DIP_HW10

December 20, 2019

1 Digital Image Processing - HW10 - 98722278 - Mohammad Doosti Lakhani

In this notebook, I have solved the assignment's problems which are as follows: 1. Answer following questions: 1. Why Convolutional Neural Networks have been used for image processing instead of fully connected neural networks? 2. What are the benefits of using Pooling layer? 3. What is the role of non-linear activation functions such as sigmoid and tanh? Is it possible to use linear activation functions? 4. What is the main reason that number of parameters in *GoogleNet* with 22 layers are much less than *AlexNet* with 8 layers?

2. Summarize Xception model
   1. LeNet
   2. AlexNet
   3. VGG
   4. ResNet
   5. Inception (GoogleNet)
   6. Xception
3. Train a Keras model on CIFAR10 dataset and report accuracy and Confusion Matrix
   1. Libraries
   2. Preparing Data
      1. Loading
      2. Normalizing
      3. Onehot Vector For Labels
   3. Setting Hyperparameters
   4. Learning Rate Decay Callbacks
   5. Defining ResNet110V2
   6. Compile Model
   7. Save Model Callbacks
   8. Cutout Regularization
   9. ImageDataGenerator
   10. Train Model
   11. Evaluate Model
       1. Last Model
       2. Best Model
       3. Confusion Matrix of Best Model
   12. 10 Worst Predictions

1.1 1 Answer following questions:

1. Why Convolutional Neural Networks have been used for image processing instead of fully connected neural networks?
2. What are the benefits of using Pooling layer?
3. What is the role of non-linear activation functions such as sigmoid and tanh? Is it possible to use linear activation functions?
4. What is the main reason that number of parameters in *GoogleNet* with 22 layers are much less than *AlexNet* with 8 layers?

Image credits mainly from the coresponding papers and this

1.1.1 1.A CNN vs FCN

The major reason of introducing CNNs is that FCNs combine all features by connecting all neurons in each layer to the all neurons of next layer while CNNs incorporate spatial features regarding of position of filters w.r.t. to input layers. Also CNNs preserve receptive fields regarding different sizes of filters which cannot be obtained in any form of fully connected neural networks. In other words, each neuron in CNN only is connected to a small chunk of input image.

Other major reason is the processing manners. CNNs can learn much more features with less number of parameters as FCNs cannot properly learn spatial features. Also because CNNs are smaller in term of features, they are fast too.

1.1.2 1.B Why Pooling layer

In summary, images are huge in size and number of features before passing them to any network. So when we start to train a network, after learning some feautures using convolutional layers, still we have huge matrix in term of spatial size so the best way to reduce considering local connectivity (each neuron only is connceted to small chunk of input image) is to taking pooling such as max or average.

The reason that this approach works is that in high spatial size matrices, a neuron can represent its locality as it is the feature of images where pixels are patially relative, so pooling just retain the most dominant information and exlcude all duplicate info which can be ignored.

1.1.3 1.C Why non-linear Activation Functions?

In term of computing convolution of FCN, logic is same, we have `w*x` for a single layer. Now let's have 3 layers. Here is the forward operation considering weights are optimal: `w3*w2*w1*x`.

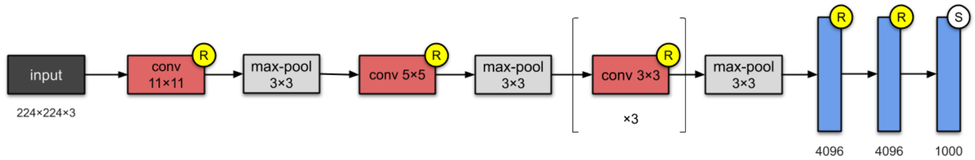
As we can see all the operations are linear between layers so we can reduce `w3*w2*w1` to `w` as convolution of multiple matrices are still linear. So the main idea of neural networks that can learn non-linearity of data has been gone!

To prevent this problem from happening, we had a non-linear function such as sigmoid or tanh to help eliminating linearity between each layer. Note that if we use linear functions, still convolving different matrices will be linear.

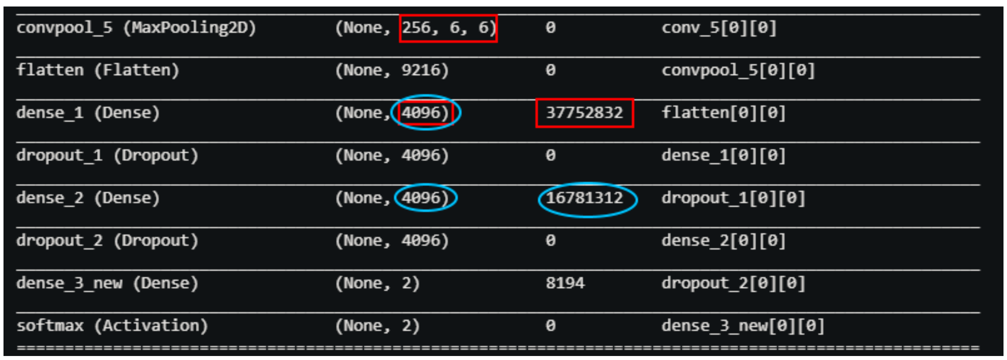
1.1.4 1.D 7M Paremeters InceptionV1 model vs. 56M parameters AlexNet

It is better to have an intuition of network before explaining the difference.

AlexNet:



alex net arch



convpool_5 (MaxPooling2D)	(None, 256, 6, 6)	0	conv_5[0][0]
flatten (Flatten)	(None, 9216)	0	convpool_5[0][0]
dense_1 (Dense)	(None, 4096)	37752832	flatten[0][0]
dropout_1 (Dropout)	(None, 4096)	0	dense_1[0][0]
dense_2 (Dense)	(None, 4096)	16781312	dropout_1[0][0]
dropout_2 (Dropout)	(None, 4096)	0	dense_2[0][0]
dense_3_new (Dense)	(None, 2)	8194	dropout_2[0][0]
softmax (Activation)	(None, 2)	0	dense_3_new[0][0]

alex net last fcn

Convolution layers in both networks almost have same amount of parameters and because *Inception V1* a.k.a. *GoogLeNet* has more conv layers, it has more parameters excluding fully connected parts.

In term of convolution parameters: 1. Inception: 3.2M 2. AlexNet: 2.5M

But what makes this huge difference is the connection between conv layers and fully connected layers and also the connection between fully connected layers itself.

In *AlexNet*, the connection between last pooling layer and first fully connected layer has about 37M parameters and on top of it, the connection between this layer and next fully connected layer has 16M parameters too so only these 2 layers of 8-layer AlexNet have more than 54M parameteres while the entire *Inception* model has 7M parameters which has been explained later. Here is an image of last pooling layer and 2 fully connected layers of *AlexNet*:

What about *Inception*? Inception V1:

About *Inception* we need to focus on two procedure: 1. `1x1` Convs: As we know in Inception module, different convs have been taken and then concatnated but before doing this, they first take `1v1` conv for sake of dimensionality reduction.

2. Using `GlobalAveragePooling` in the last layer after Inception layers helped to reduce the last convotional layer to only 1024, then a fully connected with 1000 neurons are connected at the last layer.

Note that there are 2 other auxiliry losses with same logic (after Inception4a and Inception 4d) so approximately all three pathes have about 3M parameters + 3.2M parameters of inception modules and other conv layers, we reach 7M parameters.

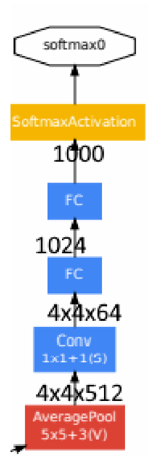
Image above shows the dimensionality of different layers at the connection of conv layers to fully connected for auxility loss 1. This is same for aux loss 2 except tensor after avg pooling is `4x4x528` instead of `4x4x512x`.

Below image also show the main path (main loss) of GoogleNet:

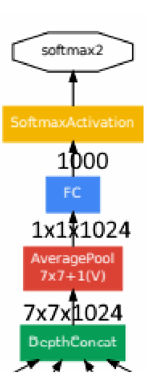
type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture

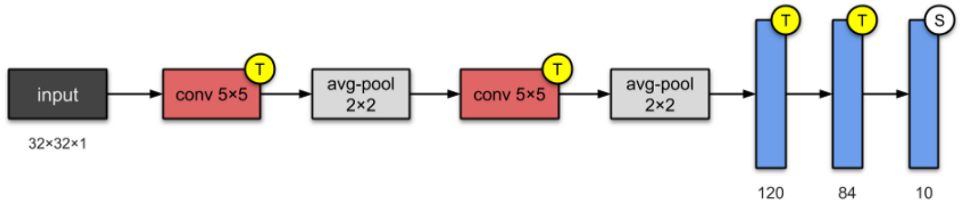
inception v1



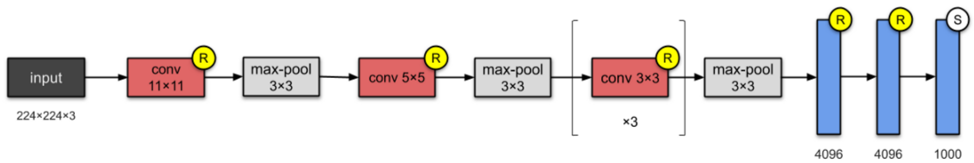
aux loss



main path



lenet



alex net

1.2 2 Summarize Xception model

1. LeNet
2. AlexNet
3. VGG
4. ResNet
5. Inception (GoogleNet)
6. Xception

Note: In all modules, circles with *T=Tanh*, *S=Sigmoid* and *R=ReLU*

1.2.1 2.A LeNet

LeNet is simplest neural network here, just 2 conv layer and 3 FCs. This model has only 0.6M parameters.

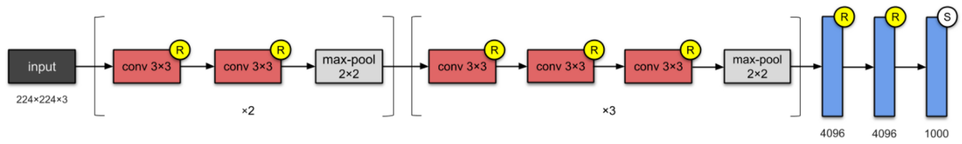
1.2.2 2.B AlexNet

AlexNet has 8 layers. What is interesting about this network is that the idea of stacking multiple convolutions then reducing the spatial size of filters has been introduced which is commonly used in almost all networks. The other important note is that they introduced ReLU activation in this paper.

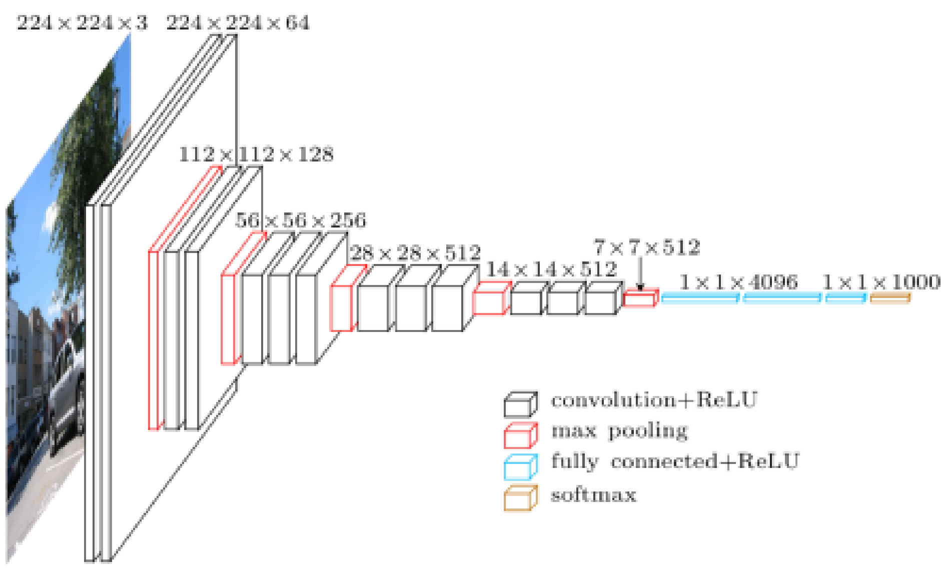
This model has huge capacity with 60M parameters and at the time of publishing, implementing this using available GPUs was challenge so they used parallel implementation of different operation that led them win challenges.

