sachmeer4u@gmail.com_25

July 1, 2019

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, Ac
# from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
# from keras.layers import Concatenate
# from keras.optimizers import Adam

!pip install -q keras

import keras
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, RMSprop
import tensorflow as tf
```

```
Using TensorFlow backend.
```

In [0]:
```
#!pip install tensorflow-gpu==1.5.0
```

In [0]:
```
#import tensorflow as tf
#tf.__version__
```

In [0]:
```
# this part will prevent tensorflow to allocate all the avaliable GPU Memory
# backend

from tensorflow import keras

from keras import backend as k

# Don't pre-allocate memory; allocate as-needed
# import tensorflow as tf

#tf.config.gpu.set_per_process_memory_fraction(0.75)
#tf.config.gpu.set_per_process_memory_growth(True)
```

```
        config = tf.ConfigProto()
        config.gpu_options.allow_growth = True

        # Create a session with the above options specified.
        # k.tensorflow_backend.set_session(tf.Session(config=config))
```

In [0]:
```
        # Hyperparameters


        #batch_size = 128
        batch_size = 64
        num_classes = 10

        epochs = 35

        l = 6

        num_filter = 35

        compression = 1
        dropout_rate = 0.3
```

In [5]:
```
        # 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 encoing
        y_train = tf.keras.utils.to_categorical(y_train, num_classes)
        y_test = tf.keras.utils.to_categorical(y_test, num_classes)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 11s 0us/step
```

In [0]:
```
        import numpy as np
        from sklearn.model_selection import train_test_split

        def preprocess_data(data_set):
            mean = np.array([125.3, 123.0, 113.9])
            std = np.array([63.0, 62.1, 66.7])

            data_set -= mean
            data_set /= std
            return data_set

        X_train = X_train.astype('float32')
        X_test = X_test.astype('float32')
        X_train = preprocess_data(X_train)
```

```
X_test = preprocess_data(X_test)


X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size = 0.10)
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator
```

1 Using IMage Augmentation

In [0]:
```
#datagen = ImageDataGenerator(
#    featurewise_center=True,
#    featurewise_std_normalization=True,
#    rotation_range=20,
#    width_shift_range=0.1,
#    height_shift_range=0.1,
#    horizontal_flip=True)

#datagen.fit(X_train)


datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.125,
    height_shift_range=0.125,
    horizontal_flip=True,
    fill_mode='nearest',
    zoom_range=0.10
)

datagen.fit(X_train)
```

In [9]:
```
X_train.shape
```

Out[9]:
```
(45000, 32, 32, 3)
```

In [10]:
```
X_test.shape
```

Out[10]:
```
(10000, 32, 32, 3)
```

In [0]:
```
# Dense Block
def denseblock(input, num_filter = 12, dropout_rate = 0.2):
    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
        if dropout_rate>0:
```

```
                Conv2D_3_3 = layers.Dropout(dropout_rate)(Conv2D_3_3)
            concat = layers.Concatenate(axis=-1)([temp,Conv2D_3_3])

