Neural Networks: Forward Pass and Backpropagation

Question 1: Forward Pass for Regression

Consider a neural network with:
Input: 1 feature, Hidden layer: 5 neurons (sigmoid activation), Output layer: 1 neuron (linear activation, no sigmoid).

Solution

The pre-activation z_{1j} for the j-th hidden neuron is:

$$z_{1j} = w_{1j}x + b_{1j}.$$

The output h_j of the j-th hidden neuron, after applying the sigmoid activation function $\sigma(\cdot)$, is:

$$h_j = \sigma(z_{1j}) = \frac{1}{1 + e^{-z_{1j}}}.$$

This calculation is performed for each of the 5 hidden neurons ($j = 1, 2, 3, 4, 5$).

The outputs from the hidden layer, h_1, h_2, h_3, h_4, h_5, serve as the inputs for the single neuron in the output layer. The output neuron has a linear activation, meaning we only compute the weighted sum:

$$\hat{y} = \sum_{i=1}^{5} v_i h_i + c.$$

—

Question 2: Gradients of the Loss

We start with the loss function for a single datapoint:

$$E = \tfrac{1}{2}(\hat{y} - y)^2.$$

So:

$$\frac{\partial E}{\partial \hat{y}} = \hat{y} - y.$$

Output-layer weights and bias

For each output weight v_i:

$$\frac{\partial E}{\partial v_i} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial v_i} = (\hat{y} - y)h_i.$$

For the output bias c:

$$\frac{\partial E}{\partial c} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial c} = \hat{y} - y.$$

Hidden-layer weights and biases

We need the derivative through the sigmoid. For hidden unit i:

$$\sigma'(z_i) = h_i(1 - h_i).$$

First compute derivative of E w.r.t. z_i:

$$\frac{\partial E}{\partial z_i} = (\hat{y} - y)v_i\,\sigma'(z_i) = (\hat{y} - y)v_i\,h_i(1 - h_i).$$

Now derivatives w.r.t. the hidden weights w_i and biases b_i. Since $z_i = w_i x + b_i$:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_i} = \Big((\hat{y} - y)v_i h_i(1 - h_i)\Big) \cdot x,$$

$$\frac{\partial E}{\partial b_i} = \frac{\partial E}{\partial z_i} \cdot \frac{\partial z_i}{\partial b_i} = (\hat{y} - y)v_i h_i(1 - h_i).$$

Conclusion – Final Derivatives

Output Layer:

$$\frac{\partial E}{\partial v_i} = (\hat{y} - y)h_i, \qquad \frac{\partial E}{\partial c} = \hat{y} - y.$$

Hidden Layer:

$$\frac{\partial E}{\partial w_i} = (\hat{y} - y)v_i h_i(1 - h_i)\, x, \qquad \frac{\partial E}{\partial b_i} = (\hat{y} - y)v_i h_i(1 - h_i).$$

—

Question 3: Backpropagation via Chain Rule

In a regression setup, the chain rule is used to propagate the error from the output back to the hidden layer.

1. **Compute the derivative of the loss with respect to the output:**

$$\frac{\partial E}{\partial \hat{y}} = \hat{y} - y.$$

2. **Propagate the error to the output weights and bias:**

$$\frac{\partial E}{\partial v_i} = (\hat{y} - y)h_i, \qquad \frac{\partial E}{\partial c} = \hat{y} - y.$$

3. **Compute the error signal at each hidden unit:**

$$\frac{\partial E}{\partial z_i} = (\hat{y} - y)v_i h_i(1 - h_i).$$

4. **Propagate the error to the hidden weights and biases:**

$$\frac{\partial E}{\partial w_i} = (\hat{y} - y)v_i h_i(1 - h_i)\, x, \qquad \frac{\partial E}{\partial b_i} = (\hat{y} - y)v_i h_i(1 - h_i).$$