1.2.3 2.C VGG

VGG has two main difference from AlexNet 1. VGG uses smaller filter sizes like 2x2 or 3x3 instead of using big ones like 11x11 at first then reduing the filter sizes and reducing volume sizes has been taken care of by maxpooling layers. 2. VGG is much deeper and the reason is that in the



VGG



VGG 2

corresponding paper (or many books) it has been shown that deeper and bigger neural networks has more capacity to learn, so why not deeper?!

They stacked much more layers of smaller filter sizes so as we can guess number of parameters increased to 138M. VGG has different models that a number follows the name VGG which demonstrates number of layers of model. Most reknown ones are VGG-16 and VGG-19.

Something is good to know is that in many different tasks, people use intermediate layers of VGG trained on ImageNet as latent vector of feature extractor or directly transfering knowledge from VGG model for their particular tasks.

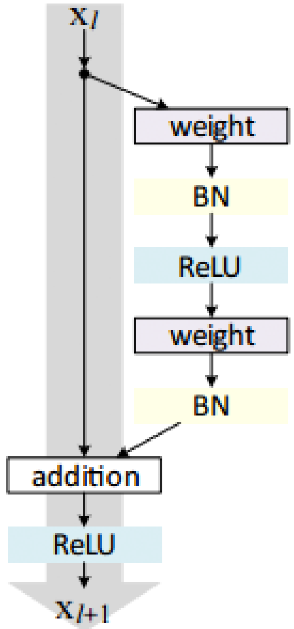
This one may show much better:

1.2.4 2.D ResNet

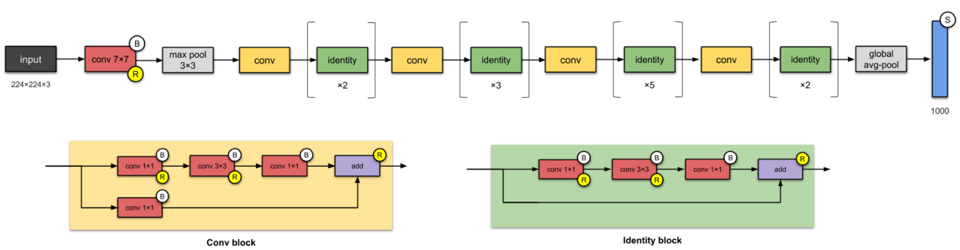
The resnet's idea is really simple in intuitive way, "go deeper and deeper but you might forget what you have learned before, so every time you go deeper, try to learn new thing, if you cannot, retain what ever you have had so far (by identity function!)".

Actually I made up the sentence above but it is absolutely true about ResNet. Here is an image that helps:

The other parts are similar to any other networks, stacking up multiple layers but this time, we stack everytime a single one of aforementioned layers called ResNet block(bottleneck). The straight lines works as identity function in simplified terms called skipping connections.



resnet bottleneck



resnet arch

This architecture helps scientists to build much deeper networks from 20 layers to nowadays 1000 layers with increase in accuracy by increasing layers thanks to those micro-architectures(resnet blocks).

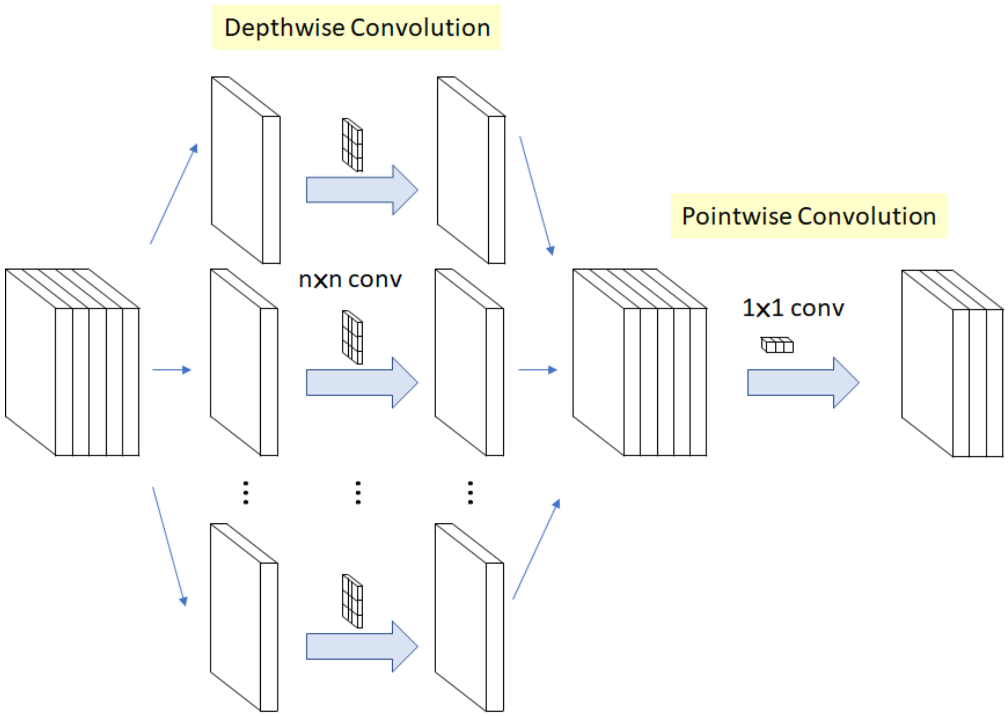
Because ResNet also uses global average pooling(we can see similar effect in Inception module) the number of parameters is much less than AlexNet. For instance, ResNet-50 with 50 layers has only 26M parameters.

1.2.5 2.E Inception-V1

please see section 1.D

1.2.6 2.F Xception

The first point I would like to focus on is that *Xception* only not introduces new artichtecture, it also depicts a new approach of taking convolutions called *Depthwise separable convolutions* which



depthwise separable convoltuions

is enormously faster than normal convolution we knew from other models such as *ResNet*, *VGG*.

What is *depthwise convolution*? Multiplication is a expensive operation for computer and in a normal convolution for a image with size of `H*H*M` and `N` filters with size of `K*K*M`, the number of multiplication will be `N*(H-K)^2*K^2*M`.

Depthwise separable convolution has two steps: 1. Depthwise convolution: Only applies convolution to a channel at a time rather than all channels so we need `M` filter with size of `K*K*1` too. The output size of this step will be in size of `(H-K)^M` which needs `M*K^2*(H-K)^2` multiplications. 2. Pointwise convolution: Linear combination of these layers with a filter with size of `1*1*M`. Assume as normal convolution we need `N` filters so we can exapnd the idea here too. The output of this step will be in size of `N*(H-K)^2` which needs `N*(H-K)^2*M` multiplications.

Finally, the speedup is equal to `(1/N) + (1/K^2)`.

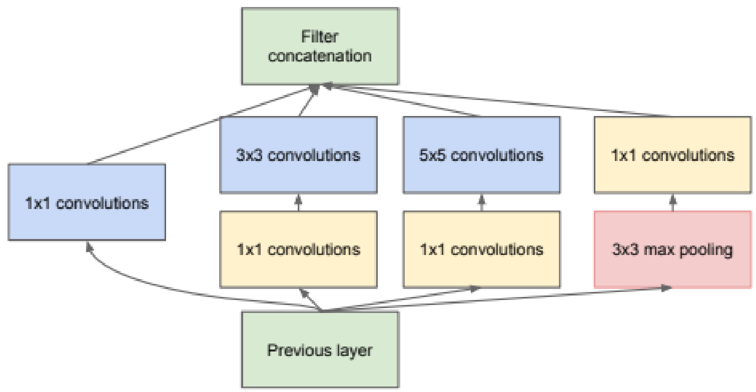
In summary, we have same ratio for number of parameters too which is big deal. This image may help to understand the operation:

Now we focus on *Xception* module. In *Xception* the order of operations in depthwise separable convolutions has been reversed which means first `1*1` convolutions have been used like below image:

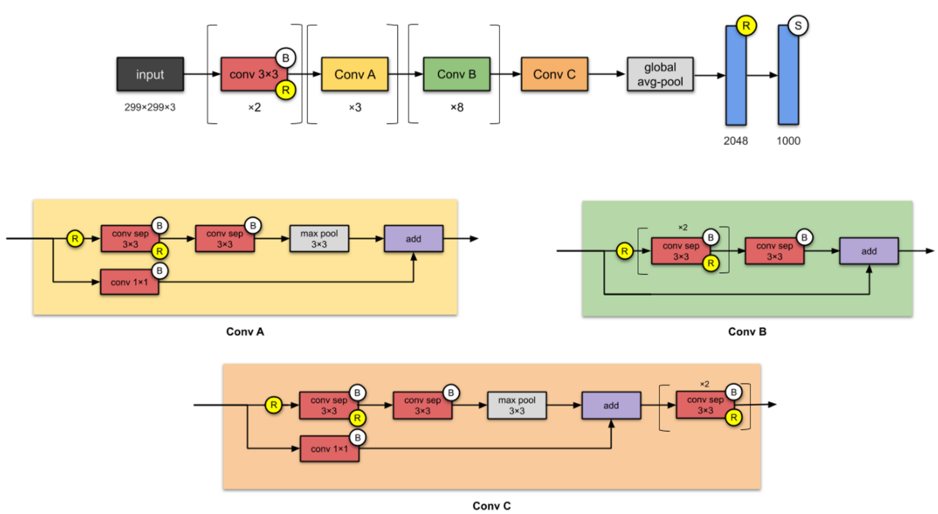
Xception is the extended version of Inception-V3 (in term of number of stacked layers) but the main difference as has been explained is using *depthwize separable convolutions eXtreme*ly which is exactly as the way I have explained in mathematically. The other parts are very similar to Inception and convetional CNNs.

This model has about 23M parameters.

Here is the final model:



simplified inception module



Xception

1.3 3 Train a Keras model on CIFAR10

1. Libraries
2. Preparing Data
    1. Loading
    2. Normalizing
    3. Onehot Vector For Labels
3. Setting Hyperparameters
4. Learning Rate Decay Callbacks
5. Defining ResNet110V2
6. Compile Model
7. Save Model Callbacks
8. Cutout Regularization
9. ImageDataGenerator
10. Train Model
11. Evaluate Model
    1. Last Model
    2. Best Model
    3. Confusion Matrix of Best Model
12. 10 Worst Predictions

1.3.1 3.A Libraries

```
In [ ]: %tensorflow_version 1.x

        from __future__ import print_function
        import seaborn as sns
        import matplotlib.pylab as plt
        import PIL
        import pandas as pd
        import numpy as np
        from keras import backend as K
        from keras.preprocessing.image import ImageDataGenerator
        from keras.optimizers import Adam
        from keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint, Learning
        from keras.layers import Dense, Conv2D, BatchNormalization, Activation
        from keras.layers import AveragePooling2D, Input, Flatten
        from keras.regularizers import l2
        from keras.models import Model
        from keras.utils import to_categorical
        import keras

        from keras.datasets import cifar10
```

1.3.2 3.B Preparing Data

1. loading

2. normalizing
3. converting labels to onehot vectors

3.B.a Loading

```
In [5]: (x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 4s 0us/step
```

3.B.b Normalizing

```
In [6]: x_train = x_train.astype('float32') / 255
        x_test = x_test.astype('float32') / 255
        x_train_mean = np.mean(x_train, axis=0)
        x_train -= x_train_mean
        x_test -= x_train_mean

        print('x_train shape:', x_train.shape)
        print('train samples', x_train.shape[0])
        print('test samples', x_test.shape[0])
        print('y_train shape:', y_train.shape)
```

```
x_train shape: (50000, 32, 32, 3)
train samples 50000
test samples 10000
y_train shape: (50000, 1)
```

3.B.c Convert Y to onehot vectors

```
In [8]: num_classes = 10
        y_train = to_categorical(y_train, num_classes)
        y_test = to_categorical(y_test, num_classes)
        print(y_train.shape)
```

```
(50000, 10)
```

