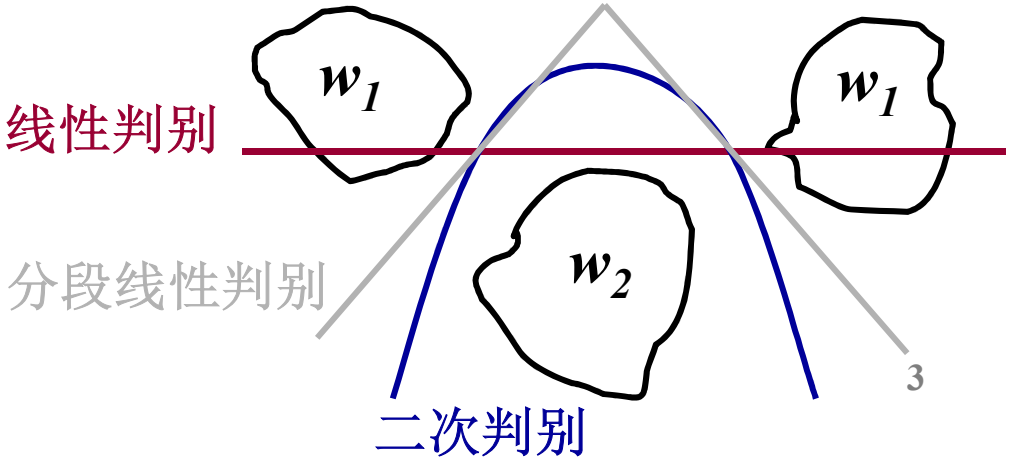
第5章 非线性分类器

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

分段线性判别函数

- 线性判别函数在进行分类决策时是最简单有效的，但在实际应用中，常常会出现不能用线性判别函数直接进行分类的情况。
- 采用广义线性判别函数的概念，可以通过增加维数来得到线性判别，但维数的大量增加会使在低维空间里在解析和计算上行得通的方法在高维空间遇到困难，增加计算的复杂性。
- 引入分段线性判别函数的判别过程，它比一般的线性判别函数的错误率小，但又比非线性判别函数简单。



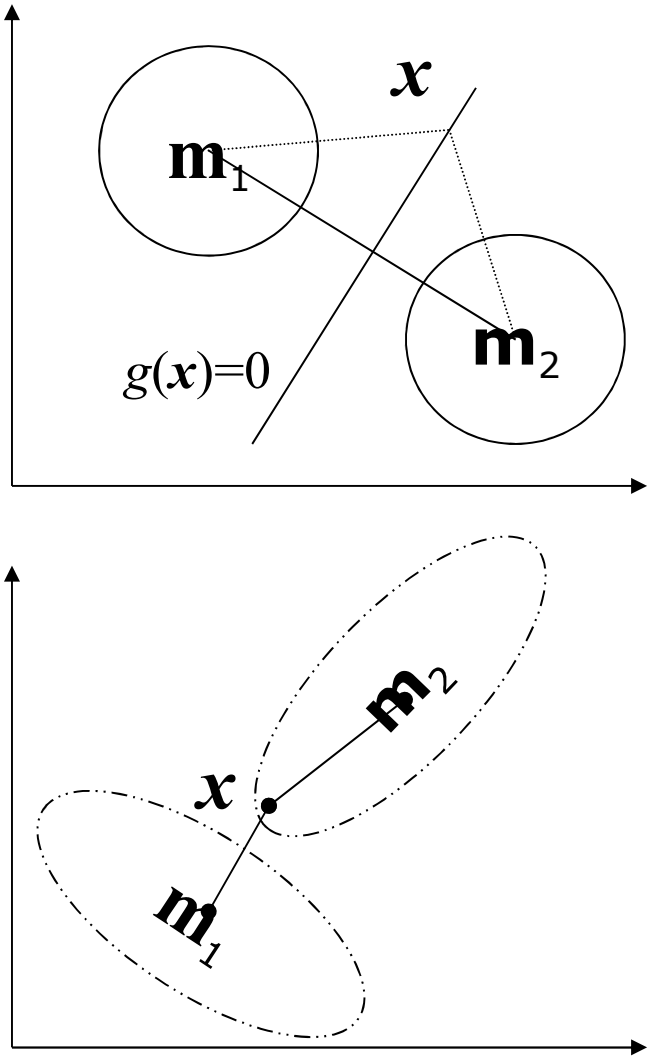
基于距离的分段线性判别函数

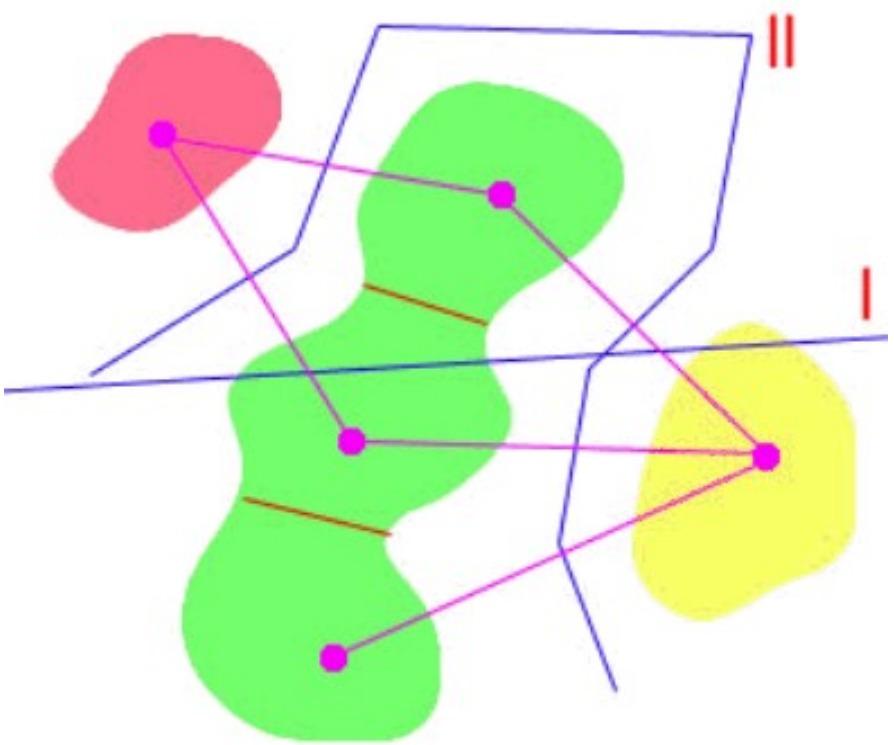
- 最小距离分类器：把各类别样本特征的均值向量作为各类的代表点(prototype)，根据待识样本到各类别代表点的最小距离判别其类别。决策面是两类别均值连线的垂直平分面
- 分段线性距离分类器：将各类别划分成相对密集的子类，每个子类以它们的均值作为代表点，然后按最小距离分类
- 判别函数定义：ω_i有l_i个子类，即属于ω_i 的决策域R_i分成l_i个子域$R_i^1, R_i^2,\ldots, R_i^{li}$)，每个子区域用均值$m_i^k$代表点

$$g_i(\mathbf{x}) = \min_{k=1,\ldots,l_i} \left\| \mathbf{x} - \mathbf{m}_i^k \right\|$$

- 判别规则：$$\text{if } g_j(\mathbf{x}) = \min_{i=1,\ldots,c} g_i(\mathbf{x}) \text{ then } \mathbf{x} \in \omega_j$$

分段线性距离分类器图例





Ⅰ： 线性距离判别

Ⅱ： 分段线性距离判别

分段线性判别函数

- 分段线性判别函数的一般形式：$g_i^k(\mathbf{x})$表示第i类第k段线性判别函数，l_i为i类所具有的判别函数个数，$\mathbf{w}_i^k$与w_{i0}^k分别是第k段的权向量与阈值权

$$g_i^k(\mathbf{x}) = \mathbf{w}_i^{(k)T}\mathbf{x} + w_{i0}^k,\ k = 1,2,...,l_i; i = 1,...,c$$

- 第i类的判别函数：$g_i(\mathbf{x}) = \max_{k=1,...,l_i} g_i^k(\mathbf{x})$

- 判别规则：$\text{if } g_j(\mathbf{x}) = \max_{i=1,...,c} g_i(\mathbf{x}) \text{ then } \mathbf{x} \in \omega_j$

- 决策面取决于相邻的决策域，如第i类的第n个子类与第j类的第m个子类相邻，则由它们共同决定的决策面方程为

$$g_i^n(\mathbf{x}) = g_j^m(\mathbf{x})$$

Outline:

