We use L to denote the number of layers in the network.

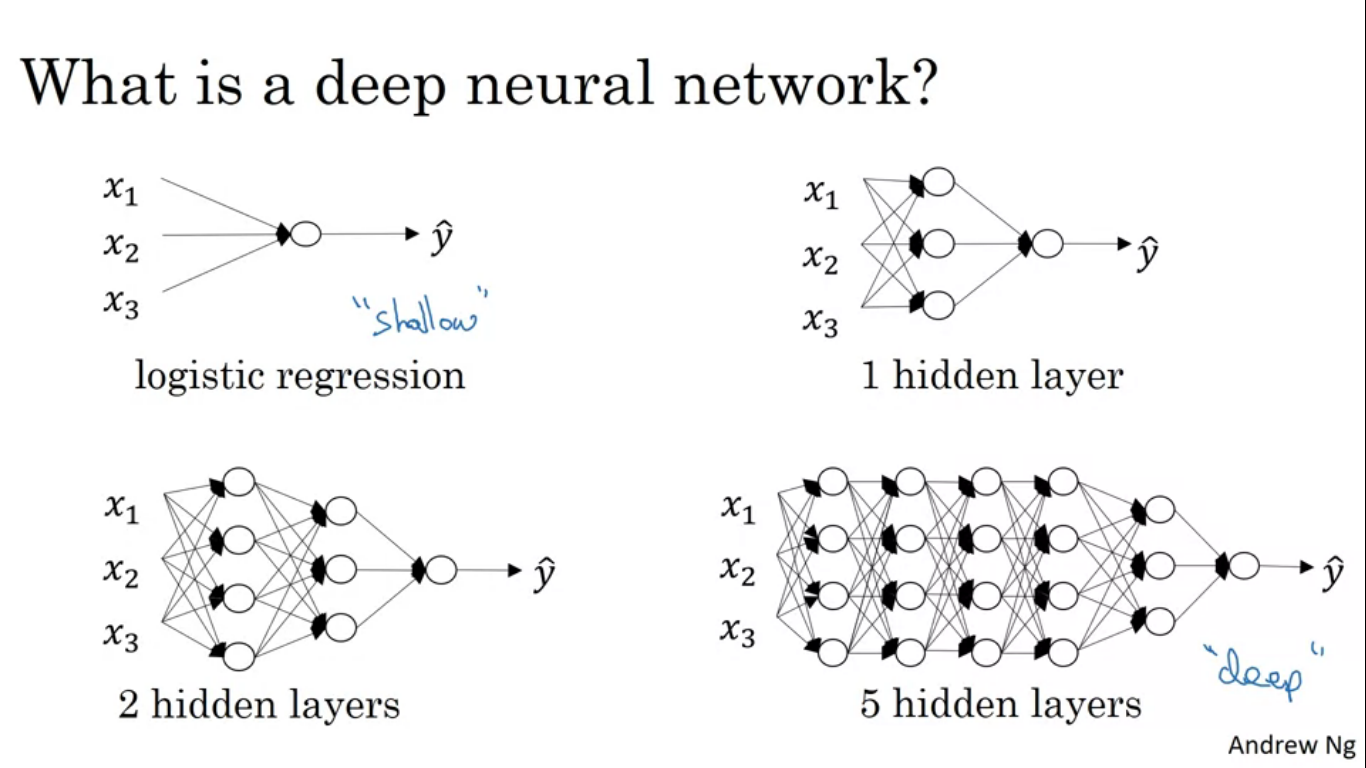
n[l] to denote number of units in layer l.

a[l] to denote the activation in layer l.

w[l] for computing the value Z[l].

Input features are called X

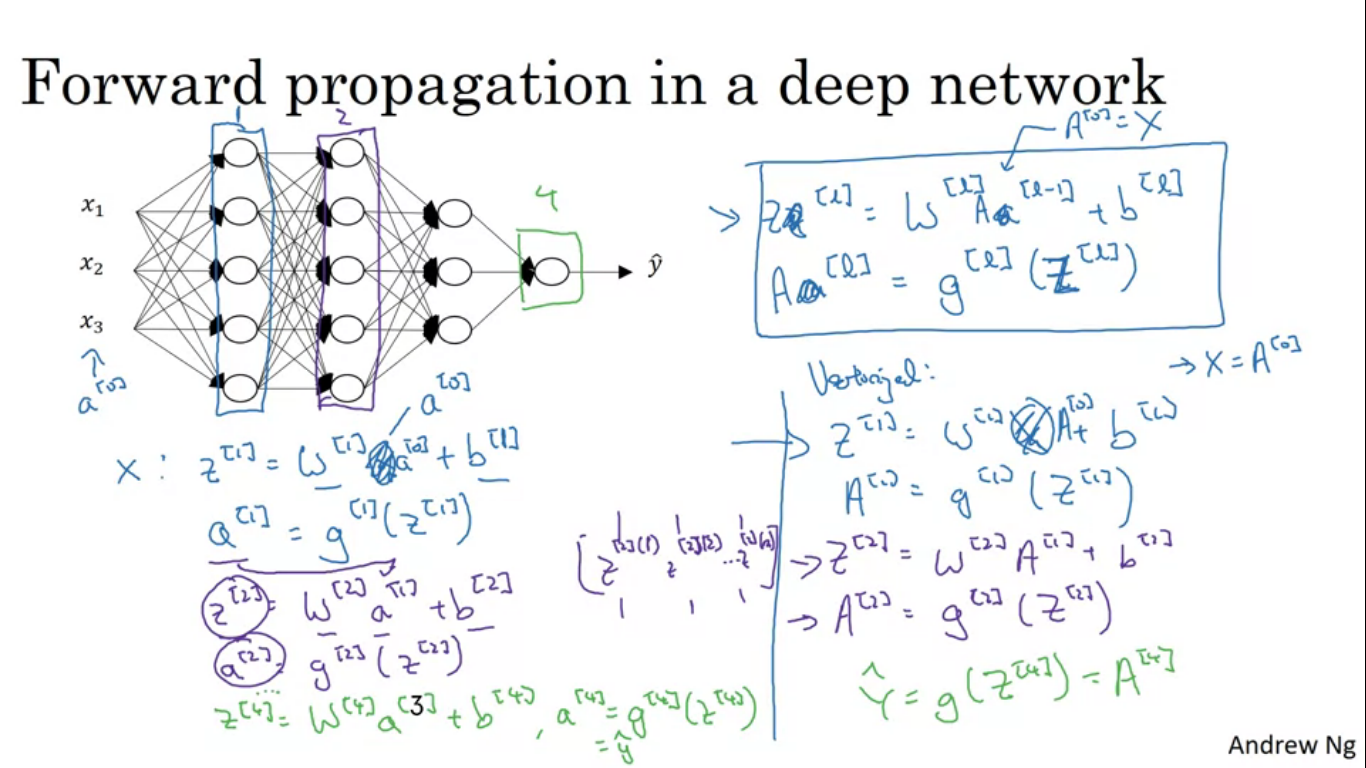
a[0]=X and a[L] = yhat.



**Forward Propagation in deep network**

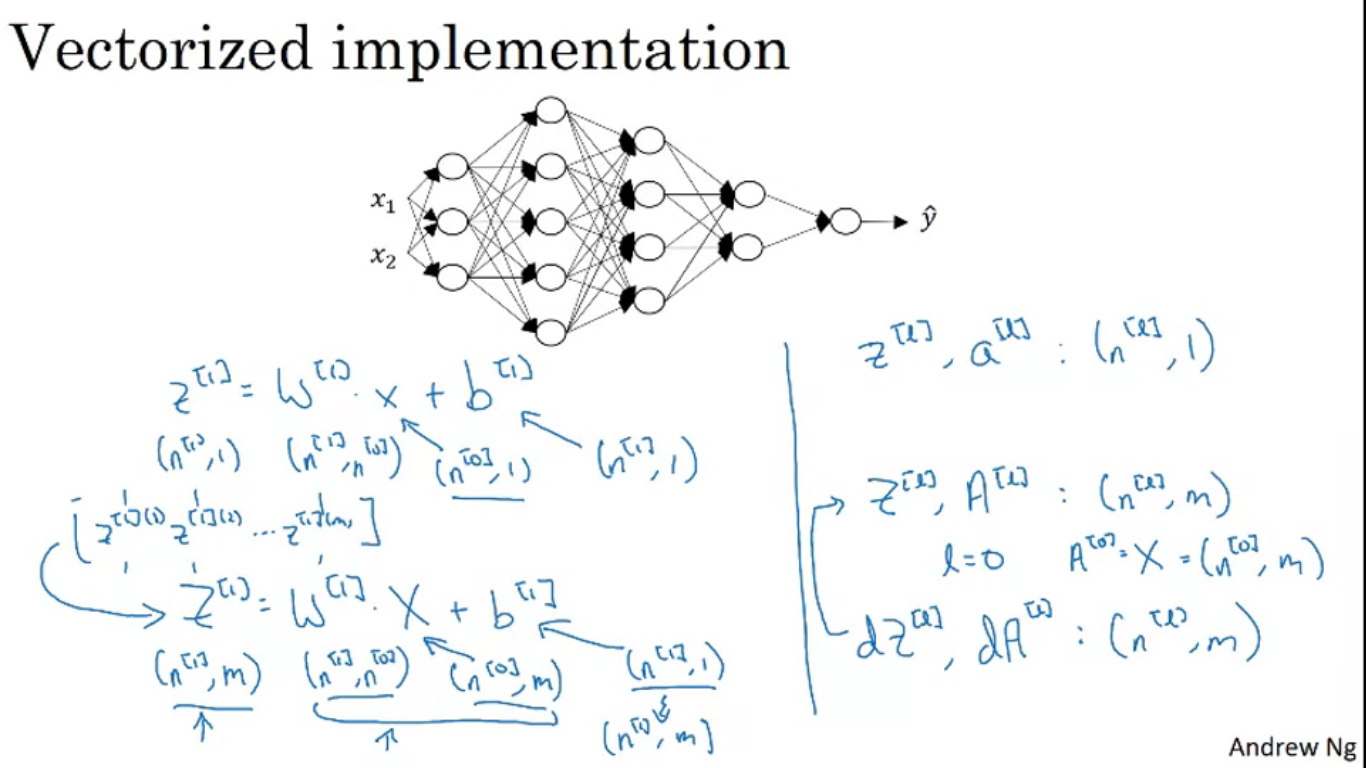
**Z[l] = w[l]a[l-1]+b[l]**

**a[l] = g[l](Z[l])**

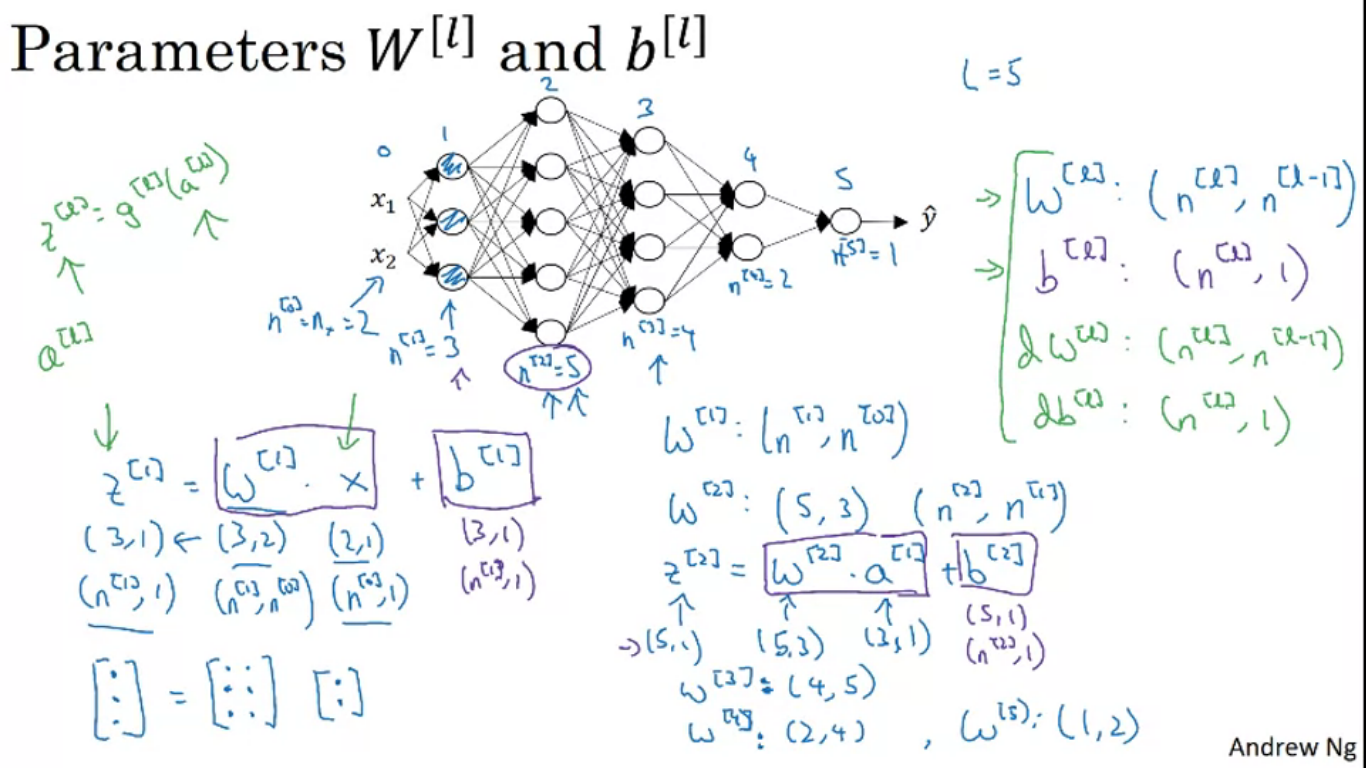
****

above done for single training example, to do it for the whole training set/ vectorized version

to compute things for layers 1 to L, we might need a for loop, to compute the activations for layer 1 to .

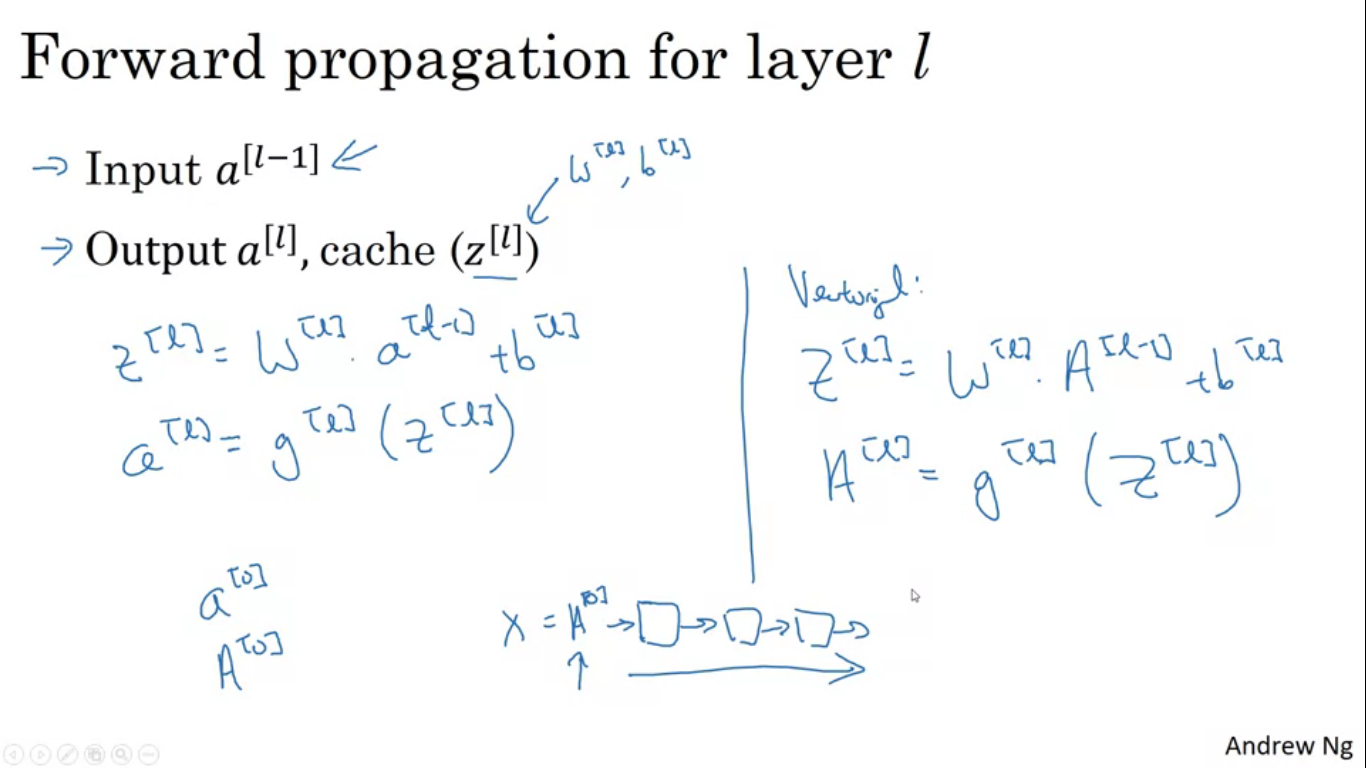


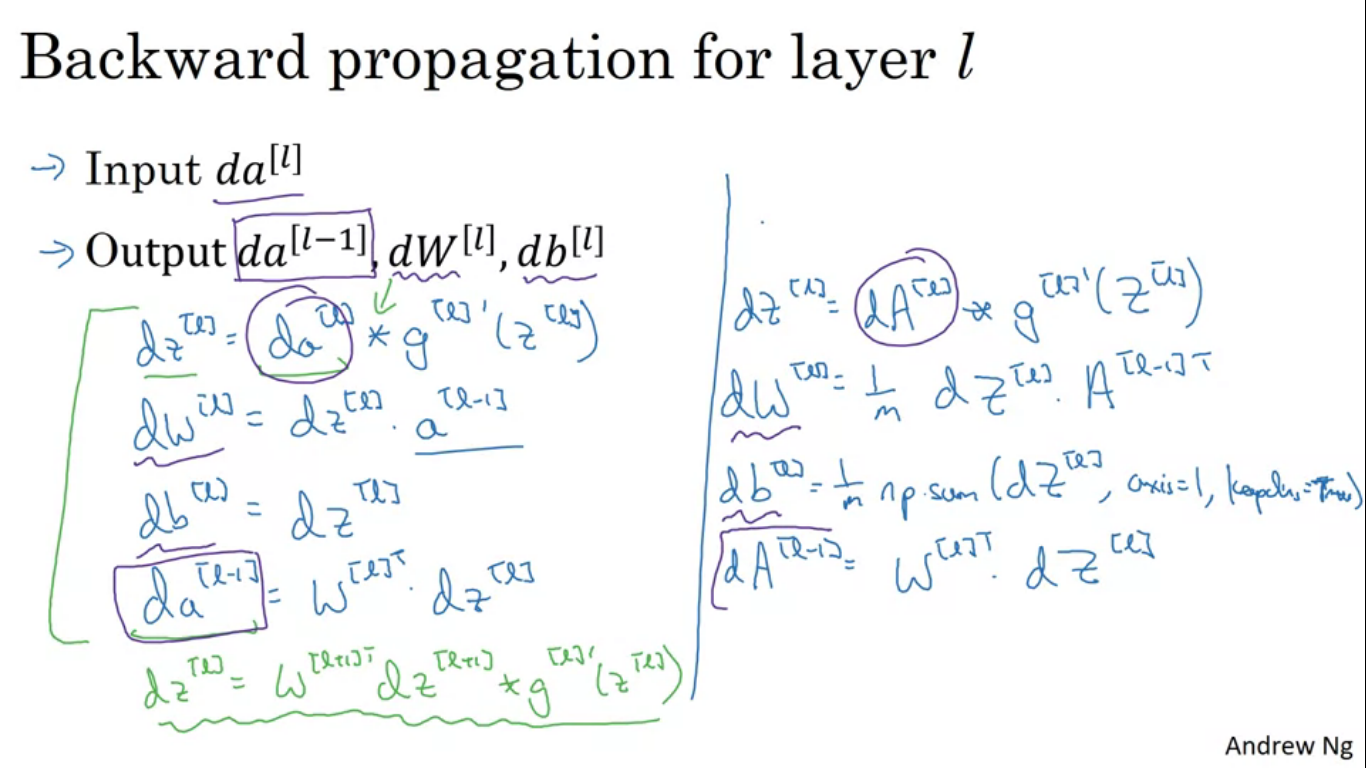
Working through the dimensions of matrix we are working with to make our code bug free

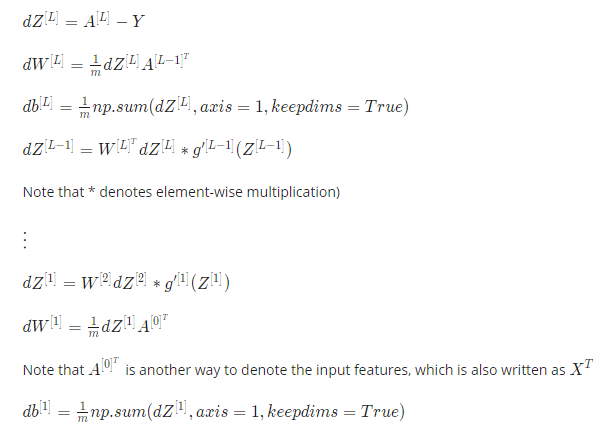


Parameters and hyperparameters optimization

Parameters are W and b. Other things we need to tell our leaning algorithm, such as learning rate alpha, or maybe number of iterations of gradient descent. Other parameters include number of hidden layers or number of hidden units , choice of activation fn. Hyper parameters control the parameters W and b.







It is usually very difficult to know what is the best value of the hyperparameters. So you might have to try out many different values.