1.3.3 3.C Setting Hyperparameters

```
In [ ]: batch_size = 32
        epochs = 200

        data_augmentation = True
        subtract_pixel_mean = True

        # resnet 110 v2
```

```
        depth = 110
        version = 2
        model_type = 'ResNet110v2'
```

```
In [ ]: input_shape = x_train.shape[1:]
```

1.3.4 3.D Learning Rate Decay Callbacks

```
In [ ]: def lr_schedule(epoch):
            """
            Learning Rate Schedule

            Learning rate is scheduled to be reduced after 80, 120, 160, 180 epochs.
            Called automatically every epoch as part of callbacks during training.


            :param epoch: The number of epochs

            :Returns: lr (float32) learning rate
            """
            lr = 1e-3
            if epoch > 180:
                lr *= 0.5e-3
            elif epoch > 160:
                lr *= 1e-3
            elif epoch > 120:
                lr *= 1e-2
            elif epoch > 80:
                lr *= 1e-1
            print('Learning rate: ', lr)
            return lr


        lr_scheduler = LearningRateScheduler(lr_schedule)
        lr_reducer = ReduceLROnPlateau(factor=np.sqrt(0.1), cooldown=0, patience=5, min_lr=0.5e
```

1.3.5 3.E Defining ResNet110V2

```
In [ ]: def resnet_layer(inputs, num_filters=16, kernel_size=3, strides=1, activation='relu', b
            """2D Convolution-Batch Normalization-Activation block

            :param inputs (tensor): input tensor from input image or previous layer
            :param num_filters (int): Conv2D number of filters
            :param kernel_size (int): Conv2D kernel dimensions
            :param strides (int): Conv2D stride dimensions
            :param activation (string): activation name
            :param batch_normalization (bool): whether to include batch normalization
            :param conv_first (bool): conv-bn-activation (True) or bn-activation-conv (False)
```

```
    :return: x (tensor) tensor as input to the next layer
    """
    conv = Conv2D(num_filters, kernel_size=kernel_size, strides=strides, padding='same

    x = inputs
    if conv_first:
        x = conv(x)
        if batch_normalization:
            x = BatchNormalization()(x)
        if activation is not None:
            x = Activation(activation)(x)
    else:
        if batch_normalization:
            x = BatchNormalization()(x)
        if activation is not None:
            x = Activation(activation)(x)
        x = conv(x)
    return x



def resnet110v2(input_shape, num_classes=10):
    """
    ResNet Version 2 Model

    Stacks of (1 x 1)-(3 x 3)-(1 x 1) BN-ReLU-Conv2D
    First shortcut connection per layer is 1 x 1 Conv2D.
    Second and onwards shortcut connection is identity.
    At the beginning of each stage, the feature map size is halved (downsampled)
    by a convolutional layer with strides=2, while the number of filter maps is
    doubled. Within each stage, the layers have the same number filters and the
    same filter map sizes.
    Features maps sizes:
    conv1  : 32x32,  16
    stage 0: 32x32,  64
    stage 1: 16x16, 128
    stage 2:  8x8,  256

    :param input_shape (tensor): shape of input tensor
    :param num_classes (int): number of classes


    :return: Keras model instance
    """

    # Start model definition.
    num_filters_in = 16
```

```
num_res_blocks = 12

inputs = Input(shape=input_shape)

x = resnet_layer(inputs=inputs, num_filters=num_filters_in, conv_first=True)

# Instantiate the resnet blocks
for stage in range(3):
    for res_block in range(num_res_blocks):
        activation = 'relu'
        batch_normalization = True
        strides = 1
        if stage == 0:
            num_filters_out = num_filters_in * 4
            if res_block == 0:  # first layer and first stage
                activation = None
                batch_normalization = False
        else:
            num_filters_out = num_filters_in * 2
            if res_block == 0:  # first layer but not first stage
                strides = 2

        # resnet blocks
        y = resnet_layer(inputs=x, num_filters=num_filters_in, kernel_size=1, stri
                         batch_normalization=batch_normalization, conv_first=False
        y = resnet_layer(inputs=y, num_filters=num_filters_in, conv_first=False)
        y = resnet_layer(inputs=y, num_filters=num_filters_out, kernel_size=1, con
        if res_block == 0:
            # linear projection residual shortcut connection to match
            x = resnet_layer(inputs=x, num_filters=num_filters_out, kernel_size=1,
        x = keras.layers.add([x, y])

    num_filters_in = num_filters_out

# add classifier
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = AveragePooling2D(pool_size=8)(x)
y = Flatten()(x)
outputs = Dense(num_classes, activation='softmax', kernel_initializer='he_normal')

# instantiate model
model = Model(inputs=inputs, outputs=outputs)
return model
```