            temp = concat

        return temp

    ## transition Blosck
    def transition(input, num_filter = 12, dropout_rate = 0.2):
        global compression
        BatchNorm = layers.BatchNormalization()(input)
        relu = layers.Activation('relu')(BatchNorm)
        Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (1,1), use_bias=Fals
        if dropout_rate>0:
             Conv2D_BottleNeck = layers.Dropout(dropout_rate)(Conv2D_BottleNeck)
        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)
        flat = layers.Flatten()(AvgPooling)
        output = layers.Dense(num_classes, activation='softmax')(flat)
        return output
```

2 Chaging the architecture

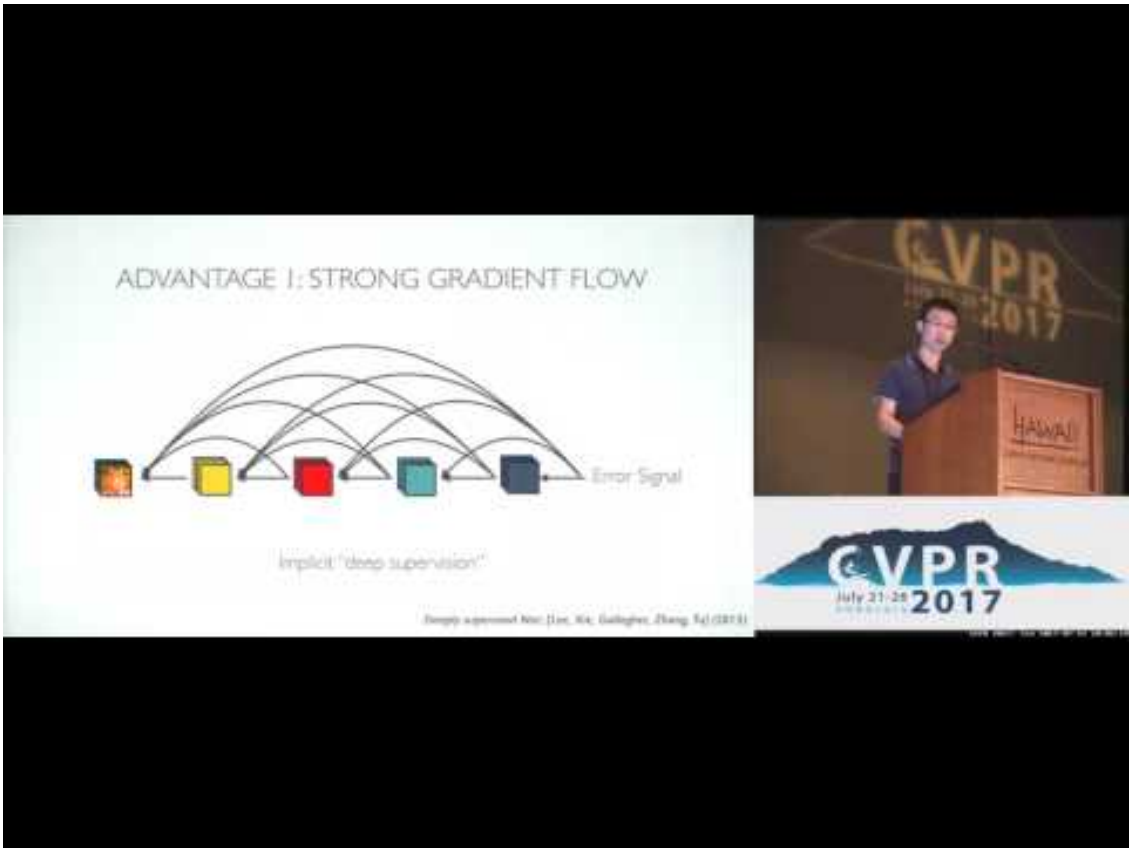
In [12]:
```
#num_filter = 12
#dropout_rate = 0.2

#l = 12

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, dropout_rate)
First_Transition = transition(First_Block, num_filter, dropout_rate)

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

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

```
        #Fourth_Block = denseblock(Third_Transition, num_filter, dropout_rate)
        #Fourth_Transition = transition(Fourth_Block, num_filter, dropout_rate)

        Last_Block = denseblock(Third_Transition,  num_filter, dropout_rate)


        output = output_layer(Last_Block)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0628 04:38:55.109632 140063208904576 deprecation.py:506] From /usr/local/lib/python3.6/dist-pa
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
```

In [0]:
```
#https://arxiv.org/pdf/1608.06993.pdf
from IPython.display import IFrame, YouTubeVideo
YouTubeVideo(id='-W6y8xnd--U', width=600)
```

