Training Deep Convolutional Neural Network

CPE393-2019 Coding in AI

Fasion MNIST dataset

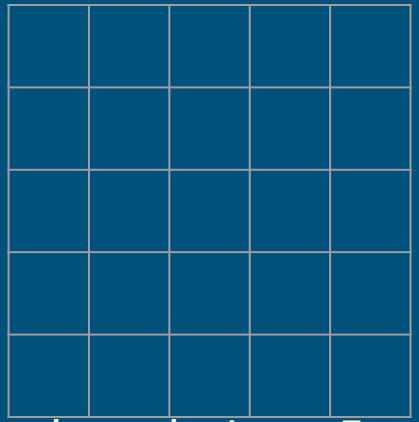
Label	Class	Label	Class
0	T-shirt/top	6	Shirt
1	Trouser	7	Sneaker
2	Pullover	8	Bag
3	Dress	9	Ankle boot
4	Coat		
5	Sandal		

Deep Convolutional Neural Network

```
cnn = models.Sequential()
cnn.add(layers.Conv2D(40, kernel_size=5, padding="same",input_shape=(28, 28, 1), activation = 'relu',name ='conv1_1'))
cnn.add(layers.MaxPool2D((2, 2)))
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(64, activation='relu'))
cnn.add(layers.BatchNormalization())
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Dense(10, activation='softmax'))
```

Deep Convolutional Neural Network

```
cnn = models.Sequential()
cnn.add(layers.Conv2D(40, kernel_size=5, padding="same",input_shape=(28, 28, 1), activation = 'relu',name ='conv1_1'))
cnn.add(layers.MaxPool2D((2, 2)))
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(64, activation='relu'))
cnn.add(layers.BatchNormalization())
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Dense(10, activation='softmax'))
```



kernel_size = 5

Padding = same can call HALF padding output of this padding will have same size of input

Padding = Valid output size don't has same size of input

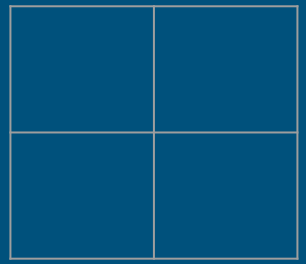
Ref :: http://cs231n.github.io/convolutional-networks/

Ref : https://keras.io/layers/convolutional/

Deep Convolutional Neural Network

Ref : https://keras.io/layers/convolutional/

```
cnn = models.Sequential()
cnn.add(layers.Conv2D(40, kernel_size=5, padding="same",input_shape=(28, 28, 1), activation = 'relu',name ='conv1_1'))
cnn.add(layers.MaxPool2D((2, 2)))
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(64, activation='relu'))
cnn.add(layers.BatchNormalization())
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Dense(10, activation='softmax'))
```

