracy: 0.8172

Epoch 2/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.4429 - accuracy: 0.8458 - val_loss: 0.4653 - val_accuracy: 0.8468

Epoch 3/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4376 - accuracy: 0.8502 - val_loss: 0.5645 - val_accuracy: 0.8268

Epoch 4/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.4248 - accuracy: 0.8526 - val_loss: 0.4453 - val_accuracy: 0.8550

Epoch 5/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4256 - accuracy: 0.8530 - val_loss: 0.5586 - val_accuracy: 0.8282

Epoch 6/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4216 - accuracy: 0.8544 - val_loss: 0.4700 - val_accuracy: 0.8488

Epoch 7/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4130 - accuracy: 0.8577 - val_loss: 0.5473 - val_accuracy: 0.8234

Epoch 8/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4074 - accuracy: 0.8586 - val_loss: 0.8300 - val_accuracy: 0.7817

Epoch 9/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.4042 - accuracy: 0.8607 - val_loss: 0.4540 - val_accuracy: 0.8574

Epoch 10/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.3978 - accuracy: 0.8646 - val_loss: 0.4242 - val_accuracy: 0.8652

Epoch 11/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.3933 - accuracy: 0.8653 - val_loss: 0.5467 - val_accuracy: 0.8394

Epoch 12/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.3899 - accuracy: 0.8647 - val_loss: 0.5973 - val_accuracy: 0.8250

Epoch 13/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.3871 - accuracy: 0.8639 - val_loss: 0.4817 - val_accuracy: 0.8457

Epoch 14/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.3842 - accuracy: 0.8672 - val_loss: 0.4311 - val_accuracy: 0.8619

Epoch 15/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.3787 - accuracy: 0.8703 - val_loss: 0.4190 - val_accuracy: 0.8667

Epoch 16/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.3755 - accuracy: 0.8713 - val_loss: 0.6326 - val_accuracy: 0.8199

Epoch 17/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.3729 - accuracy: 0.8706 - val_loss: 0.4052 - val_accuracy: 0.8658

Epoch 18/50

1563/1562 [=====] - 224s 143ms/step - loss: 0.3646 - accuracy: 0.8748 - val_loss: 0.5043 - val_accuracy: 0.8438

Epoch 19/50

1563/1562 [=====] - 223s 143ms/step - loss: 0.3678 - accuracy: 0.8727 - val_loss: 0.5111 - val_accuracy: 0.8492

Epoch 20/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3594 - accuracy: 0.8759 - val_loss: 0.4370 - val_accuracy: 0.8651

Epoch 21/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3590 - accuracy: 0.8757 - val_loss: 0.4262 - val_accuracy: 0.8652

Epoch 22/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3516 - accuracy: 0.8774 - val_loss: 0.4380 - val_accuracy: 0.8680

Epoch 23/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3508 - accuracy: 0.8792 - val_loss: 0.4342 - val_accuracy: 0.8665

Epoch 24/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3487 - accuracy: 0.8794 - val_loss: 0.4061 - val_accuracy: 0.8707

Epoch 25/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.3459 - accuracy: 0.8805 - val_loss: 0.4344 - val_accuracy: 0.8708

