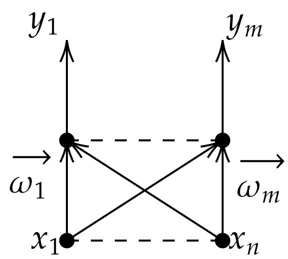
1. 简介

1.1. 单层前馈网络

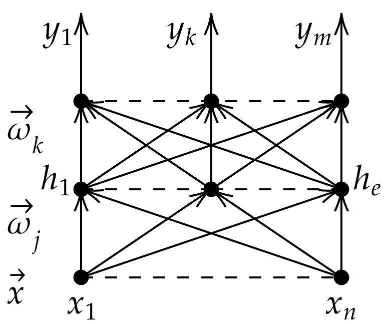


已知训练集 $\left\{\vec{x}^p, d^p\right\}_{p=i}, \vec{x}^p \in R^n, \vec{w}_j \in R^n, j = 1, 2, \cdots, m$

目标：求 $\vec{\omega}_1, \vec{\omega}_2, \cdots, \vec{\omega}_m$ 使得 J **最小**

$$J(\vec{\omega}_1, \vec{\omega}_2, \cdots, \vec{\omega}_m) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(d_j^p - y_j^p(\vec{\omega}_j)\right)^2$$

1.2. 多层前馈网络



已知训练集 $\left\{\vec{x}^p, \vec{d}^p\right\}_{p=i}^{N}, y_k^p = f\left(\vec{\omega}_k^\top, \vec{h}^p\right)$

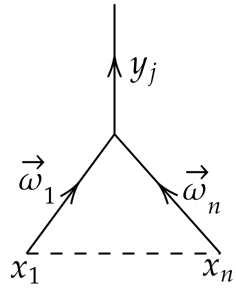
$h_j^p = f\left(\vec{\omega}_j^\top \vec{x}^p\right), \vec{\omega}_j \in R^n, \vec{\omega}_k \in R^l$

目标：求 $\vec{\omega}_j, \vec{\omega}_k$ 使得 J 最小

$$J(\vec{\omega}_j, \vec{\omega}_k) = \frac{1}{2}\sum_{p=i}^{N}\left|\left|\vec{d}^p - \vec{y}^p\right|\right|^2$$

2. 感知器

2.1. 神经元模型



$\vec{x} = (x_1, x_2, \cdots, x_n)^\top \quad \vec{\omega}_j = (\omega_{j1}, \omega_{j2}, \cdots, \omega_{jn})^\top$

$u_j = \vec{\omega}_j^\top \vec{x} \quad y_j = f(u_j) = \begin{cases} 1 & u_j \geqslant 0 \\ -1 & u_j < 0 \end{cases}$

训练集 $\left\{\vec{x}^p, d^p\right\}^N \quad \begin{cases} \vec{x}^p \in C_A & y_j = 1 \\ \vec{x}^p \in C_B & y_j = -1 \end{cases}$

2.2. 算法

目标函数：$J(\vec{\omega}_j) = \frac{1}{2}\sum_{j=1}^{N}\left(d^p - y_j^p\right)^2$

梯度法：$\Delta\vec{\omega}_j = -\eta\frac{\partial J(\vec{\omega}_j)}{\partial\vec{\omega}_j} = \eta\sum_{p=1}^{N}\left(d^p - y_j^p\right)\frac{y_j^p(\vec{\omega}_j)}{\partial\vec{\omega}_j} \quad \eta > 0$

作一个近似，$y_j^p = u_j^p = \vec{\omega}_j^\top\vec{x}^p \quad \vec{x}^p = \frac{\partial y_j^p(\vec{\omega}_j)}{\partial\vec{\omega}_j}$

则可知 $\Delta\vec{\omega}_j = \eta\sum_{i=1}^{N}\left(d^p - y_j^p\right)\vec{x}^p$

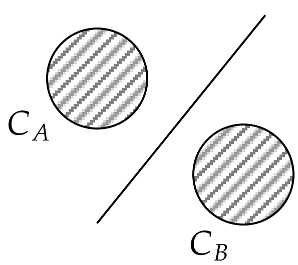
则算法的具体实现步骤如下：

- 随机产生 $\vec{\omega}_j(0)$
- 输入 $\vec{x}^p(p = 1, 2, \cdots, N)$，计算 y^p
- 调整权值 $\vec{\omega}_j(t) = \vec{\omega}_j(t-1) + \eta\sum_{p=1}^{N}\left(d^p - y_j^p\right)\vec{x}^p$
- 重复二三两步直到 $J(\vec{\omega}_j) < \varepsilon$ 即可

