1.1.1.1 Ecuația unui hiperplan

Reamintim că ecuația unui hiperplan $\mathcal{H}$ ce trece printr-un punct $\mathbf{x}_0$ şi este normal pe un vector unitar $\mathbf{u}$ se poate scrie sub forma

$$(\mathbf{u}, \mathbf{x} - \mathbf{x}_0) = \mathbf{u}'(\mathbf{x} - \mathbf{x}_0) = 0$$

cu produsul scalar uzual.

Ecuația dreptei Δ ce trece printr-un punct $\mathbf{z}_0$ şi este ortogonală pe hiperplanul $\mathcal{H}$ de ecuație se scrie

$$\mathbf{x} - \mathbf{z}_0 = t\mathbf{u}, \quad t \in \mathbb{R}$$

adică

$$\mathbf{x} = \mathbf{z}_0 + t\mathbf{u}, \quad t \in \mathbb{R}$$

Pentru a găsi intersecția lui $\mathcal{H}$ cu Δ înlocuim ecuația dreptei în ecuația hiperplanului. Obținem

$$\mathbf{u}^T(\mathbf{z}_0 + t\mathbf{u} - \mathbf{x}_0) = 0,$$

şi deci

$$t\mathbf{u}^T\mathbf{u} = \mathbf{u}^T(\mathbf{x}_0 - \mathbf{z}_0),$$

de unde, ținând cont că $\|\mathbf{u}\| = 1$, găsim

$$t = \frac{\mathbf{u}^T(\mathbf{x}_0 - \mathbf{z}_0)}{\|\mathbf{u}\|^2} = \mathbf{u}^T(\mathbf{x}_0 - \mathbf{z}_0).$$

Punctul de intersecție al dreptei cu hiperplanul $\mathcal{H}$ va fi aşadar

$$\mathbf{x}' = \mathbf{z}_0 + \mathbf{u}^T(\mathbf{x}_0 - \mathbf{z}_0)\mathbf{u}.$$

Distanța de la punctul $\mathbf{z}_0$ la hiperplan este deci

$$\begin{aligned} d(\mathcal{H}, \mathbf{z}_0) &= \|\mathbf{x}' - \mathbf{z}_0\| \\ &= |\mathbf{u}'(\mathbf{x}_0 - \mathbf{z}_0)| \cdot \|\mathbf{u}\|. \\ &= |\mathbf{u}'(\mathbf{x}_0 - \mathbf{z}_0)| \end{aligned}$$

Distanța de la originea spațiului la hiperplan se obține punând în relația de mai sus $\mathbf{z}_0 = 0$ şi deci

$$D = d(\mathcal{H}, 0) = |\mathbf{u}'\mathbf{x}_0|.$$

2.1.1Generalizations of the Perceptron Learning Rule

The perceptron learning rule may be generalized to include a variable increment ρ^k and a fixed positive margin *b*. This generalized learning rule updates the weight vector whenever $\left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k$ fails to exceed the margin *b*. Here, the algorithm for weight vector update is given by

$$\begin{cases} \mathbf{w}^1 \text{ arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho^k \mathbf{z}^k \quad \text{if } \left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k \le b \\ \mathbf{w}^{k+1} = \mathbf{w}^k \qquad \text{otherwise} \end{cases} \tag{2.16}$$

The margin *b* is useful because it gives a dead-zone robustness to the decision boundary. That is, the perceptron's decision hyperplane is constrained to lie in a region between the two classes such that sufficient clearance is realized between this hyper-plane and the extreme points (boundary patterns) of the training set. This makes the perceptron robust with respect to noisy

inputs. It can be shown (Duda and Hart, 1973) that if the training set is linearly separable and if the following three conditions are satisfied:

1. $\rho^k \geq 0$ (2.17a)

2. $\lim_{m\to\infty} \sum_{k=1}^{m} \rho^k = \infty$ (2.17b)

3. $\lim_{m\to\infty} \frac{\sum_{k=1}^{m} \left(\rho^k\right)^2}{\left(\sum_{k=1}^{m} \rho^k\right)^2} = 0$ (2.17c)

(e.g., $\rho^k = \rho / k$ or even $\rho^k = \rho k$), then $\mathbf{w}^k$ converges to a solution $\mathbf{w}^*$ that satisfies $\left(\mathbf{z}^i\right)^{\mathrm{T}} \mathbf{w}^k > b$, for i = l, 2, ... , m. Furthermore, when ρ^k is fixed at a positive constant ρ, this learning rule converges in finite time.