Out[0]:



```
In [13]: 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, 35)	945	input_1[0][0]
batch_normalization (BatchNorma	(None, 32, 32, 35)	140	conv2d[0][0]
activation (Activation)	(None, 32, 32, 35)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 32, 32, 35)	11025	activation[0][0]
dropout (Dropout)	(None, 32, 32, 35)	0	conv2d_1[0][0]
concatenate (Concatenate)	(None, 32, 32, 70)	0	conv2d[0][0] dropout[0][0]
batch_normalization_1 (BatchNor	(None, 32, 32, 70)	280	concatenate[0][0]
activation_1 (Activation)	(None, 32, 32, 70)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 32, 32, 35)	22050	activation_1[0][0]
dropout_1 (Dropout)	(None, 32, 32, 35)	0	conv2d_2[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 105)	0	concatenate[0][0] dropout_1[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 105)	420	concatenate_1[0][0]
activation_2 (Activation)	(None, 32, 32, 105)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 32, 32, 35)	33075	activation_2[0][0]
dropout_2 (Dropout)	(None, 32, 32, 35)	0	conv2d_3[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 140)	0	concatenate_1[0][0] dropout_2[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 140)	560	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 140)	0	batch_normalization_3[0][0]

```
conv2d_4 (Conv2D)               (None, 32, 32, 35)   44100       activation_3[0][0]
____________________________________________________________________________________________
dropout_3 (Dropout)             (None, 32, 32, 35)   0           conv2d_4[0][0]
____________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 32, 32, 175)  0           concatenate_2[0][0]
                                                                 dropout_3[0][0]
____________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 32, 32, 175)  700         concatenate_3[0][0]
____________________________________________________________________________________________
activation_4 (Activation)       (None, 32, 32, 175)  0           batch_normalization_4[0][0]
____________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, 32, 32, 35)   55125       activation_4[0][0]
____________________________________________________________________________________________
dropout_4 (Dropout)             (None, 32, 32, 35)   0           conv2d_5[0][0]
____________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 32, 32, 210)  0           concatenate_3[0][0]
                                                                 dropout_4[0][0]
____________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 32, 32, 210)  840         concatenate_4[0][0]
____________________________________________________________________________________________
activation_5 (Activation)       (None, 32, 32, 210)  0           batch_normalization_5[0][0]
____________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, 32, 32, 35)   66150       activation_5[0][0]
____________________________________________________________________________________________
dropout_5 (Dropout)             (None, 32, 32, 35)   0           conv2d_6[0][0]
____________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 32, 32, 245)  0           concatenate_4[0][0]
                                                                 dropout_5[0][0]
____________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 32, 32, 245)  980         concatenate_5[0][0]
____________________________________________________________________________________________
activation_6 (Activation)       (None, 32, 32, 245)  0           batch_normalization_6[0][0]
____________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, 32, 32, 35)   8575        activation_6[0][0]
____________________________________________________________________________________________
dropout_6 (Dropout)             (None, 32, 32, 35)   0           conv2d_7[0][0]
____________________________________________________________________________________________
average_pooling2d (AveragePooli (None, 16, 16, 35)   0           dropout_6[0][0]
____________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, 16, 16, 35)   140         average_pooling2d[0][0]
____________________________________________________________________________________________
activation_7 (Activation)       (None, 16, 16, 35)   0           batch_normalization_7[0][0]
____________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 16, 16, 35)   11025       activation_7[0][0]
____________________________________________________________________________________________
dropout_7 (Dropout)             (None, 16, 16, 35)   0           conv2d_8[0][0]
____________________________________________________________________________________________
concatenate_6 (Concatenate)     (None, 16, 16, 70)   0           average_pooling2d[0][0]
```

