**Linear Classifiers and Perceptrons**

**N** samples: \( \{(x_\mu, w_\mu) : \mu = 1 \leftrightarrow N\} \leq \mu \leq N \)

Can we find a linear classifier that separates the positive and negative examples?

**Example:** a plane \( \mathbf{a} \cdot \mathbf{x} = 0 \) s.t. \( \text{sign}(\mathbf{a} \cdot \mathbf{x}) = w \)

- \( \mathbf{a} \cdot \mathbf{x}_\mu > 0 \), if \( w_\mu = +1 \)
- \( \mathbf{a} \cdot \mathbf{x}_\mu < 0 \), if \( w_\mu = -1 \)

Plane goes through the origin \((\mathbf{a} \cdot \mathbf{0} = 0)\)

**Perceptron Algorithm** (1950s)

First, replace -ve examples by the examples

If \( w_\mu = -1 \), set \( x_\mu \rightarrow -x_\mu \), \( w_\mu \rightarrow -\mathbf{a} \cdot \mathbf{x}_\mu \).

(Note: require \( \text{sign}(\mathbf{a} \cdot \mathbf{x}_\mu) = w_\mu \); this is equivalent to \( \text{sign}(-\mathbf{a} \cdot \mathbf{x}_\mu) = -w_\mu \))
This reduces to finding a plane
\[ \mathbf{x} \cdot \mathbf{a} \geq 0, \quad \text{for } \mu = \text{FN} \]

Note: the vector \( \mathbf{a} \) need not be unique. It is better to try to maximize the margin (see next lecture). To find a with \( \|\mathbf{a}\| = 1 \), so that \( \mathbf{a} \cdot \mathbf{\mu} \geq \mathbf{m} \), \( \forall \mu = \text{FN} \) for the maximum value of \( \mathbf{m} \).

More geometry

Claim:
\[ \sqrt{\mathbf{a} \cdot \mathbf{a}} \text{ is a unit vector} \]
\[ \|\mathbf{a}\| = 1, \text{ then } \mathbf{a} \cdot \mathbf{y} \text{ is the signed distance of } \mathbf{y} \]
\[ \text{to the plane } \mathbf{a} \cdot \mathbf{x} = 0. \]
\( \text{(i.e. } \mathbf{a} \cdot \mathbf{y} > 0, \text{ if } \mathbf{y} \text{ is above plane}) \)
\[ \mathbf{a} \cdot \mathbf{y} < 0, \text{ if } \mathbf{y} \text{ is below plane} \]

Proof: write \( \mathbf{y} = \lambda \mathbf{a} + \mathbf{y}_p \), where \( \mathbf{y}_p \) is the projection of \( \mathbf{y} \) into the plane. By definition \( \mathbf{a} \cdot \mathbf{y}_p = 0 \), hence \( \lambda = (\mathbf{a} \cdot \mathbf{y}) / \|\mathbf{a}\|^2 = (\mathbf{a} \cdot \mathbf{y}), \quad \|\mathbf{a}\| = 1. \)
(3) **Perceptron Algorithm**

Initialize: \( a(0) = 0 \).

Loop over \( \mu = 1 \to N \)

If \( x_\mu \) is misclassified, set \( a \to a + x_\mu \)

Repeat until all samples are classified correctly.

**Novikov’s Thm.** The Perceptron algorithm will converge to a solution weight that classifies all the samples correctly (provided this is possible).

**Proof.** Let \( \hat{a} \) be a separating weight.

Let \( m = \min_{\mu=1}^{N} \hat{a} \cdot x_\mu \quad (m > 0) \)

Let \( \beta = \max_{\mu=1}^{N} |x_\mu|^2 \)

Suppose \( x_+ \) is misclassified at time \( t \).

So \( \hat{a} \cdot x_+ < 0 \)

\[ a_{t+1} - (\beta \mu) \hat{a} = a_t + (\beta \mu) x_t + x_+ \]
\[ \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 = \| \mathbf{a}_t - (\beta^2 \mathbf{m}) \mathbf{\hat{a}} \|^2 + \| \mathbf{x}_t \|^2 - 2 (\mathbf{a}_t - (\beta^2 \mathbf{m}) \mathbf{\hat{a}}) \cdot \mathbf{x}_t. \]

Using \( \| \mathbf{x}_t \|^2 \leq \beta^2 \), \( \mathbf{a}_t \cdot \mathbf{x}_t < 0 \), \( -\mathbf{a}_t \cdot \mathbf{x}_t < -m \)

It follows that
\[ \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 \leq \| \mathbf{a}_t - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 + \beta^2 - 2 \beta^2 \mathbf{m}^2 \]

Hence
\[ \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 \leq \| \mathbf{a}_t - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 - \beta^2. \]

So, each time we update a weight, we reduce the quantity \( \| \mathbf{a}_t - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 \) by a fixed amount \( \beta^2 \). \( \| \mathbf{a}_t - \beta^2 \mathbf{m} \mathbf{\hat{a}} \|^2 \) is bounded by \( \frac{\beta^4 \| \mathbf{a} \|^2}{m^2} \).