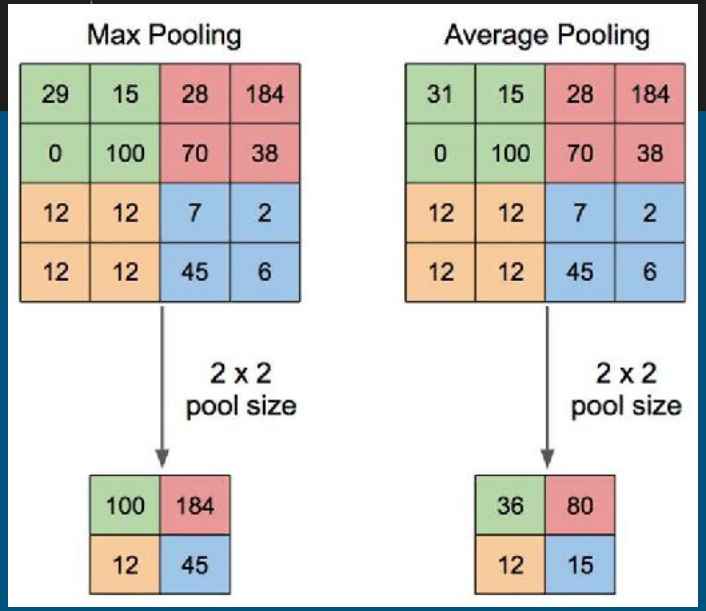


Max Pooling = 2

Pooling

Avg Pooling

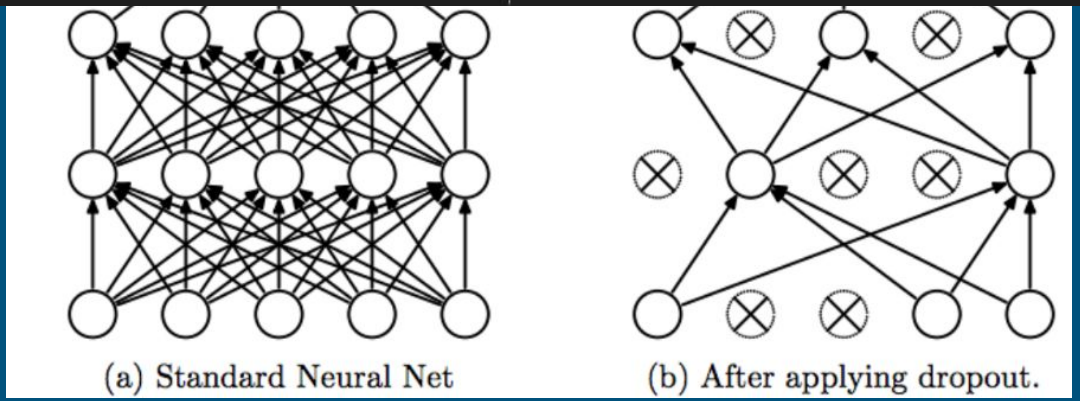
Max Pooling



Deep Convolutional Neural Network

```
cnn = models.Sequential()
cnn.add(layers.Conv2D(40, kernel_size=5, padding="same",input_shape=(28, 28, 1), activation = 'relu',name ='conv1_1'))
cnn.add(layers.MaxPool2D((2, 2)))
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(64, activation='relu'))
cnn.add(layers.BatchNormalization())
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Dense(10, activation='softmax'))
```

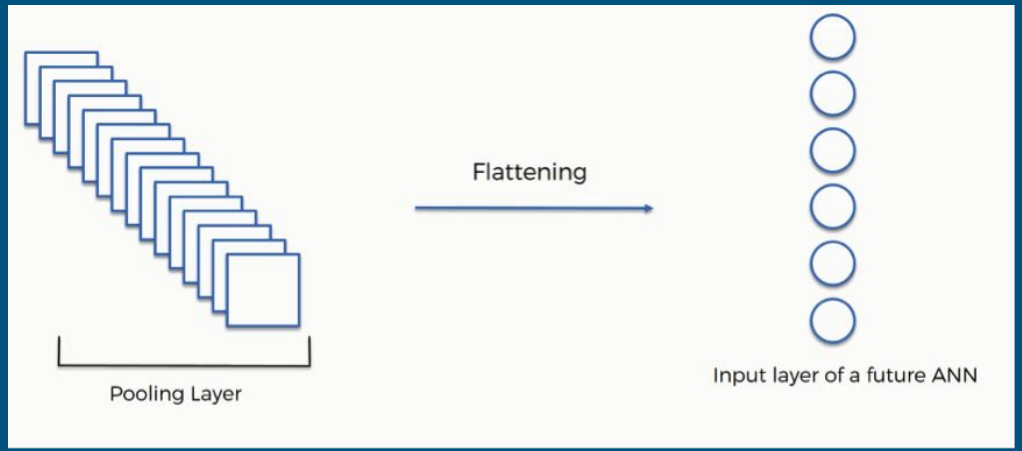
Drop out



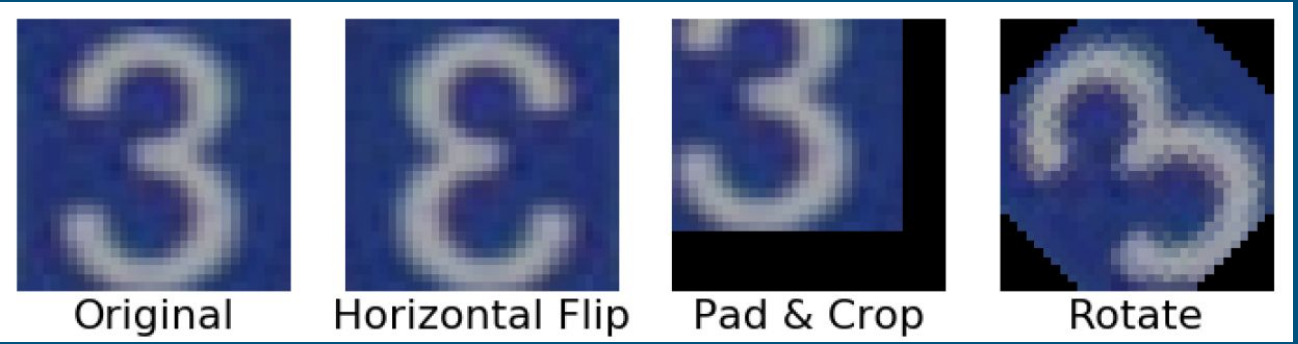
Deep Convolutional Neural Network

```
cnn = models.Sequential()
cnn.add(layers.Conv2D(40, kernel_size=5, padding="same",input_shape=(28, 28, 1), activation = 'relu',name ='conv1_1'))
cnn.add(layers.MaxPool2D((2, 2)))
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(64, activation='relu'))
cnn.add(layers.BatchNormalization())
cnn.add(layers.Dropout(0.25))
cnn.add(layers.Dense(10, activation='softmax'))
```

Flatten



Data augumentation



Data augumentation

```
image.ImageDataGenerator(rescale=1./255,
                         rotation_range=40,
                         width_shift_range=0.2,
                         height_shift_range=0.2,
                         shear_range=0.2,
                         zoom_range=0.2,
                         horizontal_flip=False)
```

