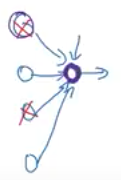
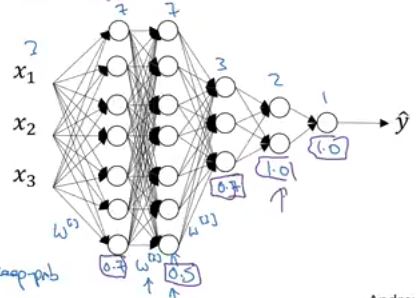
**Lec 48: Why does Droput work?**

Consider the following network with a single unit having 4 inputs:



While performing dropout (0.5) any two units of the previous layer can be dropped. Hence it is not advisable to have extremely high weight for any one particular feature. In this case, one should spread out the weights to all the units. This shrinks the squared norm of the weights. And similar to L2 regularization it prevents overfitting.

For the following network it is also possible to vary the ‘keep\_prob’ to different values for different layers:



The above is used when one is worried that some layers would overfit more than others. Consider the 2 layer having 7 hidden units, whose weights matrix is 7x7. This layer is likely to overfit, hence one can apply more dropout to this layer and apply a lower dropout to the other layers.

One problem with having different dropouts for different layers is that, there are many more hyperparameters to tune.

Dropouts are usually used in Computer vision more than other disciplines.

The downside of having dropouts is that the cost function ‘J’ is no longer a simple function which can be plotted to visualize gradient descent. This is because new terms get added for different dropouts used in each layer, which increases the hyperparameters to the function. This makes it difficult to visualize the gradient descent. One thing that can be done is to first turn off dropouts for every layer and visualize the cost function graph. If it is having a decreasing tendency, then turn on the dropout.

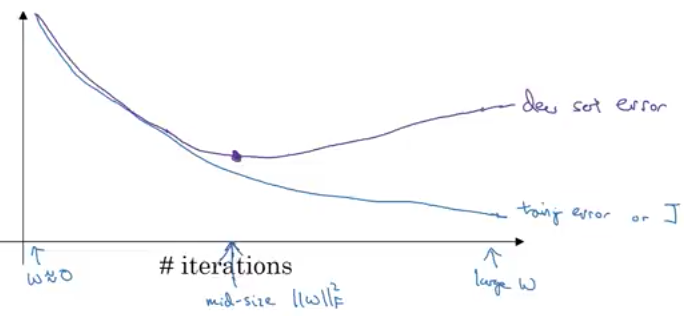
**Lec 49: Other Regularization Methods**

Apart from L2 regularization and dropouts there are many other techniques of regularization.

One technique is to get more data. But if getting data is expensive, then one can augment them. This greatly should reduce overfitting because now there are many more variations to the same data.



Early stopping is another regularization method to prevent overfitting.



Initially while training the weights are less and as you train they increase. Beyond a certain point they are so high that validation error increases. One must find the optimal point to stop iterating and save the model at that point. At that point the weights would be mid-sized hence reducing overfitting.

Early stopping has one downside:

Usually while training a network the first task is to optimize the cost function (J) and then work towards avoiding overfitting using regularization methods. While using early stopping this essence is lost, because you have to perform both cost minimization and overfitting avoidance together. As a result none of the two are optimally achieved.

The other way around this is to use L2 regularization where one has to try various values to get an optimal solution. This is computationally expensive though. In early stopping one advantage is getting the optimal set of weights in one descent without having to try various values.