CS422-01: Homework #7

Vijay K. Gurbani

Sujad Patel

November 27, 2022

Part I

Exercises

1.1

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

For hidden layer, the input is:

$$[X_1, X_2, X_3] = [xw_1, xw_2, xw_3]$$

$$\Rightarrow [4, 4, -4]$$

For hidden layer, RelU activation function is used to generate output.

$$z_1 = \sigma_{X_1} = 4$$

$$z_2 = \sigma_{X_2} = 4$$

$$z_3 = \sigma_{X_3} = 0$$

For output layer, the input is:

$$Z = \sum_{n=1}^{3} z_n w_{n+3} = (4 * 0.5) + (4 * 1) + 0 = 6$$

For output layer, Sigmoid function is used for generating the output:

$$\hat{y} = \sigma_6 = \frac{1}{1 + e^{-6}} = \mathbf{0.997}$$

(b) What is the loss or error value?

Loss is:

$$(y - \hat{y}^2)$$

$$\Rightarrow (0 - 0.997)^2 = \mathbf{0.995}$$

(c) Using backpropagation, compute the gradient of the weight vector, that is, compute the partial derivative of the error with respect to all of the weights.

Gradient for w$_4 is$:

$$\frac{\partial E}{\partial w_4} = \frac{\partial E}{\partial Z} * \frac{\partial Z}{\partial z} * \frac{\partial z}{\partial w_4}$$

$$\Rightarrow (\hat{y} - y) * (\hat{y} * (1 - \hat{y})) * z1$$

$$\Rightarrow -2(0 - 0.997) * (0.997 * (1 - 0.997)) * 4$$

$$= \mathbf{0.024}$$

Gradient for w$_5 is$:

$$\Rightarrow -2(0 - 0.997) * (0.997 * (1 - 0.997)) * 4$$

$$= \mathbf{0.024}$$

Gradient for w$_6 is$:

$$\Rightarrow (0.997 - 0) * (0.997 * (1 - 0.997)) * 0$$

$$= \mathbf{0}$$

Gradient for w$_1 is$:

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial Z} * \frac{\partial Z}{\partial z_1} * \frac{\partial z_1}{\partial w_1}$$

$$\Rightarrow (-2(\hat{y} - y) * (\hat{y} * (1 - \hat{y}))) * w_j * x$$

$$\Rightarrow -2(0 - 0.997) * (0.997 * (1 - 0.997)) * 0.5 * 4$$

$$= \mathbf{0.012}$$

Gradient for w$_2 is$:

$$\Rightarrow -2(0 - 0.997) * (0.997 * (1 - 0.997)) * 1 * 4$$

$$= \mathbf{0.024}$$

Gradient for w$_3 is$:

$$\Rightarrow -2(0 - 0.997) * (0.997 * (1 - 0.997)) * 2 * 4$$

$$= \mathbf{0.048}$$

(d) Using a learning rate of 1.0, compute new weights from the gradient. With the new weights, use forward propagation to compute the new predicted output, and the loss (error).

New weights from the gradient with learning rate(η) :$w_i = w_i - \eta\frac{\partial E}{\partial w_i}$

$$w_1 = 1 - 1 * 0.012 = \mathbf{0.988}$$

$$w_2 = 1 - 1 * 0.024 = \mathbf{0.976}$$

$$w_3 = -1 - 1 * 0.048 = \mathbf{-1.05}$$

$$w_4 = 0.5 - 1 * 0.024 = \mathbf{0.476}$$

$$w_5 = 1 - 1 * 0.024 = \mathbf{0.976}$$

$$w_6 = 2 - 1 * 0 = \mathbf{2.0}$$

Using forward propagation to generate output using new weights.

$$[X_1, X_2, X_3] = [xw_1, xw_2, xw_3]$$

$$\Rightarrow [3.952, 3.904, -4.2]$$

For hidden layer, RelU activation function is used to generate output.

$$z_1 = \sigma_{X_1} = 3.952$$

$$z_2 = \sigma_{X_2} = 3.904$$

$$z_3 = \sigma_{X_3} = 0$$

For output layer, the input is:

$$Z = \sum_{n=1}^{3} z_n w_{n+3} = (3.952 * 0.476) + (3.904 * 0.976) + 0 = 5.69$$

For output layer, Sigmoid function is used for generating the output:

$$\hat{y} = \sigma_{5.69} = \frac{1}{1 + e^{-5.69}} = \mathbf{0.996}$$

Loss is:

$$(y - \hat{y}^2)$$

$$\Rightarrow (0 - 0.996)^2 = \mathbf{0.992}$$

(e) Comment on the difference between the loss values you observe in (b) and (d)

Loss1 = 0.995, Loss2 = 0.992
Therefore, updating weights made the predicted output closer to the actual output.

1.2 Tan Chapter 4(Neural Networks)

Question 14

(a) A AND B AND C

On a graph for AND function, a line can be drawn separating 3 False points from 1 True point. So, AND function is separable, therefore the above statement is also linearly separable.

(b) NOT A AND B

NOT AND function both are separable, therefore the above statement is also linearly separable.

(c) (A OR B) AND (A OR C)

Since OR is negation of AND, OR is also linearly separable. The above statement is also linearly separable.

(d) (A XOR B) AND (A OR B)

XOR is not linearly separable, since the graph of XOR points distributed in away one line cannot separate true and false points on the graph. Therefore, the above statement is not linearly separable.

Question 15

(a) Demonstrate how the perceptron model can be used to represent the AND and OR functions between a pair of Boolean variables.

Since AND and OR are lineatly separable, they can be easily modeled according to the perceptron rules. Since the booleans functions take two inputs, there will be three parameters (w1,w2 and bias).

$$\hat{y} = S(x1w1 + x2w2 + b)$$

we can update the weights using learning rate. Compute until within the error margin.

(b) Comment on the disadvantage of using linear functions as activation functions for multi-layer neural networks.

The linear function derivative is a constant value. So, no value gained in gradient descent or error reducing. If the activation function is linear, then nesting in "n" number of the hidden layer with the same function will have no real effect. Thus, the linear function act as a disadvantage for activation functions.

1.3

Consider a dataset that has 8 predictors. You train a neural network with 3 hidden layers and an output layer that predicts a continuous 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?

Number of parameters = = (I*H) + (H*O) + (H+O); where I is Inputs, H is hidden layers and O is output

$$(8 * 28) + (28 * 1) + (28 + 1) = \mathbf{281} parameters$$