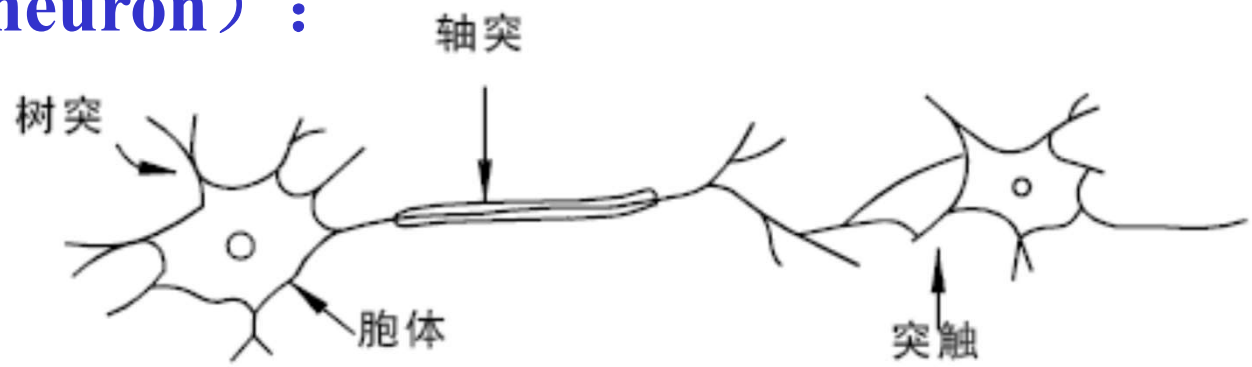
- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

神经元与感知器

神经元（neuron）：



- ✓细胞体（**cell**）、树突（**dentrite**）、轴突（**axon**）、突触（**synapses**）
- ✓神经元的作用：加工、传递信息（电脉冲信号）
- ✓神经系统：神经网：大量神经元的复杂连接
- ✓通过大量简单单元的广泛、复杂的连接而实现各种智能活动

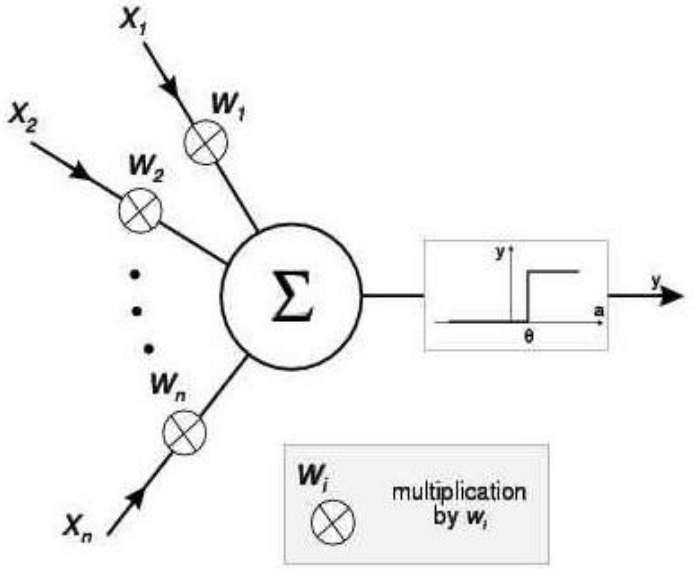
神经元与感知器

- （人工）神经网络的基本结构：
  - ✓ 大量简单的计算单元（结点）以某种形式相连接，形成一个网络，其中的某些因素，如连接强度（权值）、结点计算特性甚至网络结构等，可依某种规则随外部数据进行适当的调整，最终实现某种功能.
- 基本工作机制
  - ✓ 一个神经元有两种状态：兴奋与抑制。平时处于抑制状态的神经元，其树突和胞体接收其他神经元经由突触传来的兴奋电位，多个输入在神经元中以代数和的方式叠加；如果输入兴奋总量超过某个阈值，神经元就会被激发进入兴奋状态，发出输出脉冲，并由轴突的突触传递给其他神经元。
  - ✓ 神经元是按照“全和无”的原则工作的，只有兴奋和抑制两种状态

神经元与感知器

基本的神经元模型

McCulloch-Pitts Model (1943)



神经元的动作：

$$net = \sum_{i=1}^{c} \omega_i x_i$$

$$y = f(net)$$

当f为阈值函数时：

$$y = \mathrm{sgn}\left(\sum_{i=1}^{n} \omega_i x_i - \theta \right)$$

几何意义？

$$\left.\begin{array}{l} \theta = -\omega_0 \\ W = (\omega_0, \omega_1, \omega_2, \cdots, \omega_n)^T \\ X = (1, x_1, x_2, \cdots, x_n)^T \end{array}\right\} \Rightarrow y = \mathrm{sgn}(W^T X)$$

神经元与感知器

基本的神经元模型

McCulloch-Pitts Model (1943)

输出函数f的选取：

$$f(x)=\begin{cases}1 & (x>0)\\ 0 & (x\le 0)\end{cases} \qquad f(x)=\mathrm{sgn}(x)=\begin{cases}1 & (x\ge 0)\\ -1 & (x<0)\end{cases}$$

$$Sigmoid\ \ function: f(x)=th(x)=\frac{2}{1+e^{-2x}}-1$$

$$then\ \ y\in(-1,1)$$

$$Sigmoid\ \ function: f(x)=\frac{2}{1+e^{-x}}$$

$$then\ \ y\in(0,1)$$

神经元与感知器

基本的神经元模型
McCulloch-Pitts Model (1943)

输出函数f的选取：

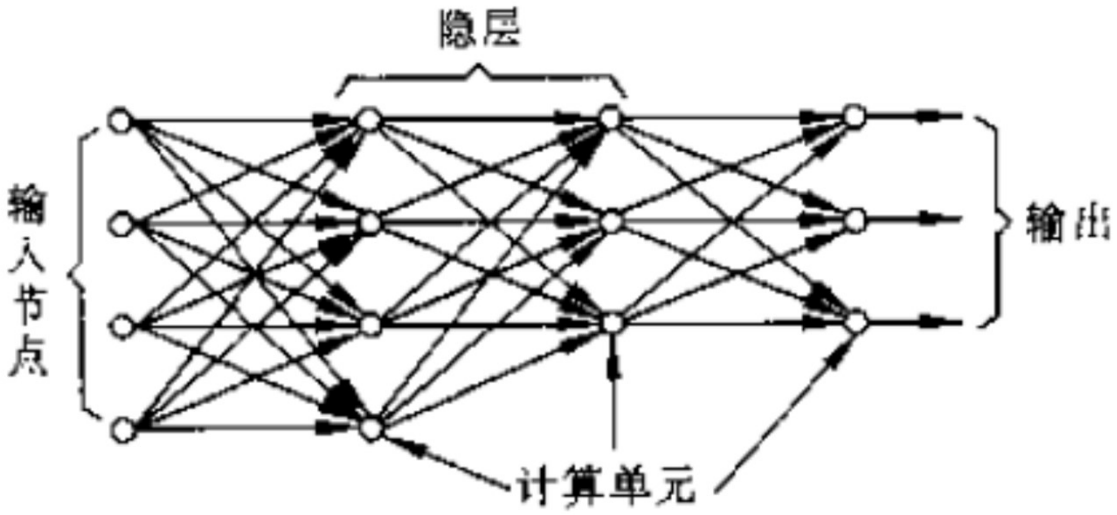
- **Sigmoid函数特征：**
  - ✓ 非线性，单调性
  - ✓ 无限次可微
  - ✓ 当权值很大时可近似阈值函数
  - ✓ 当权值很小时可近似线性函数

