Machine Learning

Assignment 6.2

Submitted By: Ranji Raj

December 4, 2020

Perceptron rule

The training error of a weight vector is defined as,

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 \tag{1}$$

where, D is the set of training examples, t_d is the target output for the training example d, and o_d is the output of the linear unit for training example d.
The weight vector is revised as,

$$\vec{w} \leftarrow \vec{w} + \Delta\vec{w} \tag{2}$$

$$\Delta\vec{w} = -\eta\nabla E(\vec{w}) \tag{3}$$

here η is a positive constant called as the *learning rate*, which determines the step size in the gradient descent search.
The negative sign indicates the movement if the weight vector in the direction that *decreases* E. Now, (2) and (3) in component form as,

$$w_i \leftarrow w_i + \Delta w_i \tag{4}$$

$$\Delta w_i = -\eta\frac{\partial E}{\partial w_i} \tag{5}$$

The vector of $\frac{\partial E}{\partial w_i}$ derivatives that form the gradient can be obtained by differentiating E from (1), as

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i}\frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\in D}\frac{\partial}{\partial w_i}(t_d - o_d)^2$$

$$= \frac{1}{2}\sum_{d\in D}2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) = \sum_{d\in D}(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w}.\vec{x_d})$$

$$\therefore \quad \frac{\partial E}{\partial w_i} = \sum_{d \in D} (t_d - o_d)(-x_{i,d}) \tag{6}$$

where, $x_{i,d}$ denotes the single input component x_i for training example d. Substituting (6) in (5) yields,

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{i,d} \tag{7}$$

Given:

x1	x2	t (target)
1	0	0
3	-1	1
2	2	0
4	4	0
1	-2	1

Table 1: Data

and threshold as,

$$w_0 + w_1 x_1 + w_2 x_2 > 0$$

Iteration 1

We set the bias input x_0=1 and calculate the error as,

x0	x1	x2	t	o	$(t-o)$
1	1	0	0	0	0
1	3	-1	1	0	1
1	2	2	0	0	0
1	4	4	0	0	0
1	1	-2	1	0	1

Table 2: Data at Iteration 1

Now, we revise the weights given: w_0=w_1=w_2=0 & $\eta = 0.1$

weights	calculations	learned weights
w0	0+0.1[(1*0)+(1*1)+(1*0)+(1*0)+(1*1)]	0.2
w1	0+0.1[(1*0)+(3*1)+(2*0)+(4*0)+(1*1)]	0.4
w2	0+0.1[(0*0)+(-1*1)+(2*0)+(4*0)+(-2*1)]	-0.3

Table 3: Weights at Iteration 1

Calculating threshold as, if

$$0.2 + 0.4 * x_1 - 0.3 * x_2 > 0$$

then $o^{(1)}$=1 otherwise, $o^{(1)}$=0

x1	x2	calculations	classification ($o^{(1)}$)
1	0	0.2+0.4*1-0.3*0=0.6	1
3	-1	0.2+0.4*3-0.3*(-1)=1.7	1
2	2	0.2+0.4*2-0.3*2=0.4	1
4	4	0.2+0.4*4-0.3*4=0.6	1
1	-2	0.2+0.4*1-0.3*(-2)=1.2	1

Table 4: Classification at Iteration 1

Iteration 2

With the new outputs we prepare data for iteration 2.

x0	x1	x2	t	$o^{(1)}$	$(t - o^{(1)})$
1	1	0	0	1	-1
1	3	-1	1	1	0
1	2	2	0	1	-1
1	4	4	0	1	-1
1	1	-2	1	1	0

Table 5: Data at Iteration 2

Now, we revise the weights given: $w_0 = 0.2, w_1 = 0.4, w_2 = -0.3$ & $\eta = 0.1$

weights	calculations	learned weights
w0	0.2+0.1[(1*-1)+(1*0)+(1*-1)+(1*-1)+(1*0)]	-0.1
w1	0.4+0.1[(1*-1)+(3*0)+(2*-1)+(4*-1)+(1*0)]	-0.3
w2	-0.3+0.1[(0*-1)+(-1*0)+(2*-1)+(4*-1)+(-2*0)]	-0.9

Table 6: Weights at Iteration 2

Calculating threshold as, if

$$-0.1 - 0.3 * x_1 - 0.9 * x_2 > 0$$

then $o^{(2)}$=1 otherwise, $o^{(2)}$=0

x1	**x2**	**calculations**	**classification** ($o^{(2)}$)
1	0	-0.1-0.3*1-0.9*0=-0.4	0
3	-1	-0.1-0.3*3-0.9*(-1)=-0.1	0
2	2	-0.1-0.3*2-0.9*2=-2.5	0
4	4	-0.1-0.3*4-0.9*4=-4.9	0
1	-2	-0.1-0.3*1-0.9*(-2)=1.4	1

Table 7: Classification at Iteration 2

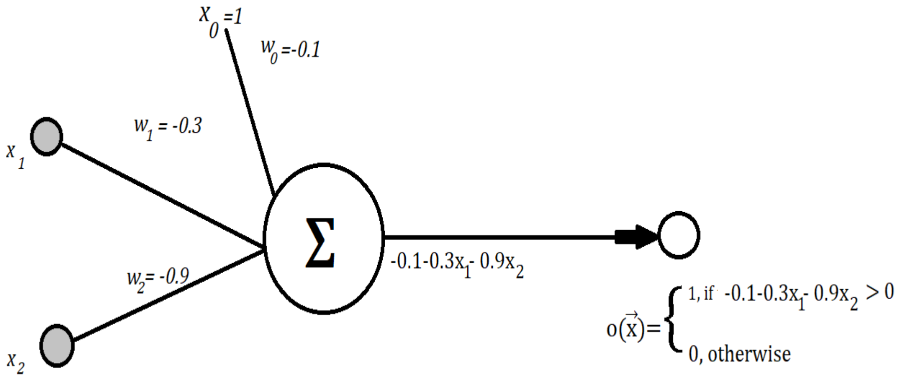

Figure 1: Final Perceptron

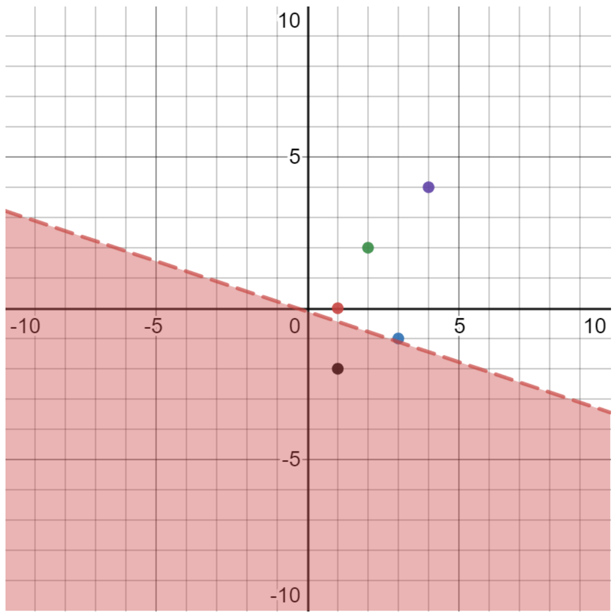

Figure 2: Classification boundary-Instances with class label **1** are put **below** the decision boundary and with class label **0** are put **above** the decision boundary.