CS3244 Machine Learning – Semester 1, 2012/13
Solution to Tutorial 4

1. What are the values of weights w_0, w_1, and w_2 for the perceptron whose decision surface is illustrated in Figure 4.3? Assume the surface crosses the x_1 axis at −1, and the x_2 axis at 2.

Answer:

The line for the decision surface corresponds to the equation $x_2 = 2x_1 + 2$, and since all points above the line should be classified as positive, we have $x_2 - 2x_1 - 2 > 0$. Hence $w_0 = -2$, $w_1 = -2$, and $w_2 = 1$.

2. Consider two perceptrons defined by the threshold expression $w_0+w_1x_1+w_2x_2>0$. Perceptron *A* has weight values

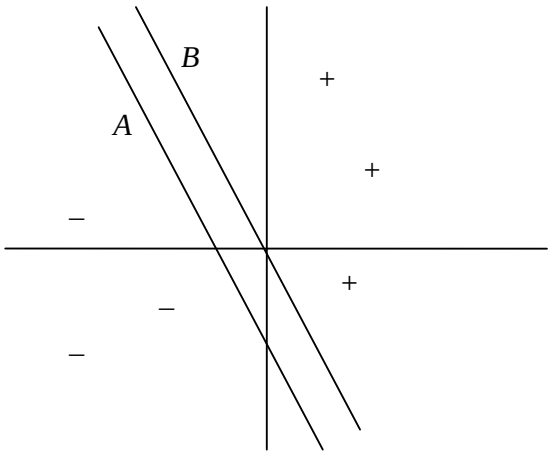
$$w_0 = 1, \quad w_1 = 2, \quad w_2 = 1$$

and Perceptron *B* has weight values

$$w_0 = 0, \quad w_1 = 2, \quad w_2 = 1$$

True or false? Perceptron *A* is *more-general-than* perceptron *B*. (*More-general-than* is defined in Chapter 2).

Answer:



True. Perceptron *A* is more general than *B*, because any point lying above *B* will also be above *A*. i.e. as per definition in chapter 2 for *more-general-than*:

$$(\forall x \in X)[(B(x) = 1) \rightarrow A(x) = 1)]$$

3. Derive a gradient descent training rule for a single unit with output *o*, where

$$o = w_0 + w_1x_1 + w_1x_1^2 + \ldots + w_nx_n + w_nx_n^2$$

Answer:

First, the error function is defined as: $E(\vec{w}) = \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2$

The update rule is the same, namely: $w_i := w_i + \Delta w_i$

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

For w_0,

$$\frac{\partial E}{\partial w_0} = \frac{\partial}{\partial w_0}\frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\in D}\frac{\partial}{\partial w_0}(t_d - o_d)^2 = \frac{1}{2}\sum_{d\in D}2(t_d - o_d)\frac{\partial}{\partial w_0}(t_d - o_d)$$
$$= \sum_{d\in D}(t_d - o_d)(-1) = -\sum_{d\in D}(t_d - o_d)$$

Thus

$$\Delta w_0 = \eta\sum_{d\in D}(t_d - o_d)$$

For $w_1, w_2, \ldots, w_n$

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

Thus

$$\Delta w_i = \eta\sum_{d\in D}(t_d - o_d)(x_{id} + x_{id}^2)$$

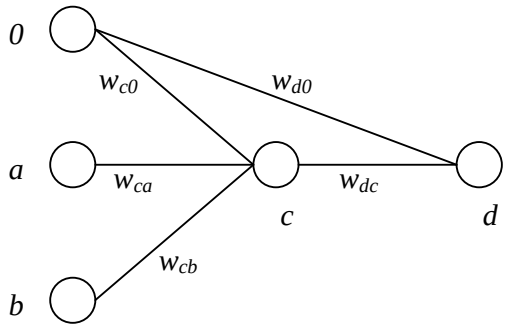
4. Consider a two-layer feedforward ANN with two inputs *a* and *b*, one hidden unit *c*, and one output unit *d*. This network has five weights (w_{ca}, w_{cb}, w_{c0}, w_{dc}, w_{d0}), where w_{x0} represents the threshold weight for unit *x*. Initialize these weights to the values (0.1, 0.1, 0.1, 0.1, 0.1), then give their values after each of the first two training iterations of the BACKPROPAGATION algorithm. Assume learning rate η = 0.3, momentum α = 0.9, incremental weight updates, and the following training examples:

a	*b*	*d*
1	0	1
0	1	0

Answer:

