PERCEPTRON LEARNING RULE CONVERGENCE THEOREM

PERCEPTRON CONVERGENCE THEOREM: Says that there if there is a weight vector $\mathbf{w}^*$ such that $f(\mathbf{w}^*\mathbf{p}(q)) = t(q)$ for all q, then for any starting vector $\mathbf{w}$, the perceptron learning rule will converge to a weight vector (not necessarily unique and not necessarily $\mathbf{w}^*$) that gives the correct response for all training patterns, and it will do so in a finite number of steps.

IDEA OF THE PROOF: The idea is to find upper and lower bounds on the length of the weight vector. If the length is finite, then the perceptron has converged, which also implies that the weights have changed a finite number of times.

PROOF:

1) Assume that the inputs to the perceptron originate from two linearly separable classes. **That is, the classes can be distinguished by a perceptron**. Let X_1 be the subset of training vectors belonging to C_1. That is, $\mathbf{p}(1)$, $\mathbf{p}(2)$,... . Let X_2 be the set of training vectors belonging to C_2. That is, $\mathbf{p}(1)$, $\mathbf{p}(2)$,... . Then we can say that $X_1 \cup X_2$ is the complete training set X.

2) Given the set of vectors X_1 and X_2 to train this perceptron to train this perceptron, the training process (as we have seen) involves the adjustment of the weight vector $\mathbf{w}$ such that C_1 and C_2 are linearly separable. That is, there exists some $\mathbf{w}$ such that

$$\begin{aligned} &3)\ \mathbf{w}^T\mathbf{p} > 0 \text{ for every input vector } \mathbf{p} \in C_1 \\ &4)\ \mathbf{w}^T\mathbf{p} < 0 \text{ for every input vector } \mathbf{p} \in C_2 \end{aligned} \qquad (1)$$

3) What need to do is find some $\mathbf{w}$ such that the above is satisfied, which is the purpose of the perceptron algorithm.

   One algorithm for adapting the weight vector for the perceptron algorithm can be formulated as follows (there are others):

   a. If the k^{th} member of the training set $\mathbf{p}(k)$ is correctly classified at the k^{th} iteration no correction is made to $\mathbf{w}$. This is done according to the following rule:

$$\begin{aligned} &\mathbf{w}(k+1) = \mathbf{w}(k) \text{ if } \mathbf{w}^T\mathbf{p}(k) > 0 \text{ and } \mathbf{p}(k) \in C_1 \\ &\mathbf{w}(k+1) = \mathbf{w}(k) \text{ if } \mathbf{w}^T\mathbf{p}(k) < 0 \text{ and } \mathbf{p}(k) \in C_2 \end{aligned} \qquad (2)$$

b. Otherwise, the weight vector of the perceptron is updated according to the following rule:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{p}(k) \text{ if } \mathbf{w}^T\mathbf{p}(k) > 0 \text{ and } \mathbf{p}(k) \in C_2$$
$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k) \text{ if } \mathbf{w}^T\mathbf{p}(k) < 0 \text{ and } \mathbf{p}(k) \in C_1 \quad (3)$$

4) The proof begins assuming that $\mathbf{w}(1) = \mathbf{0}$ (i.e., the zero vector). Suppose that $\mathbf{w}^T(k)\mathbf{p}(k) < 0$ for k = 1, 2, ..., and all the input vectors $\mathbf{p}(k) \in X_1$ (or C_1).

Here, the perceptron **incorrectly** classifies the vectors $\mathbf{p}(1)$, $\mathbf{p}(2)$,... since the second condition in Equation (1) is violated (i.e., $\mathbf{w}^T(k)\mathbf{p}(k)$ should be greater than 0). Now we can use Equation (3) to write the adjustments to the weight matrix according to the perceptron learning rule

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k) \text{ for } \mathbf{p}(k) \in C_1 \quad (4)$$

Recall, what we are doing is modifying the weight vector so that it points in the right direction.

5) Given the initial condition $\mathbf{w}(1) = \mathbf{0}$, we can iteratively solve for $\mathbf{w}(k+1)$ obtaining:

$$\mathbf{w}(k+1) = \mathbf{p}(1) + \mathbf{p}(2) + ... + \mathbf{p}(k) \quad (5)$$

CLS EXERCISE: Show $\mathbf{w}(k+1) = \mathbf{p}(1) + \mathbf{p}(2) + ... + \mathbf{p}(k)$

ANSWER:

