Perceptron Learning Rule

(1) *input*:

a training set $\left\{(x_n, y_n)\right\}_{n=1}^{N}$, where $x_n \in R^d$ and $y_n \in \{-1, +1\}$;

the maximum number of iterations T

(2) expand each x_n to $\mathbf{x}_n = \Big(-1, (x_n)_1, (x_n)_2, \ldots, (x_n)_d\Big)$

(3) initialize a $(d{+}1)$-dimensional vector $\mathbf{w}^{(0)}$ that represents $\Big(\theta^{(0)}, (w)_1^{(0)}, \ldots, w_d^{(0)}\Big)$, say,

$$\mathbf{w}^{(0)} \longleftarrow (0, 0, \ldots, 0)$$

(4) for $t = 1, 2, \ldots, T$

- randomly pick one n from $\{1, 2, \ldots, N\}$
- if $\left(y_n \neq \text{sign}\Big(\langle \mathbf{w}^{(t-1)}, \mathbf{x}_n \rangle\Big)\right)$, then

$$\mathbf{w}^{(t)} \longleftarrow \mathbf{w}^{(t-1)} + y_n \cdot \mathbf{x}_n \ ;$$

otherwise

$$\mathbf{w}^{(t)} \longleftarrow \mathbf{w}^{(t-1)} \ ,$$

where $\mathbf{w}^{(t)}$ represents $\Big(\theta^{(t)}, (w)_1^{(t)}, \ldots, w_d^{(t)}\Big)$

(5) *return*:

$g^{(T)}$, where $g^{(T)}(x) = \text{sign}\Big(\langle w^{(T)}, x \rangle - \theta^{(T)}\Big)$