The network and the sigmoid activation function sigmoid function are as follows:

$$\sigma(y) = \frac{1}{1+e^{-y}}$$



Training example 1:

The outputs of the two neurons, noting that *a*=1and *b*=0:

$$o_c = \sigma(0.1\times 1 + 0.1\times 0 + 0.1\times 1) = \sigma(0.2) = 0.5498$$
$$o_d = \sigma(0.1\times 0.5498 + 0.1\times 1) = \sigma(0.15498) = 0.53867$$

The error terms for the two neurons, noting that *d*=1:

$$\delta_d = 0.53867\times(1-0.53867)\times(1-0.53867) = 0.1146$$
$$\delta_c = 0.5498\times(1-0.5498)\times 0.1\times 0.1146 = 0.002836$$

Compute the correction terms as follows, noting that *a*=1, *b*=0 and η=0.3:

$$\Delta w_{d0} = 0.3\times 0.1146\times 1 = 0.0342$$
$$\Delta w_{dc} = 0.3\times 0.1146\times 0.5498 = 0.0189$$
$$\Delta w_{c0} = 0.3\times 0.002836\times 1 = 0.000849$$
$$\Delta w_{ca} = 0.3\times 0.002836\times 1 = 0.000849$$
$$\Delta w_{cb} = 0.3\times 0.002836\times 0 = 0$$

and the new weights become:

$$w_{d0} = 0.1 + 0.0342 = 0.1342$$
$$w_{dc} = 0.1 + 0.0189 = 0.1189$$
$$w_{c0} = 0.1 + 0.000849 = 0.100849$$
$$w_{ca} = 0.1 + 0.000849 = 0.100849$$
$$w_{cb} = 0.1 + 0 = 0.1$$

Training example 2:

The outputs of the two neurons, noting that a=0 and b=1:

$$o_c = \sigma(0.100849 \times 0 + 0.1 \times 1 + 0.100849 \times 1) = \sigma(0.200849) = 0.55$$
$$o_d = \sigma(0.1189 \times 0.55 + 0.1342 \times 1) = \sigma(0.1996) = 0.5497$$

The error terms for the two neurons, noting that d=0:

$$\delta_d = 0.5497 \times (1 - 0.5497) \times (0 - 0.5497) = -0.1361$$
$$\delta_c = 0.55 \times (1 - 0.55) \times 0.1189 \times (-0.1361) = -0.004$$

Compute the correction terms as follows, noting that a=0, b=1, η=0.3 and α=0.9:

$$\Delta w_{d0} = 0.3 \times (-0.1361) \times 1 + 0.9 \times 0.0342 = -0.01$$
$$\Delta w_{dc} = 0.3 \times (-0.1361) \times 0.55 + 0.9 \times 0.0189 = -0.0055$$
$$\Delta w_{c0} = 0.3 \times (-0.004) \times 1 + 0.9 \times 0.000849 = -0.0004$$
$$\Delta w_{ca} = 0.3 \times (-0.004) \times 0 + 0.9 \times 0.000849 = 0.00086$$
$$\Delta w_{cb} = 0.3 \times (-0.004) \times 1 + 0.9 \times 0 = -0.0012$$

and the new weights become:

$$w_{d0} = 0.1342 - 0.01 = 0.1242$$
$$w_{dc} = 0.1189 - 0.0055 = 0.1134$$
$$w_{c0} = 0.100849 - 0.0004 = 0.100849$$
$$w_{ca} = 0.100849 + 0.00086 = 0.1016$$
$$w_{cb} = 0.1 - 0.0012 = 0.0988$$

5. Revise the BACKPROPAGATION algorithm in Table 4.2 so that it operates on units using the squashing function *tanh* in place of the sigmoid function. That is, assume the output of a single unit is $o = tanh(\vec{w}.\vec{x})$. Give the weight update rule for output layer weights and hidden layer weights. Hint: $tanh'(x) = 1 - tanh^2(x)$.

Answer:

Steps T4.3 and T4.4 in Table 4.2 will become as follows, respectively:

$$\delta_k \leftarrow (1 - o_k^{\ 2})(t_k - o_k)$$

$$\delta_h \leftarrow (1 - o_h^{\ 2}) \sum_{k \in outputs} w_{kh} \delta_k$$

6. Consider the alternative error function described in Section 4.81.

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d \in D}\sum_{k \in outputs}(t_{kd} - o_{kd})^2 + \gamma \sum_{i,j} w_{ji}^2$$

Derive the gradient descent update rule for this definition of E. Show that it can be implemented by multiplying each weight by some constant before performing the standard gradient descent update given in Table 4.2.

