PERCEPTRON CONVERGENCE THEOREM

From "Fundamentals of Neural Networks - Architectures, algorithms and applications."
Laurene Fausett, Prentice-Hall, 1994., ISBN: 0-13-334186-0
(Please notice that the algorithm is presented slightly differently from our textbook)

2.3.4 Perceptron Learning Rule Convergence Theorem

The statement and proof of the perceptron learning rule convergence theorem given here are similar to those presented in several other sources [Hertz, Krogh, & Palmer, 1991; Minsky & Papert, 1988; Arbib, 1987]. Each of these provides a slightly different perspective and insights into the essential aspects of the rule. The fact that the weight vector is perpendicular to the plane separating the input patterns at each step of the learning processes [Hertz, Krogh, & Palmer, 1991] can be used to interpret the degree of difficulty of training a perceptron for different types of input.

The perceptron learning rule is as follows:

Given a finite set of P input training vectors

$$\mathbf{x}(p), \quad p = 1, \ldots, P,$$

each with an associated target value

$$t(p), \quad p = 1, \ldots, P,$$

which is either $+1$ or -1, and an activation function $y = f(y\_in)$, where

$$y = \begin{cases} 1 & \text{if } y\_in > \theta \\ 0 & \text{if } -\theta \leq y\_in \leq \theta \\ -1 & \text{if } y\_in < -\theta, \end{cases}$$

the weights are updated as follows:

If $y \neq t$, then

$$\mathbf{w}\ (\text{new}) = \mathbf{w}\ (\text{old}) + t\mathbf{x};$$

else

no change in the weights.

The perceptron learning rule convergence theorem is:

If there is a weight vector $\mathbf{w}^*$ such that $f(\mathbf{x}(p) \cdot \mathbf{w}^*) = t(p)$ for all p, 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.

The proof of the theorem is simplified by the observation that the training set can be considered to consist of two parts:

$$F^+ = \{\mathbf{x} \text{ such that the target value is } +1\}$$

and

$$F^- = \{\mathbf{x} \text{ such that the target value is } -1\}.$$

A new training set is then defined as

$$F = F^+ \cup -F^-,$$

where

$$-F^- = \{-\mathbf{x} \text{ such that } \mathbf{x} \text{ is in } F^-\}.$$

In order to simplify the algebra slightly, we shall assume, without loss of generality, that $\theta = 0$ and $\alpha = 1$ in the proof. The existence of a solution of the original problem, namely the existence of a weight vector $\mathbf{w}^*$ for which

$$\mathbf{x}\cdot\mathbf{w}^* > 0 \qquad \text{if } \mathbf{x} \text{ is in } F^+$$

and

$$\mathbf{x}\cdot\mathbf{w}^* < 0 \qquad \text{if } \mathbf{x} \text{ is in } F^-,$$

is equivalent to the existence of a weight vector $\mathbf{w}^*$ for which

$$\mathbf{x}\cdot\mathbf{w}^* > 0 \qquad \text{if } \mathbf{x} \text{ is in } F.$$

All target values for the modified training set are $+1$. If the response of the net is incorrect for a given training input, the weights are updated according to

$$\mathbf{w}(\text{new}) = \mathbf{w}(\text{old}) + \mathbf{x}.$$

Note that the input training vectors must each have an additional component (which is always 1) included to account for the signal to the bias weight.

We now sketch the proof of this remarkable convergence theorem, because of the light that it sheds on the wide variety of forms of perceptron learning that are guaranteed to converge. As mentioned, we assume that the training set has been modified so that all targets are $+1$. Note that this will involve reversing the sign of all components (including the input component corresponding to the bias) for any input vectors for which the target was originally -1.

We now consider the sequence of input training vectors for which a weight change occurs. We must show that this sequence is finite.

