Yingyi Kang

Question 1

Loss function : $J = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ (Mean Square Error Loss)

activation function for hidden layer (first layer) : $g_1(z) = \frac{e^z}{e^z+1}$

activation function for output layer (second layer) : $g_2(z) = z$ (Given this is neural network for regression)

Let $\vec{x}$ be the input features $[n \times f]$

Let $\vec{y}$ be the ~~output~~ target ~~labels~~ output values $[n \times 1]$

Define the weights [parameters] of the neural network:

First layer $\rightarrow W_1$, bias $\rightarrow \vec{b_1}$

Second layer $\rightarrow W_2$, bias $\rightarrow \vec{b_2}$

Then, the output of layer 1 is:

$\vec{Z}_1 = W_1 \cdot \vec{x} + \vec{b_1}$

$\vec{a_1} = g_1(\vec{Z_1})$

The output of layer 2 is

$\vec{Z_2} = W_2 \cdot \vec{a_1} + \vec{b_2}$

$\vec{a_2} = g_2(\vec{Z_2})$

The output of the neural network is

$\hat{y} = \vec{a_2}$

To learn the parameters (Steps to train a 2-layer neural network with backpropagation for regression):

① Provide random values for weights W_1, W_2 and biases $\vec{b_1}$, $\vec{b_2}$

② update the weights and biases backforward (from last layer to first layer) using gradient descent by:

$W_i = W_i - \alpha \frac{\partial L}{\partial W_i}$ $\vec{b_i} = \vec{b_i} - \alpha \cdot \frac{\partial L}{\partial \vec{b_i}}$ (i refers to ith layer, α is learning rate.)

③ repeat until convergence.

Derivation of update rules:

① ~~for~~ find $\frac{\partial L}{\partial W_2}$ to update the weights of the second layer.

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

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

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

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

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

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot\frac{\partial g_2(\vec{z_2})}{\partial W_2}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\frac{\partial g_2(\vec{z_2})}{\partial z_2}\cdot\frac{\partial \vec{z_2}}{\partial W_2}$$

(Since $g_2(\vec{z_2}) = z_2$)

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot 1\cdot\frac{\partial}{\partial W_2}[W_2\cdot\vec{a_1^{(i)}}+\vec{b_2}]$$

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

② find $\frac{\partial L}{\partial \vec{b_2}}$ to update the biases of the second layer

Similarly:

$$\frac{\partial L}{\partial \vec{b_2}} = \frac{\partial}{\partial \vec{b_2}}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2\right] = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial \vec{b_2}}(y_i-\hat{y}_i)^2 = \ldots$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot\frac{\partial g_2(\vec{z_2})}{\partial z_2}\cdot\frac{\partial \vec{z_2}}{\partial \vec{b_2}}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_2)\cdot 1\cdot\frac{\partial}{\partial \vec{b_2}}[W_2\vec{a_1^{(i)}}+\vec{b_2}]$$

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

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i) = \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i) = \frac{2}{n}\sum_{i=1}^{n}(\vec{a_2^{(i)}}-y_i)$$

③ find $\frac{\partial L}{\partial W_1}$ to update the weights of the first layer.

$$\frac{\partial L}{\partial W_1} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial W_1}(y_i-\hat{y}_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial(y_i-\hat{y}_i)^2}{\partial(y_i-\hat{y}_i)}\cdot\frac{\partial(y_i-\hat{y}_i)}{\partial\hat{y}_i}\cdot\frac{\partial\hat{y}_i}{\partial W_1}$$

$$= \frac{1}{n}\sum_{i=1}^{n}2(y_i-\hat{y}_i)\cdot(-1)\cdot\frac{\partial\vec{a}_2^{(i)}}{\partial W_1}$$

$$= \frac{1}{n}\sum_{i=1}^{n}2(y_i-\hat{y}_i)\cdot(-1)\cdot\frac{\partial g_2(\vec{z}_2)}{\partial\vec{z}_2}\cdot\frac{\partial\vec{z}_2}{\partial\vec{a}_1^{(i)}}\cdot\frac{\partial\vec{a}_1^{(i)}}{\partial W_1}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot 1\cdot\frac{\partial}{\partial a_1^{(i)}}[W_2\vec{a}_1^{(i)}+\vec{b}_2]\cdot\frac{\partial g_1(\vec{z}_1)}{\partial\vec{z}_1}\cdot\frac{\partial\vec{z}_1}{\partial W_1}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot W_2\cdot(1-g_1(\vec{z}_1))\cdot g_1(\vec{z}_1)\cdot\frac{\partial}{\partial W_1}(W_1\cdot\vec{X}^{(i)}+\vec{b}_1)$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot W_2\cdot(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)}\cdot\vec{X}^{(i)}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)\cdot W_2(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)}\cdot\vec{X}^{(i)} = \frac{2}{n}\sum_{i=1}^{n}(\vec{a}_2^{(i)}-y_i)\cdot W_2(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)}\cdot\vec{X}^{(i)}$$

④ find $\frac{\partial L}{\partial b_1}$ to update the biases of the first layer.

$$\frac{\partial L}{\partial b_1} = -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot\frac{\partial g_2(\vec{z}_2)}{\partial\vec{z}_2}\cdot\frac{\partial\vec{z}_2}{\partial\vec{a}_1^{(i)}}\cdot\frac{\partial\vec{a}_1^{(i)}}{\partial\vec{b}_1}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot 1\cdot\frac{\partial}{\partial\vec{a}_1^{(i)}}[W_2\vec{a}_1^{(i)}+\vec{b}_2]\cdot\frac{\partial g_1(\vec{z}_1)}{\partial\vec{z}_1}\cdot\frac{\partial\vec{z}_1}{\partial\vec{b}_1}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\cdot W_2\cdot(1-g_1(\vec{z}_1))\cdot g_1(\vec{z}_1)\cdot\frac{\partial}{\partial\vec{b}_1}(W_1\vec{X}^{i}+\vec{b}_1)$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)\,W_2\cdot(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)}\cdot 1$$

$$= \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)\,W_2(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)} = \frac{2}{n}\sum_{i=1}^{n}(\vec{a}_2^{(i)}-y_i)\cdot W_2(1-\vec{a}_1^{(i)})\cdot\vec{a}_1^{(i)}$$

Hence, plug these in the update rules, $W_i = W_i - \alpha\frac{\partial L}{\partial W_i}$ and $\vec{b}_i = \vec{b}_i - \alpha\frac{\partial L}{\partial\vec{b}_i}$.

then we can get specific update rule for each weight and bias.

The difference between network trained for binary classification using Log loss and network trained for regression using Mean Square ~~Error~~ Error Loss in the update rule:

For classification problem:

$$\frac{\partial L}{\partial W_1} = \underbrace{(\vec{a}_2 - \vec{y})\, W_2\, g'(z) \cdot \vec{X}}$$

For regression task:

$$\frac{\partial L}{\partial W_1} = \frac{2}{n}\sum_{i=1}^{n} \underbrace{(\vec{a}_2^{(i)} - y_i)\, W_2\, g'(z_i)\, \vec{X}^{(i)}}$$

From above, we can the difference.

① Due to different loss functions used in them, MSE makes the update rule have to sum up the values of $(a_2 - y) W_2 g'(z_i) X$ on each data point and then calculate the average value.

While log loss function makes the update rule just need to directly do $(a_2 - y) W_2 g'(z) X$ on the ~~vectors~~ matrix $a_2, y, W_2, g'(z), X$.

② The $g(z)$ in ~~bignary~~ binary classification is logistic sigmoid function.

While $g(z)$ in regression is identity function.

Hence, their $g'(z)$ are different.