Forward and backward pass in a neural network

Andrea Esuli

September 3, 2020

This is a step by step example of performing the forward and backward pass on a neural network.

Network

We will work with a simple two-layers network.

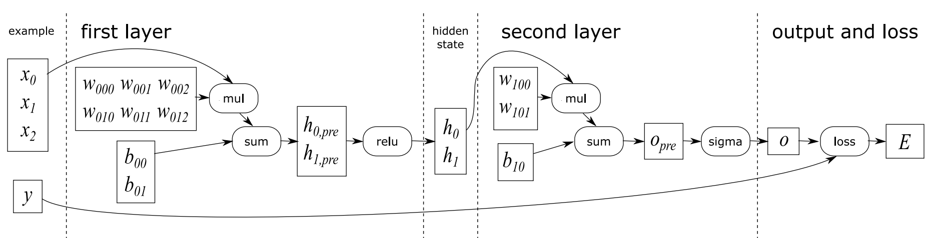

Figure 1: Network and flow of computation

First layer, two neurons with bias, ReLU activation ($\text{ReLU}(x) = \max(0, x)$).

$$W_0 = \begin{bmatrix} w_{000} & w_{001} & w_{002} \\ w_{010} & w_{011} & w_{012} \end{bmatrix} = \begin{bmatrix} 0.2 & -1.2 & 0.9 \\ -0.5 & -1.2 & 0.3 \end{bmatrix} \tag{1}$$

$$b_{\text{hid}} = \begin{bmatrix} b_{00} \\ b_{01} \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} \tag{2}$$

Second layer (output layer), one neuron with bias, sigmoid activation ($\sigma(x) = \frac{1}{1+e^{-x}}$).

$$W_{\text{out}} = \begin{bmatrix} w_{100} & w_{101} \end{bmatrix} = \begin{bmatrix} 0.8 & -1.1 \end{bmatrix} \tag{3}$$

$$b_{\text{out}} = b_{10} = -0.1 \tag{4}$$

Data

Training example, input vector and expected output.

$$x = \begin{bmatrix} 0.9 \\ 0.2 \\ 0.5 \end{bmatrix} \tag{5}$$

$$y = 0 \tag{6}$$

Forward pass

Passing input through first layer.

$$h_{\text{pre}} = W_{\text{hid}}x + b_{\text{hid}} = \begin{bmatrix} 0.39 \\ -0.54 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.29 \\ -0.34 \end{bmatrix} \tag{7}$$

$$h = \text{relu}(h_{\text{pre}}) = \begin{bmatrix} 0.29 \\ 0 \end{bmatrix} \tag{8}$$

Passing the output of first layer through the second layer.

$$o_{\text{pre}} = W_{\text{out}}h + b_{\text{out}} = 0.203 - 0.1 = 0.103 \tag{9}$$

$$o = \sigma(o_{\text{pre}}) = \frac{1}{1 + e^{-0.103}} = 0.526 \tag{10}$$

Output o is > 0.5 so the prediction would be $\hat{y} = 1$.
Computing loss.

$$\text{loss} = E = \frac{1}{2}\sum_i (y_i - o_i)^2 = \frac{1}{2}(0 - 0.526)^2 = 0.138 \tag{11}$$

Backpropagation

Computing the partial derivative (gradient) of error with respect to weights (including biases) of the network. Example for w_{100}.
Applying the chain rule.

$$\frac{\partial E}{\partial w_{100}} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial o_{\text{pre}}} \cdot \frac{\partial o_{\text{pre}}}{\partial w_{100}} \tag{12}$$

$$\frac{\partial E}{\partial o} = 2\frac{1}{2}(y - o)^{2-1} \cdot -1 = -(y - o) = o - y = 0.526 \tag{13}$$

$$\frac{\partial o}{\partial o_{\text{pre}}} = \frac{\partial \sigma(o_{\text{pre}})}{\partial o_{\text{pre}}} = \sigma(o_{\text{pre}})(1 - \sigma(o_{\text{pre}})) = 0.526(1 - 0.526) = 0.249 \tag{14}$$

$$\frac{\partial o_{\text{pre}}}{\partial w_{100}} = \frac{\partial w_{100} h_0 + w_{101} h_1 + b10}{\partial w_{100}} = h_0 = 0.29 \quad (15)$$

$$\frac{\partial E}{\partial w_{100}} = 0.526 \cdot 0.249 \cdot 0.29 = 0.038 \quad (16)$$