So we can update the weights at most \( \frac{\beta^2 \| \mathbf{a} \|^2}{m^2} \) times,

Guarantees convergence.
(5) The Perceptron are very influential and unrealistic claims were made about its effectiveness.

\underline{Perceptron Capacity:} suppose we have \( n \) samples in a \( d \)-dimensional space. Assume the points are in general position (i.e., no subset of \( d+1 \) points lies in a \( d \)-dimensional subspace).

Let \( f(n,d) \) be the fraction of the \( 2^n \) dichotomies of the \( n \) points that can be expressed by linear separation.

It can be shown that \( f(n,d) = 1 \) for \( n < d+1 \), otherwise \( f(n,d) = \frac{2^d d}{2^n} \sum_{j=0}^{d} \frac{1}{j! (n-1-j)!} \).

There is a critical value \( 2(d+1) \):

\( f(n,d) = 1 \) for \( n < 2(d+1) \) and \( f(n,d) = 0 \) for \( n \geq 2(d+1) \).

\( n \)

These values of finding a separating hyperplane by chance alignment decreases rapidly for \( n > 2(d+1) \).
(6) Perceptrons can only represent a restricted set of decision rules (e.g., separation by hyperplane). This is a limitation and a virtue. If we can find a separating hyperplane, then it is probably not due to chance alignment of the data (provided $n > 2(d+1)$), and so it is likely to generalize.

Alternative: Multilevel Perceptrons.

Advantages:
(i) Can represent almost any classifier (in theory).
(ii) Have biological plausibility (dubious).
(iii) Very popular in the 1980s.

Comment: This is still a practical technique, but it never achieved its "hype". It has been bypassed by Support Vector Machines (SVM) and AdaBoost.
(7) Multilevel Perceptrons:

Two ingredients:
1) a standard perceptron has a discrete outcome
   \[ \text{sign}(a \cdot x) \in \{ -1, 1 \} \]

Replace this by a smooth sigmoid function:
\[ \sigma_T(a \cdot x) = \frac{1}{1 + e^{-(a \cdot x) / T}} \]

Note: \( \sigma_T(a \cdot x) \to \text{step function as } T \to 0 \).

(2) Introduce hidden units:

\[ h_a = \sigma \left( \sum_i w_{ai} x_i \right) \]
\[ w_a = \sigma \left( \sum_b w_{ab} h_b \right) \]

Can represent the full output as
\[ w_x = \sigma \left( \sum_i w_{bi} \sigma \left( \sum_b w_{bi} x_i \right) \right) \]
Train the system using a set of labelled samples \( \{(X_\mu, Y_\mu) : \mu = 1, \ldots, N\} \)

where \( X_\mu = (X^1_\mu, \ldots, X^M_\mu) \) — the input unit,

\( Y_\mu = (Y^1_\mu, \ldots, Y^M_\mu) \) — the output unit.

**Note:** we allow multiple classes (i.e., not just two classes).

Define an energy goodness of fit:

\[
E[\{\theta_i, \mu \}] = \frac{1}{N} \sum_{\mu=1}^{N} \sum_{\alpha=1}^{M} \left( \sum_{b} (\theta^\alpha_{b i} - \sigma \bar{z}_{\mu b}) \right)^2
\]

Minimize \( E \) w.r.t. \( \{\theta_i, \mu \} \).

**Note:** for statisticians, this is a form of non-linear regression.

Minimize \( \Delta \theta \) loop over \( \mu \),

\[
\frac{dE}{d\theta} = -\Delta \bar{z} \quad \frac{dE}{d\mu} = -2 \bar{z}
\]

Repeat.
(9) **Multi-layer Perceptrons.**

The update equations are messy algebraically but not impractical (see Duda, Hart & Stork).

There is no guarantee that the updates will converge to a global minimum.

Very hard to prove anything about multi-layer perceptrons except that they can represent any input output function.

Main problems -> what are the hidden units?

How many should there be?