1.3.6 3.F Compile Model

```
In [ ]: model = resnet110v2(input_shape=input_shape)
        model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=lr_schedule(0)), metri
        model.summary()
        print(model_type)
```

```
Learning rate:  0.001
Model: "model_5"

__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_6 (InputLayer)            (None, 32, 32, 3)    0
__________________________________________________________________________________________________
conv2d_454 (Conv2D)             (None, 32, 32, 16)   448         input_6[0][0]
__________________________________________________________________________________________________
batch_normalization_440 (BatchN (None, 32, 32, 16)   64          conv2d_454[0][0]
__________________________________________________________________________________________________
activation_440 (Activation)     (None, 32, 32, 16)   0           batch_normalization_440[0][0]
__________________________________________________________________________________________________
conv2d_455 (Conv2D)             (None, 32, 32, 16)   272         activation_440[0][0]
__________________________________________________________________________________________________
batch_normalization_441 (BatchN (None, 32, 32, 16)   64          conv2d_455[0][0]
__________________________________________________________________________________________________
activation_441 (Activation)     (None, 32, 32, 16)   0           batch_normalization_441[0][0]
__________________________________________________________________________________________________
conv2d_456 (Conv2D)             (None, 32, 32, 16)   2320        activation_441[0][0]
__________________________________________________________________________________________________
batch_normalization_442 (BatchN (None, 32, 32, 16)   64          conv2d_456[0][0]
__________________________________________________________________________________________________
activation_442 (Activation)     (None, 32, 32, 16)   0           batch_normalization_442[0][0]
__________________________________________________________________________________________________
conv2d_458 (Conv2D)             (None, 32, 32, 64)   1088        activation_440[0][0]
__________________________________________________________________________________________________
conv2d_457 (Conv2D)             (None, 32, 32, 64)   1088        activation_442[0][0]
__________________________________________________________________________________________________
add_145 (Add)                   (None, 32, 32, 64)   0           conv2d_458[0][0]
                                                                 conv2d_457[0][0]
__________________________________________________________________________________________________
batch_normalization_443 (BatchN (None, 32, 32, 64)   256         add_145[0][0]
__________________________________________________________________________________________________
activation_443 (Activation)     (None, 32, 32, 64)   0           batch_normalization_443[0][0]
__________________________________________________________________________________________________
conv2d_459 (Conv2D)             (None, 32, 32, 16)   1040        activation_443[0][0]
__________________________________________________________________________________________________
batch_normalization_444 (BatchN (None, 32, 32, 16)   64          conv2d_459[0][0]
__________________________________________________________________________________________________
activation_444 (Activation)     (None, 32, 32, 16)   0           batch_normalization_444[0][0]
__________________________________________________________________________________________________
```

```
conv2d_460 (Conv2D)            (None, 32, 32, 16)   2320        activation_444[0][0]
__________________________________________________________________________________________________
batch_normalization_445 (BatchN (None, 32, 32, 16)   64          conv2d_460[0][0]
__________________________________________________________________________________________________
activation_445 (Activation)     (None, 32, 32, 16)   0           batch_normalization_445[0][0]
__________________________________________________________________________________________________
conv2d_461 (Conv2D)            (None, 32, 32, 64)   1088        activation_445[0][0]
__________________________________________________________________________________________________
add_146 (Add)                   (None, 32, 32, 64)   0           add_145[0][0]
                                                                 conv2d_461[0][0]
__________________________________________________________________________________________________
batch_normalization_446 (BatchN (None, 32, 32, 64)   256         add_146[0][0]
__________________________________________________________________________________________________
activation_446 (Activation)     (None, 32, 32, 64)   0           batch_normalization_446[0][0]
__________________________________________________________________________________________________
conv2d_462 (Conv2D)            (None, 32, 32, 16)   1040        activation_446[0][0]
__________________________________________________________________________________________________
batch_normalization_447 (BatchN (None, 32, 32, 16)   64          conv2d_462[0][0]
__________________________________________________________________________________________________
activation_447 (Activation)     (None, 32, 32, 16)   0           batch_normalization_447[0][0]
__________________________________________________________________________________________________
conv2d_463 (Conv2D)            (None, 32, 32, 16)   2320        activation_447[0][0]
__________________________________________________________________________________________________
batch_normalization_448 (BatchN (None, 32, 32, 16)   64          conv2d_463[0][0]
__________________________________________________________________________________________________
activation_448 (Activation)     (None, 32, 32, 16)   0           batch_normalization_448[0][0]
__________________________________________________________________________________________________
conv2d_464 (Conv2D)            (None, 32, 32, 64)   1088        activation_448[0][0]
__________________________________________________________________________________________________
add_147 (Add)                   (None, 32, 32, 64)   0           add_146[0][0]
                                                                 conv2d_464[0][0]
__________________________________________________________________________________________________
batch_normalization_449 (BatchN (None, 32, 32, 64)   256         add_147[0][0]
__________________________________________________________________________________________________
activation_449 (Activation)     (None, 32, 32, 64)   0           batch_normalization_449[0][0]
__________________________________________________________________________________________________
conv2d_465 (Conv2D)            (None, 32, 32, 16)   1040        activation_449[0][0]
__________________________________________________________________________________________________
batch_normalization_450 (BatchN (None, 32, 32, 16)   64          conv2d_465[0][0]
__________________________________________________________________________________________________
activation_450 (Activation)     (None, 32, 32, 16)   0           batch_normalization_450[0][0]
__________________________________________________________________________________________________
conv2d_466 (Conv2D)            (None, 32, 32, 16)   2320        activation_450[0][0]
__________________________________________________________________________________________________
batch_normalization_451 (BatchN (None, 32, 32, 16)   64          conv2d_466[0][0]
__________________________________________________________________________________________________
activation_451 (Activation)     (None, 32, 32, 16)   0           batch_normalization_451[0][0]
__________________________________________________________________________________________________
```

```
conv2d_467 (Conv2D)             (None, 32, 32, 64)   1088        activation_451[0][0]
__________________________________________________________________________________________________
add_148 (Add)                   (None, 32, 32, 64)   0           add_147[0][0]
                                                                 conv2d_467[0][0]
__________________________________________________________________________________________________
batch_normalization_452 (BatchN (None, 32, 32, 64)   256         add_148[0][0]
__________________________________________________________________________________________________
activation_452 (Activation)     (None, 32, 32, 64)   0           batch_normalization_452[0][0]
__________________________________________________________________________________________________
conv2d_468 (Conv2D)             (None, 32, 32, 16)   1040        activation_452[0][0]
__________________________________________________________________________________________________
batch_normalization_453 (BatchN (None, 32, 32, 16)   64          conv2d_468[0][0]
__________________________________________________________________________________________________
activation_453 (Activation)     (None, 32, 32, 16)   0           batch_normalization_453[0][0]
__________________________________________________________________________________________________
conv2d_469 (Conv2D)             (None, 32, 32, 16)   2320        activation_453[0][0]
__________________________________________________________________________________________________
batch_normalization_454 (BatchN (None, 32, 32, 16)   64          conv2d_469[0][0]
__________________________________________________________________________________________________
activation_454 (Activation)     (None, 32, 32, 16)   0           batch_normalization_454[0][0]
__________________________________________________________________________________________________
conv2d_470 (Conv2D)             (None, 32, 32, 64)   1088        activation_454[0][0]
__________________________________________________________________________________________________
add_149 (Add)                   (None, 32, 32, 64)   0           add_148[0][0]
                                                                 conv2d_470[0][0]
__________________________________________________________________________________________________
batch_normalization_455 (BatchN (None, 32, 32, 64)   256         add_149[0][0]
__________________________________________________________________________________________________
activation_455 (Activation)     (None, 32, 32, 64)   0           batch_normalization_455[0][0]
__________________________________________________________________________________________________
conv2d_471 (Conv2D)             (None, 32, 32, 16)   1040        activation_455[0][0]
__________________________________________________________________________________________________
batch_normalization_456 (BatchN (None, 32, 32, 16)   64          conv2d_471[0][0]
__________________________________________________________________________________________________
activation_456 (Activation)     (None, 32, 32, 16)   0           batch_normalization_456[0][0]
__________________________________________________________________________________________________
conv2d_472 (Conv2D)             (None, 32, 32, 16)   2320        activation_456[0][0]
__________________________________________________________________________________________________
batch_normalization_457 (BatchN (None, 32, 32, 16)   64          conv2d_472[0][0]
__________________________________________________________________________________________________
activation_457 (Activation)     (None, 32, 32, 16)   0           batch_normalization_457[0][0]
__________________________________________________________________________________________________
conv2d_473 (Conv2D)             (None, 32, 32, 64)   1088        activation_457[0][0]
__________________________________________________________________________________________________
add_150 (Add)                   (None, 32, 32, 64)   0           add_149[0][0]
                                                                 conv2d_473[0][0]
__________________________________________________________________________________________________
batch_normalization_458 (BatchN (None, 32, 32, 64)   256         add_150[0][0]
```

```
________________________________________________________________________________________________
activation_458 (Activation)     (None, 32, 32, 64)   0           batch_normalization_458[0][0]
________________________________________________________________________________________________
conv2d_474 (Conv2D)             (None, 32, 32, 16)   1040        activation_458[0][0]
________________________________________________________________________________________________
batch_normalization_459 (BatchN (None, 32, 32, 16)   64          conv2d_474[0][0]
________________________________________________________________________________________________
activation_459 (Activation)     (None, 32, 32, 16)   0           batch_normalization_459[0][0]
________________________________________________________________________________________________
conv2d_475 (Conv2D)             (None, 32, 32, 16)   2320        activation_459[0][0]
________________________________________________________________________________________________
batch_normalization_460 (BatchN (None, 32, 32, 16)   64          conv2d_475[0][0]
________________________________________________________________________________________________
activation_460 (Activation)     (None, 32, 32, 16)   0           batch_normalization_460[0][0]
________________________________________________________________________________________________
conv2d_476 (Conv2D)             (None, 32, 32, 64)   1088        activation_460[0][0]
________________________________________________________________________________________________
add_151 (Add)                   (None, 32, 32, 64)   0           add_150[0][0]
                                                                 conv2d_476[0][0]
________________________________________________________________________________________________
batch_normalization_461 (BatchN (None, 32, 32, 64)   256         add_151[0][0]
________________________________________________________________________________________________
activation_461 (Activation)     (None, 32, 32, 64)   0           batch_normalization_461[0][0]
________________________________________________________________________________________________
conv2d_477 (Conv2D)             (None, 32, 32, 16)   1040        activation_461[0][0]
________________________________________________________________________________________________
batch_normalization_462 (BatchN (None, 32, 32, 16)   64          conv2d_477[0][0]
________________________________________________________________________________________________
activation_462 (Activation)     (None, 32, 32, 16)   0           batch_normalization_462[0][0]
________________________________________________________________________________________________
conv2d_478 (Conv2D)             (None, 32, 32, 16)   2320        activation_462[0][0]
________________________________________________________________________________________________
batch_normalization_463 (BatchN (None, 32, 32, 16)   64          conv2d_478[0][0]
________________________________________________________________________________________________
activation_463 (Activation)     (None, 32, 32, 16)   0           batch_normalization_463[0][0]
________________________________________________________________________________________________
conv2d_479 (Conv2D)             (None, 32, 32, 64)   1088        activation_463[0][0]
________________________________________________________________________________________________
add_152 (Add)                   (None, 32, 32, 64)   0           add_151[0][0]
                                                                 conv2d_479[0][0]
________________________________________________________________________________________________
batch_normalization_464 (BatchN (None, 32, 32, 64)   256         add_152[0][0]
________________________________________________________________________________________________
activation_464 (Activation)     (None, 32, 32, 64)   0           batch_normalization_464[0][0]
________________________________________________________________________________________________
conv2d_480 (Conv2D)             (None, 32, 32, 16)   1040        activation_464[0][0]
________________________________________________________________________________________________
batch_normalization_465 (BatchN (None, 32, 32, 16)   64          conv2d_480[0][0]
```

```
__________________________________________________________________________________________________
activation_465 (Activation)     (None, 32, 32, 16)   0           batch_normalization_465[0][0]
__________________________________________________________________________________________________
conv2d_481 (Conv2D)             (None, 32, 32, 16)   2320        activation_465[0][0]
__________________________________________________________________________________________________
batch_normalization_466 (BatchN (None, 32, 32, 16)   64          conv2d_481[0][0]
__________________________________________________________________________________________________
activation_466 (Activation)     (None, 32, 32, 16)   0           batch_normalization_466[0][0]
__________________________________________________________________________________________________
conv2d_482 (Conv2D)             (None, 32, 32, 64)   1088        activation_466[0][0]
__________________________________________________________________________________________________
add_153 (Add)                   (None, 32, 32, 64)   0           add_152[0][0]
                                                                 conv2d_482[0][0]
__________________________________________________________________________________________________
batch_normalization_467 (BatchN (None, 32, 32, 64)   256         add_153[0][0]
__________________________________________________________________________________________________
activation_467 (Activation)     (None, 32, 32, 64)   0           batch_normalization_467[0][0]
__________________________________________________________________________________________________
conv2d_483 (Conv2D)             (None, 32, 32, 16)   1040        activation_467[0][0]
__________________________________________________________________________________________________
batch_normalization_468 (BatchN (None, 32, 32, 16)   64          conv2d_483[0][0]
__________________________________________________________________________________________________
activation_468 (Activation)     (None, 32, 32, 16)   0           batch_normalization_468[0][0]
__________________________________________________________________________________________________
conv2d_484 (Conv2D)             (None, 32, 32, 16)   2320        activation_468[0][0]
__________________________________________________________________________________________________
batch_normalization_469 (BatchN (None, 32, 32, 16)   64          conv2d_484[0][0]
__________________________________________________________________________________________________
activation_469 (Activation)     (None, 32, 32, 16)   0           batch_normalization_469[0][0]
__________________________________________________________________________________________________
conv2d_485 (Conv2D)             (None, 32, 32, 64)   1088        activation_469[0][0]
__________________________________________________________________________________________________
add_154 (Add)                   (None, 32, 32, 64)   0           add_153[0][0]
                                                                 conv2d_485[0][0]
__________________________________________________________________________________________________
batch_normalization_470 (BatchN (None, 32, 32, 64)   256         add_154[0][0]
__________________________________________________________________________________________________
activation_470 (Activation)     (None, 32, 32, 64)   0           batch_normalization_470[0][0]
__________________________________________________________________________________________________
conv2d_486 (Conv2D)             (None, 32, 32, 16)   1040        activation_470[0][0]
__________________________________________________________________________________________________
batch_normalization_471 (BatchN (None, 32, 32, 16)   64          conv2d_486[0][0]
__________________________________________________________________________________________________
activation_471 (Activation)     (None, 32, 32, 16)   0           batch_normalization_471[0][0]
__________________________________________________________________________________________________
conv2d_487 (Conv2D)             (None, 32, 32, 16)   2320        activation_471[0][0]
__________________________________________________________________________________________________
batch_normalization_472 (BatchN (None, 32, 32, 16)   64          conv2d_487[0][0]
```

```
____________________________________________________________________________________________________
activation_472 (Activation)     (None, 32, 32, 16)   0           batch_normalization_472[0][0]
____________________________________________________________________________________________________
conv2d_488 (Conv2D)             (None, 32, 32, 64)   1088        activation_472[0][0]
____________________________________________________________________________________________________
add_155 (Add)                   (None, 32, 32, 64)   0           add_154[0][0]
                                                                 conv2d_488[0][0]
____________________________________________________________________________________________________
batch_normalization_473 (BatchN (None, 32, 32, 64)   256         add_155[0][0]
____________________________________________________________________________________________________
activation_473 (Activation)     (None, 32, 32, 64)   0           batch_normalization_473[0][0]
____________________________________________________________________________________________________
conv2d_489 (Conv2D)             (None, 32, 32, 16)   1040        activation_473[0][0]
____________________________________________________________________________________________________
batch_normalization_474 (BatchN (None, 32, 32, 16)   64          conv2d_489[0][0]
____________________________________________________________________________________________________
activation_474 (Activation)     (None, 32, 32, 16)   0           batch_normalization_474[0][0]
____________________________________________________________________________________________________
conv2d_490 (Conv2D)             (None, 32, 32, 16)   2320        activation_474[0][0]
____________________________________________________________________________________________________
batch_normalization_475 (BatchN (None, 32, 32, 16)   64          conv2d_490[0][0]
____________________________________________________________________________________________________
activation_475 (Activation)     (None, 32, 32, 16)   0           batch_normalization_475[0][0]
____________________________________________________________________________________________________
conv2d_491 (Conv2D)             (None, 32, 32, 64)   1088        activation_475[0][0]
____________________________________________________________________________________________________
add_156 (Add)                   (None, 32, 32, 64)   0           add_155[0][0]
                                                                 conv2d_491[0][0]
____________________________________________________________________________________________________
batch_normalization_476 (BatchN (None, 32, 32, 64)   256         add_156[0][0]
____________________________________________________________________________________________________
activation_476 (Activation)     (None, 32, 32, 64)   0           batch_normalization_476[0][0]
____________________________________________________________________________________________________
conv2d_492 (Conv2D)             (None, 16, 16, 64)   4160        activation_476[0][0]
____________________________________________________________________________________________________
batch_normalization_477 (BatchN (None, 16, 16, 64)   256         conv2d_492[0][0]
____________________________________________________________________________________________________
activation_477 (Activation)     (None, 16, 16, 64)   0           batch_normalization_477[0][0]
____________________________________________________________________________________________________
conv2d_493 (Conv2D)             (None, 16, 16, 64)   36928       activation_477[0][0]
____________________________________________________________________________________________________
batch_normalization_478 (BatchN (None, 16, 16, 64)   256         conv2d_493[0][0]
____________________________________________________________________________________________________
activation_478 (Activation)     (None, 16, 16, 64)   0           batch_normalization_478[0][0]
____________________________________________________________________________________________________
conv2d_495 (Conv2D)             (None, 16, 16, 128)  8320        add_156[0][0]
____________________________________________________________________________________________________
conv2d_494 (Conv2D)             (None, 16, 16, 128)  8320        activation_478[0][0]
```

```
__________________________________________________________________________________________________
add_157 (Add)                   (None, 16, 16, 128)  0           conv2d_495[0][0]
                                                                 conv2d_494[0][0]
__________________________________________________________________________________________________
batch_normalization_479 (BatchN (None, 16, 16, 128)  512         add_157[0][0]
__________________________________________________________________________________________________
activation_479 (Activation)     (None, 16, 16, 128)  0           batch_normalization_479[0][0]
__________________________________________________________________________________________________
conv2d_496 (Conv2D)             (None, 16, 16, 64)   8256        activation_479[0][0]
__________________________________________________________________________________________________
batch_normalization_480 (BatchN (None, 16, 16, 64)   256         conv2d_496[0][0]
__________________________________________________________________________________________________
activation_480 (Activation)     (None, 16, 16, 64)   0           batch_normalization_480[0][0]
__________________________________________________________________________________________________
conv2d_497 (Conv2D)             (None, 16, 16, 64)   36928       activation_480[0][0]
__________________________________________________________________________________________________
batch_normalization_481 (BatchN (None, 16, 16, 64)   256         conv2d_497[0][0]
__________________________________________________________________________________________________
activation_481 (Activation)     (None, 16, 16, 64)   0           batch_normalization_481[0][0]
__________________________________________________________________________________________________
conv2d_498 (Conv2D)             (None, 16, 16, 128)  8320        activation_481[0][0]
__________________________________________________________________________________________________
add_158 (Add)                   (None, 16, 16, 128)  0           add_157[0][0]
                                                                 conv2d_498[0][0]
__________________________________________________________________________________________________
batch_normalization_482 (BatchN (None, 16, 16, 128)  512         add_158[0][0]
__________________________________________________________________________________________________
activation_482 (Activation)     (None, 16, 16, 128)  0           batch_normalization_482[0][0]
__________________________________________________________________________________________________
conv2d_499 (Conv2D)             (None, 16, 16, 64)   8256        activation_482[0][0]
__________________________________________________________________________________________________
batch_normalization_483 (BatchN (None, 16, 16, 64)   256         conv2d_499[0][0]
__________________________________________________________________________________________________
activation_483 (Activation)     (None, 16, 16, 64)   0           batch_normalization_483[0][0]
__________________________________________________________________________________________________
conv2d_500 (Conv2D)             (None, 16, 16, 64)   36928       activation_483[0][0]
__________________________________________________________________________________________________
batch_normalization_484 (BatchN (None, 16, 16, 64)   256         conv2d_500[0][0]
__________________________________________________________________________________________________
activation_484 (Activation)     (None, 16, 16, 64)   0           batch_normalization_484[0][0]
__________________________________________________________________________________________________
conv2d_501 (Conv2D)             (None, 16, 16, 128)  8320        activation_484[0][0]
__________________________________________________________________________________________________
add_159 (Add)                   (None, 16, 16, 128)  0           add_158[0][0]
                                                                 conv2d_501[0][0]
__________________________________________________________________________________________________
batch_normalization_485 (BatchN (None, 16, 16, 128)  512         add_159[0][0]
__________________________________________________________________________________________________
```

```
activation_485 (Activation)     (None, 16, 16, 128)  0           batch_normalization_485[0][0]
__________________________________________________________________________________________________
conv2d_502 (Conv2D)             (None, 16, 16, 64)   8256        activation_485[0][0]
__________________________________________________________________________________________________
batch_normalization_486 (BatchN (None, 16, 16, 64)   256         conv2d_502[0][0]
