Backprop

Lech Szymanski

August 2021

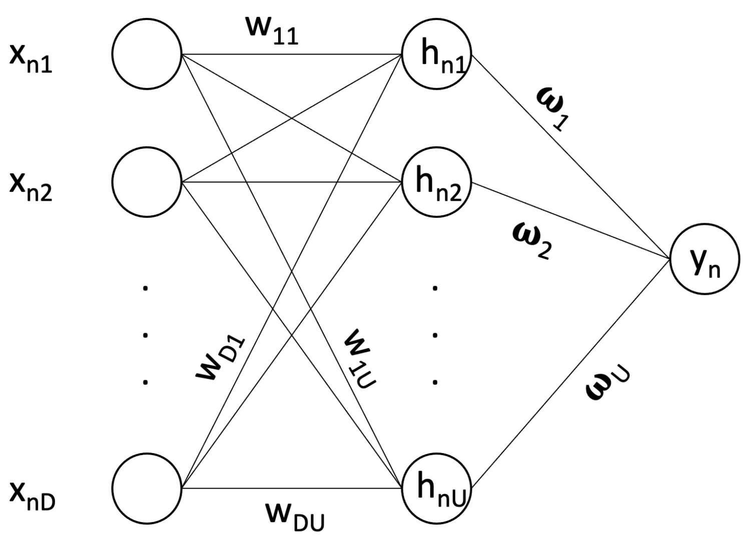

Figure 1: Your network

1 Summary

I hope this makes sense. The output computations for the feed-forward operation are given in Equations 1,2,3. The updates to weights and biases due to backprop are given in Equations 6, 8, 10 and 13. The only thing you need to decide on is how many neurons U in your hidden layer and what learning parameters μ to use (something small, like 0.001 or 0.0001 should be fine).

2 Details

OK, based on the image above, I assume your have D inputs into your network and N training points, so x_{nd} is the d-th component of then n-th input. Let's say you have U neurons in the first layer; the weights are such that w_{du} connects d-th component of the input to u-th neuron. Figure doesn't show biases, but it's a good idea to have bias, so b_u will be bias of the u-th neuron. Now, the let's define activity of neurons in the hidden layer - it's the sum of weighted inputs and bias...so, the activity of neuron u for input n is

$$v_{nu} = \sum_{d=1}^{D} x_{nd} w_{du} + b_u \tag{1}$$

Now, assuming you have a ReLU function, let's define output of the same neuron as:

$$h_{nu} = \begin{cases} v_{nu} & v_{nu} > 0 \\ 0 & v_{nu} \leq 0 \end{cases} \tag{2}$$

Next, assuming you have single output of the network, let's define it as

$$y_n = \sum_{u=1}^{U} h_{nu} \omega_u + \beta, \tag{3}$$

where ω_u is the weight connecting neuron u to the output and β is the bias of the output.

Now, suppose for each input $\mathbf{x}_n = \begin{bmatrix} x_{n1} & \ldots x_{nD} \end{bmatrix}$ you have a target output t_n. Let's define error

$$e_n = t_n - y_n \tag{4}$$

and total mean squared error (MSE) as:

$$J = \frac{1}{2N} \sum_{n=1}^{N} e_n^2 \tag{5}$$

And here comes the backprop. The update to your weights in the last layer is:

$$\omega_u := \omega_u - \mu \Delta \omega_u, \tag{6}$$

where μ is a learning parameter (some value you choose that is less than one, start with 0.001 if you don't know how to set it), and

$$\Delta \omega_u = \frac{1}{N} \sum_{n=1}^{N} e_n h_{nu} \tag{7}$$

The update to your bias on the output neuron is

$$\beta := \beta - \mu \Delta \beta, \tag{8}$$

where

$$\Delta\beta = \frac{1}{N} e_n \tag{9}$$

Next, the update for your first layer weight is:

$$w_{du} := w_{du} - \mu \Delta w_{du}, \tag{10}$$

where

$$\Delta w_{du} = \frac{1}{N} \sum_{n=1}^{N} e_n \omega_u \frac{\partial h_{nu}}{\partial v_{nu}} x_{nd}, \tag{11}$$

and

$$\frac{\partial h_{nu}}{\partial v_{nu}} = \begin{cases} 1 & v_{nu} > 0 \\ 0 & v_{nu} \leq 0 \end{cases}. \tag{12}$$

Finally, the update to your first layer neruon bias is:

$$b_u := b_u - \mu \Delta b_u, \tag{13}$$

where

$$\Delta b_u = \frac{1}{N} \sum_{n=1}^{N} e_n \omega_u \frac{\partial h_{nu}}{\partial v_{nu}}. \tag{14}$$