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

前馈神经网络

前馈多层感知器结构



$$out_i = f\left(net_i\right) = f\left(\sum_j w_{ij} out_j - \theta_i\right)$$

常用的非线性兴奋函数：

$$f\left(net_i\right) = \frac{2}{1+e^{-2net_i}} - 1$$

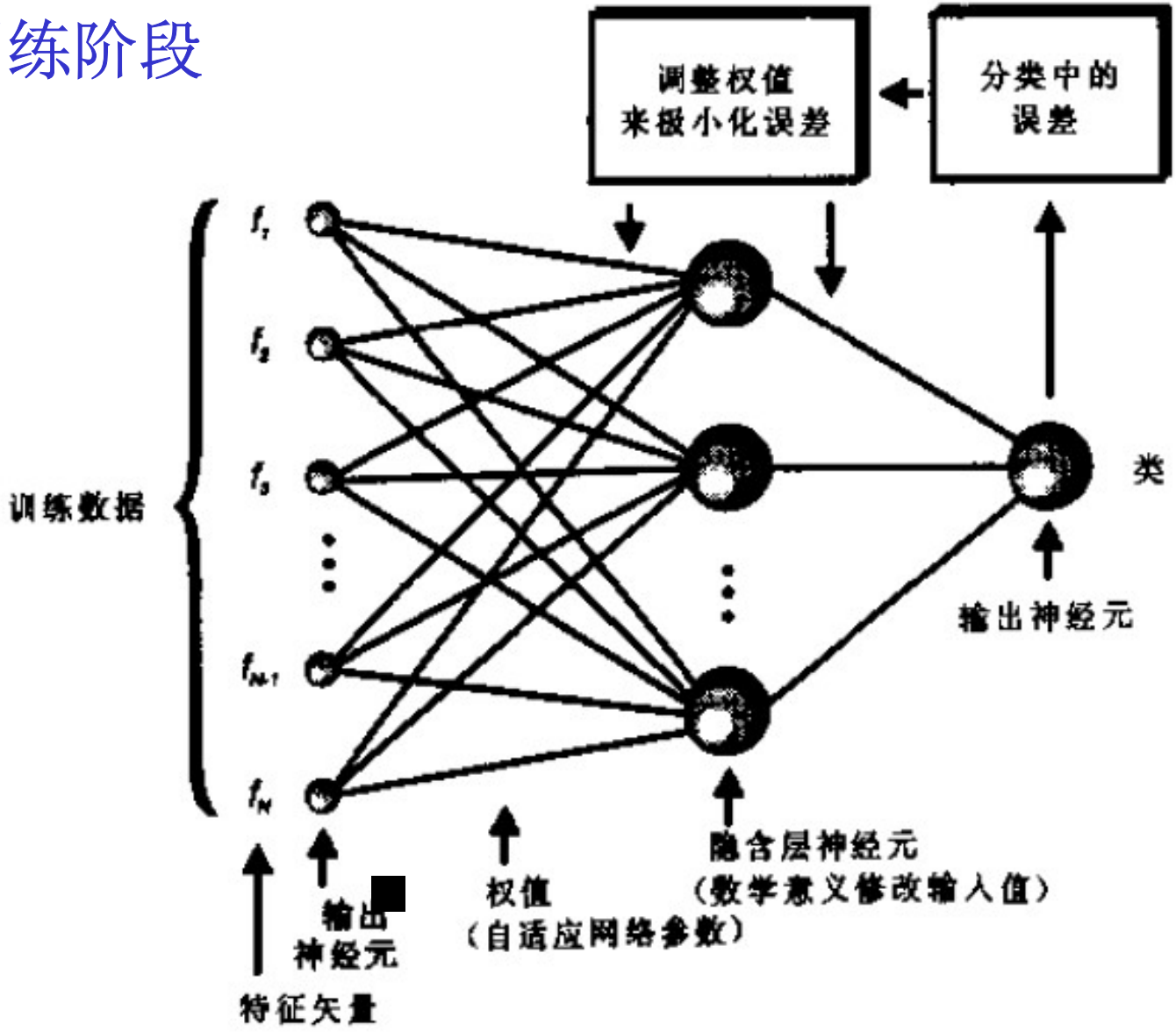
$$f\left(net_i\right) = \frac{1}{1+e^{-net_i}}$$

前馈神经网络

- 把输入模式映射到相应分类器所需知识由权值来体现
- 训练：寻找有用权值
- 网络训练过程，包括从训练集合到权值集合的映射。至少在给定误差内，该组权值可对训练集矢量正确分类。实际上，网络所学正是训练集所教
- 把神经网络应用于**PR**问题包括两个阶段：网络训练阶段、预测阶段

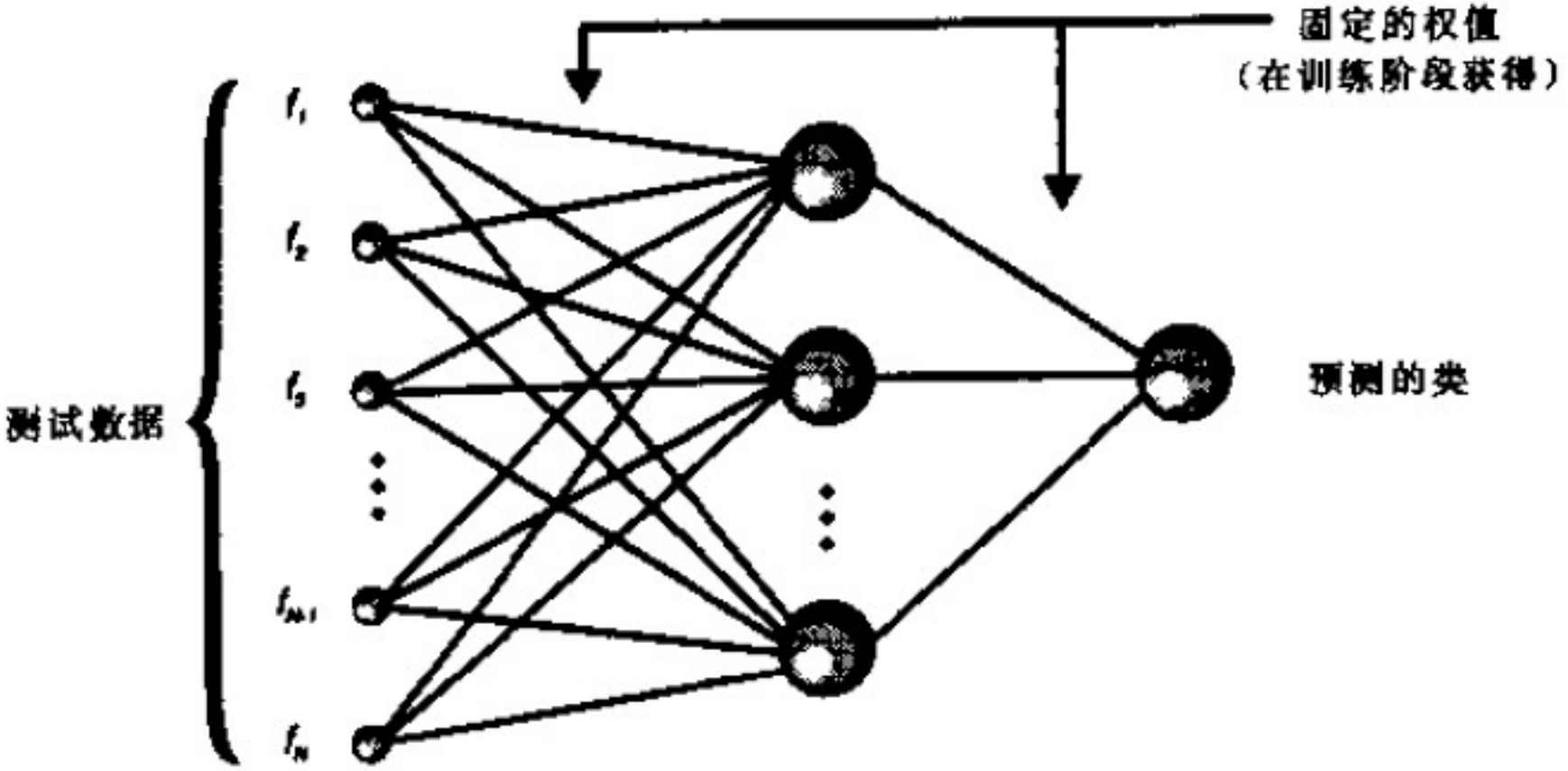
前馈神经网络

训练阶段


调整权值
来极小化误差
分类中的
误差
训练数据
类
输出神经元
隐含层神经元
(数学意义修改输入值)
权值
(自适应网络参数)
输出
神经元
特征矢量

前馈神经网络

预测阶段



Outline:

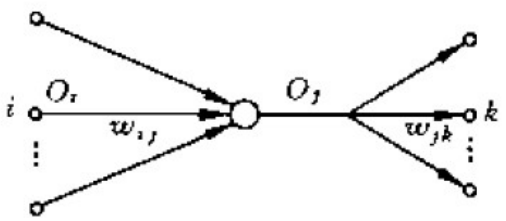
- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