Another variant of the perceptron learning rule is given by the *batch update* procedure

$$\begin{cases} \mathbf{w}^1 \quad \text{arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho \sum_{\mathbf{z} \in \mathbf{Z}\left(\mathbf{w}^k\right)} \mathbf{z} \end{cases} \tag{2.18}$$

where $\mathbf{Z}\left(\mathbf{w}^k\right)$ is the set of patterns $\mathbf{z}$ misclassified by $\mathbf{w}^k$. Here, the weight vector change $\Delta\mathbf{w} = \mathbf{w}^{k+1} - \mathbf{w}^k$ is along the direction of the resultant vector of all misclassified patterns. In general, this update procedure converges faster than the perceptron rule, but it requires more storage.

In the nonlinearly separable case, the preceding algorithms do not converge. Few theoretical results are available on the behavior of these algorithms for nonlinearly separable problems [see Minsky and Papert (1969) for some preliminary results]. For example, it is known that the length of $\mathbf{w}$ in the perceptron rule is bounded, i.e., tends to fluctuate near some limiting value $\left\|\mathbf{w}^*\right\|$. This information may be used to terminate the search for w*. Another approach is to average the

weight vectors near the fluctuation point $\mathbf{w}^*$. Butz (1967) proposed the use of a reinforcement factor γ, $0 \le \gamma \le 1$, in the perceptron learning rule. This reinforcement places **w** in a region that tends to minimize the probability of error for nonlinearly separable cases. Butz's rule is as follows:

$$\begin{cases} \mathbf{w}^1 \text{ arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho \mathbf{z}^k \quad \text{if } \left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k \le 0 \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho\gamma \mathbf{z}^k \quad \text{if } \left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k > 0 \end{cases} \tag{2.19}$$

2.1.2The Perceptron Criterion Function

It is interesting to see how the preceding error-correction rules can be derived by a gradient descent on an appropriate criterion (objective) function. For the perceptron, we may define the following criterion function (Duda and Hart, 1973):

$$J(\mathbf{w}) = -\sum_{\mathbf{z}\in Z(\mathbf{w})} \mathbf{z}^{\mathrm{T}}\mathbf{w} \qquad (2.20)$$

where $Z(\mathbf{w})$ is the set of samples misclassified by $\mathbf{w}$ (i.e., $\mathbf{z}^{\mathrm{T}}\mathbf{w} \le 0$). Note that if $Z(\mathbf{w})$ is empty, then $J(\mathbf{w}) = 0$; otherwise, $J(\mathbf{w}) > 0$. Geometrically, $J(\mathbf{w})$ is proportional to the sum of the distances from the misclassified samples to the decision boundary. The smaller J is, the better the weight vector $\mathbf{w}$ will be.

Given this objective function $J(\mathbf{w})$, the search point $\mathbf{w}^k$ can be incrementally improved at each iteration by sliding downhill on the surface defined by $J(\mathbf{w})$ in $\mathbf{w}$ space. Specifically, we may use J to perform a discrete gradient-descent search that updates $\mathbf{w}^k$ so that a step is taken downhill in the "steepest" direction along the search surface $J(\mathbf{w})$ at $\mathbf{w}^k$. This can be achieved

by making $\Delta\mathbf{w}^k$ proportional to the gradient of J at the present location $\mathbf{w}^k$; formally, we may write[12]

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \rho\nabla J\left(\mathbf{w}\right)|_{\mathbf{w}=\mathbf{w}^k} = \mathbf{w}^k - \rho\left[\frac{\partial J}{\partial w_1}\frac{\partial J}{\partial w_2}\ldots\frac{\partial J}{\partial w_{n+1}}\right]^{\mathrm{T}}|_{\mathbf{w}=\mathbf{w}^k} \quad (2.21)$$

Here, the initial search point $\mathbf{w}^1$ and the learning rate (step size) ρ are to be specified by the user. Equation (2.21) can be called the *steepest gradient descent search rule* or, simply, *gradient descent.* Next, substituting the gradient

---

[1] Discrete gradient-search methods are generally governed by the following equation:

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \alpha\mathbf{A}\nabla J|_{\mathbf{w}=\mathbf{w}^k}$$

Here, **A** is an $n\times n$ matrix and α is a real number, both are functions of $\mathbf{w}^k$. Numerous versions of gradient-search methods exist, and they differ in the way in which **A** and α are selected at $\mathbf{w}=\mathbf{w}^k$. For example, if **A** is taken to be the identity matrix, and if α is set to a small positive constant, the gradient "descent" search in Equation (2.21) is obtained. On the other hand, if α is a small negative constant, gradient "ascent" search is realized which seeks a local maximum. In either case, though, a saddle point (nonstable equilibrium) may be reached. However, the existence of noise in practical systems prevents convergence to such nonstable equilibriums.
It also should be noted that in addition to its simple structure, Equation (2.21) implements "steepest" descent. It can be shown that starting at a point **w**°, the gradient direction $\nabla J\left(\mathbf{w}^0\right)$ yields the greatest incremental increase of $J\left(\mathbf{w}\right)$ for a fixed incremental distance $\Delta\mathbf{w}^0 = \mathbf{w} - \mathbf{w}^0$. The speed of convergence of steepest descent search is affected by the choice of α, which is normally adjusted at each time step to make the most error correction subject to stability constraints.
Finally, it should be pointed out that setting A equal to the inverse of the Hessian matrix $\left[\nabla\nabla J\right]^{-1}$ and α to 1 results in the well-known Newton's search method.