```
                                                                 dropout_7[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, 16, 16, 70)   280         concatenate_6[0][0]
__________________________________________________________________________________________________
activation_8 (Activation)       (None, 16, 16, 70)   0           batch_normalization_8[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, 16, 16, 35)   22050       activation_8[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 16, 16, 35)   0           conv2d_9[0][0]
__________________________________________________________________________________________________
concatenate_7 (Concatenate)     (None, 16, 16, 105)  0           concatenate_6[0][0]
                                                                 dropout_8[0][0]
__________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None, 16, 16, 105)  420         concatenate_7[0][0]
__________________________________________________________________________________________________
activation_9 (Activation)       (None, 16, 16, 105)  0           batch_normalization_9[0][0]
__________________________________________________________________________________________________
conv2d_10 (Conv2D)              (None, 16, 16, 35)   33075       activation_9[0][0]
__________________________________________________________________________________________________
dropout_9 (Dropout)             (None, 16, 16, 35)   0           conv2d_10[0][0]
__________________________________________________________________________________________________
concatenate_8 (Concatenate)     (None, 16, 16, 140)  0           concatenate_7[0][0]
                                                                 dropout_9[0][0]
__________________________________________________________________________________________________
batch_normalization_10 (BatchNo (None, 16, 16, 140)  560         concatenate_8[0][0]
__________________________________________________________________________________________________
activation_10 (Activation)      (None, 16, 16, 140)  0           batch_normalization_10[0][0]
__________________________________________________________________________________________________
conv2d_11 (Conv2D)              (None, 16, 16, 35)   44100       activation_10[0][0]
__________________________________________________________________________________________________
dropout_10 (Dropout)            (None, 16, 16, 35)   0           conv2d_11[0][0]
__________________________________________________________________________________________________
concatenate_9 (Concatenate)     (None, 16, 16, 175)  0           concatenate_8[0][0]
                                                                 dropout_10[0][0]
__________________________________________________________________________________________________
batch_normalization_11 (BatchNo (None, 16, 16, 175)  700         concatenate_9[0][0]
__________________________________________________________________________________________________
activation_11 (Activation)      (None, 16, 16, 175)  0           batch_normalization_11[0][0]
__________________________________________________________________________________________________
conv2d_12 (Conv2D)              (None, 16, 16, 35)   55125       activation_11[0][0]
__________________________________________________________________________________________________
dropout_11 (Dropout)            (None, 16, 16, 35)   0           conv2d_12[0][0]
__________________________________________________________________________________________________
concatenate_10 (Concatenate)    (None, 16, 16, 210)  0           concatenate_9[0][0]
                                                                 dropout_11[0][0]
__________________________________________________________________________________________________
batch_normalization_12 (BatchNo (None, 16, 16, 210)  840         concatenate_10[0][0]
__________________________________________________________________________________________________
```

```
activation_12 (Activation)      (None, 16, 16, 210)  0           batch_normalization_12[0][0]
____________________________________________________________________________________________
conv2d_13 (Conv2D)              (None, 16, 16, 35)   66150       activation_12[0][0]
____________________________________________________________________________________________
dropout_12 (Dropout)            (None, 16, 16, 35)   0           conv2d_13[0][0]
____________________________________________________________________________________________
concatenate_11 (Concatenate)    (None, 16, 16, 245)  0           concatenate_10[0][0]
                                                                 dropout_12[0][0]
____________________________________________________________________________________________
batch_normalization_13 (BatchNo (None, 16, 16, 245)  980         concatenate_11[0][0]
____________________________________________________________________________________________
activation_13 (Activation)      (None, 16, 16, 245)  0           batch_normalization_13[0][0]
____________________________________________________________________________________________
conv2d_14 (Conv2D)              (None, 16, 16, 35)   8575        activation_13[0][0]
____________________________________________________________________________________________
dropout_13 (Dropout)            (None, 16, 16, 35)   0           conv2d_14[0][0]
____________________________________________________________________________________________
average_pooling2d_1 (AveragePoo (None, 8, 8, 35)     0           dropout_13[0][0]
____________________________________________________________________________________________
batch_normalization_14 (BatchNo (None, 8, 8, 35)     140         average_pooling2d_1[0][0]
____________________________________________________________________________________________
activation_14 (Activation)      (None, 8, 8, 35)     0           batch_normalization_14[0][0]
____________________________________________________________________________________________
conv2d_15 (Conv2D)              (None, 8, 8, 35)     11025       activation_14[0][0]
____________________________________________________________________________________________
dropout_14 (Dropout)            (None, 8, 8, 35)     0           conv2d_15[0][0]
____________________________________________________________________________________________
concatenate_12 (Concatenate)    (None, 8, 8, 70)     0           average_pooling2d_1[0][0]
                                                                 dropout_14[0][0]
____________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, 8, 8, 70)     280         concatenate_12[0][0]
____________________________________________________________________________________________
activation_15 (Activation)      (None, 8, 8, 70)     0           batch_normalization_15[0][0]
____________________________________________________________________________________________
conv2d_16 (Conv2D)              (None, 8, 8, 35)     22050       activation_15[0][0]
____________________________________________________________________________________________
dropout_15 (Dropout)            (None, 8, 8, 35)     0           conv2d_16[0][0]
____________________________________________________________________________________________
concatenate_13 (Concatenate)    (None, 8, 8, 105)    0           concatenate_12[0][0]
                                                                 dropout_15[0][0]
____________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 8, 8, 105)    420         concatenate_13[0][0]
____________________________________________________________________________________________
activation_16 (Activation)      (None, 8, 8, 105)    0           batch_normalization_16[0][0]
____________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 8, 8, 35)     33075       activation_16[0][0]
____________________________________________________________________________________________
dropout_16 (Dropout)            (None, 8, 8, 35)     0           conv2d_17[0][0]
```