Epoch 26/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3431 - accuracy: 0.8802 - val_loss: 0.3738 - val_accuracy: 0.8767
Epoch 27/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3339 - accuracy: 0.8836 - val_loss: 0.4750 - val_accuracy: 0.8541
Epoch 28/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3387 - accuracy: 0.8820 - val_loss: 0.4335 - val_accuracy: 0.8626
Epoch 29/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3289 - accuracy: 0.8863 - val_loss: 0.5189 - val_accuracy: 0.8440
Epoch 30/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3284 - accuracy: 0.8867 - val_loss: 0.4083 - val_accuracy: 0.8726
Epoch 31/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3252 - accuracy: 0.8862 - val_loss: 0.4942 - val_accuracy: 0.8559
Epoch 32/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3224 - accuracy: 0.8880 - val_loss: 0.4219 - val_accuracy: 0.8714
Epoch 33/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3205 - accuracy: 0.8903 - val_loss: 0.4560 - val_accuracy: 0.8683
Epoch 34/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3165 - accuracy: 0.8908 - val_loss: 0.5106 - val_accuracy: 0.8424
Epoch 35/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3238 - accuracy: 0.8886 - val_loss: 0.4943 - val_accuracy: 0.8617
Epoch 36/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3153 - accuracy: 0.8911 - val_loss: 0.4072 - val_accuracy: 0.8748
Epoch 37/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3155 - accuracy: 0.8896 - val_loss: 0.3964 - val_accuracy: 0.8721
Epoch 38/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3138 - accuracy: 0.8922 - val_loss: 0.4638 - val_accuracy: 0.8659
Epoch 39/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3103 - accuracy: 0.8933 - val_loss: 0.3942 - val_accuracy: 0.8738
Epoch 40/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3084 - accuracy: 0.8924 - val_loss: 0.4204 - val_accuracy: 0.8745
Epoch 41/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3022 - accuracy: 0.8958 - val_loss: 0.4075 - val_accuracy: 0.8733
Epoch 42/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3023 - accuracy: 0.8965 - val_loss: 0.3931 - val_accuracy: 0.8693
Epoch 43/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2950 - accuracy: 0.8971 - val_loss: 0.5113 - val_accuracy: 0.8585
Epoch 44/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.3038 - accuracy: 0.8936 - val_loss: 0.4713 - val_accuracy: 0.8678
Epoch 45/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2916 - accuracy: 0.8974 - val_loss: 0.3931 - val_accuracy: 0.8784
Epoch 46/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2923 - accuracy: 0.8990 - val_loss: 0.3997 - val_accuracy: 0.8799
Epoch 47/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2885 - accuracy: 0.8999 - val_loss: 0.3992 - val_accuracy: 0.8768
Epoch 48/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2906 - accuracy: 0.8989 - val_loss: 0.3701 - val_accuracy: 0.8855
Epoch 49/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2889 - accuracy: 0.8988 - val_loss: 0.3935 - val_accuracy: 0.8857
Epoch 50/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2843 - accuracy: 0.9010 - val_loss: 0.4761 - val_accuracy: 0.8691

Out[27]:

<tensorflow.python.keras.callbacks.History at 0x18181cd5160>

Epoch: 100 done

In [28]:

```
model.save_weights('Denset_Checkpt1.h5')
```

In [29]:

```
model.fit_generator(train_datagen.flow(X_train, y_train, batch_size=32),
                    epochs=50,
                    steps_per_epoch = X_train.shape[0]/32,
                    verbose=1,
                    validation_data=test_datagen.flow(X_test, y_test, batch_size=32),
                    callbacks=[csv_logger])
```

W0321 19:28:33.631057 9596 data_adapter.py:1091] sample_weight modes were coerced from

```
...
to
['...']
```

W0321 19:28:33.649005 9596 data_adapter.py:1091] sample_weight modes were coerced from

```
...
to
['...']
```

Train for 1562.5 steps, validate for 313 steps

Epoch 1/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2992 - accuracy: 0.8974 - val_loss: 0.3389 - val_accuracy: 0.8907

Epoch 2/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2966 - accuracy: 0.8983 - val_loss: 0.4106 - val_accuracy: 0.8785

Epoch 3/50

1563/1562 [=====] - 223s 142ms/step - loss: 0.2931 - accuracy: 0.8985 - val_loss: 0.3354 - val_accuracy: 0.8920

Epoch 4/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2946 - accuracy: 0.8981 - val_loss: 0.4036 - val_accuracy: 0.8782

Epoch 5/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2886 - accuracy: 0.9002 - val_loss: 0.3798 - val_accuracy: 0.8878

Epoch 6/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2875 - accuracy: 0.9010 - val_loss: 0.3944 - val_accuracy: 0.8784

Epoch 7/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2782 - accuracy: 0.9032 - val_loss: 0.3948 - val_accuracy: 0.8856

Epoch 8/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2779 - accuracy: 0.9030 - val_loss: 0.3621 - val_accuracy: 0.8852

Epoch 9/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2782 - accuracy: 0.9035 - val_loss: 0.3893 - val_accuracy: 0.8803

Epoch 10/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2739 - accuracy: 0.9040 - val_loss: 0.3843 - val_accuracy: 0.8840

Epoch 11/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2745 - accuracy: 0.9039 - val_loss: 0.3379 - val_accuracy: 0.8967

Epoch 12/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2664 - accuracy: 0.9083 - val_loss: 0.3664 - val_accuracy: 0.8827

Epoch 13/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2722 - accuracy: 0.9058 - val_loss: 0.3468 - val_accuracy: 0.8927

