

# A Brief Introduction to Perceptrons

Perceptrons are a form of supervised learning from examples.

## Linear Threshold Functions

Suppose that each example consists of a label or target  $T$  (either  $+1$  or  $-1$ ) and a vector  $\mathbf{I}$  of  $n$  values  $(I_1, I_2, \dots, I_n)$  for  $n$  inputs. Let  $(\mathbf{I}, T)$  denote a labeled example, i.e., the vector and its label. Let  $\mathbf{W} = (W_1, W_2, \dots, W_n)$  be a weight vector of  $n$  values. Let  $\mathbf{W} \cdot \mathbf{I} = \sum_{j=1}^n W_j I_j$  be the dot product of  $\mathbf{W}$  and  $\mathbf{I}$ . Then  $\mathbf{W}$  can be used represent a linear threshold function  $f$  in the following way:

$$f(\mathbf{I}) = \begin{cases} +1 & \text{if } \mathbf{W} \cdot \mathbf{I} \geq 0 \\ -1 & \text{if } \mathbf{W} \cdot \mathbf{I} < 0 \end{cases}$$

Assume that one of features is permanently set to 1. The corresponding weight is called the *bias* or the *threshold*.

## Perceptron Algorithm

PERCEPTRON

$\mathbf{W} \leftarrow \mathbf{0}$

**loop** until convergence or out of patience

  get an example  $(\mathbf{I}, T)$

**if**  $T = 1$  and  $\mathbf{W} \cdot \mathbf{I} < 1$

**then**  $\mathbf{W} \leftarrow \mathbf{W} + \alpha \mathbf{I}$

**if**  $T = -1$  and  $\mathbf{W} \cdot \mathbf{I} > -1$

**then**  $\mathbf{W} \leftarrow \mathbf{W} - \alpha \mathbf{I}$

In other words, if the current weight vector  $\mathbf{W}$  misclassifies a labeled example  $(\mathbf{I}, T)$  (or is in the  $(-1, 1)$  range), then *update* the weights by adding/subtracting  $\alpha \mathbf{I}$  to/from  $\mathbf{W}$  depending on whether the example is positive or negative.

If the examples can be classified correctly by a linear threshold function, then PERCEPTRON will converge to a linear threshold function that correctly classifies the examples. The convergence is very efficient under certain conditions [2].

**Theorem 1** *If inputs are restricted to the interval  $[-1, 1]$ , and if there is a vector  $\mathbf{U}$  such that  $\mathbf{U} \cdot \mathbf{I} \geq 1$  for positive examples and  $\mathbf{U} \cdot \mathbf{I} \leq -1$  for negative examples, then PERCEPTRON will make at most  $(\mathbf{U} \cdot \mathbf{U})(2 + \alpha n)/\alpha$  updates.*

**Proof:** Let  $\mathbf{W}$  be the current weight vector. The inequality is proved by using  $\mathbf{U} \cdot \mathbf{W} \leq \|\mathbf{W}\| \|\mathbf{U}\|$  from vector arithmetic, where  $\|\mathbf{W}\| = \sqrt{\mathbf{W} \cdot \mathbf{W}}$ . First, consider  $\mathbf{U} \cdot \mathbf{W}$ .

If  $\mathbf{W} \cdot \mathbf{I} < 1$  on a positive example  $\mathbf{I}$ , then the new vector is  $\mathbf{W} + \alpha \mathbf{I}$ . Because  $\mathbf{U} \cdot \mathbf{I} \geq 1$  for positive examples, it follows that:

$$(\mathbf{W} + \alpha \mathbf{I}) \cdot \mathbf{U} = \mathbf{W} \cdot \mathbf{U} + \alpha \mathbf{I} \cdot \mathbf{U} \geq \mathbf{W} \cdot \mathbf{U} + \alpha$$

