Neural Network Concepts and Terms

# Basics of Neural Networks

1. Neuron – receives the input, processes it, and either generated an output or sends it to other neurons
2. Weights – initialized randomly and multiplied to each input; higher weight mean higher importance
3. Bias – added to the product of the input and weight and provides the final linear transformation of the input
4. Activation Function – the activation function is a non-linear transformation applied to the linearly transformed input and translates the input to output. Some common activation functions are
   1. Sigmoid – produced a smooth range of values from 0 to 1
   2. Rectified Linear Units (ReLU) – contains a constant derivative that provides values from 0 to infinity
   3. Softmax – usually used for classification problems and is similar to the sigmoid except the outputs all sum to 1
5. Neural Network – backbone of deep learning and its goal is to find an approximation of an unknown factor
6. Input/Output/Hidden Layer – the input layer is where the information enters the NN and the output is where the solution comes out. The hidden layer is where all the processing happens
7. Multi Layer Perceptron (MLP) – a neural network where all the layers are fully connected to each other
8. Forward Propagation – the forward movement through the NN
9. Cost Function – NN tried to predict the output as close as possible to the actual value. This accuracy is measured by cost function, also called the loss function. Minimizing this is the main objective
10. Gradient Descent – optimization algorithm for minimizing the cost.
11. Learning Rate – the amount of minimization in the cost function in each iteration; The rate of descent towards the minima of the cost function
12. Backpropagation – once the error of the network is determined it is backpropagated, or fed backwards, to the NN along with the gradient of the cost function to update the weights of the NN
13. Batches – an input that is divided into chunks before being fed into the NN. This makes the model more generalized
14. Epochs – a single training iteration of all batches in both forward and back propagation
15. Dropout – regularization of the NN to prevent overfitting; certain neurons in the hidden layer are dropped; various architectures of the NN are trained in different combinations and the output of all the various architectures are used to determine the final output
16. Batch Normalization – ensures that the distribution of data is the same as the next layer hoped for

# Convolutional Neural Networks

1. Filters – weight matrix which is multiplied by part of the image to generate a convolved output; the filter values are updated like weights during backpropagation
2. Convolutional Neural Network (CNN) – applied to image data; normal NN produce huge amounts of parameters, CNN reduce the amount of parameters
3. Pooling – done to reduce the number of parameters and prevent overfitting; pooling layer of filer size (2,2) with MAX operation is the most common
4. Padding – adding a layer of zeroes around the image so the output image has the same size as the input
5. Data Augmentation – addition of new data derived from the given data which might prove beneficial for prediction

# Recurrent Neural Network

1. Recurrent Neuron – The output of the neuron is sent back through the NN for t time; this give a more generalized output
2. Recurrent Neural Network (RNN) – used especially for sequential data where the previous output is used to predict the next one; these loops are in the hidden layer and give them the ability to store information for some time to predict the output; the output goes to the next layer only after completing all time stamps
   1. Backpropagation Through Time – used in RNN and is like backpropagation only the error is sent back through the unfolded network and updates those values
3. Vanishing Gradient Problem – arises when the gradient of the activation function is very small and tend to make the low weights so small the ‘vanish’ the further they go into the network; ReLU activation function avoids this
4. Exploding Gradient Problem – exact opposite of the vanishing gradient problem