Transfer learning

```
from keras.applications import vgg16

vgg = vgg16.VGG16(include_top=False,
                  weights='imagenet',
                  input_shape=(150,150,3))
```

Ref :: https://keras.io/applications/

Others ::

Xception

VGG16

VGG19

ResNet, ResNetV2

InceptionV3

InceptionResNetV2

MobileNet

MobileNetV2

DenseNet

Transfer learning

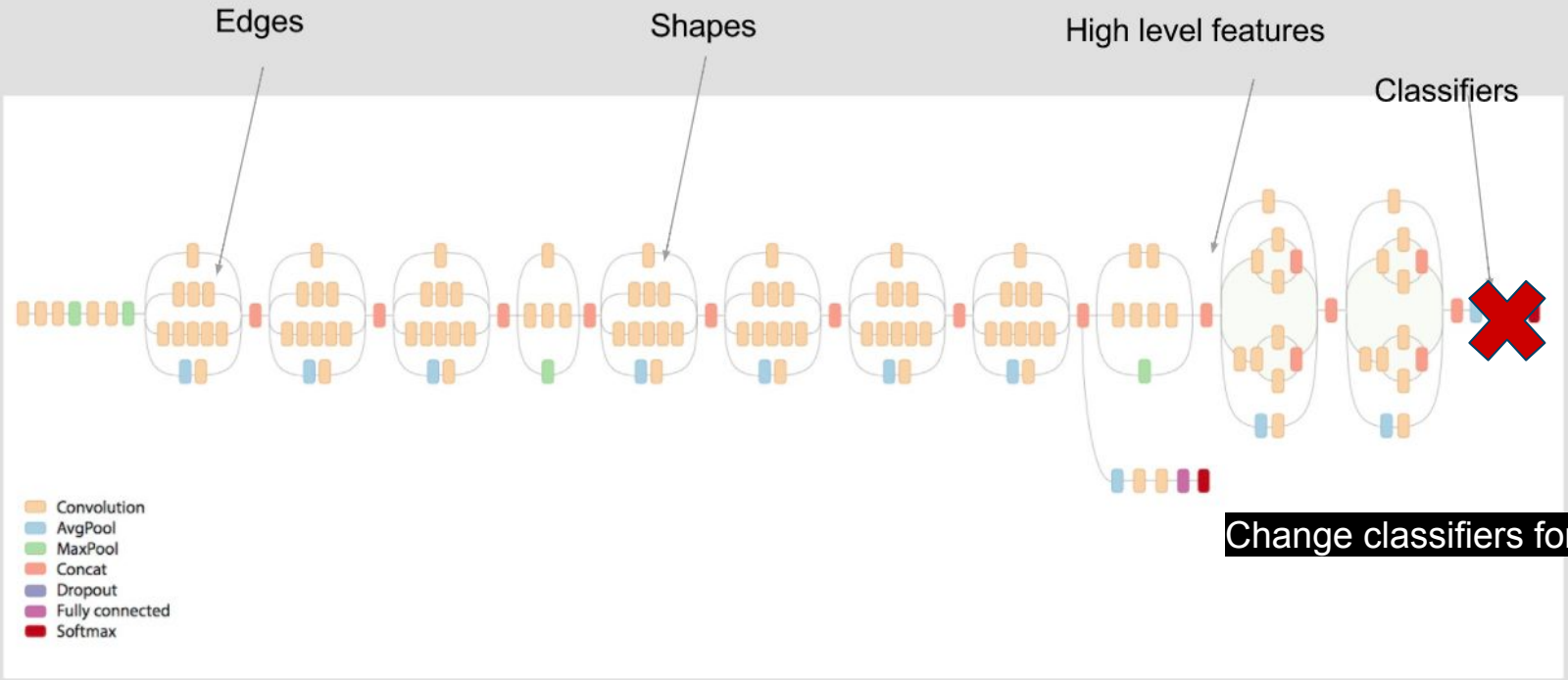
```
from keras.applications import vgg16

vgg = vgg16.VGG16(include_top=False,
                  weights='imagenet',
                  input_shape=(150,150,3))
```

include_top: whether to include the fully-connected layer at the top of the network.

weights: one of None (random initialization) or 'imagenet' (pre-training on ImageNet).

What does the layers learn?



Change classifiers for our task

Ref : https://medium.com/@14prakash/transfer-learning-using-keras-d804b2e04ef8

Transfer learning : Freeze layer

```
prev_cnn.trainable = False
```

Trainable = Fasle == No train / Change weight in this layer

```
# Freeze a specific layers
# Freeze first 3 layer
for i in range(3):
    prev_cnn.layers[i].trainable = False
```

Transfer learning : At own FC Layer for our task

```
# Time to create a new model
new_cnn = models.Sequential()

# Add convolutional layer as a pre-train network
new_cnn.add(prev_cnn)

# Define fully-connect layer
new_cnn.add(layers.Dense(256,activation='elu'))
new_cnn.add(layers.Dropout(0.2))
new_cnn.add(layers.Dense(10,activation='softmax',name='output'))

# Show how your network looklike
new_cnn.summary()
```