```
concatenate_14 (Concatenate)    (None, 8, 8, 140)    0           concatenate_13[0][0]
                                                                 dropout_16[0][0]
batch_normalization_17 (BatchNo (None, 8, 8, 140)    560         concatenate_14[0][0]
activation_17 (Activation)      (None, 8, 8, 140)    0           batch_normalization_17[0][0]
conv2d_18 (Conv2D)              (None, 8, 8, 35)     44100       activation_17[0][0]
dropout_17 (Dropout)            (None, 8, 8, 35)     0           conv2d_18[0][0]
concatenate_15 (Concatenate)    (None, 8, 8, 175)    0           concatenate_14[0][0]
                                                                 dropout_17[0][0]
batch_normalization_18 (BatchNo (None, 8, 8, 175)    700         concatenate_15[0][0]
activation_18 (Activation)      (None, 8, 8, 175)    0           batch_normalization_18[0][0]
conv2d_19 (Conv2D)              (None, 8, 8, 35)     55125       activation_18[0][0]
dropout_18 (Dropout)            (None, 8, 8, 35)     0           conv2d_19[0][0]
concatenate_16 (Concatenate)    (None, 8, 8, 210)    0           concatenate_15[0][0]
                                                                 dropout_18[0][0]
batch_normalization_19 (BatchNo (None, 8, 8, 210)    840         concatenate_16[0][0]
activation_19 (Activation)      (None, 8, 8, 210)    0           batch_normalization_19[0][0]
conv2d_20 (Conv2D)              (None, 8, 8, 35)     66150       activation_19[0][0]
dropout_19 (Dropout)            (None, 8, 8, 35)     0           conv2d_20[0][0]
concatenate_17 (Concatenate)    (None, 8, 8, 245)    0           concatenate_16[0][0]
                                                                 dropout_19[0][0]
batch_normalization_20 (BatchNo (None, 8, 8, 245)    980         concatenate_17[0][0]
activation_20 (Activation)      (None, 8, 8, 245)    0           batch_normalization_20[0][0]
conv2d_21 (Conv2D)              (None, 8, 8, 35)     8575        activation_20[0][0]
dropout_20 (Dropout)            (None, 8, 8, 35)     0           conv2d_21[0][0]
average_pooling2d_2 (AveragePoo (None, 4, 4, 35)     0           dropout_20[0][0]
batch_normalization_21 (BatchNo (None, 4, 4, 35)     140         average_pooling2d_2[0][0]
```

activation_21 (Activation)	(None, 4, 4, 35)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 4, 4, 35)	11025	activation_21[0][0]
dropout_21 (Dropout)	(None, 4, 4, 35)	0	conv2d_22[0][0]
concatenate_18 (Concatenate)	(None, 4, 4, 70)	0	average_pooling2d_2[0][0] dropout_21[0][0]
batch_normalization_22 (BatchNo	(None, 4, 4, 70)	280	concatenate_18[0][0]
activation_22 (Activation)	(None, 4, 4, 70)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 4, 4, 35)	22050	activation_22[0][0]
dropout_22 (Dropout)	(None, 4, 4, 35)	0	conv2d_23[0][0]
concatenate_19 (Concatenate)	(None, 4, 4, 105)	0	concatenate_18[0][0] dropout_22[0][0]
batch_normalization_23 (BatchNo	(None, 4, 4, 105)	420	concatenate_19[0][0]
activation_23 (Activation)	(None, 4, 4, 105)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 4, 4, 35)	33075	activation_23[0][0]
dropout_23 (Dropout)	(None, 4, 4, 35)	0	conv2d_24[0][0]
concatenate_20 (Concatenate)	(None, 4, 4, 140)	0	concatenate_19[0][0] dropout_23[0][0]
batch_normalization_24 (BatchNo	(None, 4, 4, 140)	560	concatenate_20[0][0]
activation_24 (Activation)	(None, 4, 4, 140)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 4, 4, 35)	44100	activation_24[0][0]
dropout_24 (Dropout)	(None, 4, 4, 35)	0	conv2d_25[0][0]
concatenate_21 (Concatenate)	(None, 4, 4, 175)	0	concatenate_20[0][0] dropout_24[0][0]
batch_normalization_25 (BatchNo	(None, 4, 4, 175)	700	concatenate_21[0][0]
activation_25 (Activation)	(None, 4, 4, 175)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 4, 4, 35)	55125	activation_25[0][0]

