Perceptron Algorithm

Andersen Ang

First created: 2013-Sep. Last update: 2017-Feb-1

Assume we have n inputs with n weights, i.e. we have a input vector and a weight vector

$$\mathbf{x} = \{x_1, x_2, ..., x_n\} \qquad \mathbf{w} = \{w_1, w_2, ..., w_n\}$$

We can form a *sum* of these 2 vector by an *integration function*

$$sum = f(\mathbf{x}, \mathbf{w})$$

For example, if the integration function is *dot product* , then

$$sum = f_{Dot}(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T\mathbf{w} = \sum x_i w_i = x_1 w_1 + x_2 w_2 + ... + x_n w_n$$

Then , the additional *thershold term* with it's weight is

$$x_{Th} \qquad w_{Th}$$

Let the thershold term be -1

$$-1 \qquad w_{Th}$$

Then the total sum (using dot product as integration function)is thus

$$sum_T = \mathbf{x}^T\mathbf{w} - x_{Th}w_{Th} = \mathbf{x}^T\mathbf{w} - w_{Th}$$

i.e the extened input vector is $\mathbf{x} = \{x_1, x_2, ..., x_n, -1\}$, and the extended weight vector is $\mathbf{w} = \{w_1, w_2, ..., w_n, w_{Th}\}$

Then such sum has to be pass to the activation function inorder to get the output

$$y = f_A(sum_T) = f_A\left(\mathbf{x}^T\mathbf{w} - w_{Th}\right)$$

For binary classification, the sign function can be used as activation function

$$y = \operatorname{sgn}\left(\mathbf{x}^T\mathbf{w} - w_{Th}\right)$$

Since perceptron is an *supervised learning* algorithm, thus there is *training set* $T(\mathbf{x}, y^d)$

Since the neural network initally works with random generated weight vectors, thus the output is a guess, and there is error

$$e = y^d - y$$

The error term can be used to "correct" or "update" the weight vector

The updated new weight can be expressed as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta\mathbf{w}_k$$

Where $\Delta\mathbf{w}_k$ is error time the current input

$$\Delta\mathbf{w}_k = e\mathbf{x}_k$$

The learning speed can be controlled by learning rate constant

$$\Delta\mathbf{w}_k = e\mathbf{x}_k\eta$$

Thus

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta\left(y^d - y\right)\mathbf{x}_k$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta\left(y^d - \text{sgn}\left(\mathbf{x}_k^T\mathbf{w_k} - w_{Th}\right)\right)\mathbf{x}_k$$

Therefore, given training set $T(\mathbf{x}, y)$. We first randomly generate a weight vector $\mathbf{w}$. Then use that training set to update $\mathbf{w}$

$-END-$