Neural Networks and Fully Connected Neural Network Architecture

Fully Connected Neural Network Architecture

Introduction

Basically the architecture is how to arrange the different number of hidden layers and neurons.

Normally, hidden layers have multiple neurons and output layers have either one or multiple outputs corresponding to the number of classes that need to be classified. This can be thought of as replacing the output layer with a softmax function.

Anything more than 1 hidden layer is referred to as a deep neural network.

More layers may however lead to overfitting.

The output or activation of each layer is the same as the number of neurons.

Each neuron is like a linear classifier, therefore each neuron must have the same number of inputs as the previous layer.

So!

If the input has 4 dimensions, the next layer of neurons each have 4 inputs.

If that layer is three neurons, the next layer has 3 inputs and so on.

The correct architecture can be ascertained by looking at the accuracy of each architecture.

However deeper networks are harder to train due to the disappearing or vanishing gradient, aka. To perform gradient descent during back-propagation an activation function is used but the step on the gradient grows infinitely smaller.

To rectify this the ReLu function can be used. The ReLu function, short for Rectified, Linear Unit function is zero when any input is less than zero. It is only used on hidden layers (don’t want to short inputs?).

Networks have layers that help with training:

* Dropout layers help with overfitting
* Batch Normalization improves training
* Skip connections allow the training of deeper networks by connecting deeper layers during training

The hidden layers replace the kernels in SVM.