$\mathbf{w}(1) = \mathbf{0}$
$\mathbf{w}(2) = \mathbf{w}(1) + \mathbf{p}(1) = \mathbf{p}(1)$
$\mathbf{w}(3) = \mathbf{w}(2) + \mathbf{p}(2) = \mathbf{w}(1) + \mathbf{p}(1) + \mathbf{p}(2) = \mathbf{p}(1) + \mathbf{p}(2)$
$\mathbf{w}(4) = \mathbf{w}(3) + \mathbf{p}(3) = \mathbf{w}(2) + \mathbf{p}(2) + \mathbf{p}(3) = \mathbf{w}(1) + \mathbf{p}(1) + \mathbf{p}(2) + \mathbf{p}(3)$
$= \mathbf{p}(1) + \mathbf{p}(2) + \mathbf{p}(3)$

So, in general:

$\mathbf{w}(k+1) = \mathbf{p}(1) + \mathbf{p}(2) + ... + \mathbf{p}(k)$

Since C_1 and C_2 are assumed to be linearly separable, then there exists a $\mathbf{w_o}$ that can correctly classify all input vectors belonging to C_1 and C_2.

In this proof, we have assumed the existence of $\mathbf{w_o}$ for which

$$\mathbf{w_o}^T\mathbf{p}(k) > 0 \text{ if } \mathbf{p}(k) \in C_1$$

and

$$\mathbf{w_o}^T\mathbf{p}(k) < 0 \text{ if } \mathbf{p}(k) \in C_2$$

This is equivalent to the existence of a weight vector $\mathbf{w_o}$ for which

$$\mathbf{w_o}^T\mathbf{p}(k) > 0 \text{ if } \mathbf{p}(k) \in C_1 \cup C_2 \text{ or } X$$

The reason is that the training set can be considered to consist of two parts:

$$C_1 = \{\mathbf{p} \text{ such that the target value is } 1\}$$

and

$$C_2 = \{\mathbf{p} \text{ such that the target value is } 0\}$$

So we can think of the training set X as

$$X = C_1 \cup C_2$$

where

$$C_2 = \{-\mathbf{p} \text{ such that } \mathbf{p} \in C_2\}$$

Now, if the response for the network is incorrect, then this allows the weights to be updated according to:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k)$$

since $-\mathbf{w_o}^T\mathbf{p}(k) < 0 \Rightarrow \mathbf{w_o}^T\mathbf{p}(k) > 0$ if $\mathbf{p}(k) \in C_2$. Now in this case the perceptron is updated by $\mathbf{w}(k+1) = \mathbf{w}(k) - (-\mathbf{p}(k)) = \mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k)$ since $\mathbf{w}^T\mathbf{p}(k) > 0$ and $\mathbf{p}(k) \in C_2$.

For the solution $\mathbf{w_o}$, we can define some $\alpha > 0$ as

$$\alpha = \min_{\mathbf{p}(k) \in X} \mathbf{w_o}^T\mathbf{p}(k) \qquad (6)$$

which is just the minimum (scalar) value of all $\mathbf{w_o}^T\mathbf{p}(k)$ for all $\mathbf{p} \in X$.

6) By multiplying each side of Equation (5) by $\mathbf{w_o}^T$, we get

$$\mathbf{w_o}^T\mathbf{w}\ (k+1) = \mathbf{w_o}^T\mathbf{p}(1) + \mathbf{w_o}^T\mathbf{p}(2) + \ldots + \mathbf{w_o}^T\mathbf{p}(k)$$

7) And, by applying Equation (6), we get

$$\mathbf{w_o}^T\mathbf{w}\ (k+1) \geq k\alpha \tag{7}$$

since $\mathbf{w_o}^T\mathbf{w}(k+1)$ has to be greater than or equal to $k \times \min_{\mathbf{p}(k) \in X} \mathbf{w_o}^T\mathbf{p}(k)$

8) Next, use the Cauchy-Swartz inequality for $\mathbf{w_o}^T$ and $\mathbf{w_o}^T(k+1)$, which states for any two vectors **x** and **y**:

$$[\mathbf{x} \cdot \mathbf{y}]^2 \leq ||\mathbf{x}||^2\ ||\mathbf{y}||^2$$

or

$$||\mathbf{x}||^2 \geq [\mathbf{x} \cdot \mathbf{y}]^2 / ||\mathbf{y}||^2$$

where $||\mathbf{x}||$ is the Euclidian norm.

Example. Suppose **x** = [-1 0 4]. Then, $\|\mathbf{x}\| = \sqrt{(-1) + 0^2 + 4^2} = \sqrt{17}$.