$$\nabla J\left(\mathbf{w}^k\right) = -\sum_{\mathbf{z}\in Z\left(\mathbf{w}^k\right)} \mathbf{z} \qquad (2.22)$$

into Equation (2.21) leads to the weight update rule

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \rho \sum_{\mathbf{z}\in Z\left(\mathbf{w}^k\right)} \mathbf{z} \qquad (2.23)$$

The learning rule given in Equation (2.23) is identical to the multiple-sample (batch) perceptron rule of Equation (2.18). The original perceptron learning rule of Equation (2.3) can be thought of as an "incremental" gradient descent search rule for minimizing the perceptron criterion function in Equation (2.20). Following a similar procedure as in Equations (2.21) through (2.23) it can be shown that

$$J\left(\mathbf{w}\right) = -\sum_{\mathbf{z}^{\mathrm{T}}\mathbf{w}\le b} \left(\mathbf{z}^{\mathrm{T}}\mathbf{w} - b\right) \qquad (2.24)$$

is the appropriate criterion function for the modified perceptron rule in Equation (2.16).

Before moving on, it should be noted that the gradient of J in Equation (2.22) is not mathematically precise. Owing to the piecewise linear nature of J, sudden changes in the gradient of J occur every time the perceptron output y goes through a transition at $\left(\mathbf{z}^k\right)^{\mathrm{T}}\mathbf{w}=0$. Therefore, the gradient of J is not defined at "transition" points $\mathbf{w}$ satisfying $\left(\mathbf{z}^k\right)^{\mathrm{T}}\mathbf{w}=0$, k = 1, 2, ... , m. However, because of the discrete nature of Equation (2.21), the likelihood of $\mathbf{w}^k$ overlapping with one of these transition points is negligible, and thus we may still express ∇J as in Equation (2.22).

2.1.3Mays' Learning Rule

The criterion functions in Equations (2.20) and (2.24) are by no means the only functions that are minimized when $\mathbf{w}$ is a solution vector. For example, an alternative function is the quadratic function

$$J\left(\mathbf{w}\right)=\frac{1}{2}\sum_{\mathbf{z}^{\mathrm{T}}\mathbf{w}\le b}\left(\mathbf{z}^{\mathrm{T}}\mathbf{w}-b\right)^2 \qquad (2.25)$$

where b is a positive constant margin. Like the previous criterion functions, the function $J(\mathbf{w})$ in Equation (2.25) focuses attention on the misclassified samples. Its major difference is that its gradient is continuous, whereas the gradient of the perceptron criterion function, with or without the use of margin, is not. Unfortunately, the present function can be dominated by the input vectors with the largest magnitudes. We may eliminate this undesirable effect by dividing by $\|\mathbf{z}\|^2$:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{\mathbf{z}^T\mathbf{w} \le b} \frac{\left(\mathbf{z}^T\mathbf{w} - b\right)^2}{\|\mathbf{z}\|^2} \qquad (2.26)$$

The gradient of $J(\mathbf{w})$ in Equation (2.26) is given by

$$\nabla J(\mathbf{w}) = \sum_{\mathbf{z}^T\mathbf{w} \le b} \frac{\mathbf{z}^T\mathbf{w} - b}{\|\mathbf{z}\|^2} \mathbf{z} \qquad (2.27)$$

which, upon substituting in Equation (2.21), leads to the following learning rule

$$\begin{cases} \mathbf{w}^1 \quad \text{arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho \sum_{\mathbf{z}^{\mathrm{T}}\mathbf{w} \le b} \dfrac{b - \mathbf{z}^{\mathrm{T}}\mathbf{w}^k}{\left\|\mathbf{z}\right\|^2}\mathbf{z} \end{cases} \tag{2.28}$$

If we consider the incremental update version of Equation (2.28), we arrive at Mays' rule (Mays, 1964):

$$\begin{cases} \mathbf{w}^1 \quad \text{arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \rho \dfrac{b - \left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k}{\left\|\mathbf{z}\right\|^2}\mathbf{z}^k \quad \text{if } \left(\mathbf{z}^k\right)^{\mathrm{T}} \mathbf{w}^k \le b \\ \mathbf{w}^{k+1} = \mathbf{w}^k \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2.29}$$

If the training set is linearly separable, Mays' rule converges in a finite number of iterations, for $0 < \rho < 2$ (Duda and Hart, 1973). In the case of a nonlinearly separable training set, the training procedure in Equation (2.29) will never converge. To fix this problem, a decreasing learning rate such as $\rho^k = \rho / k$ may be used to force convergence to some approximate separating surface.