Learning rate is a parameter of the training process.
This is a very high learning rate, select to make the correction based on a single example more evident.

$$\mu = 0.1 \quad (17)$$

Weight is changed by combining gradient and learning rate so as to reduce error.

$$w_{100}^* = w_{100} - \mu \frac{\partial E}{\partial w_{100}} = 0.7 - 0.1 \cdot 0.038 = 0.696 \quad (18)$$

Partial derivatives can be reused to compute correction for the other weights in the same layer.

$$\frac{\partial E}{\partial w_{101}} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial o_{\text{pre}}} \cdot \frac{\partial o_{\text{pre}}}{\partial w_{101}} = 0.526 \cdot 0.249 \cdot 0 = 0 \quad (19)$$

Gradient for w_{101} is zero because ReLU of first layer gave $h_1 = 0$. Weight does not change.

$$w_{101}^* = w_{101} - \mu \frac{\partial E}{\partial w_{101}} = -1.1 - 0.1 \cdot 0 = -1.1 \quad (20)$$

Bias b_{out} changes in the same way of weights, as it is just a weight with constant input equal to one.

$$\frac{\partial E}{\partial b_{10}} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial o_{\text{pre}}} \cdot \frac{\partial o_{\text{pre}}}{\partial b_{10}} = 0.526 \cdot 0.249 \cdot 1 = 0.131 \quad (21)$$

$$b_{10}^* = b_{10} - \mu \frac{\partial E}{\partial b_{10}} = -0.1 - 0.1 \cdot 0.131 = -0.113 \quad (22)$$

We compute hidden layer gradients, using chain rule.

$$\frac{\partial E}{\partial w_{000}} = \frac{\partial E}{\partial h_0} \cdot \frac{\partial h_0}{\partial h_{\text{pre},0}} \cdot \frac{\partial h_{\text{pre},0}}{\partial w_{000}} \quad (23)$$

We can reuse gradients from output layers.

$$\frac{\partial E}{\partial h_0} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial o_{\text{pre}}} \cdot \frac{\partial o_{\text{pre}}}{\partial h_0} = 0.526 \cdot 0.249 \cdot w_{100} = 0.526 \cdot 0.249 \cdot 0.7 = 0.092 \quad (24)$$

ReLU derivative on non-negative values is 1.

$$\frac{\partial h_0}{\partial h_{\text{pre},0}} = 1 \quad (25)$$

$$\frac{\partial h_{\text{pre},0}}{\partial w_{000}} = x_0 = 0.9 \quad (26)$$

$$\frac{\partial E}{\partial w_{000}} = 0.092 \cdot 1 \cdot 0.9 = 0.082 \quad (27)$$

Weight update.

$$w^*_{000} = w_{000} - \mu \frac{\partial E}{\partial w_{000}} = 0.2 - 0.1 \cdot 0.082 = 0.191 \quad (28)$$

Same goes for all other weights and biases for the first layer.
Note that:

$$\frac{\partial E}{\partial h_1} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial o_{\text{pre}}} \cdot \frac{\partial o_{\text{pre}}}{\partial h_1} = 0.526 \cdot 0.249 \cdot w_{101} = 0.526 \cdot 0.249 \cdot -1.1 = -0.144 \quad (29)$$

Let's compute all remaining gradients.

$$\frac{\partial E}{\partial w_{001}} = \frac{\partial E}{\partial h_0} \cdot \frac{\partial h_0}{\partial h_{\text{pre},0}} \cdot \frac{\partial h_{\text{pre}}, 0}{\partial w_{001}} = 0.092 \cdot 1 \cdot 0.2 = 0.018 \quad (30)$$

$$\frac{\partial E}{\partial w_{002}} = \frac{\partial E}{\partial h_0} \cdot \frac{\partial h_0}{\partial h_{\text{pre},0}} \cdot \frac{\partial h_{\text{pre}}, 0}{\partial w_{002}} = 0.092 \cdot 1 \cdot 0.5 = 0.046 \quad (31)$$