Answer:

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

$$\Delta w_{ji} = -\eta \frac{\partial E(\vec{w})}{\nabla \partial w_{ji}}$$

$$\frac{\partial E(\vec{w})}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \frac{1}{2}\sum_{d \in D}\sum_{k \in outputs}(t_{kd} - o_{kd})^2 + \frac{\partial}{\partial w_{ji}} \gamma \sum_{i,j} w_{ji}^2$$

The first term in the R.H.S of the above equation can be derived in the same manner as in equation (4.27), while we continue to work on the 2nd term. For output nodes, leads to:

$$\frac{\partial E(\vec{w})}{\partial w_{ji}} = -(t_j - o_j) o_j (1 - o_j) x_{ji} + 2\gamma w_{ji}$$

$$w_{ji} \leftarrow w_{ji} + \eta (t_j - o_j) o_j (1 - o_j) x_{ji} - 2\eta\gamma w_{ji}$$

$$w_{ji} \leftarrow \beta w_{ji} + \eta \delta_j x_{ji}$$

where $\beta = 1 - 2\eta\gamma$ and $\delta_j = (t_j - o_j) o_j (1 - o_j)$

Similarly, for hidden units, we can derive:

$$w_{ji} \leftarrow \beta w_{ji} + \eta \delta_j x_{ji}$$

where $\beta = 1 - 2\eta\gamma$ and $\delta_j = o_j (1 - o_j) \sum_{k \in Downstream(j)} \delta_k w_{kj}$

The above shows the update rule can be implemented by multiplying each weight by some constant before performing the gradient descent update given in Table 4.2.

7. Assume the following error function:

$$E(w) = 2\sigma^2 - \lambda w + \frac{1}{2}\mu w^2$$

where σ, λ and μ are constants. The weight w is updated according to gradient descent with a positive learning rate η. Write down the update equation for $w(k+1)$ given $w(k)$. Find the optimum weight w that gives the minimal error $E(w)$. What is the value of the minimal $E(w)$? (8 marks)

Answer:

$$\frac{\partial E}{\partial w} = -\lambda + \mu w$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} = \eta(\lambda - \mu w)$$

$$w(k+1) = w(k) + \eta(\lambda - \mu w)$$

When $E(w)$ becomes the smallest, $\frac{\partial E}{\partial w} = 0$

Thus, optimal $w_{optimal} = \frac{\lambda}{\mu}$

Minimal error:

$$E(w_{optimal}) = 2\sigma^2 - \frac{\lambda^2}{\mu} + \frac{\lambda^2}{2\mu} = 2\sigma^2 - \frac{\lambda^2}{2\mu}$$

8. WEKA outputs the following confusion matrix after training a J48 decision tree classifier with the contact-lenses dataset. (a) Count the number of True Positives, True Negatives, False Positives and False Negatives for each the three classes, i.e. soft, hard and none. (b) Calculate the TP rate (Recall), FP rate, Precision and F-measure for each class.

```
  a  b  c   <-- classified as
  4  0  1 |  a = soft
  0  1  3 |  b = hard
  1  2 12 |  c = none
```

Answer:

soft:

(a) TP = 4
TN = 18
FP = 1
FN = 1

(b) TP rate = Recall = TP / (TP + FN) = 4/5 = 0.8
FP rate = FP / (FP + TN) = 1/19 = 0.053
Precision = TP / (TP + FP) = 4/5 = 0.8
F-Measure = 2×0.8×0.8/(0.8+0.8) = 0.8

hard:

(a) TP = 1
TN = 18
FP = 2
FN = 3

(b) TP rate = Recall = TP / (TP + FN) = 1 / 4 = 0.25
FP rate = FP / (FP + TN) = 2/20 = 0.1
Precision = TP / (TP + FP) = 1 / 3= 0.333
F-Measure = 2×0.25×0.333/(0.25+0.333) = 0.286

none:

(a) TP = 12
TN = 5
FP = 4
FN = 3

(b) TP rate = Recall = TP / (TP + FN) = 12 / 15 = 0.8
FP rate = FP / (FP + TN) = 4/9 = 0.444
Precision = TP / (TP + FP) = 12 / 16 = 0.75
F-Measure = 2×0.8×0.75/(0.8+0.75) = 0.774