Session 9: Classifier 2 - Convolutional Neural Networks

Objectives

- Learn the basics of CNN based classifiers
- Train a CNN classifier and test its accuracy

ref:
http://cv-tricks.com/tensorflow-tutorial/training-convolutional-neural-network-for-image-classification/

What are Neural Networks?

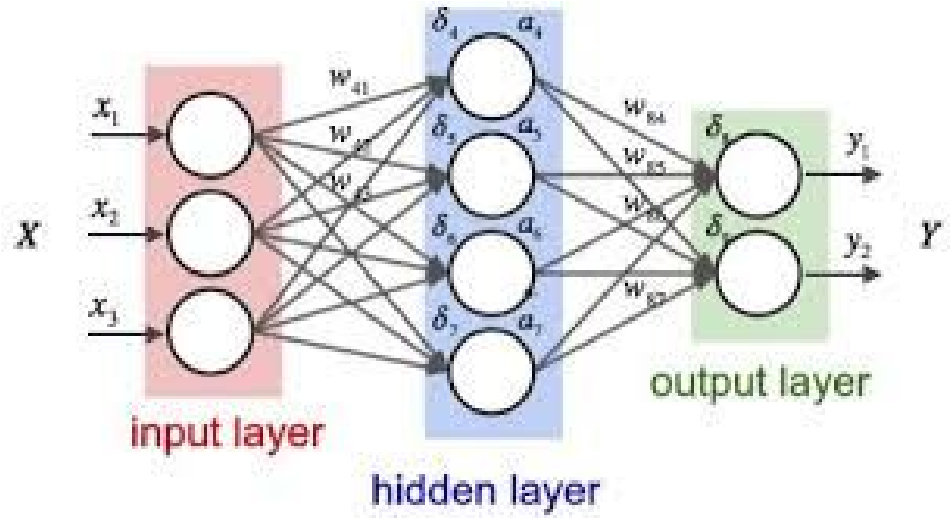
- Neural Networks are essentially mathematical models to solve an optimization problem.
- They are made of neurons, the basic computation unit of neural networks.
- A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b).
- This value is passed to a non-linear function called activation function(f) to produce the final output(activation) of a neuron.
- There are many kinds of activation functions. One of the popular activation function is Sigmoid, which is:

$$y = \frac{1}{1+e^{-z}}$$

- Other activation functions include Rectifified Linear Unit (RELU) and TanH

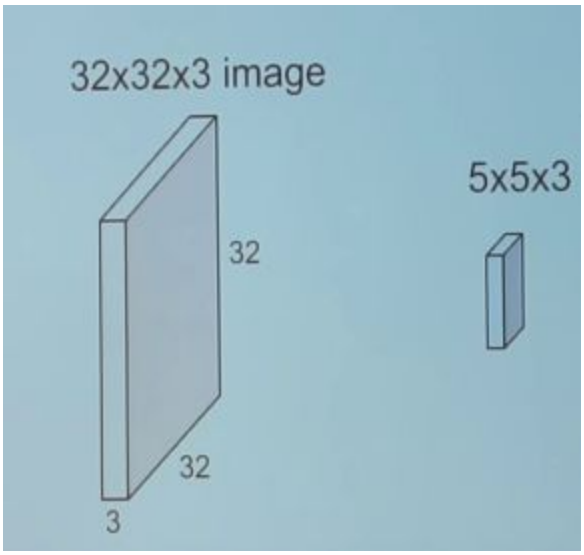
Layers:

If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks.

