Contents

1 Learning Goals

By the end of the exercise, you should be able to

- Describe components of a perceptron.
- Construct a perceptron to represent simple linear functions such as AND, OR, and NOT.
- Represent the XOR function using a three-layer feed-forward perceptron network.
- Explain why the back-propagation algorithm can be interpreted as a version of the gradient descent optimization algorithm.
- Execute the back-propagation algorithm given the update rules of the weights.

2 Representing the XOR function using a three-layered feed-forward network

The XOR function is defined by the following truth table.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

XOR can be modeled by using a neural network with one hidden layer.

- Two input units.
- Two hidden units
- One output unit
- The activation function is the Sigmoid function.

To describe the back-propagation algorithm, we first introduce some notation.

- x_1, x_2 denotes the values of the input units.
  h_1, h_2 denotes the values of the hidden units.
  o_1 denotes the values of the output unit.
  y denoted the actual value(true label).
- $w1_{ij}$ is the weight on line between input unit x_i and hidden unit h_j.
  $w2_{j1}$ is the weight on line between hidden unit h_j and output unit o_1.

To measure the error between the desired output values and the actual output values, we will use the squared difference function.

error $= \dfrac{1}{2}(y - o_1)^2.$

3 Practice Questions

Question 1:

Calculating the derivative of sigmoid function $f(x) = \dfrac{1}{1 + e^{-x}}$ respect to x.

Question 2: Consider a neural network with 2 input units, 2 hidden units and 1 output unit.

Derive the gradient of $w2_{j1}$, where $w2_{j1}$ represents the weight on line between the j^{th} hidden unit and the output unit o_1.

The predicted value from the output unit is o_1, the expected output(true label) is y, and the output from the j^{th} hidden unit is h_j.

Question 3:

We would like to learn the XOR function using a multi-layer neural network. We are running the back-propagation algorithm on the neural network and we are currently at the n-th iteration.

The current values of the parameters are as follows.

- $w1_{01} = 1, w1_{11} = 1, w1_{21} = -1$
- $w1_{02} = 1, w1_{12} = -1, w1_{22} = 1$
- $w2_{01} = 1, w2_{11} = -1, w2_{21} = -1$

The next set of inputs is $x_1 = 0, x_2 = 1$, and the true label is $y = 1$. The learning rate is $\alpha = 0.1$.

Calculating the updated values of these parameters after this iteration.