Let the starting weights be denoted by $\mathbf{w}(0)$, the first new weights by $\mathbf{w}(1)$, etc. If $\mathbf{x}(0)$ is the first training vector for which an error has occurred, then

$$\mathbf{w}(1) = \mathbf{w}(0) + \mathbf{x}(0) \qquad (\text{where, by assumption, } \mathbf{x}(0)\cdot\mathbf{w}(0) \leq 0).$$

If another error occurs, we denote the vector $\mathbf{x}(1)$; $\mathbf{x}(1)$ may be the same as $\mathbf{x}(0)$ if no errors have occurred for any other training vectors, or $\mathbf{x}(1)$ may be different from $\mathbf{x}(0)$. In either case,

$$\mathbf{w}(2) = \mathbf{w}(1) + \mathbf{x}(1) \qquad (\text{where, by assumption, } \mathbf{x}(1)\cdot\mathbf{w}(1) \leq 0).$$

At any stage, say, k, of the process, the weights are changed if and only if the current weights fail to produce the correct (positive) response for the current input vector, i.e., if $\mathbf{x}(k-1)\cdot\mathbf{w}(k-1) \leq 0$. Combining the successive weight changes gives

$$\mathbf{w}(k) = \mathbf{w}(0) + \mathbf{x}(0) + \mathbf{x}(1) + \mathbf{x}(2) + \cdots + \mathbf{x}(k-1).$$

We now show that k cannot be arbitrarily large.

Let $\mathbf{w}^*$ be a weight vector such that $\mathbf{x}\cdot\mathbf{w}^* > 0$ for all training vectors in F. Let $m = \min\{\mathbf{x}\cdot\mathbf{w}^*\}$, where the minimum is taken over all training vectors in F; this minimum exists as long as there are only finitely many training vectors. Now,

$$\begin{aligned}\mathbf{w}(k)\cdot\mathbf{w}^* &= [\mathbf{w}(0) + \mathbf{x}(0) + \mathbf{x}(1) + \mathbf{x}(2) + \cdots + \mathbf{x}(k-1)]\cdot\mathbf{w}^* \\ &\geq \mathbf{w}(0)\cdot\mathbf{w}^* + km\end{aligned}$$

since $\mathbf{x}(i)\cdot\mathbf{w}^* \geq m$ for each i, $1 \leq i \leq P$.

The Cauchy-Schwartz inequality states that for any vectors $\mathbf{a}$ and $\mathbf{b}$,

$$(\mathbf{a}\cdot\mathbf{b})^2 \leq \|\mathbf{a}\|^2 \|\mathbf{b}\|^2,$$

or

$$\|\mathbf{a}\|^2 \geq \frac{(\mathbf{a}\cdot\mathbf{b})^2}{\|\mathbf{b}\|^2} \qquad (\text{for } \|\mathbf{b}\|^2 \neq 0).$$

Therefore,

$$\begin{aligned}\|\mathbf{w}(k)\|^2 &\geq \frac{(\mathbf{w}(k)\cdot\mathbf{w}^*)^2}{\|\mathbf{w}^*\|^2} \\ &\geq \frac{(\mathbf{w}(0)\cdot\mathbf{w}^* + km)^2}{\|\mathbf{w}^*\|^2}.\end{aligned}$$

This shows that the squared length of the weight vector grows faster than k^2, where k is the number of time the weights have changed.

However, to show that the length cannot continue to grow indefinitely, consider

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{x}(k-1),$$

together with the fact that

$$\mathbf{x}(k-1)\cdot\mathbf{w}(k-1) \leq 0.$$

By simple algebra,

$$\begin{aligned}\|\mathbf{w}(k)\|^2 &= \|\mathbf{w}(k-1)\|^2 + 2\mathbf{x}(k-1)\cdot\mathbf{w}(k-1) + \|\mathbf{x}(k-1)\|^2 \\ &\leq \|\mathbf{w}(k-1)\|^2 + \|\mathbf{x}(k-1)\|^2.\end{aligned}$$