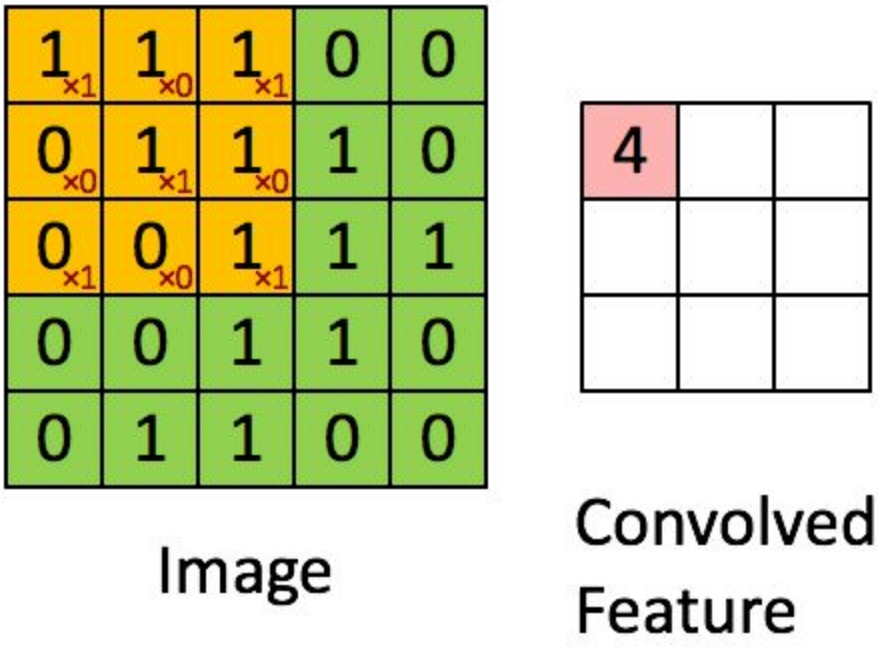


Convolutional Layer

In a convolutional layer, all neurons apply convolution operation to the inputs. The most important parameter in a convolutional neuron is the filter size, let's say we have a layer with filter size 5*5*3. Also, assume that the input that's fed to convolutional neuron is an input image of size of 32*32 with 3 channels.



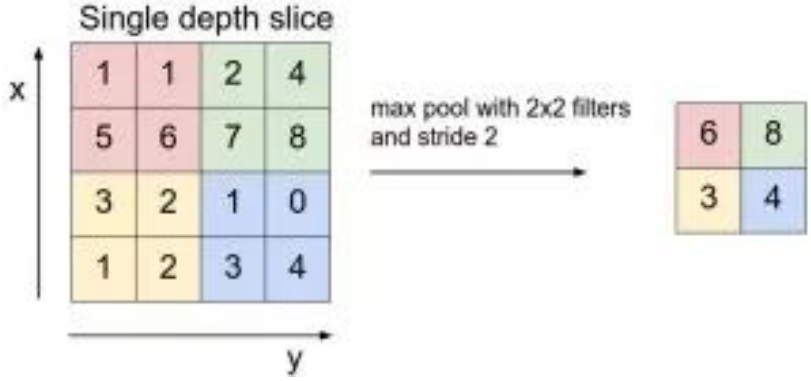
We shall slide convolutional filter over whole input image to calculate this output across the image:



In this case, we slide our window by 1 pixel at a time. If some cases, people slide the windows by more than 1 pixel. This number is called **stride**.

Pooling Layer

- Pooling layer is mostly used immediately after the convolutional layer to reduce the spatial size (only width and height, not depth).
- This reduces the number of parameters, hence computation is reduced.
- Less number of parameters also reduces overfitting.
- The most common form of pooling is Max pooling where we take a filter of size F*F and apply the maximum operation over the F*F sized part of the image.



Fully Connected Layer

If each neuron in a layer receives input from all the neurons in the previous layer, then this layer is called fully connected layer. The output of this layer is computed by matrix multiplication followed by bias offset.

Training

When we are trying to train a neural network, there are two fundamental things we need to do:

- Architecture
- Weights

The Architecture of the network:

- How do you arrange layers? which layers to use?
- How many neurons to use in each layer etc.?
- There are many standard architectures which work great for many standard problems, e.g. AlexNet, GoogleNet, InceptionResnet, VGG etc.

Correct weights/parameters:

Once you have decided the architecture of the network; the second biggest variable is the weights(w) and biases(b) or the parameters of the network. The objective of the training is to get the best possible values of the all these parameters which solve the problem reliably. For example, when we are trying to build the classifier between stop sign and warning sign, we are looking to find parameters such that output layer gives out higher probability of stop (than other classes) for all images of stop signs and higher probability of warning sign for all images of warning signs.

Backward Propagation

- You can find the best set of parameters using a process called **Backward propagation**.
- Start with a random set of parameters and keep changing these weights such that for every training image we get the correct output.
- There are many optimizer methods to change the weights that are mathematically quick in finding the correct weights, e.g. Gradient Descent.

Batch Training

For computational simplicity, not all training data is fed to the network at once. Rather, let's say we have total 1600 images, we divide them in small batches say of size 16 or 32 called **batch-size**. Hence, it will take 100 or 50 rounds (iterations) for complete data to be used for training. This is called one **epoch**, i.e. in one epoch the networks sees all the training images once.

Exercise 9-1: Train a CNN classifier using TensorFlow and verify its accuracy

Open the folder ex-9-1-CNN-Train.

TO-DO:
Copy the generated `data.p` and `test.p` files into working folder.

```
cp ../../ex-8-0-prepdata_classification/*.p .
```

Study the code in `signs-cnn.py`, which trains a CNN classifier and verifies the accuracy of the model. The CNN model is based on Yann LeCun's LeNet-5 model for handwritten and machine-printed character recognition.

TO-DO:
Complete the missing layers for the convolutional neural network