Epoch 14/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2631 - accuracy: 0.9098 - val_loss: 0.4081 - val_accuracy: 0.8746

Epoch 15/50

1563/1562 [=====] - 222s 142ms/step - loss: 0.2652 - accuracy: 0.9075 - val_loss: 0.4081 - val_accuracy: 0.8746

```
1563/1562 [=====] - 222s 142ms/step - loss: 0.2653 - accuracy: 0.9075 - val_lo
ss: 0.3440 - val_accuracy: 0.8878
Epoch 16/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2663 - accuracy: 0.9074 - val_lo
ss: 0.3715 - val_accuracy: 0.8897
Epoch 17/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2635 - accuracy: 0.9088 - val_lo
ss: 0.3515 - val_accuracy: 0.8916
Epoch 18/50
1563/1562 [=====] - 224s 143ms/step - loss: 0.2638 - accuracy: 0.9088 - val_lo
ss: 0.4148 - val_accuracy: 0.8711
Epoch 19/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2603 - accuracy: 0.9094 - val_lo
ss: 0.4037 - val_accuracy: 0.8812
Epoch 20/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2570 - accuracy: 0.9097 - val_lo
ss: 0.3574 - val_accuracy: 0.8919
Epoch 21/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2590 - accuracy: 0.9097 - val_lo
ss: 0.3898 - val_accuracy: 0.8819
Epoch 22/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2578 - accuracy: 0.9106 - val_lo
ss: 0.4071 - val_accuracy: 0.8868
Epoch 23/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2487 - accuracy: 0.9124 - val_lo
ss: 0.3912 - val_accuracy: 0.8819
Epoch 24/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2496 - accuracy: 0.9133 - val_lo
ss: 0.4227 - val_accuracy: 0.8750
Epoch 25/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2508 - accuracy: 0.9127 - val_lo
ss: 0.3445 - val_accuracy: 0.8932
Epoch 26/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2495 - accuracy: 0.9123 - val_lo
ss: 0.4402 - val_accuracy: 0.8742
Epoch 27/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2480 - accuracy: 0.9132 - val_lo
ss: 0.4640 - val_accuracy: 0.8731
Epoch 28/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2448 - accuracy: 0.9141 - val_lo
ss: 0.4153 - val_accuracy: 0.8811
Epoch 29/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2493 - accuracy: 0.9118 - val_lo
ss: 0.3398 - val_accuracy: 0.8981
Epoch 30/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2442 - accuracy: 0.9133 - val_lo
ss: 0.3875 - val_accuracy: 0.8858
Epoch 31/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2426 - accuracy: 0.9152 - val_lo
ss: 0.3650 - val_accuracy: 0.8931
Epoch 32/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2389 - accuracy: 0.9178 - val_lo
ss: 0.3438 - val_accuracy: 0.8968
Epoch 33/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2386 - accuracy: 0.9179 - val_lo
ss: 0.3871 - val_accuracy: 0.8890
Epoch 34/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2398 - accuracy: 0.9163 - val_lo
ss: 0.3298 - val_accuracy: 0.8950
Epoch 35/50
1563/1562 [=====] - 221s 141ms/step - loss: 0.2409 - accuracy: 0.9170 - val_lo
ss: 0.3384 - val_accuracy: 0.9005
Epoch 36/50
1563/1562 [=====] - 221s 142ms/step - loss: 0.2410 - accuracy: 0.9159 - val_lo
ss: 0.3076 - val_accuracy: 0.9045
Epoch 37/50
1563/1562 [=====] - 221s 142ms/step - loss: 0.2302 - accuracy: 0.9204 - val_lo
ss: 0.5778 - val_accuracy: 0.8503
Epoch 38/50
1563/1562 [=====] - 221s 141ms/step - loss: 0.2288 - accuracy: 0.9201 - val_lo
ss: 0.4000 - val_accuracy: 0.8858
Epoch 39/50
1563/1562 [=====] - 221s 142ms/step - loss: 0.2318 - accuracy: 0.9201 - val_lo
ss: 0.3146 - val_accuracy: 0.9014
Epoch 40/50
1563/1562 [=====] - 221s 142ms/step - loss: 0.2277 - accuracy: 0.9213 - val_lo
ss: 0.3881 - val_accuracy: 0.8876
Epoch 41/50
```



```

Epoch 41/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2315 - accuracy: 0.9188 - val_loss: 0.3346 - val_accuracy: 0.9024
Epoch 42/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2305 - accuracy: 0.9190 - val_loss: 0.3295 - val_accuracy: 0.9002
Epoch 43/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2267 - accuracy: 0.9212 - val_loss: 0.3730 - val_accuracy: 0.8894
Epoch 44/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2300 - accuracy: 0.9199 - val_loss: 0.3888 - val_accuracy: 0.8871
Epoch 45/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2263 - accuracy: 0.9205 - val_loss: 0.3949 - val_accuracy: 0.8831
Epoch 46/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2247 - accuracy: 0.9214 - val_loss: 0.3954 - val_accuracy: 0.8842
Epoch 47/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2242 - accuracy: 0.9217 - val_loss: 0.3367 - val_accuracy: 0.8984
Epoch 48/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2192 - accuracy: 0.9215 - val_loss: 0.3432 - val_accuracy: 0.8938
Epoch 49/50
1563/1562 [=====] - 222s 142ms/step - loss: 0.2170 - accuracy: 0.9231 - val_loss: 0.3952 - val_accuracy: 0.8843
Epoch 50/50
1563/1562 [=====] - 223s 142ms/step - loss: 0.2180 - accuracy: 0.9247 - val_loss: 0.3744 - val_accuracy: 0.8871

