# Perceptrons and Perceptron Learning

Perceptron Pete has created a nice module (the PANDa) that he claims can be used ready-to-go as a logical AND unit, or after some training, as something to compute whatever binary logic function you want.

*x*0

*x*i ∈ {0, 1}

*w*0=1

*y* ∈ {0, 1}

*x*1

Σ

⎦⎡θ=0

*w*1=1

*w*2=−1.6

*x*2=1

1. Show that the PANDa correctly computes the logical AND function by filling in the following table:

*x*0 *x*1 *x*2 W • X *y*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | -1.6 | 0 |
| 0 | 1 | 1 | -0.6 | 0 |
| 1 | 0 | 1 | -0.6 | 0 |
| 1 | 1 | 1 | 0.4 | 1 |

2. Now, Xtreme Xerxes wants a module to compute the following function, and so he proceeds to train the PANDa above by using the training sequence 〈 *a, b, c, d, a, b, c, d,* … 〉 with learning rate α = 0.5.

*x*0 *x*1 *x*2  W • X *y*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *a* | 0 | 0 | 1 | -1.6 | 0 |
| *b* | 0 | 1 | 1 | -0.6 | 1 |
| *c* | 1 | 0 | 1 | -0.6 | 1 |
| *d* | 1 | 1 | 1 | 0.4 | 0 |

1. Show the weight vector W after the first step of training (after training example *a*).
2. Show the weight vector W after the first 2 steps of training (after training examples *a* and *b*).
3. Show the weight vector W after the first 3 steps of training (after training examples *a, b* and *c*).
4. Show the weight vector W after the first 4 steps of training (after training examples, *a, b, c* and *d*).
5. Give an estimate of the number of steps that will be needed to complete the training.

# UPDATE FORMULA FOR PERCEPTRON TRAINING:

if X is classified incorrectly, do the update

s = {+1 if Wi • X < 0; −1 otherwise}

Wi+1 = Wi + s α X