output = model(input)

Backpropagation

<https://www.youtube.com/watch?v=Ilg3gGewQ5U>

The scenario here is that we have an MNIST network. The input is 784 neurons, 2 hidden layers with 16 neurons, and an output layer of 10 neurons each representing a single digit from MNIST.

We pass a number 2 digit. The output at the end is 0.2 after passed through the network. The target we want is 1.0. For this example, we will ignore the other classes for now. We want to increase the output (0.2) to the target (1.0). We do this by looking at the neurons that are fed into this output neuron. We have w0 which is the weight for neuron 0 in this hidden layer. We have a0 which is the collective activation from the neurons fed into neuron 0 in this hidden layer. All of these neurons are summed along with a bias for the target neuron.

How do we update the 0.2 to change into 1.0? Increase b and w or change a. The higher weighted neurons have a bigger effect and influence on the target neuron itself.

Essentially we want to propagate backwards, the changes we want to make for a neuron. For example, one neuron in the second hidden layer will be updated in 10 different ways. Each way corresponding to each of the output neurons.

<https://www.youtube.com/watch?v=tIeHLnjs5U8> 3Blue1Brown Backprop calculus

Once we have all the nudges from each training data. We average over all of the training data to get the negative gradient of the cost function.

How much does the cost C0 change with respect to the weight of the neuron before it?

[Single neuron network, 2 hidden layers, 0.48 2nd layer, 0.86 output]

“We don’t care which neurons are nudged up and down for gradient descent, we care about the ones that give the most bang for your buck”.