__________________________________________________________________________________________________
activation_486 (Activation)     (None, 16, 16, 64)   0           batch_normalization_486[0][0]
__________________________________________________________________________________________________
conv2d_503 (Conv2D)             (None, 16, 16, 64)   36928       activation_486[0][0]
__________________________________________________________________________________________________
batch_normalization_487 (BatchN (None, 16, 16, 64)   256         conv2d_503[0][0]
__________________________________________________________________________________________________
activation_487 (Activation)     (None, 16, 16, 64)   0           batch_normalization_487[0][0]
__________________________________________________________________________________________________
conv2d_504 (Conv2D)             (None, 16, 16, 128)  8320        activation_487[0][0]
__________________________________________________________________________________________________
add_160 (Add)                   (None, 16, 16, 128)  0           add_159[0][0]
                                                                 conv2d_504[0][0]
__________________________________________________________________________________________________
batch_normalization_488 (BatchN (None, 16, 16, 128)  512         add_160[0][0]
__________________________________________________________________________________________________
activation_488 (Activation)     (None, 16, 16, 128)  0           batch_normalization_488[0][0]
__________________________________________________________________________________________________
conv2d_505 (Conv2D)             (None, 16, 16, 64)   8256        activation_488[0][0]
__________________________________________________________________________________________________
batch_normalization_489 (BatchN (None, 16, 16, 64)   256         conv2d_505[0][0]
__________________________________________________________________________________________________
activation_489 (Activation)     (None, 16, 16, 64)   0           batch_normalization_489[0][0]
__________________________________________________________________________________________________
conv2d_506 (Conv2D)             (None, 16, 16, 64)   36928       activation_489[0][0]
__________________________________________________________________________________________________
batch_normalization_490 (BatchN (None, 16, 16, 64)   256         conv2d_506[0][0]
__________________________________________________________________________________________________
activation_490 (Activation)     (None, 16, 16, 64)   0           batch_normalization_490[0][0]
__________________________________________________________________________________________________
conv2d_507 (Conv2D)             (None, 16, 16, 128)  8320        activation_490[0][0]
__________________________________________________________________________________________________
add_161 (Add)                   (None, 16, 16, 128)  0           add_160[0][0]
                                                                 conv2d_507[0][0]
__________________________________________________________________________________________________
batch_normalization_491 (BatchN (None, 16, 16, 128)  512         add_161[0][0]
__________________________________________________________________________________________________
activation_491 (Activation)     (None, 16, 16, 128)  0           batch_normalization_491[0][0]
__________________________________________________________________________________________________
conv2d_508 (Conv2D)             (None, 16, 16, 64)   8256        activation_491[0][0]
__________________________________________________________________________________________________
batch_normalization_492 (BatchN (None, 16, 16, 64)   256         conv2d_508[0][0]
__________________________________________________________________________________________________
```

```
activation_492 (Activation)     (None, 16, 16, 64)   0           batch_normalization_492[0][0]
__________________________________________________________________________________________________
conv2d_509 (Conv2D)             (None, 16, 16, 64)   36928       activation_492[0][0]
__________________________________________________________________________________________________
batch_normalization_493 (BatchN (None, 16, 16, 64)   256         conv2d_509[0][0]
__________________________________________________________________________________________________
activation_493 (Activation)     (None, 16, 16, 64)   0           batch_normalization_493[0][0]
__________________________________________________________________________________________________
conv2d_510 (Conv2D)             (None, 16, 16, 128)  8320        activation_493[0][0]
__________________________________________________________________________________________________
add_162 (Add)                   (None, 16, 16, 128)  0           add_161[0][0]
                                                                 conv2d_510[0][0]
__________________________________________________________________________________________________
batch_normalization_494 (BatchN (None, 16, 16, 128)  512         add_162[0][0]
__________________________________________________________________________________________________
activation_494 (Activation)     (None, 16, 16, 128)  0           batch_normalization_494[0][0]
__________________________________________________________________________________________________
conv2d_511 (Conv2D)             (None, 16, 16, 64)   8256        activation_494[0][0]
__________________________________________________________________________________________________
batch_normalization_495 (BatchN (None, 16, 16, 64)   256         conv2d_511[0][0]
__________________________________________________________________________________________________
activation_495 (Activation)     (None, 16, 16, 64)   0           batch_normalization_495[0][0]
__________________________________________________________________________________________________
conv2d_512 (Conv2D)             (None, 16, 16, 64)   36928       activation_495[0][0]
__________________________________________________________________________________________________
batch_normalization_496 (BatchN (None, 16, 16, 64)   256         conv2d_512[0][0]
__________________________________________________________________________________________________
activation_496 (Activation)     (None, 16, 16, 64)   0           batch_normalization_496[0][0]
__________________________________________________________________________________________________
conv2d_513 (Conv2D)             (None, 16, 16, 128)  8320        activation_496[0][0]
__________________________________________________________________________________________________
add_163 (Add)                   (None, 16, 16, 128)  0           add_162[0][0]
                                                                 conv2d_513[0][0]
__________________________________________________________________________________________________
batch_normalization_497 (BatchN (None, 16, 16, 128)  512         add_163[0][0]
__________________________________________________________________________________________________
activation_497 (Activation)     (None, 16, 16, 128)  0           batch_normalization_497[0][0]
__________________________________________________________________________________________________
conv2d_514 (Conv2D)             (None, 16, 16, 64)   8256        activation_497[0][0]
__________________________________________________________________________________________________
batch_normalization_498 (BatchN (None, 16, 16, 64)   256         conv2d_514[0][0]
__________________________________________________________________________________________________
activation_498 (Activation)     (None, 16, 16, 64)   0           batch_normalization_498[0][0]
__________________________________________________________________________________________________
conv2d_515 (Conv2D)             (None, 16, 16, 64)   36928       activation_498[0][0]
__________________________________________________________________________________________________
batch_normalization_499 (BatchN (None, 16, 16, 64)   256         conv2d_515[0][0]
__________________________________________________________________________________________________
```

```
activation_499 (Activation)     (None, 16, 16, 64)   0           batch_normalization_499[0][0]
__________________________________________________________________________________________________
conv2d_516 (Conv2D)             (None, 16, 16, 128)  8320        activation_499[0][0]
__________________________________________________________________________________________________
add_164 (Add)                   (None, 16, 16, 128)  0           add_163[0][0]
                                                                 conv2d_516[0][0]
__________________________________________________________________________________________________
batch_normalization_500 (BatchN (None, 16, 16, 128)  512         add_164[0][0]
__________________________________________________________________________________________________
activation_500 (Activation)     (None, 16, 16, 128)  0           batch_normalization_500[0][0]
__________________________________________________________________________________________________
conv2d_517 (Conv2D)             (None, 16, 16, 64)   8256        activation_500[0][0]
__________________________________________________________________________________________________
batch_normalization_501 (BatchN (None, 16, 16, 64)   256         conv2d_517[0][0]
__________________________________________________________________________________________________
activation_501 (Activation)     (None, 16, 16, 64)   0           batch_normalization_501[0][0]
__________________________________________________________________________________________________
conv2d_518 (Conv2D)             (None, 16, 16, 64)   36928       activation_501[0][0]
__________________________________________________________________________________________________
batch_normalization_502 (BatchN (None, 16, 16, 64)   256         conv2d_518[0][0]
__________________________________________________________________________________________________
activation_502 (Activation)     (None, 16, 16, 64)   0           batch_normalization_502[0][0]
__________________________________________________________________________________________________
conv2d_519 (Conv2D)             (None, 16, 16, 128)  8320        activation_502[0][0]
__________________________________________________________________________________________________
add_165 (Add)                   (None, 16, 16, 128)  0           add_164[0][0]
                                                                 conv2d_519[0][0]
__________________________________________________________________________________________________
batch_normalization_503 (BatchN (None, 16, 16, 128)  512         add_165[0][0]
__________________________________________________________________________________________________
activation_503 (Activation)     (None, 16, 16, 128)  0           batch_normalization_503[0][0]
__________________________________________________________________________________________________
conv2d_520 (Conv2D)             (None, 16, 16, 64)   8256        activation_503[0][0]
__________________________________________________________________________________________________
batch_normalization_504 (BatchN (None, 16, 16, 64)   256         conv2d_520[0][0]
__________________________________________________________________________________________________
activation_504 (Activation)     (None, 16, 16, 64)   0           batch_normalization_504[0][0]
__________________________________________________________________________________________________
conv2d_521 (Conv2D)             (None, 16, 16, 64)   36928       activation_504[0][0]
__________________________________________________________________________________________________
batch_normalization_505 (BatchN (None, 16, 16, 64)   256         conv2d_521[0][0]
__________________________________________________________________________________________________
activation_505 (Activation)     (None, 16, 16, 64)   0           batch_normalization_505[0][0]
__________________________________________________________________________________________________
conv2d_522 (Conv2D)             (None, 16, 16, 128)  8320        activation_505[0][0]
__________________________________________________________________________________________________
add_166 (Add)                   (None, 16, 16, 128)  0           add_165[0][0]
                                                                 conv2d_522[0][0]
```

```
__________________________________________________________________________________________________
batch_normalization_506 (BatchN (None, 16, 16, 128)  512         add_166[0][0]
__________________________________________________________________________________________________
activation_506 (Activation)     (None, 16, 16, 128)  0           batch_normalization_506[0][0]
__________________________________________________________________________________________________
conv2d_523 (Conv2D)             (None, 16, 16, 64)   8256        activation_506[0][0]
__________________________________________________________________________________________________
batch_normalization_507 (BatchN (None, 16, 16, 64)   256         conv2d_523[0][0]
__________________________________________________________________________________________________
activation_507 (Activation)     (None, 16, 16, 64)   0           batch_normalization_507[0][0]
__________________________________________________________________________________________________
conv2d_524 (Conv2D)             (None, 16, 16, 64)   36928       activation_507[0][0]
__________________________________________________________________________________________________
batch_normalization_508 (BatchN (None, 16, 16, 64)   256         conv2d_524[0][0]
__________________________________________________________________________________________________
activation_508 (Activation)     (None, 16, 16, 64)   0           batch_normalization_508[0][0]
__________________________________________________________________________________________________
conv2d_525 (Conv2D)             (None, 16, 16, 128)  8320        activation_508[0][0]
__________________________________________________________________________________________________
add_167 (Add)                   (None, 16, 16, 128)  0           add_166[0][0]
                                                                 conv2d_525[0][0]
__________________________________________________________________________________________________
batch_normalization_509 (BatchN (None, 16, 16, 128)  512         add_167[0][0]
__________________________________________________________________________________________________
activation_509 (Activation)     (None, 16, 16, 128)  0           batch_normalization_509[0][0]
__________________________________________________________________________________________________
conv2d_526 (Conv2D)             (None, 16, 16, 64)   8256        activation_509[0][0]
__________________________________________________________________________________________________
batch_normalization_510 (BatchN (None, 16, 16, 64)   256         conv2d_526[0][0]
__________________________________________________________________________________________________
activation_510 (Activation)     (None, 16, 16, 64)   0           batch_normalization_510[0][0]
__________________________________________________________________________________________________
conv2d_527 (Conv2D)             (None, 16, 16, 64)   36928       activation_510[0][0]
__________________________________________________________________________________________________
batch_normalization_511 (BatchN (None, 16, 16, 64)   256         conv2d_527[0][0]
__________________________________________________________________________________________________
activation_511 (Activation)     (None, 16, 16, 64)   0           batch_normalization_511[0][0]
__________________________________________________________________________________________________
conv2d_528 (Conv2D)             (None, 16, 16, 128)  8320        activation_511[0][0]
__________________________________________________________________________________________________
add_168 (Add)                   (None, 16, 16, 128)  0           add_167[0][0]
                                                                 conv2d_528[0][0]
__________________________________________________________________________________________________
batch_normalization_512 (BatchN (None, 16, 16, 128)  512         add_168[0][0]
__________________________________________________________________________________________________
activation_512 (Activation)     (None, 16, 16, 128)  0           batch_normalization_512[0][0]
__________________________________________________________________________________________________
conv2d_529 (Conv2D)             (None, 8, 8, 128)    16512       activation_512[0][0]
```

```
____________________________________________________________________________________________________
batch_normalization_513 (BatchN (None, 8, 8, 128)    512         conv2d_529[0][0]
____________________________________________________________________________________________________
activation_513 (Activation)     (None, 8, 8, 128)    0           batch_normalization_513[0][0]
____________________________________________________________________________________________________
conv2d_530 (Conv2D)             (None, 8, 8, 128)    147584      activation_513[0][0]
____________________________________________________________________________________________________
batch_normalization_514 (BatchN (None, 8, 8, 128)    512         conv2d_530[0][0]
____________________________________________________________________________________________________
activation_514 (Activation)     (None, 8, 8, 128)    0           batch_normalization_514[0][0]
____________________________________________________________________________________________________
conv2d_532 (Conv2D)             (None, 8, 8, 256)    33024       add_168[0][0]
____________________________________________________________________________________________________
conv2d_531 (Conv2D)             (None, 8, 8, 256)    33024       activation_514[0][0]
____________________________________________________________________________________________________
add_169 (Add)                   (None, 8, 8, 256)    0           conv2d_532[0][0]
                                                                 conv2d_531[0][0]
____________________________________________________________________________________________________
batch_normalization_515 (BatchN (None, 8, 8, 256)    1024        add_169[0][0]
____________________________________________________________________________________________________
activation_515 (Activation)     (None, 8, 8, 256)    0           batch_normalization_515[0][0]
____________________________________________________________________________________________________
conv2d_533 (Conv2D)             (None, 8, 8, 128)    32896       activation_515[0][0]
____________________________________________________________________________________________________
batch_normalization_516 (BatchN (None, 8, 8, 128)    512         conv2d_533[0][0]
____________________________________________________________________________________________________
activation_516 (Activation)     (None, 8, 8, 128)    0           batch_normalization_516[0][0]
____________________________________________________________________________________________________
conv2d_534 (Conv2D)             (None, 8, 8, 128)    147584      activation_516[0][0]
____________________________________________________________________________________________________
batch_normalization_517 (BatchN (None, 8, 8, 128)    512         conv2d_534[0][0]
____________________________________________________________________________________________________
activation_517 (Activation)     (None, 8, 8, 128)    0           batch_normalization_517[0][0]
____________________________________________________________________________________________________
conv2d_535 (Conv2D)             (None, 8, 8, 256)    33024       activation_517[0][0]
____________________________________________________________________________________________________
add_170 (Add)                   (None, 8, 8, 256)    0           add_169[0][0]
                                                                 conv2d_535[0][0]
____________________________________________________________________________________________________
batch_normalization_518 (BatchN (None, 8, 8, 256)    1024        add_170[0][0]
____________________________________________________________________________________________________
activation_518 (Activation)     (None, 8, 8, 256)    0           batch_normalization_518[0][0]
____________________________________________________________________________________________________
conv2d_536 (Conv2D)             (None, 8, 8, 128)    32896       activation_518[0][0]
____________________________________________________________________________________________________
batch_normalization_519 (BatchN (None, 8, 8, 128)    512         conv2d_536[0][0]
____________________________________________________________________________________________________
activation_519 (Activation)     (None, 8, 8, 128)    0           batch_normalization_519[0][0]
```

```
__________________________________________________________________________________________________
conv2d_537 (Conv2D)             (None, 8, 8, 128)    147584      activation_519[0][0]
__________________________________________________________________________________________________
batch_normalization_520 (BatchN (None, 8, 8, 128)    512         conv2d_537[0][0]
__________________________________________________________________________________________________
activation_520 (Activation)     (None, 8, 8, 128)    0           batch_normalization_520[0][0]
__________________________________________________________________________________________________
conv2d_538 (Conv2D)             (None, 8, 8, 256)    33024       activation_520[0][0]
__________________________________________________________________________________________________
add_171 (Add)                   (None, 8, 8, 256)    0           add_170[0][0]
                                                                 conv2d_538[0][0]
__________________________________________________________________________________________________
batch_normalization_521 (BatchN (None, 8, 8, 256)    1024        add_171[0][0]
__________________________________________________________________________________________________
activation_521 (Activation)     (None, 8, 8, 256)    0           batch_normalization_521[0][0]
__________________________________________________________________________________________________
conv2d_539 (Conv2D)             (None, 8, 8, 128)    32896       activation_521[0][0]
__________________________________________________________________________________________________
batch_normalization_522 (BatchN (None, 8, 8, 128)    512         conv2d_539[0][0]
__________________________________________________________________________________________________
activation_522 (Activation)     (None, 8, 8, 128)    0           batch_normalization_522[0][0]
__________________________________________________________________________________________________
conv2d_540 (Conv2D)             (None, 8, 8, 128)    147584      activation_522[0][0]
__________________________________________________________________________________________________
batch_normalization_523 (BatchN (None, 8, 8, 128)    512         conv2d_540[0][0]
__________________________________________________________________________________________________
activation_523 (Activation)     (None, 8, 8, 128)    0           batch_normalization_523[0][0]
__________________________________________________________________________________________________
conv2d_541 (Conv2D)             (None, 8, 8, 256)    33024       activation_523[0][0]
__________________________________________________________________________________________________
add_172 (Add)                   (None, 8, 8, 256)    0           add_171[0][0]
                                                                 conv2d_541[0][0]
__________________________________________________________________________________________________
batch_normalization_524 (BatchN (None, 8, 8, 256)    1024        add_172[0][0]
__________________________________________________________________________________________________
activation_524 (Activation)     (None, 8, 8, 256)    0           batch_normalization_524[0][0]
__________________________________________________________________________________________________
conv2d_542 (Conv2D)             (None, 8, 8, 128)    32896       activation_524[0][0]
__________________________________________________________________________________________________
batch_normalization_525 (BatchN (None, 8, 8, 128)    512         conv2d_542[0][0]
__________________________________________________________________________________________________
activation_525 (Activation)     (None, 8, 8, 128)    0           batch_normalization_525[0][0]
__________________________________________________________________________________________________
conv2d_543 (Conv2D)             (None, 8, 8, 128)    147584      activation_525[0][0]
__________________________________________________________________________________________________
batch_normalization_526 (BatchN (None, 8, 8, 128)    512         conv2d_543[0][0]
__________________________________________________________________________________________________
activation_526 (Activation)     (None, 8, 8, 128)    0           batch_normalization_526[0][0]
```

```
__________________________________________________________________________________________________
conv2d_544 (Conv2D)             (None, 8, 8, 256)    33024       activation_526[0][0]
__________________________________________________________________________________________________
add_173 (Add)                   (None, 8, 8, 256)    0           add_172[0][0]
                                                                 conv2d_544[0][0]
__________________________________________________________________________________________________
batch_normalization_527 (BatchN (None, 8, 8, 256)    1024        add_173[0][0]
__________________________________________________________________________________________________
activation_527 (Activation)     (None, 8, 8, 256)    0           batch_normalization_527[0][0]
__________________________________________________________________________________________________
conv2d_545 (Conv2D)             (None, 8, 8, 128)    32896       activation_527[0][0]
__________________________________________________________________________________________________
batch_normalization_528 (BatchN (None, 8, 8, 128)    512         conv2d_545[0][0]
__________________________________________________________________________________________________
activation_528 (Activation)     (None, 8, 8, 128)    0           batch_normalization_528[0][0]
__________________________________________________________________________________________________
conv2d_546 (Conv2D)             (None, 8, 8, 128)    147584      activation_528[0][0]
__________________________________________________________________________________________________
batch_normalization_529 (BatchN (None, 8, 8, 128)    512         conv2d_546[0][0]
__________________________________________________________________________________________________
activation_529 (Activation)     (None, 8, 8, 128)    0           batch_normalization_529[0][0]
__________________________________________________________________________________________________
conv2d_547 (Conv2D)             (None, 8, 8, 256)    33024       activation_529[0][0]
__________________________________________________________________________________________________
add_174 (Add)                   (None, 8, 8, 256)    0           add_173[0][0]
                                                                 conv2d_547[0][0]
__________________________________________________________________________________________________
batch_normalization_530 (BatchN (None, 8, 8, 256)    1024        add_174[0][0]
__________________________________________________________________________________________________
activation_530 (Activation)     (None, 8, 8, 256)    0           batch_normalization_530[0][0]
__________________________________________________________________________________________________
conv2d_548 (Conv2D)             (None, 8, 8, 128)    32896       activation_530[0][0]
__________________________________________________________________________________________________
batch_normalization_531 (BatchN (None, 8, 8, 128)    512         conv2d_548[0][0]
__________________________________________________________________________________________________
activation_531 (Activation)     (None, 8, 8, 128)    0           batch_normalization_531[0][0]
__________________________________________________________________________________________________
conv2d_549 (Conv2D)             (None, 8, 8, 128)    147584      activation_531[0][0]
__________________________________________________________________________________________________
batch_normalization_532 (BatchN (None, 8, 8, 128)    512         conv2d_549[0][0]
__________________________________________________________________________________________________
activation_532 (Activation)     (None, 8, 8, 128)    0           batch_normalization_532[0][0]
__________________________________________________________________________________________________
conv2d_550 (Conv2D)             (None, 8, 8, 256)    33024       activation_532[0][0]
__________________________________________________________________________________________________
add_175 (Add)                   (None, 8, 8, 256)    0           add_174[0][0]
                                                                 conv2d_550[0][0]
__________________________________________________________________________________________________
```

