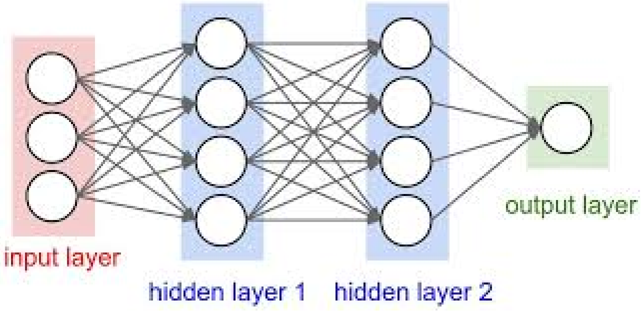
COVOLUTIONAL NEURAL NETWORKS FOR OBJECT RECOGNITION

1. **Task**

Given an image, the system should output the object class present in the image. Clearly this is a classification problem.

2. **Classification and Neural Networks**

Classification is the supervised problem of predicting output labels given an input.
Neural network is a classification algorithm that mimcs the human brain and helps to figure out complex decision boundaries.



A neural network has many layers of nodes, each node being a computative unit. The first layer is the input layer and the last layer is the output layer that outputs the probability of various output labels for the given input.
Training of neural networks is done using backpropagation which involves backpropagating the errors throughout the neural network, updating the weights of each layer in each iteration.

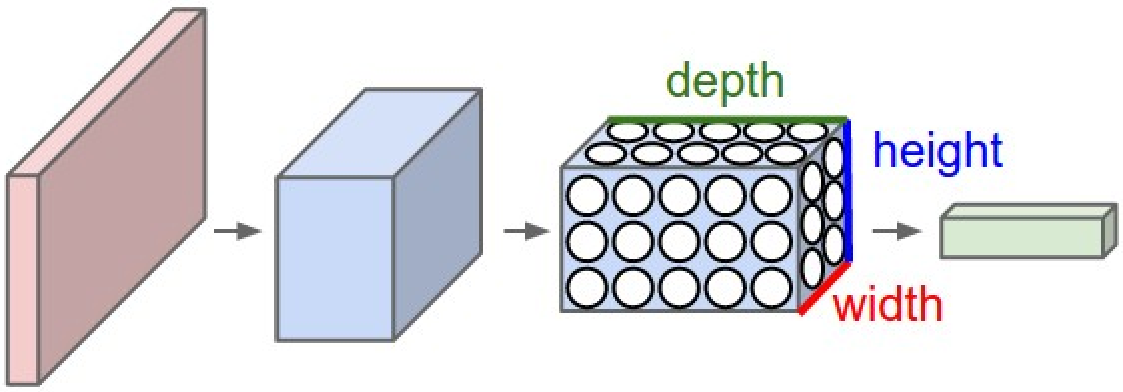
Steps involved in training the neural network:
1. Forward propagate throughout the network from the input
2. Calculate the loss and compute the gradients
3. Backpropagate the error
4. Update the weights of each layer during backpropagation

For more detailed understanding of neural networks and their intuition use the following links:

http://cs231n.github.io/neural-networks-1/
http://cs231n.github.io/neural-networks-2/
http://cs231n.github.io/neural-networks-3/

3. **Convolutional Neural Network**

Since the input is an image is of dimensions like 256x256x3 or 512x512x3, the dimensions of the input layer would be 196608 or 786432 input parameters respectively. This is computationally infeasible. To avoid this, we exploit the fact that our input are images. This lead to the invention of the convolutional neural networks.



The Convolutional Neural Network consists of three types of layers:

1. Convolutional Layer:

The Conv layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. The convolution layer consists of a set of filters and these filters are slided over the image and at each point the convolution of the filter with the image is computed. The CONV layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on a first layer of a ConvNet might have size 5x5x3 (i.e. 5 pixels width and height, and 3 because images have depth 3, the color channels). During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As we slide the filter over the width and height of the input volume we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Now, we will have an entire set of filters in each CONV layer (e.g. 12 filters), and each of them will produce a separate 2-dimensional activation map. We will stack these activation maps along the depth dimension and produce the output volume.

2. Pooling Layer:

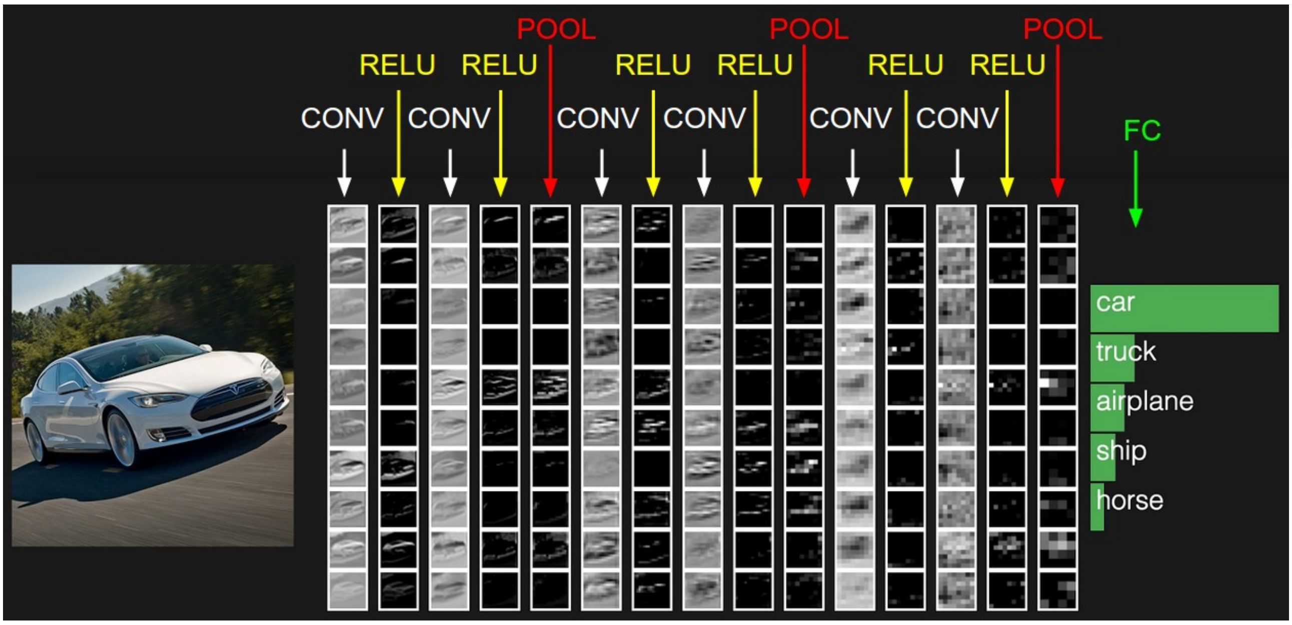
It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.

3. Fully Connected Layer:

The Fully Connected Layer is the typical neural network that consists of various layers with nodes.

For more information on how backprop works in a CNN and how the filters are learned:
http://cs231n.github.io/convolutional-networks/

Object Classifcation and CNN Structure:



Experiments conducted:

1) Object Classification:

Used the CIFAR100 dataset consisting of 50,000 images of 32x32x3. Trained for 200 epochs and achieved an accuracy of 65%

2) Human Emotion Detection:

Used the Jaffe Face Dataset and ran a simple Convolutional Architecture on it. Achieved an accuracy of 33%