```
dropout_25 (Dropout)            (None, 4, 4, 35)     0           conv2d_26[0][0]
__________________________________________________________________________________________________
concatenate_22 (Concatenate)    (None, 4, 4, 210)    0           concatenate_21[0][0]
                                                                 dropout_25[0][0]
__________________________________________________________________________________________________
batch_normalization_26 (BatchNo (None, 4, 4, 210)    840         concatenate_22[0][0]
__________________________________________________________________________________________________
activation_26 (Activation)      (None, 4, 4, 210)    0           batch_normalization_26[0][0]
__________________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, 4, 4, 35)     66150       activation_26[0][0]
__________________________________________________________________________________________________
dropout_26 (Dropout)            (None, 4, 4, 35)     0           conv2d_27[0][0]
__________________________________________________________________________________________________
concatenate_23 (Concatenate)    (None, 4, 4, 245)    0           concatenate_22[0][0]
                                                                 dropout_26[0][0]
__________________________________________________________________________________________________
batch_normalization_27 (BatchNo (None, 4, 4, 245)    980         concatenate_23[0][0]
__________________________________________________________________________________________________
activation_27 (Activation)      (None, 4, 4, 245)    0           batch_normalization_27[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 245)    0           activation_27[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 980)          0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           9810        flatten[0][0]
==================================================================================================
Total params: 978,260
Trainable params: 970,420
Non-trainable params: 7,840
__________________________________________________________________________________________________
```

```
In [0]: # determine Loss function and Optimizer


        #rms = RMSprop(lr=0.001,decay=1e-6)

        #aam_opt = Adam(lr=0.1,decay=1e-6)



        model.compile(loss='categorical_crossentropy',
                      optimizer=Adam(),
                      metrics=['accuracy'])
```

```
In [0]: ####  https://www.kaggle.com/genesis16/densenet-93-accuracy
```

```
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlyStoppi

reduce_lr = ReduceLROnPlateau(monitor = 'val_loss', factor = 0.1, patience = 5, min_lr

early_stop = EarlyStopping(monitor = "val_loss", patience = 10)

def decay_fn(epoch, lr):
    if epoch < 50:
        return 0.001
    elif epoch >= 50 and epoch < 75:
        return 0.0001
    else:
        return 0.00001

lr_scheduler = LearningRateScheduler(decay_fn)

csv_logger = CSVLogger('training.log')
```

In [16]:
```
#model.fit(X_train, y_train,
#                    batch_size=batch_size,
#                    epochs=epochs,
#                    verbose=1,
#                    validation_data=(X_test, y_test))

history = model.fit_generator(
    datagen.flow(X_train, y_train, batch_size=32),
    steps_per_epoch=(len(X_train)/batch_size)*5,
    epochs=epochs,
    verbose = 1,
    validation_data=(X_val, y_val),
    callbacks = [lr_scheduler, csv_logger]
)
```