batch_normalization_533 (BatchN	(None, 8, 8, 256)	1024	add_175[0][0]
activation_533 (Activation)	(None, 8, 8, 256)	0	batch_normalization_533[0][0]
conv2d_551 (Conv2D)	(None, 8, 8, 128)	32896	activation_533[0][0]
batch_normalization_534 (BatchN	(None, 8, 8, 128)	512	conv2d_551[0][0]
activation_534 (Activation)	(None, 8, 8, 128)	0	batch_normalization_534[0][0]
conv2d_552 (Conv2D)	(None, 8, 8, 128)	147584	activation_534[0][0]
batch_normalization_535 (BatchN	(None, 8, 8, 128)	512	conv2d_552[0][0]
activation_535 (Activation)	(None, 8, 8, 128)	0	batch_normalization_535[0][0]
conv2d_553 (Conv2D)	(None, 8, 8, 256)	33024	activation_535[0][0]
add_176 (Add)	(None, 8, 8, 256)	0	add_175[0][0] conv2d_553[0][0]
batch_normalization_536 (BatchN	(None, 8, 8, 256)	1024	add_176[0][0]
activation_536 (Activation)	(None, 8, 8, 256)	0	batch_normalization_536[0][0]
conv2d_554 (Conv2D)	(None, 8, 8, 128)	32896	activation_536[0][0]
batch_normalization_537 (BatchN	(None, 8, 8, 128)	512	conv2d_554[0][0]
activation_537 (Activation)	(None, 8, 8, 128)	0	batch_normalization_537[0][0]
conv2d_555 (Conv2D)	(None, 8, 8, 128)	147584	activation_537[0][0]
batch_normalization_538 (BatchN	(None, 8, 8, 128)	512	conv2d_555[0][0]
activation_538 (Activation)	(None, 8, 8, 128)	0	batch_normalization_538[0][0]
conv2d_556 (Conv2D)	(None, 8, 8, 256)	33024	activation_538[0][0]
add_177 (Add)	(None, 8, 8, 256)	0	add_176[0][0] conv2d_556[0][0]
batch_normalization_539 (BatchN	(None, 8, 8, 256)	1024	add_177[0][0]
activation_539 (Activation)	(None, 8, 8, 256)	0	batch_normalization_539[0][0]
conv2d_557 (Conv2D)	(None, 8, 8, 128)	32896	activation_539[0][0]

```
batch_normalization_540 (BatchN (None, 8, 8, 128)    512         conv2d_557[0][0]
__________________________________________________________________________________________________
activation_540 (Activation)     (None, 8, 8, 128)    0           batch_normalization_540[0][0]
__________________________________________________________________________________________________
conv2d_558 (Conv2D)             (None, 8, 8, 128)    147584      activation_540[0][0]
__________________________________________________________________________________________________
batch_normalization_541 (BatchN (None, 8, 8, 128)    512         conv2d_558[0][0]
__________________________________________________________________________________________________
activation_541 (Activation)     (None, 8, 8, 128)    0           batch_normalization_541[0][0]
__________________________________________________________________________________________________
conv2d_559 (Conv2D)             (None, 8, 8, 256)    33024       activation_541[0][0]
__________________________________________________________________________________________________
add_178 (Add)                   (None, 8, 8, 256)    0           add_177[0][0]
                                                                 conv2d_559[0][0]
__________________________________________________________________________________________________
batch_normalization_542 (BatchN (None, 8, 8, 256)    1024        add_178[0][0]
__________________________________________________________________________________________________
activation_542 (Activation)     (None, 8, 8, 256)    0           batch_normalization_542[0][0]
__________________________________________________________________________________________________
conv2d_560 (Conv2D)             (None, 8, 8, 128)    32896       activation_542[0][0]
__________________________________________________________________________________________________
batch_normalization_543 (BatchN (None, 8, 8, 128)    512         conv2d_560[0][0]
__________________________________________________________________________________________________
activation_543 (Activation)     (None, 8, 8, 128)    0           batch_normalization_543[0][0]
__________________________________________________________________________________________________
conv2d_561 (Conv2D)             (None, 8, 8, 128)    147584      activation_543[0][0]
__________________________________________________________________________________________________
batch_normalization_544 (BatchN (None, 8, 8, 128)    512         conv2d_561[0][0]
__________________________________________________________________________________________________
activation_544 (Activation)     (None, 8, 8, 128)    0           batch_normalization_544[0][0]
__________________________________________________________________________________________________
conv2d_562 (Conv2D)             (None, 8, 8, 256)    33024       activation_544[0][0]
__________________________________________________________________________________________________
add_179 (Add)                   (None, 8, 8, 256)    0           add_178[0][0]
                                                                 conv2d_562[0][0]
__________________________________________________________________________________________________
batch_normalization_545 (BatchN (None, 8, 8, 256)    1024        add_179[0][0]
__________________________________________________________________________________________________
activation_545 (Activation)     (None, 8, 8, 256)    0           batch_normalization_545[0][0]
__________________________________________________________________________________________________
conv2d_563 (Conv2D)             (None, 8, 8, 128)    32896       activation_545[0][0]
__________________________________________________________________________________________________
batch_normalization_546 (BatchN (None, 8, 8, 128)    512         conv2d_563[0][0]
__________________________________________________________________________________________________
activation_546 (Activation)     (None, 8, 8, 128)    0           batch_normalization_546[0][0]
__________________________________________________________________________________________________
conv2d_564 (Conv2D)             (None, 8, 8, 128)    147584      activation_546[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_547 (BatchN (None, 8, 8, 128)    512         conv2d_564[0][0]
__________________________________________________________________________________________________
activation_547 (Activation)     (None, 8, 8, 128)    0           batch_normalization_547[0][0]
__________________________________________________________________________________________________
conv2d_565 (Conv2D)             (None, 8, 8, 256)    33024       activation_547[0][0]
__________________________________________________________________________________________________
add_180 (Add)                   (None, 8, 8, 256)    0           add_179[0][0]
                                                                 conv2d_565[0][0]
__________________________________________________________________________________________________
batch_normalization_548 (BatchN (None, 8, 8, 256)    1024        add_180[0][0]
__________________________________________________________________________________________________
activation_548 (Activation)     (None, 8, 8, 256)    0           batch_normalization_548[0][0]
__________________________________________________________________________________________________
average_pooling2d_5 (AveragePoo (None, 1, 1, 256)    0           activation_548[0][0]
__________________________________________________________________________________________________
flatten_5 (Flatten)             (None, 256)          0           average_pooling2d_5[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 10)           2570        flatten_5[0][0]
==================================================================================================
Total params: 3,323,210
Trainable params: 3,302,442
Non-trainable params: 20,768
__________________________________________________________________________________________________
ResNet110v2
```

