CS 422 HW7 (A20516822)

Sunday, 30 November 2020 1:12 AM

Exercise 1

1.1 (3 points) Consider the neural network shown in the following figure.

The weight matrix W_1 is 2×3, W_2 is 3×2, and W_3 is 2×1. All the hidden layer units and the output layer use the logistic sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ as the activation function. The input vector is $x = [1, -1, 0.5]^T$.

Recall that $\mathrm{MSE}(y, \hat{y})$ is defined as a function $f(y, \hat{y})$ = $(y - \hat{y})^2$.

Suppose we defined $\mathrm{MSE}(y, \hat{y}) = f(y, \hat{y})$ = $(y - \hat{y})^2$.

The ground truth vector is $y = [0, 1]^T$.

Compute the derivatives of $\mathrm{MSE}(y, \hat{y})$ with respect to W_1, W_2, and W_3 using the back-propagation algorithm. Assume a learning rate $\eta = 1$.

Answer:

(a) Give forward propagation to compute the predicted output.

Answer:

$$W_1 = \begin{bmatrix} 1, & -1, & 0.5, \\ 1, & 1, & 2 \end{bmatrix}^T$$

$$x = Y$$

We have 3 hidden layers with $\sigma(z)$ activation function.

$$Z_{h_1} = \mathrm{mat}(W_1 * x) = \mathrm{mat}(W_1 * Y)$$

$$Z_{h_2} = \mathrm{mat}(W_2 * y_{h_1}) = \mathrm{mat}(W_2 * Y)$$

$$Z_{h_3} = \mathrm{mat}(W_3 * y_{h_2}) = \mathrm{mat}(W_3 * Y)$$

Four hidden layer is output layer, without Sigmoid activation function for output layer.

$$\hat{y} = \sigma(W_4 * Z_{h_3} + W_4 * Z_{h_3} + W_4 * Z_{h_3})$$

$$= \begin{bmatrix} \sigma(0.75 * Y) \\ \sigma(0.5) \end{bmatrix}$$

$$= \begin{bmatrix} 0.7944 \\ 0.7773 \end{bmatrix}$$

(b) What is the loss or error value?

Answer:

Squared error $= (y - \hat{y})^2$

$$= \begin{bmatrix} 0 - 0.7944 \\ 1 - 0.7773 \end{bmatrix}^2$$

$$= \begin{bmatrix} 0.7944 \\ 0.7773 \end{bmatrix}$$

(c) Using back-propagation, compute the gradient of weight vector that $\frac{\partial E}{\partial W_1}$ compute the partial derivative of the error with respect to all of the weights.

Answer:

$$W = [1, 1, -1, 0.5, 1, 2]^T$$

We have 6 total σz.

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3}, \frac{\partial E}{\partial w_4}, \frac{\partial E}{\partial w_5}, \frac{\partial E}{\partial w_6} \right]$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$= \left[-2(y_1 - \sigma_1(z_1)) \right] * \left[\sigma_1(1 - \sigma_1(z_1)) \right] * \left[x_{(1,1)} \right] \quad \text{where } x_{(1,1)} = \text{input function}$$

$$= \left[-2(0 - 0.7944) \right] * \left[(0.7944)(1 - 0.7944) \right] * \left[1 \right]$$

$$= 0.0104$$

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial w_2}$$

$$= \left[-2(y_1 - \sigma_1(z_1)) \right] * \left[\sigma_1(1 - \sigma_1(z_1)) \right] * \left[x_{(1,2)} \right] \quad \text{where } x_{(1,2)} = \text{input function}$$

$$= \left[-2(0 - 0.7944) \right] * \left[(0.7944)(1 - 0.7944) \right] * \left[-1 \right]$$

$$= 0.0104$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial w_3}$$

$$= \left[-2(y_1 - \sigma_1(z_1)) \right] * \left[\sigma_1(1 - \sigma_1(z_1)) \right] * \left[x_{(1,3)} \right] \quad \text{where } x_{(1,3)} = \text{input function}$$

$$= \left[-2(0 - 0.7944) \right] * \left[(0.7944)(1 - 0.7944) \right] * \left[0.5 \right]$$

$$= 0.0052$$

$$\frac{\partial E(z_1)}{\partial w_4} = \begin{cases} 1 & \text{if } z_{(1,1)} > \\ 0 & \text{otherwise} \end{cases}$$

