

## **Problem solution fit:**

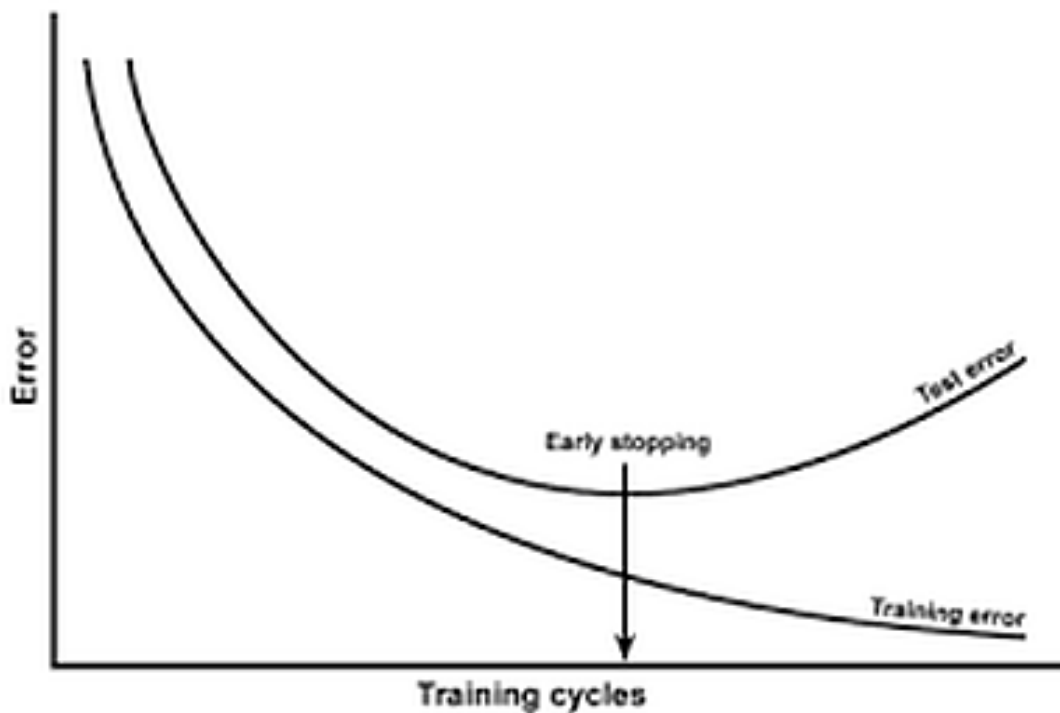
**The five techniques to prevent overfitting while training neural networks are:**

### **1. Simplifying The Model**

The first step when dealing with overfitting is to decrease the complexity of the model. To decrease the complexity, we can simply remove layers or reduce the number of neurons to make the network smaller. While doing this, it is important to calculate the input and output dimensions of the various layers involved in the neural network. There is no general rule on how much to remove or how large your network should be. But, if your neural network is overfitting, try making it smaller.

### **2. Early Stopping**

Early stopping is a form of regularization while training a model with an iterative method, such as gradient descent. Since all the neural networks learn exclusively by using gradient descent, early stopping is a technique applicable to all the problems. This method update the model so as to make it better fit the training data with each iteration. Up to a point, this improves the model's performance on data on the test set. Past that point however, improving the model's fit to the training data leads to increased generalization error. Early stopping rules provide guidance as to how many iterations can be run before the model begins to overfit.



This technique is shown in the above diagram. As we can see, after some iterations, test error has started to increase while the training error is still decreasing. Hence the model is overfitting. So to combat this, we stop the model at the point when this starts to happen.

### **3. Use Data Augmentation**

In the case of neural networks, data augmentation simply means increasing size of the data that is increasing the number of images present in the dataset. Some of the popular image augmentation techniques are flipping, translation, rotation, scaling, changing brightness, adding noise etcetera. For a more complete reference, feel free to checkout [alumentations](#) and [imgaug](#).

This technique is shown in the above diagram. As we can see, using data augmentation a lot of similar images can be generated. This helps in increasing the dataset size and thus reduce overfitting. The reason is that, as we add more data, the model is unable to overfit all the samples, and is forced to generalize.

#### **4. Use Regularization**

Regularization is a technique to reduce the complexity of the model. It does so by adding a penalty term to the loss function. The most common techniques are known as L1 and L2 regularization:

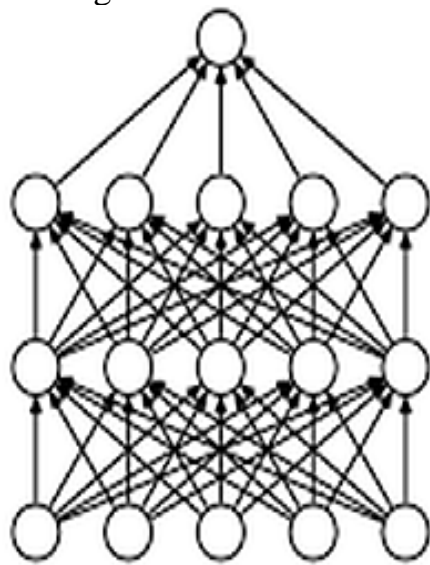
So which technique is better at avoiding overfitting? The answer is — it depends. If the data is too complex to be modelled accurately then L2 is a better choice as it is able to learn inherent patterns present in the data. While L1 is better if the data is simple enough to be modelled accurately. For most of the computer vision problems that I have encountered, L2 regularization almost always gives better results.

However, L1 has an added advantage of being robust to outliers. So the correct choice of regularization depends on the problem that we are trying to solve.

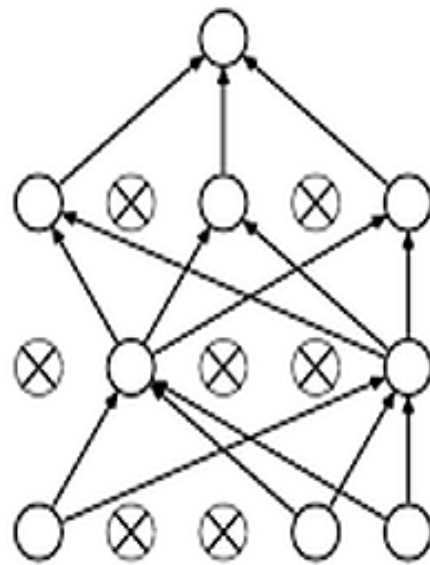
#### **5. Use Dropouts**

Dropout is a regularization technique that prevents neural networks from overfitting.

Regularization methods like L1 and L2 reduce overfitting by modifying the cost function. Dropout on the other hand, modify the network itself. It randomly drops neurons from the neural network during training in each iteration. When we drop different sets of neurons, it's equivalent to training different neural networks. The different networks will overfit in different ways, so the net effect of dropout will be to reduce overfitting.



(a) Standard Neural Net



(b) After applying dropout.