1.3.7 3.G Save Model Callbacks

In [ ]:
```python
import os
def prepare_directory(model_type):
    save_dir = os.path.join(os.getcwd(), 'saved_model')
    model_name = 'cifar10_%s_model.{epoch:03d}.h5' % model_type
    if not os.path.isdir(save_dir):
        os.makedirs(save_dir)
    filepath = os.path.join(save_dir, model_name)
    return filepath

filepath = prepare_directory(model_type)

checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_acc', verbose=1, save_bes
```

gather all callbacks

In [ ]:
```python
callbacks = [checkpoint, lr_reducer, lr_scheduler]
```

set ImageDataGenerator to use data augmentation

1.3.8 3.H Cutout Regularization

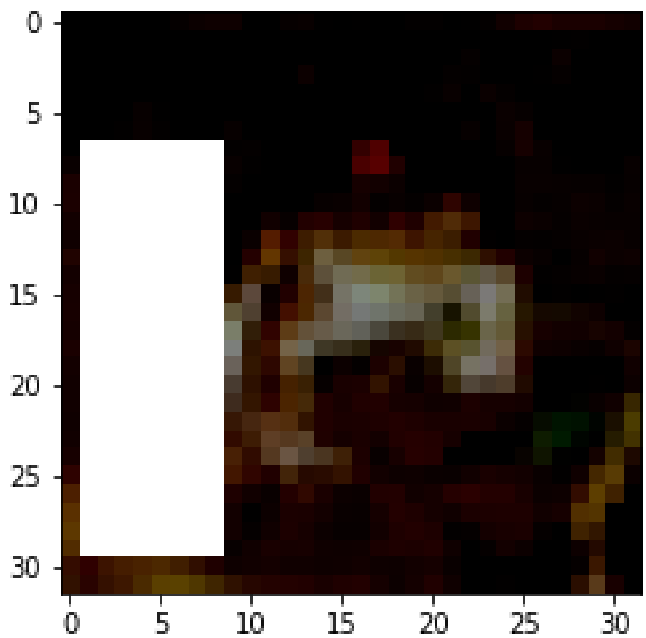
In [ ]:
```
import numpy as np
def get_random_eraser(p=0.5, s_l=0.02, s_h=0.4, r_1=0.3, r_2=1/0.3, v_l=0, v_h=255, pi
    def eraser(input_img):
        img_h, img_w, img_c = input_img.shape
        p_1 = np.random.rand()
        if p_1 > p:
            return input_img
        while True:
            s = np.random.uniform(s_l, s_h) * img_h * img_w
            r = np.random.uniform(r_1, r_2)
            w = int(np.sqrt(s / r))
            h = int(np.sqrt(s * r))
            left = np.random.randint(0, img_w)
            top = np.random.randint(0, img_h)

            if left + w <= img_w and top + h <= img_h:
                break
        if pixel_level:
            c = np.random.uniform(v_l, v_h, (h, w, img_c))
        else:
            c = np.random.uniform(v_l, v_h)
        input_img[top:top + h, left:left + w, :] = c
        return input_img
    return eraser
```

In [ ]:
```
from copy import deepcopy
z = deepcopy(x_train[0])
plt.imshow(get_random_eraser()(z))
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7f0898ed5cf8>
```



1.3.9 3.I ImageDataGenerator

```
In [ ]: datagen = ImageDataGenerator(
                rotation_range=20,
                width_shift_range=0.1,
                height_shift_range=0.1,
                fill_mode='nearest',
                cval=0.,
                zoom_range=0.1,
                horizontal_flip=True,
                rescale=None,
                preprocessing_function=get_random_eraser(v_l=0, v_h=1, pixel_level=False))

        datagen.fit(x_train)
```