利用BP算法进行网络训练

- 三层前馈网络可以逼近任意的多元非线性函数
- **BP算法思想：**
  - ✓从后向前逐层传播输出层的误差，以间接算出隐层误差。算法包含两个阶段：阶段一（正向过程）：输入信息从输入层经隐层逐层计算各单元的输出值；阶段二（反向传播过程）：输出误差逐层向前算出隐层各单元的误差，并用此误差修正前层权值。

BP法

在反向传播算法中通常采用梯度法修正权值，为此要求输出函数可微，通常采用 Sigmoid 函数作为输出函数。不失其普遍性，我们研究处于某一层的第 j 个计算单元，脚标 i 代表其前层第 i 个单元，脚标 k 代表后层第 k 个单元，O_j 代表本层输出，w_{ij}是前层到本层的权值



反向传播算法中的音量约定

当输入某个样本时，从前到后对每层各单元作如下计算（正向算法）

$$net_j = \sum_i w_{ij} O_i$$

$$O_j = f(net_j)$$

对于输出层而言，$\hat{y}_j = O_j$ 是实际输出值，y_j 是理想输出值，此样本下的误差

$$E = \frac{1}{2}\sum_j (y_j - \hat{y}_j)^2$$

为使式子简化，定义局部梯度

$$\delta_j = \frac{\partial E}{\partial net_j}$$

BP法

考虑权值 w_{ij} 对误差的影响，可得

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} = \delta_j O_i$$

权值修正应使误差最快地减小，修正量为

$$\Delta w_{ij} = -\eta\delta_j O_i$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

如果节点 j 是输出单元，则

$$O_j = \hat{y}_j$$

$$\delta_j = \frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial net_j} = -(y_j - \hat{y}_j)f'(net_j)$$

如果节点 j 不是输出单元，由图 11.6 可知，O_j 对后层的全部节点都有影响。因此，

$$\delta_j = \frac{\partial E}{\partial net_j} = \sum_k \frac{\partial E}{\partial net_k}\cdot\frac{\partial net_k}{\partial O_j}\cdot\frac{\partial O_j}{\partial net_j}$$

$$= \sum_k \delta_k w_{jk} f'(net_j)$$

BP法

对于 Sigmoid 函数

$$y = f(x) = \frac{1}{1 + e^{-x}}$$

有

$$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = y(1-y)$$

或者当

$$y = f(x) = \text{th}x$$

时有

$$f'(x) = 1 - \text{th}^2 x = 1 - y^2$$

$$\Delta w_{ij}(t) = -\eta \delta_j O_i + \alpha \Delta w_{ij}(t-1)$$

反向传播算法(BP)步骤

(1) 选定权系数初始值。

(2) 重复下述过程直至收敛(对各样本依次计算)。

①从前向后各层计算各单元 O_j

$$net_j = \sum_i w_{ij} O_i$$

$$O_j = 1/(1 - e^{-net_j})$$

② 对输出层计算 δ_j

$$\delta_j = (y - O_j) O_j (1 \cdots O_j)$$

③ 从后向前计算各隐层 δ_j

$$\delta_j = O_j (1 - O_j) \sum_k w_{jk} \delta_k$$

④ 计算并保存各权值修正量

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) + \eta \delta_j O_i$$

⑤ 修正权值

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

BP法

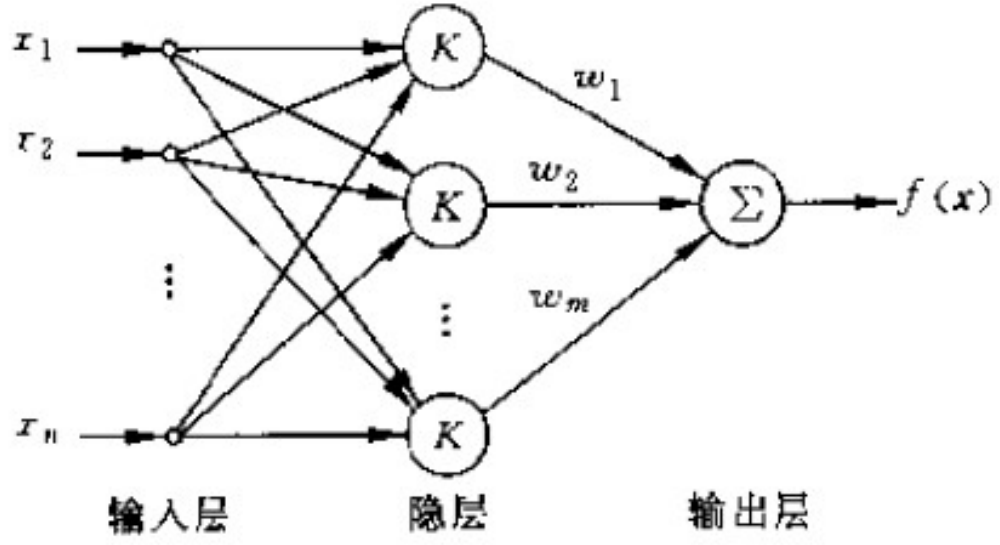
BP神经网络的输出层与输入层单元数是由问题的本身决定的。在作为模式识别时输入单元数是特征维数，输出单元数是类数。

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

径向基函数网络

- 特点：
  - ✓前馈型神经网络，只有一个隐层，隐层单元采用径向基函数作为其输出特征，输入层到隐层之间的权值均固定为1；输出节点为线性求和单元，隐层到输出节点之间的权值可调，因此输出为隐层的加权求和。



所谓径向基函数(Radial Basis Function 简称 RBF)，就是某种沿径向对称的标量函数。通常定义为空间中任一点 x 到某一中心 x_c 之间欧氏距离的单调函数，可记作 $k(\|x-x_c\|)$，其作用往往是局部的，即常 x 远离 x_c 时函数取值很小。最常用的径向基函数是高斯核函数，形式为

$$k(\|x-x_c\|)=\exp\left\{-\frac{\|x-x_c\|^2}{2\sigma^2}\right\}$$

其中 x_c 为核函数中心，σ 为函数的宽度参数，控制了函数的径向作用范围。在 RBF 网络中，这两个参数往往是可调的。

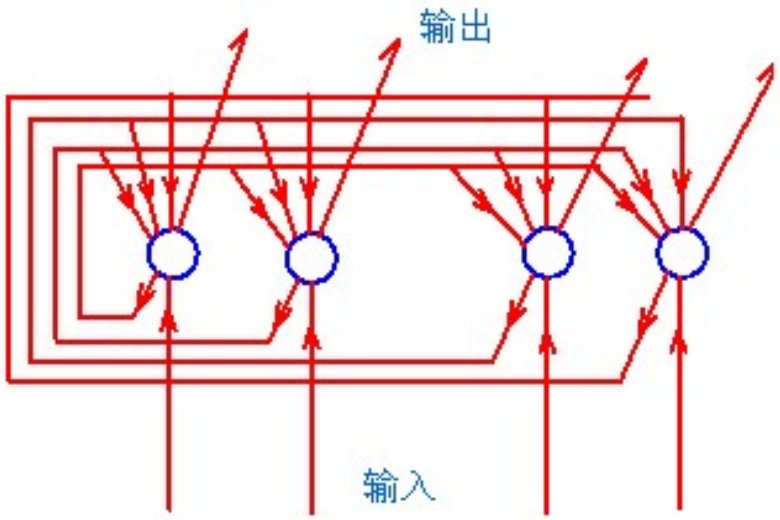
➢RBF网络的理解：

- ✓把网络看成对未知函数 $f(x)$的逼近器。一般任何函数都可表示成一组基函数的加权和，这相当于用隐层单元的输出函数构成一组基函数来逼近 $f(x)$
- ✓在**RBF**网络中，从输入层到隐层的基函数输出是一种非线性映射，而输出则是线性的。这样，**RBF**网络可以看成是首先将原始的非线性可分的特征空间变换成另一空间，通过合理选择这一变换使在新空间中原问题线性可分，然后用一个线性单元来解决问题

Outline:

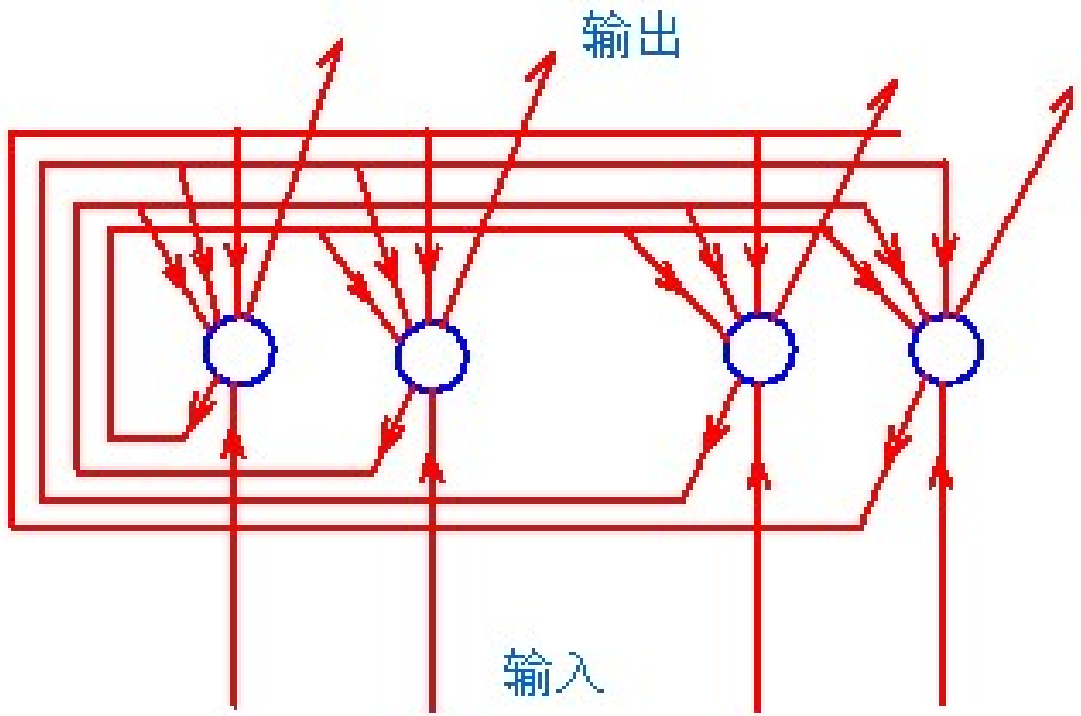
- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

Hopfield网络



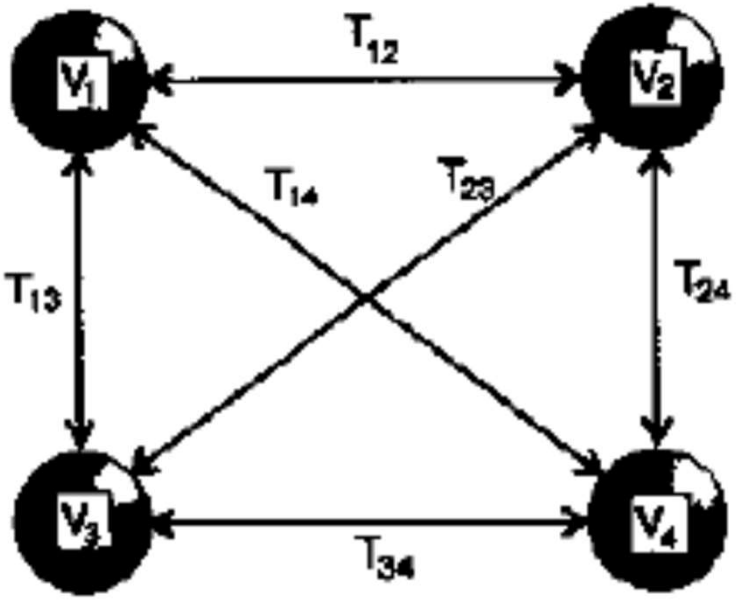
与前述的前馈网络不同，Hopfield 网络是一种反馈网络。反馈网络的基本单元是与前馈网络类似的神经元，其特性可以是阈值函数或 sigmoid 函数。反馈网络的结构是单层的，各单元地位平等，每个神经元都可以与所有其他神经元连接。如果考虑一个二层前馈网络，其输出层与输入层的神经元数相同，每一个输出都直接连接（反馈）到相对应的一个输入上，该网络就等价于一个反馈网络。人们通常把反馈网络看成动态系统，主要关心其随时间变化的动态过程。反馈网络具有一般非线性系统的许多性质，如稳定性问题、各种类型的吸引子以及混沌现象等，在某些情况下还有随机性、不可预测性。

Hopfield网络



- 神经元之间相互耦合， 动力系统
- 存储信息，并具有联想记忆功能

➢Hopfield示例



拟存储的模式

$$a^{(1)}=\begin{bmatrix}1\\0\\1\\0\end{bmatrix}\quad a^{(2)}=\begin{bmatrix}0\\1\\0\\1\end{bmatrix}$$

$$T=\begin{bmatrix}0&-2&2&-2\\-2&0&-2&2\\2&-2&0&-2\\-2&2&-2&0\end{bmatrix}$$

输入一矢量:$[1\ \ 1\ \ 1\ \ 0]^T$ → $[1\ \ 0\ \ 1\ \ 0]^T$

神经网络PR的典型做法

- 对网络进行训练学习
- 多输出型

网络的每一个输入节点对应样本一个特征，而输出层节点数等于类别数，一个输出节点对应一个类。在训练阶段，如果输入训练样本的类别标号是 i，则训练时的期望输出设为第 i 个节点为 1，而其余输出节点均为 0。在识别阶段，当一个未知类别的样本作用到输入端时，考查各输出节点的输出，并将这个样本的类别判定为与输出值最大的那个节点对应的类别。在某些情况下，如果输出最大的节点与其他节点输出的差距较小（小于某个域值），则可以作出拒绝决策。这是用多层感知器进行模式识别的最基本方式。

神经网络PR的典型做法

- 对网络进行训练学习
- 单输出型

很多实验表明，在多输出方式中，由于网络要同时适应所有类别，势必需要更多的隐层节点；而且学习过程往往收敛较慢，此时可以采用多个多输入单输出形式的网络，让每个网络只完成识别两类分类，即判断样本是否属于某个类别。这样可以克服类别之间的耦合，经常可以得到更好的结果。

具体作法是，网络的每一个输入节点对应样本一个特征，而输出层节点只有一个。为每个类建立一个这样的网络（网络的隐层节点数可以不同）。对每一类进行分别训练，将属于这一类的样本的期望输出设为 1，而把属于其他类的样本的期望输出设为 0。在识别阶段，将未知类别的样本输入到每一个网络，如果某个网络的输出接近 1（或大于某个域值，比如 0.5），则判断该样本属于这一类；而如果有多个网络的输出均大于域值，则或者将类别判断为具有最大输出的那一类，或者作出拒绝；当所有网络的输出均小于域值时也可采取类似的决策方法。

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

支持向量机

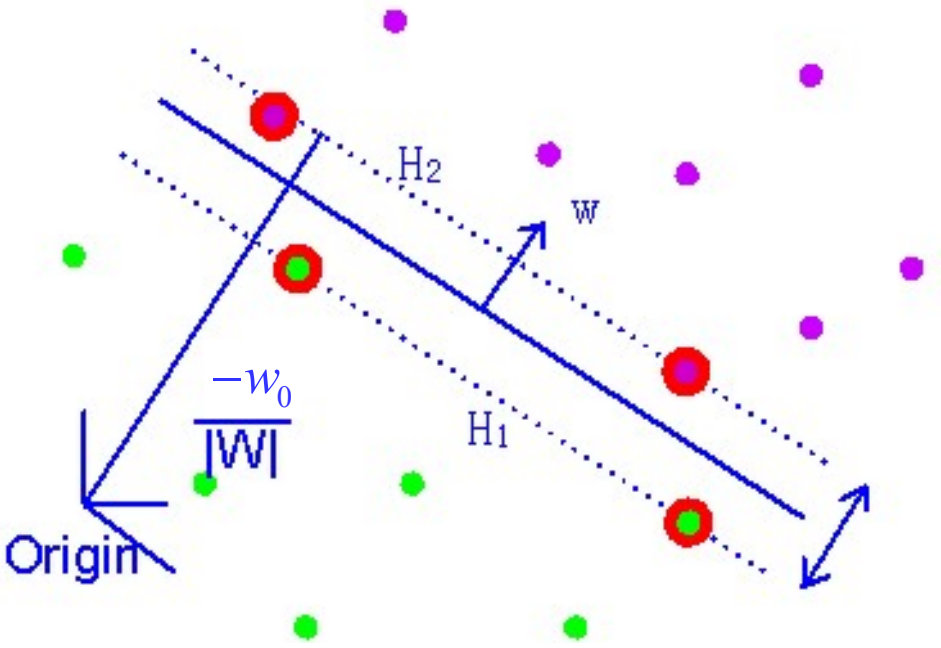
支持向量机(**Surpport Vector Machines**)简称**SVM**，是统计学习理论中最年轻的内容，也是最实用的部分。其核心内容是在**1995** 年左右,由**Vapnik**和**Chervonenkis**提出的，目前仍处在不断发展阶段。

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

支持向量机

线性可分条件下的支持向量机最优分界面



- **SVM的思路：**
  - ✓ 由于两类别训练样本线性可分，因此在两个类别的样本集之间存在一个隔离带。
  - ✓ 支持向量机的最佳准则：最大间隔准则

支持向量机

- 样本集的表示：$\{x_i, y_i\}, \quad y_i \in \{-1, +1\}$

分界面H：$W^T x_i + w_0 = 0$

令：
$$\begin{cases} \mathrm{W}^T x_i + w_0 \geq +1 & \text{for } y_i = +1 \\ W^T x_i + w_0 < -1 & \text{for } y_i = -1 \end{cases}$$

$$\forall i, y_i \left(\mathrm{W}^T x_i + w_0 \right) \geq +1$$

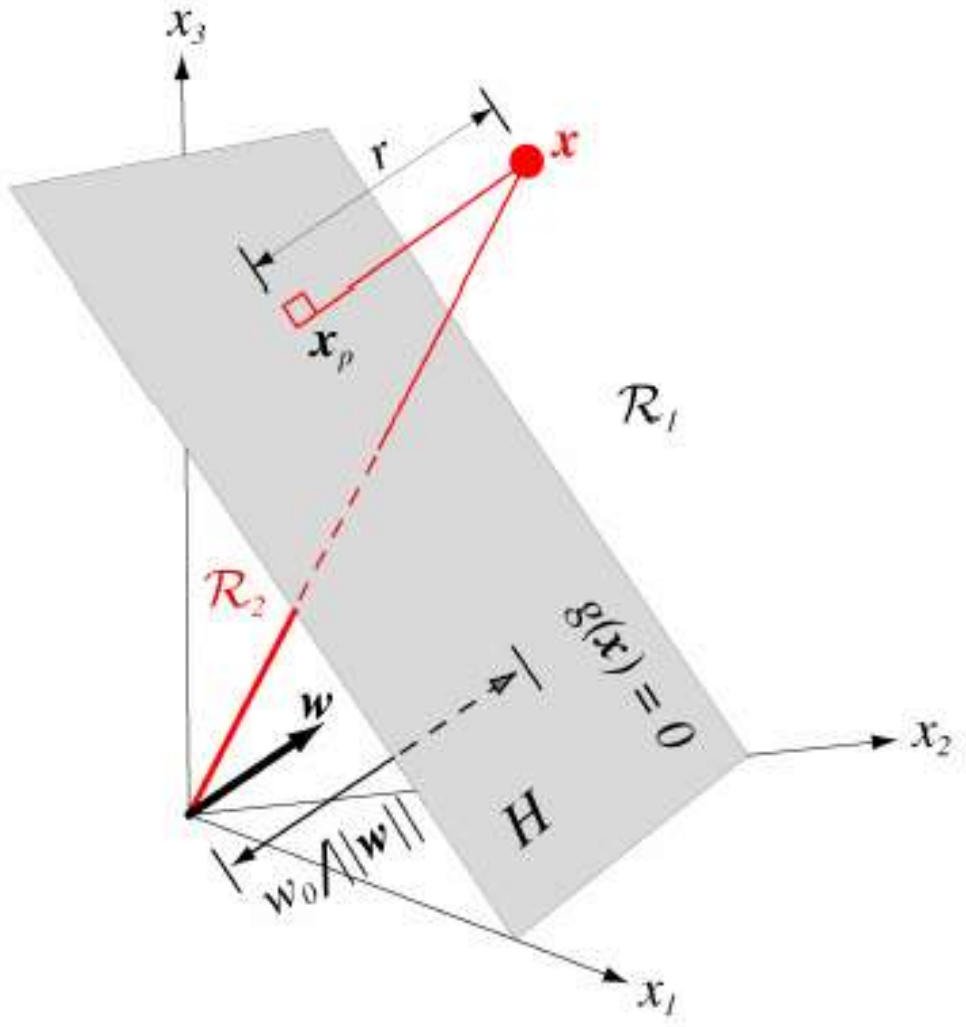
分类面

点x_0到平面

$$\langle W, x\rangle + w_0 = 0$$

的距离为

$$d = \frac{|\langle W, x_0\rangle + w_0|}{\|W\|}$$



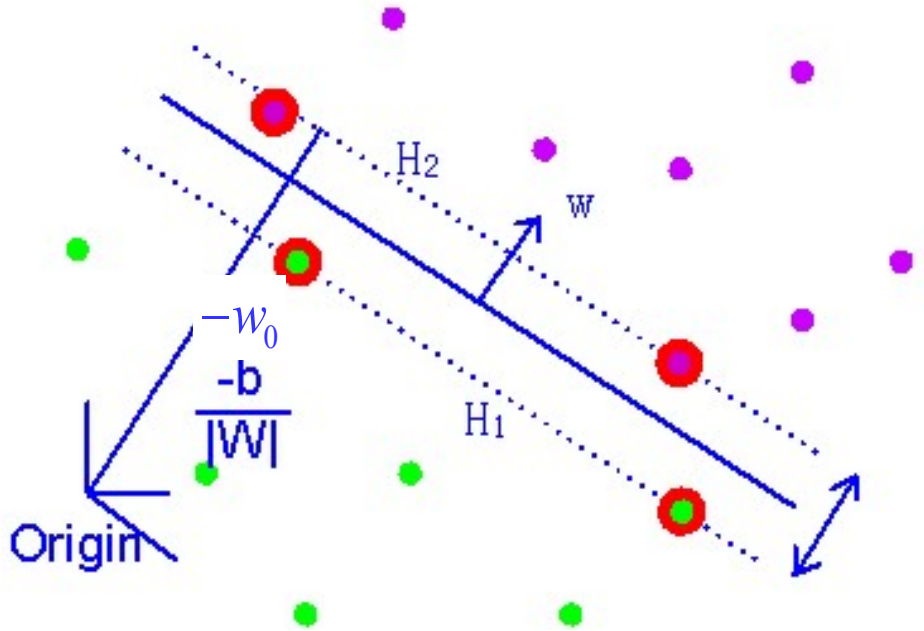
最大间隔(margin)

分类面方程为

$$W^T x + w_0 = 0$$

支撑面之间的距离叫做分类间隔

$$\gamma = \frac{2}{\|W\|}$$



线性可分的最优分类模型

$$\min \frac{1}{2} W^T W$$

$$\text{s.t. } y_i(W^T x_i + w_0) \geq 1, i = 1, 2, \cdots, N$$

作广义Lagrange乘子函数

$$L = \frac{W^T W}{2} - \sum_{i=1}^{N} \alpha_i (y_i (W^T x_i + w_0) - 1)$$

由KKT条件, 有

$$\frac{\partial L}{\partial W} = W - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad W = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$y_i(w^T x_i + w_0) - 1 \geq 0, \forall i$$

$$\alpha_i(y_i(W^T x_i + w_0) - 1) = 0, \forall i$$

$$\alpha_i \geq 0$$

得到最佳的α_i：α_i^*

$$W^* = \sum_i \alpha_i^* y_i x_i$$

将 $W=\sum_{i=1}^{N}\alpha_i y_i x_i$ 代入目标函数，由对偶理论知，系数可由如下二次规划问题解得

$$L_D=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}y_i y_j\alpha_i\alpha_j\left\langle x_i,x_j\right\rangle$$

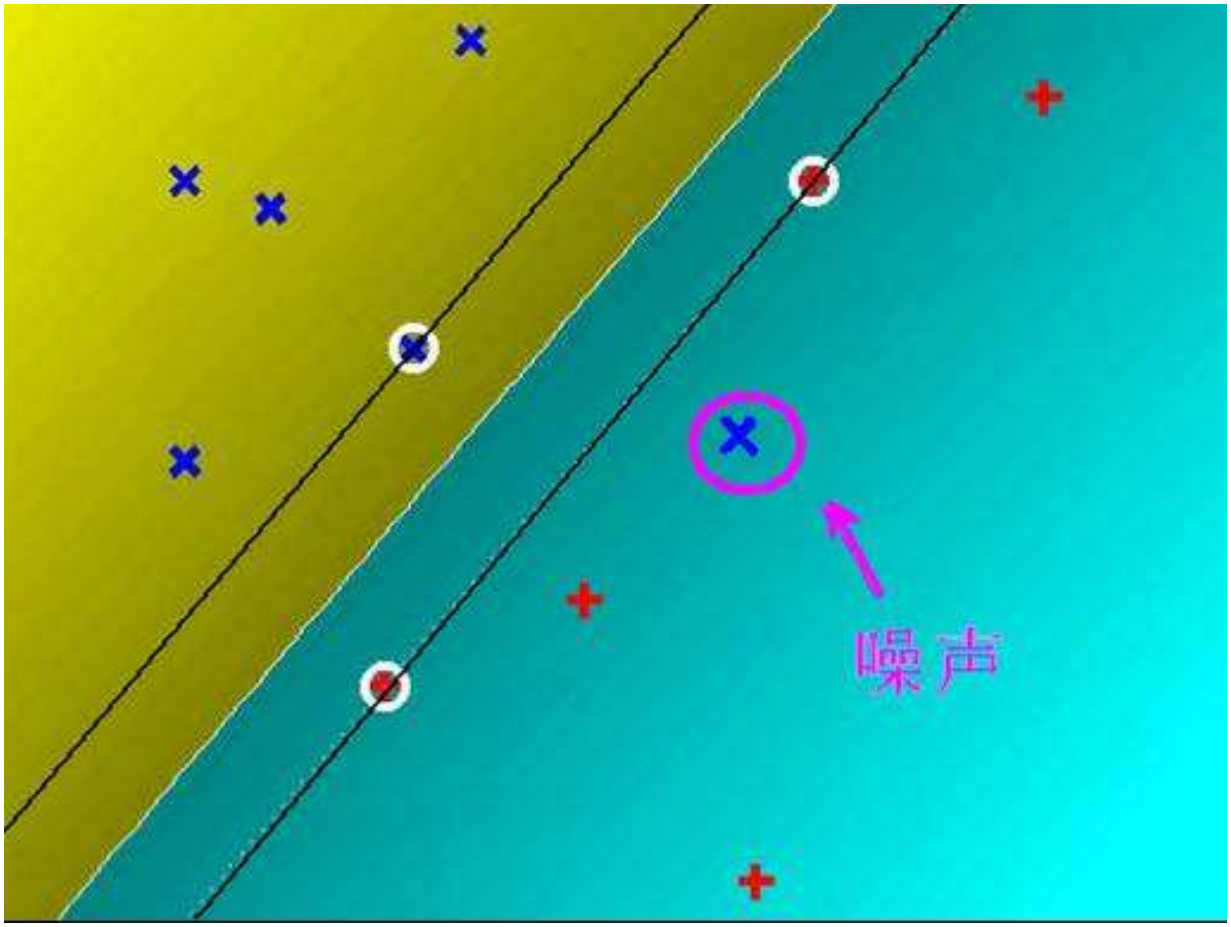
$$\max\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}y_i y_j\alpha_i\alpha_j\left\langle x_i,x_j\right\rangle$$

$$\text{s.t.}\sum_{i=1}^{N}\alpha_i y_i=0$$

Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

线性不可分条件下的广义最优分界面



线性不可分条件下的广义最优分界面

样本集不是线性可分，就是说对样本集

$$(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N),\quad x_i\in R^d,y_i\in\{+1,-1\} \tag{4-89}$$

不等式

$$y_i[(w\cdot x_i)+b]-1\geqslant 0,\quad i=1,2,\cdots,N \tag{4-90}$$

不可能被所有样本同时满足。

假定某个样本 x_k 不满足式(4-90)的条件，即 $y_k[(w\cdot x_k)+b]-1<0$，那么总可以在不等式的左侧加上一个正数 ξ_k，使得新的不等式 $y_k[(w\cdot x_k)+b]-1+\xi_k\geqslant 0$ 成立。

从这个思路出发，对每一个样本引入一个非负的松弛变量 ξ_i，$i=1,\cdots,N$，就可以把式(4-90)的不等式约束条件变为

$$y_i[(w\cdot x_i)+b]-1+\xi_i\geqslant 0,\quad i=1,2,\cdots,N$$

如果样本 x_j 被正确分类，即 $y_j[(w\cdot x_j)+b]-1\geqslant 0$，则 $\xi_j=0$；而如果有一个错分样本，则这个样本对应的 $y_j[(w\cdot x_j)+b]-1<0$，对应的松弛变量 $\xi_j>0$。

线性不可分条件下的广义最优分界面

所有样本的松弛因子之和 $\sum_{i=1}^{N}\xi_i$ 可以反映在整个训练样本集上的错分程度，错分样本数越多，则 $\sum_{i=1}^{N}\xi_i$ 越大；同时，如果样本错误的程度越大（即在错误的方向上远离分类面），则 $\sum_{i=1}^{N}\xi_i$ 也越大。显然，我们希望 $\sum_{i=1}^{N}\xi_i$ 尽可能小。因此，可以在线性可分情况下的目标函数 $\frac{1}{2}\|w\|^2$ 上增加对错误的惩罚项，定义下面的广义最优分类面的目标函数

$$\min_{w,b}\frac{1}{2}(w\cdot w)+C\left(\sum_{i=1}^{N}\xi_i\right)$$

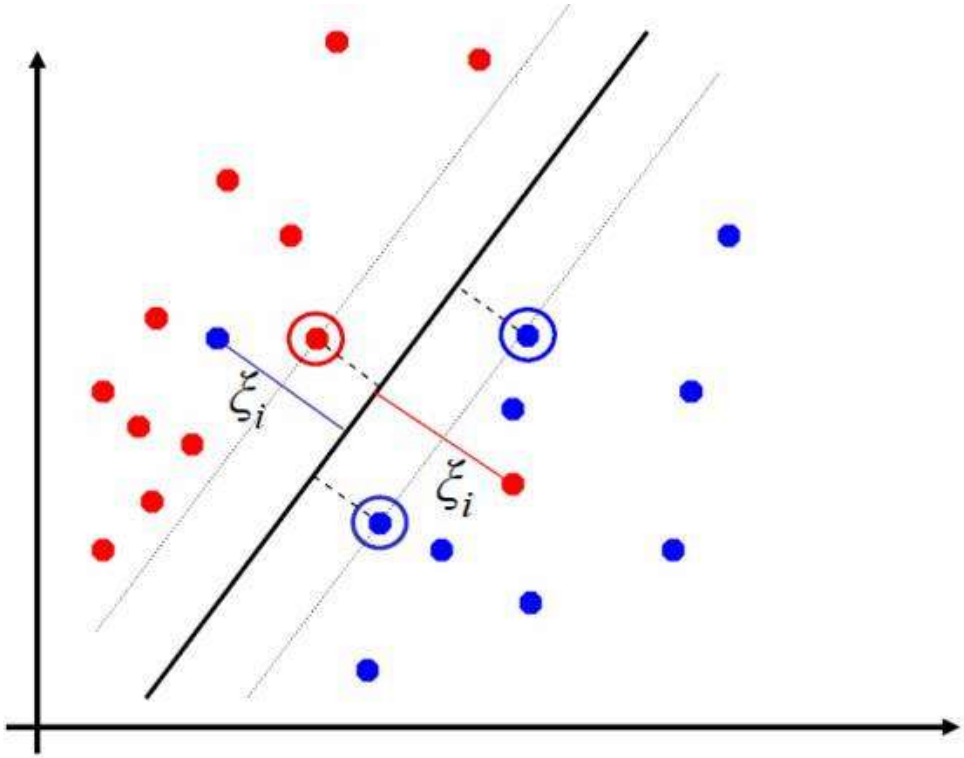
线性不可分的情况

$$y_i(W^T x_i + w_0) \geq 1$$

引入松弛变量

$$\{\xi_i\}_{i=1}^{N}$$

$$\begin{cases} W^T x_i + w_0 \geq 1 - \xi_i & y_i = +1 \\ W^T x_i + w_0 \leq -1 + \xi_i & y_i = -1 \end{cases}$$



不可分的解方程

$$\min \quad f(W)=\frac{1}{2}\|W\|^2+C\left(\sum_{i=1}^{N}\xi_i\right)$$

subject to

$$y_i\left(W^T x_i+w_0\right)\geq +1-\xi_i$$

$$\xi_i\geq 0$$

作Lagrange函数

$$L=\frac{1}{2}\|W\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{n}\alpha_i(y_i(W^T x_i+w_0)-1+\xi_i)-\sum_{i=1}^{N}\mu_i\xi_i$$