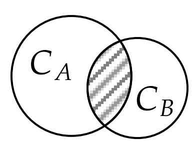
权调节的两种方式：

- 批处理
- 非批处理，一个输入样本可调整一次

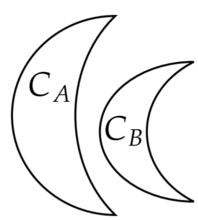
线性可分/不可分：



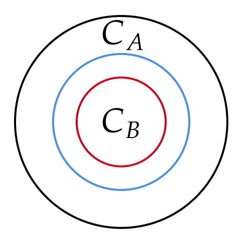
线性可分



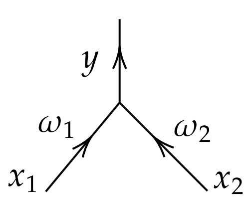
线性不可分



线性不可分

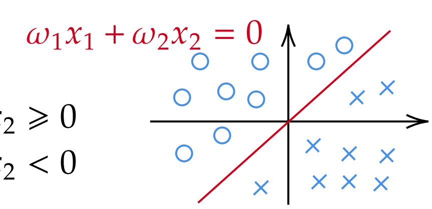


线性不可分



$$u = \omega_1 x_1 + \omega_2 x_2$$

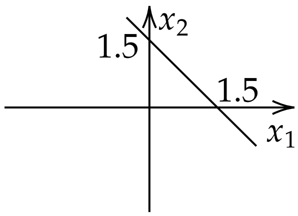
$$y = f(u) = \begin{cases} 1 & \omega_1 x_1 + \omega_2 x_2 \geqslant 0 \\ -1 & \omega_1 x_1 + \omega_2 x_2 < 0 \end{cases}$$



例：异或 $\vec{x} = (x_1, x_2)^\top \;\Rightarrow\; \begin{cases} (1,1)^\top \quad (-1,-1)^\top \quad d = -1 \\ (1,-1)^\top \quad (-1,1)^\top \quad d = 1 \end{cases}$

解：划分过程分为三步，如下

❖ 神经元 i，线性划分出 $(1,1)^\top, d = -1$

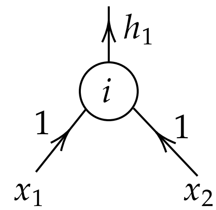


直线：$x_1 + x_2 - 1.5 = 0$

$\omega_{i1} = 1 \quad \omega_{i2} = 1 \quad \theta_i = 1.5$

对于 i：$(x_1, x_2)^\top = (1,1)^\top$时, $h_1 = 1$

其余 $h_1 = -1$



❖ 神经元 j，线性划分出 $(-1,-1)^\top, d = -1$

同上理可知，直线：$x_1 + x_2 + 1 = 0 \quad \omega_{j1} = 1 \quad \omega_{j2} = 1 \quad \theta_j = -1$

对于 j：$(x_1, x_2)^\top = (-1,-1)$ 时，$h_2 = -1$，其余 $h_2 = 1$

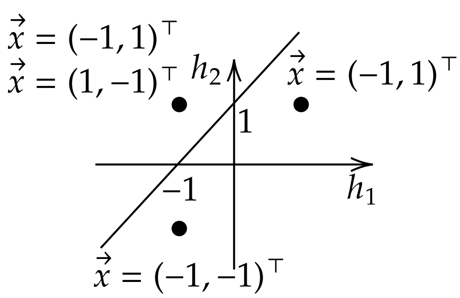
由上述划分结果可以知道：

➢ $\vec{x}^\top = (1,1)^\top \qquad h_1 = 1 \quad h_2 = 1 \quad d = -1$

➢ $\vec{x}^\top = (-1,1)^\top \; or \; (-1,-1)^\top \quad h_1 = -1 \quad h_2 = 1 \quad d = 1$