Since $z_{(1,1)} > 1$

$$\frac{\partial E(z_1)}{\partial w_4} = 1$$

$$z_{(1,1)} = x_1 * Y$$

$$\frac{\partial z_{(1,1)}}{\partial w_4} = x_1 * Y$$

$$\therefore \frac{\partial E}{\partial w_4} = 0.0104 * 1 * Y$$

$$= 0.0104$$

$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1(z_1)}{\partial z_1} \frac{\partial z_{(1,1)}}{\partial w_5}$$

$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1(z_1)}{\partial z_1} \frac{\partial z_{(1,1)}}{\partial w_5}$$

$$= \left[-2(y_1 - \sigma_1(z_1)) \right] * \left[\sigma_1(1 - \sigma_1(z_1)) \right] * \left[w_5 \right]$$

$$= \left[-2(0 - 0.7944) \right] * \left[(0.7944)(1 - 0.7944) \right] * \left[1 \right]$$

$$= 0.0104$$

$$\frac{\partial E(z_1)}{\partial w_6} = \begin{cases} 1 & \text{if } z_{(1,1)} > \\ 0 & \text{otherwise} \end{cases}$$

Since $z_{(1,1)} = w_1 * x_1 \rightarrow 1$

$$\frac{\partial E(z_1)}{\partial w_6} = 1$$

$$\frac{\partial z_{(1,1)}}{\partial w_6} = x_1$$

$$\therefore \frac{\partial z_{(1,1)}}{\partial w_6} = x_1$$

$$\therefore \frac{\partial E}{\partial w_6} = 0.0104 * 1 * Y$$

$$= 0.0104$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1(z_1)}{\partial z_1} \frac{\partial z_{(1,1)}}{\partial w_3}$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial \sigma_1} \frac{\partial \sigma_1(z_1)}{\partial z_1} \frac{\partial z_{(1,1)}}{\partial w_3}$$

$$= \left[-2(y_1 - \sigma_1(z_1)) \right] * \left[\sigma_1(1 - \sigma_1(z_1)) \right] * \left[x_{(1,3)} \right]$$

$$= \left[-2(0 - 0.7944) \right] * \left[(0.7944)(1 - 0.7944) \right] * \left[0.5 \right]$$

$$= 0.0052$$

$$\frac{\partial E(z_1)}{\partial w_3} = \begin{cases} 1 & \text{if } z_{(1,1)} > \\ 0 & \text{otherwise} \end{cases}$$

Since $z_{(1,1)} = w_1 * x_1 \rightarrow 1$

$$\frac{\partial E(z_1)}{\partial w_3} = 1$$

$$\frac{\partial z_{(1,1)}}{\partial w_3} = x_1$$

$$\therefore \frac{\partial z_{(1,1)}}{\partial w_3} = x_1$$

$$\therefore \frac{\partial E}{\partial w_3} = 0.0052 * 1 * Y$$

$$= 0$$

$$\therefore \nabla E = \left[-0.0104, 0.0104, 0, 0.0104, 0.0104, 0 \right]$$

(d) Using a learning rate of 1.0, compute new weights for the equations with that over weights, use forward propagation to compute the new predicted output, and the loss (error).

Answer:

$$\mathrm{weight}_{\mathrm{new}} = \mathrm{weight}_{\mathrm{old}} - \eta \nabla E$$

$$= \begin{bmatrix} 1, & -1, & 0.5, & 1, & 2 \end{bmatrix} - 1 * \begin{bmatrix} -0.0104, & 0.0104, & 0, & 0.0104, & 0.0104, & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.9944, & 0.9894, & -1, & 0.9894, & 0.9894, & 2 \end{bmatrix}$$

$$Z_{h_1} = \mathrm{mat}(W_1 * y_{h_1}) = \mathrm{mat}(0.9944 * Y)$$

$$= 3.9824$$

$$Z_{h_2} = \mathrm{mat}(W_2 * y_{h_2}) = \mathrm{mat}(0.9894 * Y)$$

$$= 3.9824$$

$$Z_{h_3} = \mathrm{mat}(W_3 * y_{h_3}) = \mathrm{mat}(0 * Y)$$

$$= 0$$

$$\hat{y} = \sigma(W_4 * Z_{h_3} + W_4 * Z_{h_3} + W_4 * Z_{h_3})$$

$$= \begin{bmatrix} \sigma(0.9944 * 3.9824 + 0.9894 * 3.9824 + 0 * 0) \\ \sigma(-3.9824) \end{bmatrix}$$

$$= \begin{bmatrix} 0.9944 \\ 0.9773 \end{bmatrix}$$

Squared Error $= (0 - \hat{y})^2$

$$= \begin{bmatrix} 0.9944 \\ 0.9773 \end{bmatrix}^2$$

(e) Compare the difference between the loss values you obtained in (b) & (d).