```

Out[29]:

<tensorflow.python.keras.callbacks.History at 0x18181d19390>

Epoch: 150 done

In [30]:

```
tf.keras.backend.set_value(model.optimizer.lr, 0.01)
```

In [31]:

```

model.fit_generator(train_datagen.flow(X_train, y_train, batch_size=32),
                    epochs=5,
                    steps_per_epoch = X_train.shape[0]/32,
                    verbose=1,
                    validation_data=test_datagen.flow(X_test, y_test, batch_size=32),
                    callbacks=[csv_logger])

```

```

W0321 22:33:33.005362 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']
W0321 22:33:33.024336 9596 data_adapter.py:1091] sample_weight modes were coerced from
...
to
['...']

```

Train for 1562.5 steps, validate for 313 steps

```

Epoch 1/5
1563/1562 [=====] - 225s 144ms/step - loss: 0.1958 - accuracy: 0.9319 - val_loss: 0.2911 - val_accuracy: 0.9109
Epoch 2/5
1563/1562 [=====] - 226s 144ms/step - loss: 0.1761 - accuracy: 0.9385 - val_loss: 0.2952 - val_accuracy: 0.9091
Epoch 3/5
1563/1562 [=====] - 225s 144ms/step - loss: 0.1699 - accuracy: 0.9409 - val_loss: 0.2909 - val_accuracy: 0.9120
Epoch 4/5
1563/1562 [=====] - 222s 142ms/step - loss: 0.1695 - accuracy: 0.9408 - val_loss: 0.2977 - val_accuracy: 0.9114

```

```
ss: 0.2951 - val_accuracy: 0.9124
```

Epoch 5/5

```
1563/1562 [=====] - 222s 142ms/step - loss: 0.1654 - accuracy: 0.9423 - val_loss: 0.2951 - val_accuracy: 0.9124
```

Out[31]:

```
<tensorflow.python.keras.callbacks.History at 0x18185e980f0>
```

Epoch: 155 done

Stopped trained epoch here to avoid overfitting

Achieved above 90% test accuracy as per assignment task

In [32]:

```
model.save_weights('Denset_Checkpt2.h5')
```

In []:

In [33]:

```
# Test the model
score = model.evaluate(X_test/255, y_test, verbose=1)
```

```
10000/10000 [=====] - 11s 1ms/sample - loss: 0.2955 - accuracy: 0.9124
19s - loss: 0.2592 - accuracy: 0. - ETA: 16s - loss: 0.2564 - accuracy - ETA: 8s - loss: 0.3054 - accuracy: 0. - ETA: 8s - loss: 0.3025 - accuracy - ETA: 7s - loss: 0.3097 - accuracy: 0. - ETA: 7s - loss: 0.3051 - - ETA: 6s - - ETA: 5s - loss: - ETA: 4s - - ETA: 2s - loss: - ETA: 1s - - ETA: 0s - loss: 0.2999 - accuracy - ETA: 0s - loss: 0.2976 - accuracy: - ETA: 0s - loss: 0.2959 - accuracy: 0.
```

In [34]:

```
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.29551431020498276
Test accuracy: 0.9124
```

In []: