Analysis of the Perceptron

The learning rule for a single output Perceptron with output in {0, 1} is similar to the Widrow-Hoff (or Delta) rule –

$$\underline{w}^{new} = \underline{w}^{old} + \left(y - y'\right) \cdot \underline{x}$$

In order to facilitate our analysis of the Perceptron we shall redefine the binary {0, 1} output value set to be {-1, +1}. The modified learning rule for a single output Perceptron with output in {-1, +1} and a learning rate, η, therefore becomes –

$$\underline{w}^{new} = \underline{w}^{old} + \eta \cdot \left(y - y'\right) \cdot \underline{x}$$

Strictly speaking η should be 0.5, of course, but we can use this term to vary the proportion of the error which we remove at each learning step if we do not insist on fixing it at 0.5. We must, however, ensure that $0 < \eta < 1$.

Since the Perceptron's output should now be -1 if the weighted sum of the inputs is negative and +1 otherwise we can combine these two cases into a single relationship –

$$\underline{w} \cdot \underline{x} \cdot y > 0$$

We note that the case when the weighted sum is 0 is now anomalous but the bias input of the Perceptron can be used to ensure that this situation doesn't arise, even when all of the actual inputs are required to be 0.

If we now define –

$$\underline{z} = \underline{x} \cdot y$$

then every solution must satisfy the relationship –

$$\underline{w} \cdot \underline{z} > 0$$

Multiple Output Perceptrons

We now consider the more general form of Perceptron with more than one output unit. The architecture of this system is simply an array of Perceptrons which all receive the same inputs (although multiplied by their own weights) and generate their own particular output.

y and *y'* now become vectors and *w* and *z* become matrices –

$$\underline{w}^{new} = \underline{w}^{old} + \eta \cdot \left(1 - \underline{w}^{old} \cdot \underline{z}\right) \cdot \underline{z}$$

Let us now relax the condition that the actual output values must be -1 or +1. Suppose we say that the outputs can be any size, *Ns*, above or below 0, where *N* is the number of inputs and *s* is the margin size (cf. the arbitrary overshoot target in the Widrow-Hoff Rule). We now have –

$$\underline{w} \cdot \underline{z} > Ns$$

as our criterion for solvability and we can rewrite the Perceptron Learning Rule as –

$$\underline{w}^{new} = \underline{w}^{old} + \eta \cdot \Theta\left(Ns - \underline{w}^{old} \cdot \underline{z}\right) \cdot \underline{z}$$

or

$$\Delta w_{ij} = \eta \cdot \Theta\left(Ns - y_i y_i^{'}\right) \cdot y_i x_j$$

for individual weight changes.

Note that the Heaviside function is defined as –

$$\Theta(x) = 1 \quad \text{when } x>0$$
$$= 0 \quad \text{otherwise}$$

Perceptron Convergence Theorem

Theorem

If a solution to a classification problem exists then the general Perceptron learning rule will find it in a finite number of steps.

Proof

Recall that a weight, w_{ik}, is updated only when

$$y_i y_i^{'} > Ns$$

i.e.

$$y_i \sum_k w_{ik} x_k > Ns$$

is NOT satisfied. If it is satisfied then we don't need to update the weight.

We prove the result for a single output and since each Perceptron has its own individual weight vector this will generalise to all outputs.

Let M^p denote the number of times that a pattern , p, has been used to update the weights at any stage in the learning process. At this time,

$$\underline{w} = \eta \cdot \sum_p M^p \cdot \underline{z}^p$$

i.e. $\underline{w}$ is the sum, over all patterns, of the weight changes for each pattern.

Now, consider the quantity –

$$\underline{w} \cdot \underline{w}^{*}$$

where $\underline{w}^*$ is a solution (i.e. a set of weight values that performs the required task).

$$\underline{w} \cdot \underline{w}^{*} = \eta \cdot \sum_{p} M^{p} \cdot \underline{z}^{p} \cdot \underline{w}^{*}$$

$$\geq \eta \cdot \left(\sum_{p} M^{p} \right) \cdot \min_{p} \left(\underline{z}^{p} \cdot \underline{w}^{*} \right)$$

i.e. it is bigger than if all of the changes had been the size of the smallest change.

In other words, if the total number of weight changes en route to a solution kept on increasing , then so would the value of $\underline{w}.\underline{w}^*$ so

$\underline{w} \cdot \underline{w}^{*}$ grows at the same rate as M, where $M = \sum_{p} M^{p}$.

Call this **RESULT A**
XXXXXXXXXXX

Now, consider the change in the magnitude squared of the weight vector at a single update caused by a pattern, *a* –

$$\begin{aligned}\Delta\left|\underline{\mathbf{w}}\right|^2 &= \left(\underline{\mathbf{w}} + \eta\underline{\mathbf{z}}^{\mathbf{a}}\right)^2 - \underline{\mathbf{w}}^2 \\ &= \eta^2\left(\underline{\mathbf{z}}^{\mathbf{a}}\right)^2 + 2\eta\,\underline{\mathbf{w}}\cdot\underline{\mathbf{z}}^{\mathbf{a}}\end{aligned}$$

But

$$\mathbf{Ns} \geq \underline{\mathbf{w}}\cdot\underline{\mathbf{z}}^{\mathbf{a}}$$

and each

$$\mathbf{z}_{\mathbf{k}}^{\mathbf{a}} = \pm\mathbf{1}$$

so

$$\left(\underline{\mathbf{z}}^{\mathbf{a}}\right)^2 = \mathbf{N}$$

and therefore –

$$\begin{aligned}\Delta\left|\underline{\mathbf{w}}\right|^2 &\leq \eta^2\mathbf{N} + 2\eta\mathbf{Ns} \\ &= \mathbf{N}\eta\left(\eta + 2\mathbf{s}\right)\end{aligned}$$

After *M* steps of incremental changes like this we have –

$$\left|\underline{\mathbf{w}}\right|^2 \leq \mathbf{MN}\eta\left(\eta + 2\mathbf{s}\right)$$

and therefore $|\underline{w}|$ grows no faster than $\sqrt{M}$.

Call this **RESULT B**
XXXXXXXXXXX

RESULT A and **RESULT B** together mean that

$$\frac{\underline{\mathbf{w}} \cdot \underline{\mathbf{w}}^{*}}{\left|\underline{\mathbf{w}}\right|}$$

grows at least as fast as $\sqrt{M}$.

So, if M kept on increasing then so would the above quotient.

But it can't - $\underline{w}^*$ is a solution and so won't change and $\underline{w}/|\underline{w}|$ is normalised so that can't grow either.

We therefore have to conclude that M cannot keep increasing and this means that the number of steps which the learning algorithm takes must be finite.

QED.