Thus, we can say:

$$||\mathbf{w_o}^T||^2\ ||\mathbf{w}(k+1)||^2 \geq [\mathbf{w_o}^T\mathbf{w}\ (k+1)]^2 \tag{8}$$

9) From Equation (7), we know by applying Cauchy-Swartz that

$$\mathbf{w_o}^T\mathbf{w}\,(k+1) \geq k\alpha \quad \Rightarrow$$

$$||\mathbf{w_o}^T||^2\,||\mathbf{w}\,(k+1)||^2 \geq [k\alpha]^2 \quad \Rightarrow$$

$$||\mathbf{w_o}^T||^2\,||\mathbf{w}(k+1)||^2 \geq k^2\alpha^2$$

and we get:

$$\|\mathbf{w}(k+1)\|^2 \geq \frac{k^2\alpha^2}{\|\mathbf{w_o}\|^2} \qquad (9)$$

which shows that the squared length of the weight vector ($||\mathbf{w}(k+1)||^2$) grows by a factor of k^2, where k is the number of time the weights have changed.

10) So, what we have established from Equation (9) is a **lower-bound** in the terms of the squared Euclidian norm of the weight vector **w** at iteration k + 1.

But there is another aspect we must consider. In order to show that the weights cannot continue to grow indefinitely, we must establish an **upper-bound** for the weight vector **w**.

11) To find an **upper-bound**, we now do the following. Write Equation (4) as the following (where q is the number of patterns in *X*):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k) \text{ for } k = 1,...,q \quad \mathbf{p}(k) \in X$$

12) After taking the square of the Euclidian norm, we get

$$||\mathbf{w}(k+1)||^2 = ||\mathbf{w}(k)||^2 + 2\mathbf{w}^T(k)\,\mathbf{p}(k) + ||\mathbf{p}(k)||^2 \qquad (10)$$

13) But assuming that the perceptron incorrectly classifies an input vector belonging to *X* (i.e., we have to make the adjustment $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k)$) since $\mathbf{w}^T(k)\,\mathbf{p}(k) < 0$, then from Equation (10), it follows that:

$$||\mathbf{w}(k+1)||^2 \leq ||\mathbf{w}(k)||^2 + ||\mathbf{p}(k)||^2$$

CLS EXERCISE: Why can we make this claim?

ANSWER:

Because since the input vector $\mathbf{p}(k)$ was incorrectly classified, this implies that $\mathbf{w}^T(k)\,\mathbf{p}(k) < 0$. Thus, given that $||\mathbf{w}(k+1)||^2 = ||\mathbf{w}(k)||^2 + 2\mathbf{w}^T(k)\,\mathbf{p}(k) + ||\mathbf{p}(k)||^2$, then clearly $2\mathbf{w}^T(k)\,\mathbf{p}(k)$ must be < 0 also. Thus, we can claim that:

$$||\mathbf{w}(k+1)||^2 \leq ||\mathbf{w}(k)||^2 + ||\mathbf{p}(k)||^2 .$$

We can rewrite the above as:

$$||\mathbf{w}(k+1)||^2 - ||\mathbf{w}(k)||^2 \leq ||\mathbf{p}(k)||^2 \tag{11}$$

14) Adding these quantities for k = 1,...,q (where q is the number of patterns in C_1) and using the initial condition that $\mathbf{w}(1) = \mathbf{0}$, it can be shown (**this will be for homework**) that:

$$\begin{aligned} \|\mathbf{w}(k+1)\|^2 &\leq \sum_{n=1}^{k} \|\mathbf{p}(n)\|^2 \\ &\leq k\beta \end{aligned} \tag{12}$$

where $\beta > 0$ and is defined by the following:

$$\beta = \max_{\mathbf{p}(k) \in X} \|\mathbf{p}(k)\|^2 \tag{13}$$

15) Equation (13) says that the Euclidian norm of the weight vector $\mathbf{w}(k+1)$, grows at most linearly with the number of iterations, k, given the input patterns. In other words,

$$\|\mathbf{w}(k+1)\|^2 \leq k\beta . \tag{14}$$

16) Thus, we have established both upper and lower bounds for a perceptron to classify correctly the input patterns $\mathbf{p}(k)$ for k = 1,...,q for two class C_1 and C_2 given they are lineary separable.

17) Observe that $\|\mathbf{w}(k+1)\|^2 \leq k\beta$ conflicts with the earlier result of $\|\mathbf{w}(k+1)\|^2 \geq \frac{k^2\alpha^2}{\|\mathbf{w_o}\|^2}$; namely, it can be shown that they don't agree for ||**w**(k+1)|| for large values of k.

So we can state that k cannot be larger than some k_{max} for which Equations (9) and (14) are **both** satisfied; and they must be equal to determine the maximum number of iterations for the two linearly separable classes to converge.

To determine this, set Equations (9) and (14) equal to each other and substitute and solve for k_{max}. Thus we get,

$$\frac{k_{max}^2\alpha^2}{\|\mathbf{w_o}\|^2} = k_{max}\beta\,,$$

$$k_{max} = \frac{\|\mathbf{w_o}\|^2\beta}{\alpha^2}.$$

which means that changing the weights of the perceptron must terminate after at most k_{max} iterations, which means that the machine has solved the (linearly separable) problem correctly.