➢ $\vec{x}^\top = (-1,-1)^\top \quad h_1 = -1 \quad h_2 = -1 \quad d = 1$

❖ 神经元 k 进行划分 (h_1, h_2)

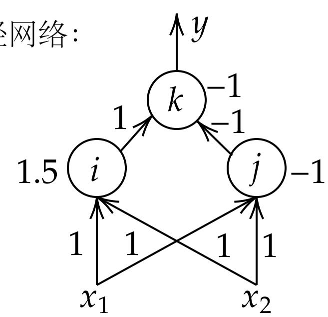


$h_1 - h_2 + 1 = 0$

$\omega_{k1} = 1 \quad \omega_{k2} = -1$

$\theta_k = -1$

总神经网络：



3. Back-Prepagation(BP)网络

3.1. 模型

$$y_k = \frac{1}{1 + exp(-u_k)} \quad h_j = \frac{1}{1 + exp(-u_j)} \quad \vec{u}_k = \vec{\omega}_k^\top \vec{h} \quad \vec{u}_j = \vec{\omega}_j^\top \vec{x}$$

目标函数：$\mathrm{J}(\vec{\omega}_k, \vec{\omega}_j) = \frac{1}{2}\sum_{p=1}^{N} \left|\left| \vec{d}^p - \vec{y}^p \right|\right|^2$，求 $\vec{\omega}_k, \vec{\omega}_j$ 使得 J 最小

3.2. 推导

3.2.1. $\Delta\vec{\omega}_k$

$$\Delta\vec{\omega}_k = -\eta\frac{\partial J}{\partial\vec{\omega}_k} = \eta\sum_{p=1}^{N}\left(d_k^p - y_k^p\right)\frac{\partial y_k^p}{\partial u_k^p}\frac{\partial u_k^p}{\partial\vec{\omega}_k}$$
$$= \eta\sum_{p=1}^{N}\left(d_k^p - y_k^p\right)\frac{1}{1+exp\left(-u_k^p\right)}\left(1-\frac{1}{1+exp\left(-u_k^p\right)}\right)\vec{h}$$
$$= \eta\sum_{p=1}^{N}\left(d_k^p - y_k^p\right)y_k^p\left(1-y_k^p\right)\vec{h} \qquad \eta > 0$$

3.2.2. $\Delta\vec{w}_j$

$$\Delta\vec{\omega}_j = -\eta\frac{\partial(\vec{\omega}_k\cdot\vec{w}_j)}{\partial\vec{\omega}_j} = \eta\sum_{p=1}^{N}\sum_{i=1}^{m}\left(d_k^p - y_k\right)\frac{\partial y_k}{\partial u_k^p}\frac{\partial u_k^p}{\partial\vec{\omega}_j}$$

$$\frac{\partial u_k^p}{\partial\vec{\omega}_j} = \frac{\partial\left(\vec{\omega}_k^{\top}\vec{h}\right)}{\partial\vec{\omega}_j} = \frac{\partial\left(\sum_{q=1}^{l}\omega_{kq}h_q^p\right)}{\partial\vec{\omega}_j} = \omega_{kj}\frac{\partial h_j}{\partial\vec{\omega}_j} = \omega_{kj}\frac{\partial h_j}{\partial u_j}\frac{\partial u_j}{\partial\omega_j} = \omega_{kj}h_j(1-h_j)\vec{x}$$

$$\Rightarrow\ \Delta\vec{\omega}_j = \eta\sum_{p=1}^{N}\sum_{k=1}^{m}\left(d_k^p - y_k^p\right)y_k^p\left(1-y_k^p\right)\omega_{kj}h_j(1-h_j)\vec{x}$$

3.3. 步骤

3.3.1. 随机选取初始权重

3.3.2. 前向计算 $h_j, y_k,\ \ j = 1,2,3,\ \cdots l,\ \cdots \quad k = 1,2,3,\ \cdots, m,\ \cdots$

3.3.3. 依次调整 $\vec{\omega}_k, \vec{\omega}_j$

3.3.4. 重复二和三两步，直到 $J(\vec{\omega}_k, \vec{\omega}_j) < \varepsilon$