Similarly, if  $\mathbf{W} \cdot \mathbf{I} > -1$  on a negative example  $\mathbf{I}$ , then the new vector is  $\mathbf{W} - \alpha\mathbf{I}$ . Because  $\mathbf{U} \cdot \mathbf{I} \leq -1$  for negative examples, it follows that:

$$(\mathbf{W} - \alpha\mathbf{I}) \cdot \mathbf{u} = \mathbf{W} \cdot \mathbf{U} - \alpha\mathbf{I} \cdot \mathbf{U} \geq \mathbf{W} \cdot \mathbf{U} + \alpha$$

Thus, every time the perceptron algorithm updates the weights, then  $\mathbf{W} \cdot \mathbf{U}$  increases by at least  $\alpha$ . If  $\mathbf{W}$  is initially zero, then after  $k$  updates,  $\mathbf{W} \cdot \mathbf{U}$  is at least  $k\alpha$ .

Now, consider  $\|\mathbf{W}\|$ . Note that  $\|\mathbf{W}\| = \sqrt{\mathbf{W} \cdot \mathbf{W}}$ . Also note that because each input value is in the interval  $[-1, 1]$ , then  $\mathbf{I} \cdot \mathbf{I} \leq n$ .

If  $\mathbf{W} \cdot \mathbf{I} < 1$  on a positive example, then the new vector is  $\mathbf{W} + \alpha\mathbf{I}$ . It follows that:

$$(\mathbf{W} + \alpha\mathbf{I}) \cdot (\mathbf{W} + \alpha\mathbf{I}) = \mathbf{W} \cdot \mathbf{W} + 2\alpha\mathbf{W} \cdot \mathbf{I} + \alpha^2\mathbf{I} \cdot \mathbf{I} \leq \mathbf{W} \cdot \mathbf{W} + 2\alpha + \alpha^2n$$

Similarly, If  $\mathbf{W} \cdot \mathbf{I} > -1$  on a negative example, then the new vector is  $\mathbf{W} - \alpha\mathbf{I}$ . It follows that:

$$(\mathbf{W} - \alpha\mathbf{I}) \cdot (\mathbf{W} - \alpha\mathbf{I}) = \mathbf{W} \cdot \mathbf{W} - 2\alpha\mathbf{W} \cdot \mathbf{I} + \alpha^2\mathbf{I} \cdot \mathbf{I} \leq \mathbf{W} \cdot \mathbf{W} + 2\alpha + \alpha^2n$$

Thus, every time the perceptron algorithm updates the weights,  $\|\mathbf{W}\|^2$  increases by at most  $2\alpha + \alpha^2n$ . If  $\mathbf{W}$  is initially the zero vector, then after  $k$  updates,  $\|\mathbf{W}\|^2$  is at most  $2k\alpha + k\alpha^2n$ , so  $\|\mathbf{W}\|$  is at most  $\sqrt{2k\alpha + k\alpha^2n}$ .

From vector arithmetic,  $\mathbf{W} \cdot \mathbf{U} \leq \|\mathbf{W}\|\|\mathbf{U}\|$ . Based on the above results, we have:

$$k\alpha \leq \mathbf{W} \cdot \mathbf{U} \leq \|\mathbf{W}\|\|\mathbf{U}\| \leq \sqrt{2k\alpha + k\alpha^2n} \|\mathbf{U}\|$$

$k\alpha \leq \sqrt{2k\alpha + k\alpha^2n} \|\mathbf{U}\|$  implies  $k \leq (\mathbf{U} \cdot \mathbf{U})(2 + \alpha n)/\alpha$ , which proves the theorem.

**End Proof.**

If the examples cannot be classified correctly by a linear threshold function, and if the examples are sampled with replacement, then PERCEPTRON will eventually generate the optimal linear threshold function [1]. This convergence is very inefficient.

Much of my research is on this subject. You can look on my web page for some of my results.

## References

- [1] S. I. Gallant. Perceptron-based learning algorithms. *IEEE Trans. on Neural Networks*, 1:179–191, 1990.
- [2] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, Massachusetts, 1969.