Assignment 2 Songhwan Kim

Q1 Input → Hidden 1 → output 2 layer.

Let x be the input

y be the output

First layer → W_1, bias → b_1

Second layer → W_2, bias → b_2

the output layer 1 is:

$z_1 = W_1 x + b_1$

$a_1 = \sigma(z_1)$ where "a" is activation function. (Sigmaid)

The output layer 2 is:

$z_2 = W_2 a_1 + b_2$

$a_2 = z_2$.

The output of the neural network is

$\hat{y} = a_2$

Task: Back propagation for regression where loss function is the Mean Square Error loss.

$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

To learn the parameters.

① Provide random values for weights W_1, W_2 and biases b_1, b_2

② Update the weights using gradient descent by

$W_i := W_i - \alpha \cdot \frac{\partial L}{\partial W_i}$ where i refers to the ith layer

$b_i := b_i - \alpha \cdot \frac{\partial L}{\partial b_i}$

③ Repeat until convergence.

To find $\frac{\partial L}{\partial W_2}$ to update the second layer

$$\frac{\partial L}{\partial W_2} = \frac{\partial}{\partial W_2}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2\right]$$

- L is MSE Loss function.

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial W_2}(y_i-\hat{y}_i)^2$$

$$= \frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\frac{\partial}{\partial W_2}(y_i-\hat{y}_i) = \frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\left(\frac{\partial}{\partial W_2}y_i - \frac{\partial}{\partial W_2}\hat{y}_i\right)$$

$$= \frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\left(-\frac{\partial}{\partial W_2}\hat{y}_i\right) = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\left(\frac{\partial}{\partial W_2}\hat{y}_i\right)$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\circ a_1$$

$\frac{\partial}{\partial W_2}\hat{y} = \frac{\partial \hat{y}}{\partial W_2}$ · $\hat{y} = a_2 = z_2$

↳ identity function for output layer

$$= \frac{\partial a_2}{\partial W_2} = \frac{\partial z_2}{\partial W_2}$$

$$= \frac{\partial}{\partial W_2}(W_2 a_1 + b_2)$$

$$= a_1$$

- $z_2 = W_2 a_1 + b_2$ from previous page

Similarly $\frac{\partial L}{\partial b_2} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)$

To find $\frac{\partial L}{\partial W_1}$, we use chain rule.

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_2}\cdot\frac{\partial z_2}{\partial a_1}\cdot\frac{\partial a_1}{\partial z_1}\cdot\frac{\partial z_1}{\partial W_1}$$

$z_1 = W_1 x + b_1$

$a_2 = z_2$
a_2 is dependent on z_2

$z_2 = W_2 a_1 + b_2$
z_2 is dependent on a_1

$a_1 = \sigma(z_1)$
a_1 is dependent on z_1

Rewrite it as

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_2}\cdot\frac{\partial z_2}{\partial a_1}\cdot\frac{\partial a_1}{\partial z_1}\cdot\frac{\partial z_1}{\partial W_1}$$

$\frac{\partial L}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_2} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)$, $\frac{\partial z_2}{\partial a_1} = W_2$, $\frac{\partial a_1}{\partial z_1} = \sigma'(z_1)$, $\frac{\partial z_1}{\partial W_1} = x$

Therefore,

$$\frac{\partial L}{\partial W_1} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot W_2\circ\sigma'(z_1)\cdot x$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_2}\cdot\frac{\partial z_2}{\partial W_2}$$

$\frac{\partial L}{\partial a_2}\cdot\frac{\partial a_2}{\partial z_2} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)$, $\frac{\partial z_2}{\partial W_2} = a_1$

Similarly

$$\frac{\partial L}{\partial b_1} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot W_2\circ\sigma'(z_1)$$

The difference from the update rule for the network trained for binary classification using log loss.

For classification problem

$$\frac{\partial L}{\partial w_1} = (a_2 - y) \cdot w_2 \cdot \sigma'(z_1) \cdot x$$

For Regression problem

$$\frac{\partial L}{\partial w_1} = -\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) w_2 \cdot \sigma'(z_1) \cdot x$$

The difference of update rule is that the classification problem uses log loss to output a probability value between 0 and 1. In the other hands, the regression problem uses MSE loss to find an optimal value.