Answer:

Comparing the values of squared error of loss obtained in (b) & (d), we can see that the error reduced by 1.02 with the weight updated. Though it cannot be concluded as a significant decrease in error or loss value, but to further literature, and weight are a significant decrease.

1.2 Two classes

(a) For each of the Boolean function given below, state whether the function is linearly separable.

(i) A AND B AND C

A	B	C	A AND B	AND C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Since the boolean function two NOT and AND operator, we can say it is linearly separable.

(ii) NOT A AND B

A	NOT A	B	NOT A AND B
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

Since the boolean function two NOT and AND operator, we can say it is linearly separable.

(iii) (A OR B) AND (A OR C)

A	B	A OR B	A	C	A OR C
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	1

A	b	A AND b
0	0	0
0	1	1
1	1	1
1	1	1

Since the operator two AND and OR function, it is linearly separable.

(iv) (A OR B) AND (A OR B)

A	B	A OR B	A	B	A OR B	A AND b
0	0	0	0	0	0	0
0	1	1	0	1	1	1
1	0	1	1	0	1	1
1	1	1	1	1	1	1

Since, it has XOR function, it is not linearly separable.

(b) (i-iv) Demonstrate how the perceptron model can be used to implement the AND and OR boolean functions. a pair of boolean operations.

Answer:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Let activation function, is activation function.

$$\sigma(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$$

Let $b_1 = -1.5$, $w_1 = 1$, $w_2 = 1$ Let $b_2 = -0.5$, $w_1 = 1$, $w_2 = 1$

$\mathrm{AND}(x_1, x_2) = \sigma(b_1 + w_1 x_1 + w_2 x_2) = \sigma(-1.5 + x_1 + x_2)$ $\mathrm{OR}(x_1, x_2) = \sigma(b_2 + w_1 x_1 + w_2 x_2) = \sigma(-0.5 + x_1 + x_2)$

$\mathrm{AND}(0, 0) = \sigma(-1.5 + 0 + 0) = \sigma(-1.5) = 0$ $\mathrm{OR}(0, 0) = \sigma(-0.5 + 0 + 0) = \sigma(-0.5) = 0$

$\mathrm{AND}(0, 1) = \sigma(-1.5 + 0 + 1) = \sigma(-0.5) = 0$ $\mathrm{OR}(0, 1) = \sigma(-0.5 + 0 + 1) = \sigma(0.5) = 1$

$\mathrm{AND}(1, 0) = \sigma(-1.5 + 1 + 0) = \sigma(-0.5) = 0$ $\mathrm{OR}(1, 0) = \sigma(-0.5 + 1 + 0) = \sigma(0.5) = 1$

$\mathrm{AND}(1, 1) = \sigma(-1.5 + 1 + 1) = \sigma(0.5) = 1$ $\mathrm{OR}(1, 1) = \sigma(-0.5 + 1 + 1) = \sigma(1.5) = 1$

(b) (v) Compared to the disadvantage of using linear functions as activation functions for multi-layer neural network.

Answer:

Using linear functions as activation functions in a series of layers of neural network is redundant because the final output will still be a linear combination of the original inputs, which the point the purpose of multi-layer neural network. Introducing non-linear activation is also useful for modeling non-linear relationships between input and activation. Thus, using linear function in each layer neural network will act as a perceptron.

2. (10 points) Consider a dataset that has 8 features. You train a neural network with 3 hidden layers and an output layer that produces a scalar value (a regression problem). The first hidden layer has 16 neurons, the second has 8 neurons, and the third has 4 neurons. In this network, how many total parameters will you have?

Answer:

Dataset has 8 feature, so d_0 input $(I) = 8$

the 1st hidden layer $= 16$

the 2nd neuron in 1st hidden layer, $(h_1) = 8$

the 3rd neuron in 1st hidden layer, $(h_2) = 8$

the 4th neuron in 1st hidden layer, $(h_3) = 4$

the 1st neuron in output layer, $(O) = 1$

the 1st parameter $= (2 * H_1) + (I_1 * H_1) + (I_2 * H_2) + (I_3 * H_3)$

$$+ (h_1 * h_2) + (h_2 * h_3) + (h_3 * O) + (h_1 * O)$$

$$= 323$$