Backprop Example from Class

Dan Ventura

March 5, 2009

Abstract

Here are the weight updates according to Backpropagation for the MLP example we presented in class. Note that for readability we sometimes round to three significant digits, but all calculations were done without rounding.

1 The setup

Given the network shown in Figure 1, the training example $(0.3, 0.7) \rightarrow 0$, and a learning rate $\eta = 0.1$, compute the weight updates $\triangle w_{12}$, $\triangle w_{13}$, $\triangle w_{24}$, $\triangle w_{25}$, $\triangle w_{34}$, $\triangle w_{35}$.

2 The Forward Pass

We will represent the network weights as two matrices, W_1, W_2 and input and hidden unit activations as vectors and compute the forward activity of the network as $\sigma(W_2\sigma(W_1\vec{x}))$. So, computing the innermost function, which gives us the dot product of the inputs with the hidden layer weights gives

$$\vec{y} = W_1\vec{x} = \left[\begin{array}{cc} 0.23 & -0.79 \\ 0.1 & 0.21 \end{array}\right]\left[\begin{array}{c} 0.3 \\ 0.7 \end{array}\right] = \left[\begin{array}{c} -0.484 \\ 0.177 \end{array}\right]$$

We then squish this to yield the output vector for the hidden layer nodes (here, we apply the $\sigma()$ function to each member of the vector, yielding another vector of the same dimension):

$$\vec{z} = \sigma(\vec{y}) = \sigma(\left[\begin{array}{c} -0.484 \\ 0.177 \end{array}\right]) = \left[\begin{array}{c} 0.38130803 \\ 0.54413484 \end{array}\right]$$

This vector $\vec{z} = \sigma(W_1\vec{x})$ acts as inputs to the output layer, and the process is repeated to compute the final network output:

$$\vec{a} = W_2\vec{z} = \left[\begin{array}{cc} -0.12 & -0.88 \end{array}\right]\left[\begin{array}{c} 0.381 \\ 0.544 \end{array}\right] = \left[\begin{array}{c} -0.5245956 \end{array}\right]$$

and finally, squishing again, we have the output of the network for the input vector $[0.3, 0.7]^T$:

$$\vec{o} = \sigma(\vec{a}) = \sigma(\left[\begin{array}{c} -0.525 \end{array}\right]) = \left[\begin{array}{c} 0.37177825 \end{array}\right]$$

3 Backward Error Propagation

Now, to compute weight updates, we compute error and propagate it backward through the network, starting at the output layer. Recall that the weight update equation is

$$\triangle w_{ji} = \eta \delta_j x_i$$

For the output layer,

$$\delta_1 = (t_1 - o_1) o_1 (1 - o_1) = (0 - 0.372) 0.372 (1 - 0.372) = -0.0868322$$

(remember that the minus sign that is produced by taking the derivative is negated by the minus sign that results from going *down* the gradient) and using

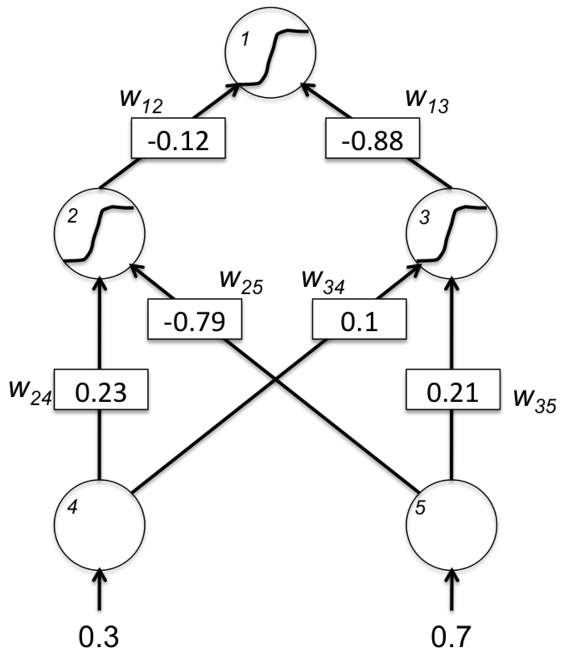

Figure 1: Example network from in-class example. The inputs are shown; the target output value is 0. Note that the sigmoid is applied at both the hidden and output layer units.

this value, we can now compute weight updates for the output layer weights as follows:

$$\triangle w_{12} = \eta \delta_1 x_2 = 0.1(-0.087)0.381 = -0.00331098$$

$$\triangle w_{13} = \eta \delta_1 x_3 = 0.1(-0.087)0.544 = -0.00472484$$

Notice that these updates (at least the direction of the updates) makes sense. The output of the network (0.372) is larger than the target. Therefore, we want to make it smaller. We make it smaller by making the weights coming into the output node smaller, and therefore, our weight update is negative. Also, notice that the update's magnitude is very small; training an MLP with backprop is slow (and this is with a relatively large learning rate).

Now, we compute the indirect errors at the hidden layers by propagating the output δ values back. In this case, there is only a single output node, so we have

$$\triangle \delta_2 = \delta_1 w_{12} o_2 (1 - o_2) = (-0.087)(-0.12)(0.381)(1 - 0.381) = 0.00245817$$

which allows us to compute the weight updates:

$$\triangle w_{24} = \eta \delta_2 x_4 = 0.1(0.002)0.3 = 0.0000737452$$

$$\triangle w_{25} = \eta \delta_2 x_5 = 0.1(0.002)0.7 = 0.000172072$$

and

$$\triangle \delta_3 = \delta_1 w_{13} o_3 (1 - o_3) = (-0.087)(-0.88)(0.544)(1 - 0.544) = 0.01895425$$

which allows us to compute the last two weight updates:

$$\triangle w_{34} = \eta \delta_3 x_4 = 0.1(0.019)0.3 = 0.000568627$$

$$\triangle w_{35} = \eta \delta_3 x_5 = 0.1(0.019)0.7 = 0.001326797$$