1.3.10 3.J Train Model

```
In [89]: model.fit_generator(datagen.flow(x_train, y_train, batch_size=batch_size),
                             validation_data=(x_test, y_test), epochs=epochs, verbose=1, worke
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/ops/math_
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend
```

```
Epoch 1/200
Learning rate:  0.001
1563/1563 [==============================] - 736s 471ms/step - loss: 2.5089 - acc: 0.4288 - val

Epoch 00001: val_acc improved from -inf to 0.46530, saving model to /content/saved_model/cifar
Epoch 2/200
Learning rate:  0.001
1563/1563 [==============================] - 668s 428ms/step - loss: 1.6728 - acc: 0.5514 - val

Epoch 00002: val_acc improved from 0.46530 to 0.50460, saving model to /content/saved_model/ci
Epoch 3/200
Learning rate:  0.001
1563/1563 [==============================] - 661s 423ms/step - loss: 1.4663 - acc: 0.6035 - val

Epoch 00003: val_acc improved from 0.50460 to 0.54330, saving model to /content/saved_model/ci
Epoch 4/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 1.3443 - acc: 0.6397 - val

Epoch 00004: val_acc improved from 0.54330 to 0.57700, saving model to /content/saved_model/ci
Epoch 5/200
Learning rate:  0.001
1563/1563 [==============================] - 664s 425ms/step - loss: 1.2577 - acc: 0.6681 - val

Epoch 00005: val_acc improved from 0.57700 to 0.64620, saving model to /content/saved_model/ci
Epoch 6/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 426ms/step - loss: 1.1900 - acc: 0.6888 - val

Epoch 00006: val_acc improved from 0.64620 to 0.64650, saving model to /content/saved_model/ci
Epoch 7/200
Learning rate:  0.001
1563/1563 [==============================] - 664s 425ms/step - loss: 1.1353 - acc: 0.7021 - val

Epoch 00007: val_acc improved from 0.64650 to 0.67940, saving model to /content/saved_model/ci
Epoch 8/200
Learning rate:  0.001
1563/1563 [==============================] - 661s 423ms/step - loss: 1.0915 - acc: 0.7131 - val

Epoch 00008: val_acc improved from 0.67940 to 0.72560, saving model to /content/saved_model/ci
Epoch 9/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 423ms/step - loss: 1.0495 - acc: 0.7275 - val

Epoch 00009: val_acc did not improve from 0.72560
Epoch 10/200
Learning rate:  0.001
```

```
1563/1563 [==============================] - 667s 427ms/step - loss: 1.0190 - acc: 0.7342 - val

Epoch 00010: val_acc did not improve from 0.72560
Epoch 11/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.9953 - acc: 0.7428 - val

Epoch 00011: val_acc did not improve from 0.72560
Epoch 12/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 425ms/step - loss: 0.9653 - acc: 0.7523 - val

Epoch 00012: val_acc improved from 0.72560 to 0.73550, saving model to /content/saved_model/ci
Epoch 13/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.9463 - acc: 0.7571 - val

Epoch 00013: val_acc improved from 0.73550 to 0.77580, saving model to /content/saved_model/ci
Epoch 14/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.9306 - acc: 0.7644 - val

Epoch 00014: val_acc did not improve from 0.77580
Epoch 15/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 425ms/step - loss: 0.9081 - acc: 0.7692 - val

Epoch 00015: val_acc improved from 0.77580 to 0.79450, saving model to /content/saved_model/ci
Epoch 16/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 423ms/step - loss: 0.8867 - acc: 0.7762 - val

Epoch 00016: val_acc did not improve from 0.79450
Epoch 17/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 425ms/step - loss: 0.8761 - acc: 0.7804 - val

Epoch 00017: val_acc did not improve from 0.79450
Epoch 18/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.8550 - acc: 0.7835 - val

Epoch 00018: val_acc improved from 0.79450 to 0.80160, saving model to /content/saved_model/ci
Epoch 19/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 424ms/step - loss: 0.8470 - acc: 0.7880 - val

Epoch 00019: val_acc did not improve from 0.80160
```

```
Epoch 20/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.8328 - acc: 0.7918 - val

Epoch 00020: val_acc did not improve from 0.80160
Epoch 21/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 424ms/step - loss: 0.8248 - acc: 0.7935 - val

Epoch 00021: val_acc improved from 0.80160 to 0.80270, saving model to /content/saved_model/ci
Epoch 22/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 424ms/step - loss: 0.8126 - acc: 0.7986 - val

Epoch 00022: val_acc improved from 0.80270 to 0.81100, saving model to /content/saved_model/ci
Epoch 23/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.8095 - acc: 0.7982 - val

Epoch 00023: val_acc did not improve from 0.81100
Epoch 24/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.7983 - acc: 0.8012 - val

Epoch 00024: val_acc did not improve from 0.81100
Epoch 25/200
Learning rate:  0.001
1563/1563 [==============================] - 661s 423ms/step - loss: 0.7898 - acc: 0.8066 - val

Epoch 00025: val_acc did not improve from 0.81100
Epoch 26/200
Learning rate:  0.001
1563/1563 [==============================] - 664s 425ms/step - loss: 0.7797 - acc: 0.8088 - val

Epoch 00026: val_acc did not improve from 0.81100
Epoch 27/200
Learning rate:  0.001
1563/1563 [==============================] - 667s 427ms/step - loss: 0.7751 - acc: 0.8099 - val

Epoch 00027: val_acc improved from 0.81100 to 0.84430, saving model to /content/saved_model/ci
Epoch 28/200
Learning rate:  0.001
1563/1563 [==============================] - 668s 427ms/step - loss: 0.7702 - acc: 0.8101 - val

Epoch 00028: val_acc did not improve from 0.84430
Epoch 29/200
Learning rate:  0.001
1563/1563 [==============================] - 662s 424ms/step - loss: 0.7594 - acc: 0.8139 - val
```

```
Epoch 00029: val_acc did not improve from 0.84430
Epoch 30/200
Learning rate:  0.001
1563/1563 [==============================] - 664s 425ms/step - loss: 0.7575 - acc: 0.8137 - val

Epoch 00030: val_acc did not improve from 0.84430
Epoch 31/200
Learning rate:  0.001
1563/1563 [==============================] - 668s 427ms/step - loss: 0.7533 - acc: 0.8178 - val

Epoch 00031: val_acc did not improve from 0.84430
Epoch 32/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 425ms/step - loss: 0.7410 - acc: 0.8198 - val

Epoch 00032: val_acc did not improve from 0.84430
Epoch 33/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.7369 - acc: 0.8247 - val

Epoch 00033: val_acc did not improve from 0.84430
Epoch 34/200
Learning rate:  0.001
1563/1563 [==============================] - 665s 425ms/step - loss: 0.7355 - acc: 0.8219 - val

Epoch 00034: val_acc did not improve from 0.84430
Epoch 35/200
Learning rate:  0.001
1563/1563 [==============================] - 668s 427ms/step - loss: 0.7338 - acc: 0.8223 - val

Epoch 00035: val_acc did not improve from 0.84430
Epoch 36/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.7272 - acc: 0.8244 - val

Epoch 00036: val_acc did not improve from 0.84430
Epoch 37/200
Learning rate:  0.001
1563/1563 [==============================] - 663s 424ms/step - loss: 0.7205 - acc: 0.8290 - val

Epoch 00037: val_acc did not improve from 0.84430
Epoch 38/200
Learning rate:  0.001
1095/1563 [====================>...] - ETA: 3:12 - loss: 0.7156 - acc: 0.8279Buffered data was
```

1.3.11 3.K Evaluate Model

1. Last Model
2. Best Model
3. Confusion Matrix of Best

3.K.a Last Model

In [90]:
```
scores = model.evaluate(x_test, y_test, verbose=1)
print('Test loss:', scores[0])
print('Test accuracy:', scores[1])
```

```
10000/10000 [==============================] - 25s 2ms/step
Test loss: 0.8169160022735595
Test accuracy: 0.8198
```

3.K.b Best Model

In [11]:
```
from keras.models import load_model
best_model = load_model('drive/My Drive/IUST-DIP/cifar10_ResNet110v2_model.039.h5')
scores = best_model.evaluate(x_test, y_test, verbose=1)
print('Test loss:', scores[0])
print('Test accuracy:', scores[1])
```

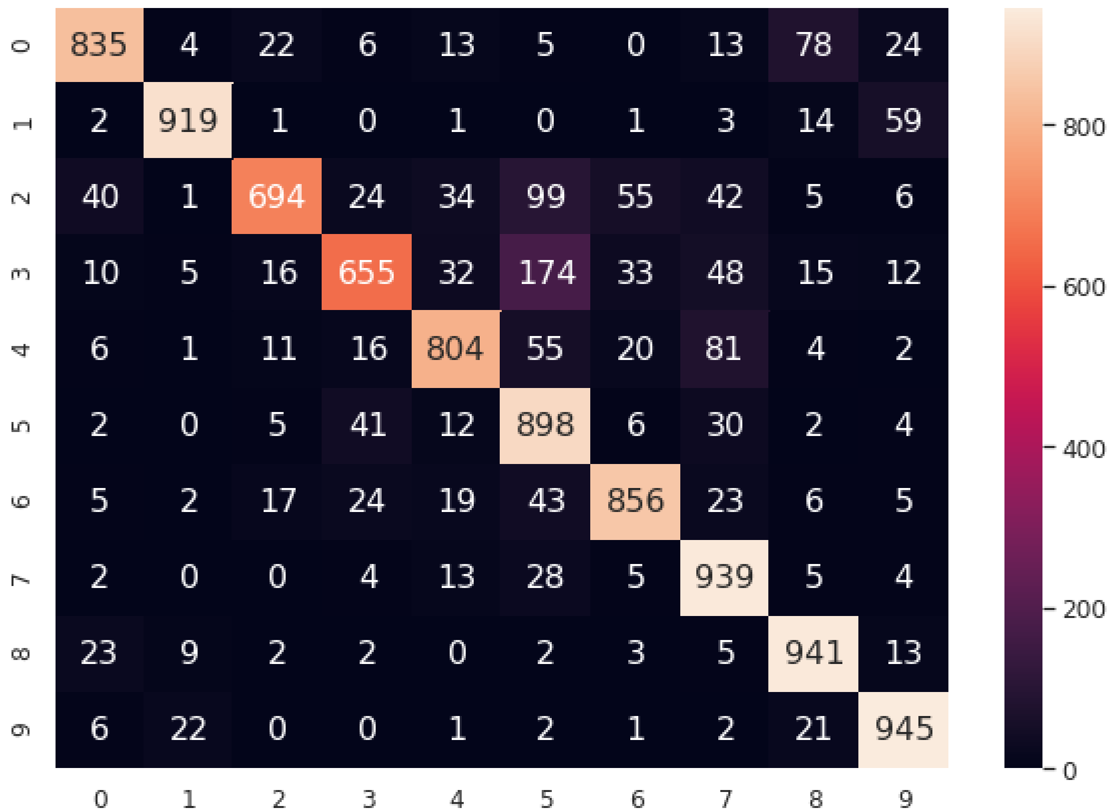
```
10000/10000 [==============================] - 13s 1ms/step
Test loss: 0.6827598980903625
Test accuracy: 0.8486
```

3.K.c Confusion Matrix of Best Model

In [28]:
```
from sklearn.metrics import confusion_matrix
pred = best_model.predict(x_test, verbose=1)
cm = confusion_matrix(np.argmax(y_test, axis=-1), np.argmax(pred, axis=-1))

df_cm = pd.DataFrame(cm, range(10), range(10))
plt.figure(figsize = (10,7))
sns.set(font_scale=1.1)
sns.heatmap(df_cm, annot=True, annot_kws={"size": 16}, fmt='g')

plt.show()
```



1.3.12 3.L 10 Worst Predictions

```
In [ ]: wrongs = np.argmax(best_model.predict(x_test), axis=-1) != np.argmax(y_test, axis=-1)
        wrongs_idx = (wrongs*1).nonzero()[0]
        wrongs_probs = best_model.predict(x_test[wrongs_idx])
        wrongs_true_prob = np.diag(wrongs_probs[:, np.argmax(y_test, axis=-1)])
        worst_preds_idx = np.argpartition(wrongs_true_prob, 10)[:10]
        worst_preds_idx = wrongs_idx[worst_preds_idx]
```

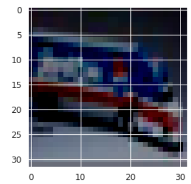
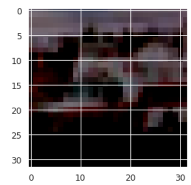
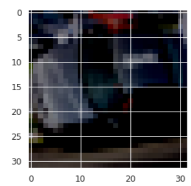
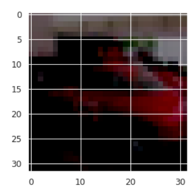
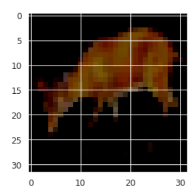
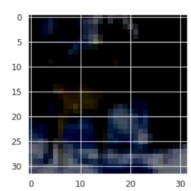
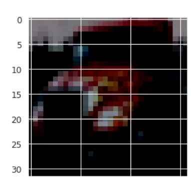
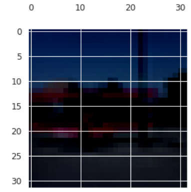
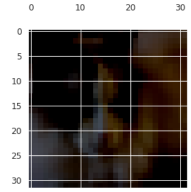
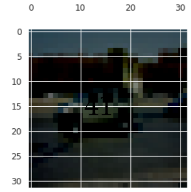
```
In [94]: fig = plt.figure(figsize=(50, 50))  # width, height in inches

         for i, idx in enumerate(worst_preds_idx):
             sub = fig.add_subplot(len(worst_preds_idx), 1, i + 1)
             sub.imshow(x_test[idx,:,:], interpolation='nearest')
             print('Predicted:', np.argmax(pred[idx], axis=-1), '------ True class', np.argmax
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255]


Predicted: 0 ------ True class 9
Predicted: 9 ------ True class 1
Predicted: 9 ------ True class 1
Predicted: 9 ------ True class 1
Predicted: 7 ------ True class 4
Predicted: 5 ------ True class 4
Predicted: 1 ------ True class 9
Predicted: 9 ------ True class 0
Predicted: 5 ------ True class 3
Predicted: 9 ------ True class 1
```


0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30

0 5 10 15 20 25 30
0 10 20 30