Now let $M = \max \{\| \mathbf{x} \|^2 \text{ for all } \mathbf{x} \text{ in the training set}\}$; then

$$\begin{aligned}
\|\mathbf{w}(k)\|^2 &\leq \|\mathbf{w}(k-1)\|^2 + \|\mathbf{x}(k-1)\|^2 \\
&\leq \|\mathbf{w}(k-2)\|^2 + \|\mathbf{x}(k-2)\|^2 + \|\mathbf{x}(k-1)\|^2 \\
&\vdots \\
&\leq \|\mathbf{w}(0)\|^2 + \|\mathbf{x}(0)\|^2 + \cdots + \|\mathbf{x}(k-1)\|^2 \\
&\leq \|\mathbf{w}(0)\|^2 + kM.
\end{aligned}$$

Thus, the squared length grows less rapidly than linearly in k.

Combining the inequalities

$$\|\mathbf{w}(k)\|^2 \geq \frac{(\mathbf{w}(0)\mathbf{w}^* + km)^2}{\|\mathbf{w}^*\|^2}$$

and

$$\|\mathbf{w}(k)\|^2 \leq \|\mathbf{w}(0)\|^2 + kM$$

shows that the number of times that the weights may change is bounded. Specifically,

$$\frac{(\mathbf{w}(0)\cdot\mathbf{w}^* + km)^2}{\|\mathbf{w}^*\|^2} \leq \|\mathbf{w}(k)\|^2 \leq \|\mathbf{w}(0)\|^2 + kM.$$

Again, to simplify the algebra, assume (without loss of generality) that $\mathbf{w}(0) = 0$. Then the maximum possible number of times the weights may change is given by

$$\frac{(km)^2}{\|\mathbf{w}^*\|^2} \leq kM,$$

or

$$k \leq \frac{M \|\mathbf{w}^*\|^2}{m^2}.$$

Since the assumption that $\mathbf{w}^*$ exists can be restated, without loss of generality, as the assumption that there is a solution weight vector of unit length (and the definition of m is modified accordingly), the maximum number of weight updates is M/m^2. Note, however, that many more computations may be required, since very few input vectors may generate an error during any one epoch of training. Also, since $\mathbf{w}^*$ is unknown (and therefore, so is m), the number of weight updates cannot be predicted from the preceding inequality.

The foregoing proof shows that many variations in the perceptron learning rule are possible. Several of these variations are explicitly mentioned in Chapter 11 of Minsky and Papert (1988).

The original restriction that the coefficients of the patterns be binary is un-

necessary. All that is required is that there be a finite maximum norm of the training vectors (or at least a finite upper bound to the norm). Training may take a long time (a large number of steps) if there are training vectors that are very small in norm, since this would cause small m to have a small value. The argument of the proof is unchanged if a nonzero value of θ is used (although changing the value of θ may change a problem from solvable to unsolvable or vice versa). Also, the use of a learning rate other than 1 will not change the basic argument of the proof (see Exercise 2.8). Note that there is no requirement that there can be only finitely many training vectors, as long as the norm of the training vectors is bounded (and bounded away from 0 as well). The actual target values do not matter, either; the learning law simply requires that the weights be incremented by the input vector (or a multiple of it) whenever the response of the net is incorrect (and that the training vectors can be stated in such a way that they all should give the same response of the net).

Variations on the learning step include setting the learning rate α to any nonnegative constant (Minsky starts by setting it specifically to 1), setting α to $1/\|\mathbf{x}\|$ so that the weight change is a unit vector, and setting α to $(\mathbf{x}\cdot\mathbf{w})/\|\mathbf{x}\|^2$ (which makes the weight change just enough for the pattern $\mathbf{x}$ to be classified correctly at this step).

Minsky sets the initial weights equal to an arbitrary training pattern. Others usually indicate small random values.

Note also that since the procedure will converge from an arbitrary starting set of weights, the process is error correcting, as long as the errors do not occur too often (and the process is not stopped before error correction occurs).