Learning a Perceptron : What is a problem

- Training sample set

$$\{(x_i, t_i)\ ,\ i = 1,2,...,N, x_i \in R^d\ ,\ t_i = \{-1,1\}\}$$

- Training task

Learn $(w_1, w_2, \cdots, w_d, b)$ such that

$$\begin{cases} \sum_{k=1}^{d} w_k \bullet x_{ki} + b \geq 0 & if \quad t_i = +1 \\ \sum_{k=1}^{d} w_k \bullet x_{ki} + b \leq 0 & if \quad t_i = -1 \end{cases} \quad for\ all\ i$$

- Training task

Learn (W^T, b) such that

$$\begin{cases} W^T x_i + b \geq 0 & if \quad t_i = +1 \\ W^T x_i + b < 0 & if \quad t_i = -1 \end{cases} \quad for\ all\ i$$

- Training task

Learn (W^T, b) such that

$$\begin{cases} (W^T, b)\begin{bmatrix} x_i \\ 1 \end{bmatrix} \geq 0 & if \quad t_i = 1 \\ (W^T, b)\begin{bmatrix} x_i \\ 1 \end{bmatrix} < 0 & if \quad t_i = -1 \end{cases} \quad for\ all\ i$$

Learn $a = (W^T, b)$ such that

$$a^T y_i \geq 0 \quad for\ all\ i$$

where

$$y_i = \begin{cases} \begin{bmatrix} x_i \\ 1 \end{bmatrix} & if \quad t_i = +1 \\ -\begin{bmatrix} x_i \\ 1 \end{bmatrix} & if \quad t_i = -1 \end{cases}$$

Geometric explanation

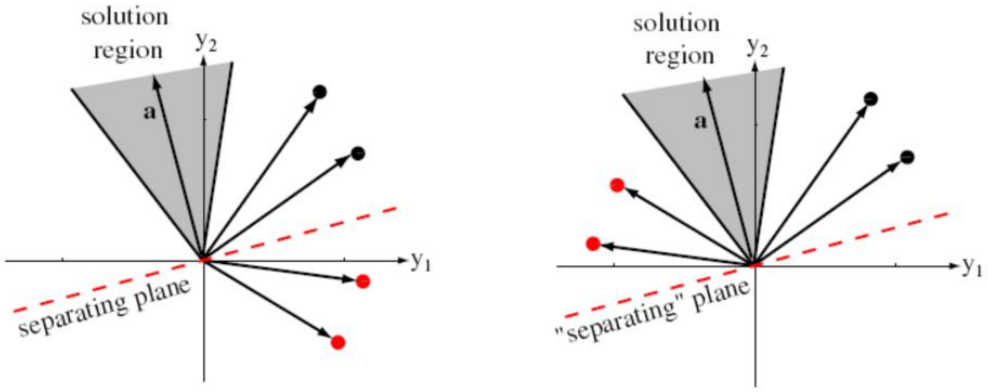


Figure 5.8: Four training samples (black for ω_1, red for ω_2) and the solution region in feature space. The figure on the left shows the raw data; the solution vectors leads to a plane that separates the patterns from the two categories. In the figure on the right, the red points have been "normalized" — i.e., changed in sign. Now the solution vector leads to a plane that places all "normalized" points on the same side.

How to learn a Perceptron

- Objective function

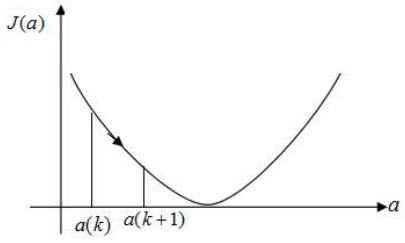
$$\mathrm{J}(a) = \sum_{y \in E} (-a^t y)$$

where E is the set of samples misclassified

- Learning method: steepest descent method

$$a(k+1) = a(k) - \eta(k) \bullet \frac{\partial J(a(k))}{\partial a}$$

- Algorithm



Perceptron learning algorithm

Algorithm 4 (Fixed-increment single-sample Perceptron)

```
1 begin initialize  a, k = 0
2         do  k ← (k + 1) mod n
3             if  y_k is misclassified by a  then  a ← a + y_k
4         until all patterns properly classified
5   return a
6 end
```

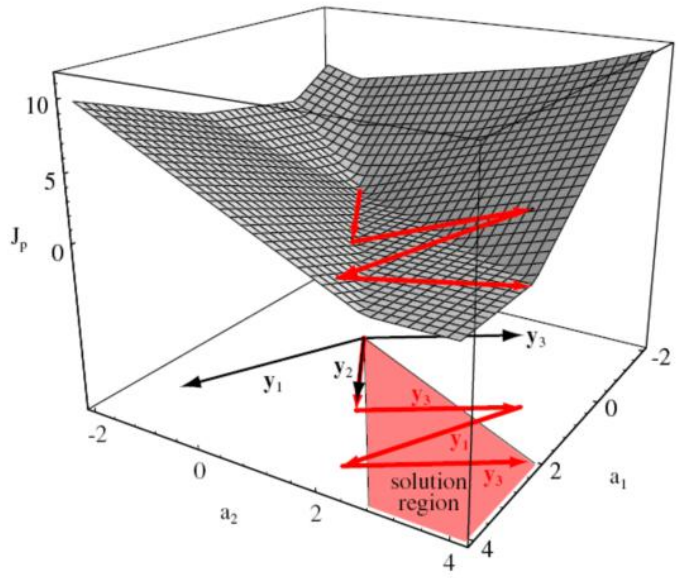


Figure 5.12: The Perceptron criterion, J_p is plotted as a function of the weights a_1 and a_2 for a three-pattern problem. The weight vector begins at $\mathbf{0}$, and the algorithm sequentially adds to it vectors equal to the "normalized" misclassified patterns themselves. In the example shown, this sequence is $\mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_1, \mathbf{y}_3$, at which time the vector lies in the solution region and iteration terminates. Note that the second update (by $\mathbf{y}_3$) takes the candidate vector *farther* from the solution region than after the first update (cf. Theorem 5.1. (In an alternate, batch method, *all* the misclassified points are added at each iteration step leading to a smoother trajectory in weight space.)