ECE 457B Assignment 1

Tzu-Chun Chou
20718966
t6chou@uwaterloo.ca

Contents

*

1 Question 1

1.1 a

Question 1 a

Given $\Delta w = -\eta \nabla_w E(w)$

$$E(w) = \frac{1}{2}\left(t^{(k)} - s\left(\sum_i w_i x_i^{(k)}\right)\right)^2$$

$$S(net) = \frac{1}{1+e^{-(net)}}$$

Chain rule: $y = f(g(x)) \rightarrow y' = f'(g(x)) \cdot g'(x)$

Let $f'(g(x)) = \dfrac{\partial E(w)}{\partial\left(\sum_i w_i x_i^{(k)}\right)}$

$$= -\left(t^{(k)} - s\right) \cdot S'\left(\sum_i (w_i x_i^{(k)})\right)$$

$$= -\left(t^{(k)} - s\right) \cdot \frac{\partial s}{\partial w} \frac{1}{1+e^{-net}}$$

$$= -\left(t^{(k)} - s\right) \cdot \frac{e^{-net}}{(1+e^{-net})}$$

$$= -\left(t^{(k)} - s\right) \left(s^2 \cdot e^{-\sum_i w_i x_i^{(k)}}\right) \quad ①$$

Let $g'(x) = \dfrac{\partial\left(\sum_i w_i x_i^{(k)}\right)}{\partial w_i} = x_i^{(k)}$ ②

Plug in ① and ② into $\Delta w = -\eta \nabla_w E(w)$

$$\Delta w = -\eta \nabla_w E(w)$$

$$= -\eta \cdot f'(g(x)) \cdot g'(x)$$

$$= -\eta \cdot -\left(t^{(k)} - s\right)\left(s^2 \cdot e^{-\sum_i w_i x_i^{(k)}}\right)\left(x_i^{(k)}\right)$$

$$= \eta \cdot \left(t^{(k)} - s\right)\left(s^2 \cdot e^{-\sum_i w_i x_i^{(k)}}\right) x_i^{(k)}$$

1.2 b

Please refer to the attached python files.

1.3 c

Perceptron: Below is an example of valid boundary classified by Perceptron. The red dots corresponds to coordinates in Class1 and the green dots corresponds to coordinates in Class2.

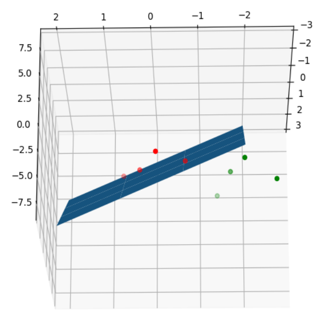


Figure 1: Boundary Equation = 1.02x - 1.85y - 0.315z = 5.22

Adaline: Below is an example of valid boundary classified by Adaline.The red dots corresponds to coordinates in Class1 and the green dots corresponds to coordinates in Class2.

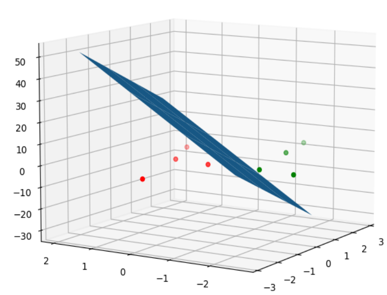


Figure 2: Boundary Equation = 4.28x - 6.52y - 0.085z = 0.60

1.4 d

Yes, the classifier boundary is still valid for both Perceptron and Adaline. As shown below:

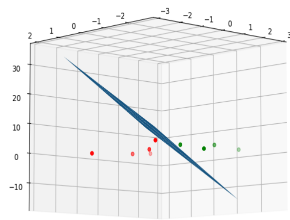


Figure 3: Updated Boundary Classification - Perceptron

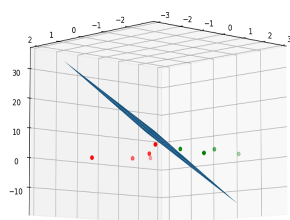


Figure 4: Updated Boundary Classification - Adaline

1.5 e

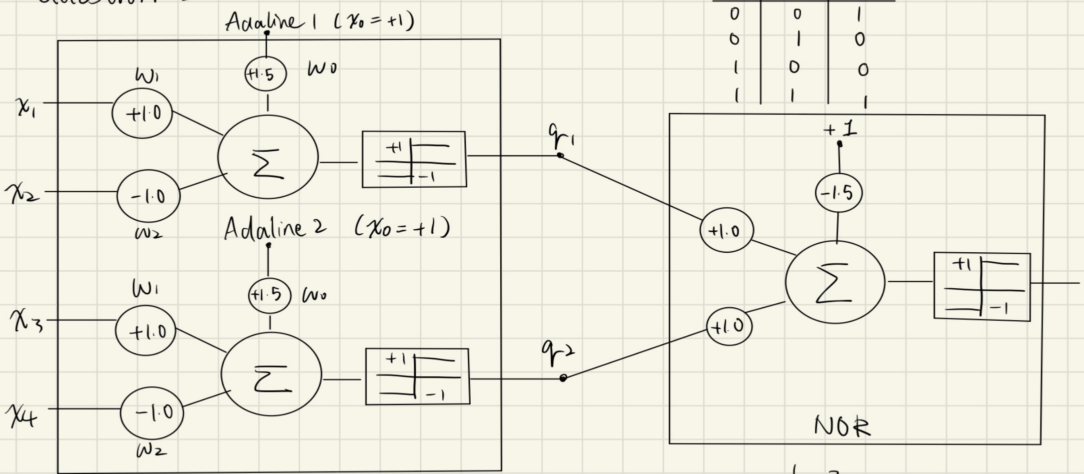
The Adaline structure with the LMS learning algorithm has better capabilities than Perceptron that uses Hebbian learning rule because the Adaline uses sigmoid(),a linear activation function, which enables continuous weight adjustments during training, whereas Perceptron uses a step activation function which is dicrete. Furthermore, Hebbian learning rule relies on the difference between the input and the target output to adjust the weights which is less efficient the LMS algorithm. Adaline uses gradient descent to adjust the weights, resulting in generally faster convergence to the optimal solution than Perceptron.

2 Question 2

Question 2.

XNOR

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

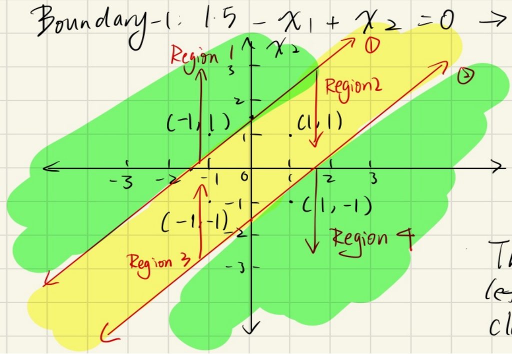


Adaline-1: $W = [1.5, 1.0, -1.0]$, $x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$

Boundary-1: $1.5 + x_1 - x_2 = 0 \rightarrow x_2 = x_1 + 1.5$ ①

Adaline-2: $W = [1.5, 1.0, -1.0]$, $x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$

Boundary-1: $1.5 - x_1 + x_2 = 0 \rightarrow x_1 = x_2 + 1.5$ ②



Region 1: (−1, 1)
Region 2: (1, 1), (−1, −1), (1, −1)
Region 3: (−1, 1), (−1, 1), (1, 1)
Region 4: (1, −1)

The compound boundary shown on the left is an example of valid XNOR classification.