Forward and Backpropagation for a Single Hidden Layer Neural Network

Tran Quoc Thai
Student ID: 2370759

April 9, 2025

1 Forward Pass

Given the input vector:

$$\boldsymbol{x} = \begin{bmatrix} 0.375 \\ 0.583 \end{bmatrix}, \tag{1}$$

and weight matrices:

$$\boldsymbol{W}_1 = \begin{bmatrix} 0.3 & 0.5 \\ 0.4 & -0.2 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}. \tag{2}$$

The hidden layer pre-activation values are:

$$\boldsymbol{z}_1 = \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1 = \begin{bmatrix} 0.3 & 0.5 \\ 0.4 & -0.2 \end{bmatrix} \begin{bmatrix} 0.375 \\ 0.583 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}. \tag{3}$$

Computing each component:

$$z_{1,1} = (0.3 \times 0.375) + (0.5 \times 0.583) + 0.1 = 0.42575, \tag{4}$$
$$z_{1,2} = (0.4 \times 0.375) + (-0.2 \times 0.583) - 0.1 = 0.0156. \tag{5}$$

Applying the sigmoid activation function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \tag{6}$$

we obtain:

$$h_1 = \sigma(0.42575) = 0.60486, \tag{7}$$
$$h_2 = \sigma(0.0156) = 0.5039. \tag{8}$$

For the output layer:

$$\boldsymbol{W}_2 = \begin{bmatrix} 0.6 & -0.4 \end{bmatrix}, \quad b_2 = 0.2. \tag{9}$$

Computing the pre-activation output:

$$z_2 = (0.6 \times 0.60486) + (-0.4 \times 0.5039) + 0.2 = 0.4027. \tag{10}$$

Applying sigmoid activation:

$$\hat{y} = \sigma(0.4027) = 0.5993. \tag{11}$$

2 Backward Pass

Target value: $y = 1$. Error derivative with respect to output:

$$\delta_2 = \hat{y} - y = 0.5993 - 1 = -0.4007. \tag{12}$$

Gradient for output weights:

$$\nabla_{W_2} = \delta_2 \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = -0.4007 \begin{bmatrix} 0.60486 \\ 0.5039 \end{bmatrix} = \begin{bmatrix} -0.2425 \\ -0.2018 \end{bmatrix}. \tag{13}$$

Gradient for output bias:

$$\nabla_{b_2} = \delta_2 = -0.4007. \tag{14}$$

Hidden layer error:

$$\delta_1 = \delta_2 \boldsymbol{W}_2 \odot \sigma'(\boldsymbol{z}_1) = -0.4007 \begin{bmatrix} 0.6 \\ -0.4 \end{bmatrix} \odot \begin{bmatrix} 0.2390 \\ 0.24998 \end{bmatrix}. \tag{15}$$

Computing components:

$$\delta_{1,1} = (-0.4007 \times 0.6) \times 0.2390 = -0.0575, \tag{16}$$
$$\delta_{1,2} = (-0.4007 \times -0.4) \times 0.24998 = 0.0400. \tag{17}$$

Gradient for first-layer weights:

$$\nabla_{W_1} = \begin{bmatrix} \delta_{1,1} \\ \delta_{1,2} \end{bmatrix} \times \boldsymbol{x}^T = \begin{bmatrix} -0.0575 \\ 0.0400 \end{bmatrix} \times \begin{bmatrix} 0.375 & 0.583 \end{bmatrix}. \tag{18}$$

Computing:

$$\nabla_{W_1} = \begin{bmatrix} -0.0575 \times 0.375 & -0.0575 \times 0.583 \\ 0.0400 \times 0.375 & 0.0400 \times 0.583 \end{bmatrix} \tag{19}$$
$$= \begin{bmatrix} -0.0216 & -0.0335 \\ 0.0150 & 0.0233 \end{bmatrix}. \tag{20}$$

Gradient for first-layer bias:

$$\nabla_{b_1} = \begin{bmatrix} -0.0575 \\ 0.0400 \end{bmatrix}. \tag{21}$$

3 Weight Updates

Using learning rate $\alpha = 0.1$:

$$W_2 = W_2 - \alpha \nabla_{W_2}, \quad b_2 = b_2 - \alpha \nabla_{b_2}, \tag{22}$$
$$W_1 = W_1 - \alpha \nabla_{W_1}, \quad b_1 = b_1 - \alpha \nabla_{b_1}. \tag{23}$$

Computing updates:

$$W_2 = \begin{bmatrix} 0.6 & -0.4 \end{bmatrix} - 0.1 \times \begin{bmatrix} -0.2425 & -0.2018 \end{bmatrix} = \begin{bmatrix} 0.6243 & -0.3798 \end{bmatrix}, \tag{24}$$
$$b_2 = 0.2 - 0.1 \times (-0.4007) = 0.2401. \tag{25}$$

Applying updates for W_1 and b_1 similarly.