$$\frac{\partial E}{\partial w_{010}} = \frac{\partial E}{\partial h_1} \cdot \frac{\partial h_1}{\partial h_{\text{pre},1}} \cdot \frac{\partial h_{\text{pre}}, 1}{\partial w_{010}} = -0.144 \cdot 0 \cdot 0.9 = 0 \quad (32)$$

$$\frac{\partial E}{\partial w_{011}} = \frac{\partial E}{\partial h_1} \cdot \frac{\partial h_1}{\partial h_{\text{pre},1}} \cdot \frac{\partial h_{\text{pre}}, 1}{\partial w_{011}} = -0.144 \cdot 0 \cdot 0.2 = 0 \quad (33)$$

$$\frac{\partial E}{\partial w_{012}} = \frac{\partial E}{\partial h_1} \cdot \frac{\partial h_1}{\partial h_{\text{pre},1}} \cdot \frac{\partial h_{\text{pre}}, 1}{\partial w_{012}} = -0.144 \cdot 0 \cdot 0.5 = 0 \quad (34)$$

$$\frac{\partial E}{\partial b_{00}} = \frac{\partial E}{\partial h_0} \cdot \frac{\partial h_0}{\partial h_{\text{pre},0}} \cdot \frac{\partial h_{\text{pre}}, 0}{\partial b_{00}} = 0.092 \cdot 1 \cdot 1 = 0.092 \quad (35)$$

$$\frac{\partial E}{\partial b_{01}} = \frac{\partial E}{\partial h_1} \cdot \frac{\partial h_1}{\partial h_{\text{pre},1}} \cdot \frac{\partial h_{\text{pre}}, 1}{\partial b_{01}} = -0.144 \cdot 0 \cdot 1 = 0 \quad (36)$$

Update of weights and biases.

$$w^*_{001} = w_{001} - \mu \frac{\partial E}{\partial w_{001}} = -1.2 - 0.1 \cdot 0.018 = -1.202 \quad (37)$$

$$w^*_{002} = w_{002} - \mu \frac{\partial E}{\partial w_{002}} = 0.9 - 0.1 \cdot 0.046 = 0.895 \quad (38)$$

$$w^*_{010} = w_{010} - \mu \frac{\partial E}{\partial w_{010}} = -0.5 - 0.1 \cdot 0 = -0.5 \quad (39)$$

$$w^*_{011} = w_{011} - \mu \frac{\partial E}{\partial w_{011}} = -1.2 - 0.1 \cdot 0 = -1.2 \quad (40)$$

$$w^*_{012} = w_{012} - \mu \frac{\partial E}{\partial w_{012}} = 0.3 - 0.1 \cdot 0 = 0.3 \quad (41)$$

$$b^*_{00} = b_{00} - \mu \frac{\partial E}{\partial b_{00}} = -0.1 - 0.1 \cdot 0.092 = -0.109 \quad (42)$$

$$b^*_{01} = b_{01} - \mu \frac{\partial E}{\partial b_{01}} = 0.2 - 0.1 \cdot 0 = 0.2 \quad (43)$$

We have made small corrections. Is there a reduction in error?

Lets' repeat the forward pass.

$$h^*_{\text{pre}} = W^*_{\text{hid}}x + b^*_{\text{hid}} = \begin{bmatrix} 0.379 \\ -0.54 \end{bmatrix} + \begin{bmatrix} -0.109 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.27 \\ -0.34 \end{bmatrix} \quad (44)$$

$$h^* = \text{relu}(h^*_{\text{pre}}) = \begin{bmatrix} 0.27 \\ 0 \end{bmatrix} \quad (45)$$

$$o^*_{\text{pre}} = W^*_{\text{out}}h^* + b^*_{\text{out}} = 0.188 - 0.113 = 0.075 \quad (46)$$

$$o^* = \sigma(o^*_{\text{pre}}) = \frac{1}{1 + e^{-0.075}} = 0.518 \quad (47)$$

$$\text{loss}^* = E^* = \frac{1}{2}\sum_i (y_i - o^*_i)^2 = \frac{1}{2}(0 - 0.518)^2 = 0.134 \quad (48)$$

Yes, we moved our prediction a bit closer to the correct one and we reduced the error.
Repeating these steps more times (and with more examples) will properly fit the network.