## *HYPERPARAMETER TUNING OF A NETWORK*

Tuning the hyperparameters effectively can lead to a massive improvement of the model.  Following are a few common hyperparameters we frequently work with in a deep neural network:

* Learning rate – α
* Momentum – β
* Adam’s hyperparameter – β1, β2, ε
* Number of hidden layers
* Number of hidden units for different layers
* Learning rate decay
* Mini-batch size

## Learning rate usually proves to be the most important among the above. This is followed by the number of hidden units, momentum, mini-batch size, the number of hidden layers, and then the learning rate decay.

## Now, the steps involved in hypertuning the parameters are :-

## Step 1 — Deciding on the network topology (not really considered optimization but is obviously very important)

## Step 2 — Adjusting the learning rate

## Learning rate controls the weight at the end of each batch

## The learning rate hyperparameter goes into the optimizer function which we will see below. Keras has a default learning rate scheduler in the optimizer that decreases the learning rate during the stochastic gradient descent optimization algorithm.

## C:\Users\Dell\Desktop\1_Dbr6vHLUeUk-fzDf9VWbUw.png

## Step 3 — Choosing an optimizer and a loss function

## When constructing a model and using it to make our predictions, we want to measure our success or failure by defining a “loss” function (or objective function). The goal of optimization is to efficiently calculate the parameters/weights that minimize this loss function.

## Step 4 — Deciding on the batch size and number of epochs

The **batch size** defines the number of samples that will be propagated through the network.

For instance, let’s say you have 1000 training samples and you want to set up a batch size equal to 100. The algorithm takes the first 100 samples (from 1st to 100th) from the training dataset and trains the network. Next, it takes the second 100 samples (from 101st to 200th) and trains the network again. We can keep doing this procedure until we have propagated all samples through the network.