```
Epoch 1/35
3516/3515 [==============================] - 542s 154ms/step - loss: 1.3503 - acc: 0.5097 - val
Epoch 2/35
3516/3515 [==============================] - 528s 150ms/step - loss: 0.9411 - acc: 0.6689 - val
Epoch 3/35
3516/3515 [==============================] - 526s 149ms/step - loss: 0.7894 - acc: 0.7240 - val
Epoch 4/35
3516/3515 [==============================] - 526s 149ms/step - loss: 0.6992 - acc: 0.7554 - val
Epoch 5/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.6359 - acc: 0.7796 - val
Epoch 6/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.5880 - acc: 0.7963 - val
Epoch 7/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.5505 - acc: 0.8084 - val
Epoch 8/35
```

```
3516/3515 [==============================] - 525s 149ms/step - loss: 0.5220 - acc: 0.8190 - val
Epoch 9/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.4961 - acc: 0.8269 - val
Epoch 10/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.4769 - acc: 0.8335 - val
Epoch 11/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.4579 - acc: 0.8396 - val
Epoch 12/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.4383 - acc: 0.8481 - val
Epoch 13/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.4251 - acc: 0.8531 - val
Epoch 14/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.4115 - acc: 0.8572 - val
Epoch 15/35
3516/3515 [==============================] - 524s 149ms/step - loss: 0.3971 - acc: 0.8616 - val
Epoch 16/35
3516/3515 [==============================] - 524s 149ms/step - loss: 0.3931 - acc: 0.8623 - val
Epoch 17/35
3516/3515 [==============================] - 527s 150ms/step - loss: 0.3807 - acc: 0.8681 - val
Epoch 18/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3702 - acc: 0.8719 - val
Epoch 19/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3645 - acc: 0.8736 - val
Epoch 20/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.3507 - acc: 0.8778 - val
Epoch 21/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3450 - acc: 0.8795 - val
Epoch 22/35
3516/3515 [==============================] - 527s 150ms/step - loss: 0.3385 - acc: 0.8817 - val
Epoch 23/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3349 - acc: 0.8829 - val
Epoch 24/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.3294 - acc: 0.8852 - val
Epoch 25/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3226 - acc: 0.8884 - val
Epoch 26/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.3173 - acc: 0.8895 - val
Epoch 27/35
3516/3515 [==============================] - 524s 149ms/step - loss: 0.3120 - acc: 0.8909 - val
Epoch 28/35
3516/3515 [==============================] - 525s 149ms/step - loss: 0.3062 - acc: 0.8933 - val
Epoch 29/35
3516/3515 [==============================] - 526s 150ms/step - loss: 0.3020 - acc: 0.8948 - val
Epoch 30/35
3516/3515 [==============================] - 529s 150ms/step - loss: 0.2990 - acc: 0.8953 - val
Epoch 31/35
3516/3515 [==============================] - 531s 151ms/step - loss: 0.2925 - acc: 0.8981 - val
Epoch 32/35
```

```
3516/3515 [==============================] - 534s 152ms/step - loss: 0.2885 - acc: 0.8986 - val
Epoch 33/35
3516/3515 [==============================] - 535s 152ms/step - loss: 0.2847 - acc: 0.9006 - val
Epoch 34/35
3516/3515 [==============================] - 533s 152ms/step - loss: 0.2831 - acc: 0.9017 - val
Epoch 35/35
3516/3515 [==============================] - 530s 151ms/step - loss: 0.2793 - acc: 0.9030 - val
```

```
In [17]: # Test the model
         score = model.evaluate(X_test, y_test, verbose=1)
         print('Test loss:', score[0])
         print('Test accuracy:', score[1])
```

```
10000/10000 [==============================] - 10s 1ms/sample - loss: 0.3223 - acc: 0.9041
Test loss: 0.32226717756986617
Test accuracy: 0.9041
```

```
In [18]: # Save the trained weights in to .h5 format
         model.save_weights("DNST_model.h5")
         print("Saved model to disk")
```

```
Saved model to disk
```

3 Conclusion

1. The most important parameters are : the value "l" which is number of layers in dense block and "k" which is denoted number of filters.
2. I tried many values of l and k . Ultimately l=6 and k=35 proved to be effective. Also, the total number of parameters increase rapidly with these two parameters.
3. Also, the process of early stopping as well as using decay function for learning rate also helped.

```
In [0]:
```