最优性条件

由KKT条件

$$\frac{\partial L}{\partial W} = W - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad W = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \Rightarrow \quad 0 \le \alpha_i \le C$$

$$\alpha_i \ge 0, \mu_i \ge 0$$

$$y_i(W^T x_i + w_0) - 1 + \xi_i \ge 0 \qquad \text{若 } y_i(w^T x_i + b) + \xi_i > 1 \Rightarrow \alpha_i = 0;$$

$$\alpha_i(y_i(W^T x_i + w_0) - 1 + \xi_i) = 0 \qquad \text{若 } \xi_i > 0 \Rightarrow \alpha_i = C.$$

$$\mu_i \xi_i = 0$$

系数的解方程

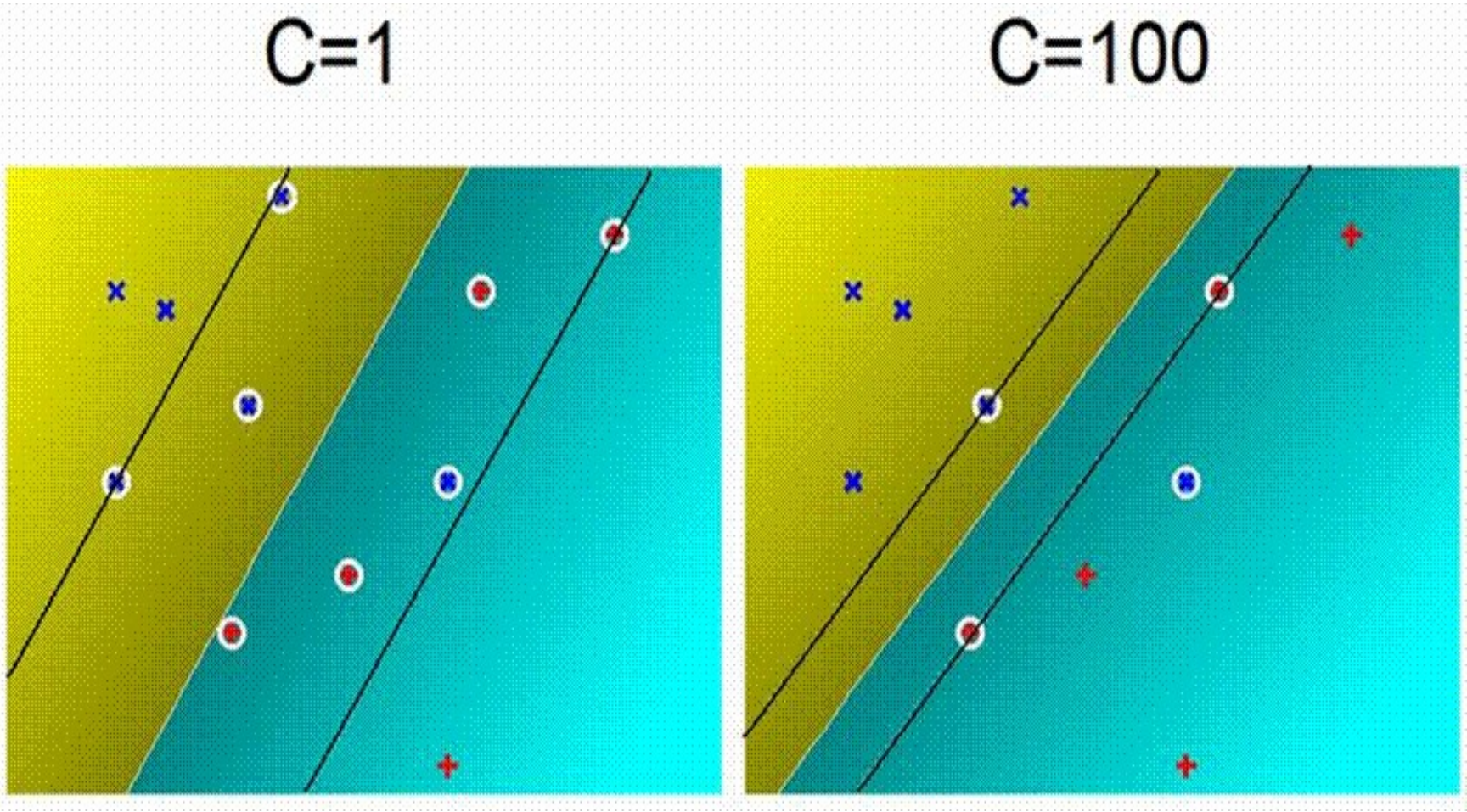
$$\max Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left\langle x_i , x_j \right\rangle$$

subject to

$$(1) \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$(2) 0 \le \alpha_i \le C \qquad \text{for } i = 1, 2, ..., N$$

c不同带来的影响

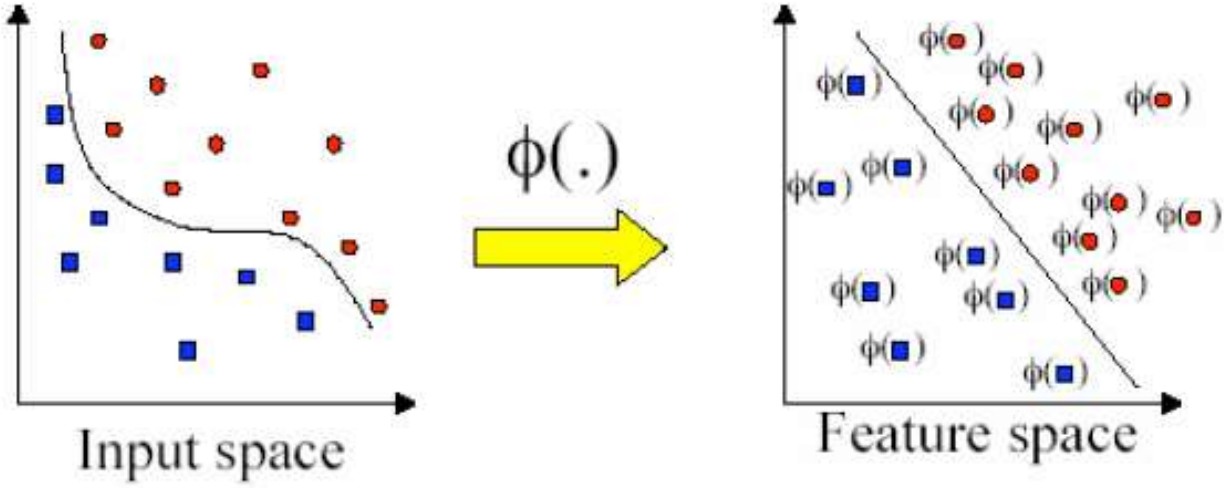


Outline:

- 分段线性判别函数
- 多层感知器神经网络
  - ✓ 神经元与感知器
  - ✓ 前馈神经网络
  - ✓ 利用BP算法进行网络训练
  - ✓ 径向基函数网络
  - ✓ Hopfield网络
- SVM
  - ✓ 线性可分条件下的SVM最优分界面
  - ✓ 线性不可分条件下的广义最优分界面
  - ✓ 特征映射法，解决非线性判别分类问题

特征映射法,解决非线性判别分类问题

- 基本思想：
  - ✓选择非线性映射$\Phi(X)$将x映射到高维特征空间Z，在Z中构造最优超平面



➢SVM中两个式子：

$$W^* = \sum_i a_i^* y_i x_i$$

a_i^*是下式求极大值的解

$$L_D = \sum_i a_i - \frac{1}{2}\sum a_i a_j y_i y_j \langle x_i \cdot x_j \rangle$$

$$W^T X = \left\langle \left(\sum_i a_i^* y_i x_i \right) \cdot X \right\rangle = \sum_i a_i^* y_i \langle x_i \cdot X \rangle$$

$$W^T X = \left\langle \left(\sum_i a_i^* y_i x_i \right) \cdot X \right\rangle = \sum_i a_i^* y_i \left\langle x_i \cdot X \right\rangle$$

- 将原特征向量用映射的方式转换成

$$x_i \to f(x_i)$$

则：

$$L_D = \sum_i a_i - \frac{1}{2} \sum a_i a_j y_i y_j \left\langle f(x_i) \cdot f(x_j) \right\rangle$$

分类界面方程：

$$\sum_i^n a_i^* y_i \left\langle f(x_i) \cdot f(x) \right\rangle + w_0^* = 0$$

新空间的内积也是原特征的函数，可记作：

$$K(x_i, x_j) \overset{def}{=} \left\langle f(x_i) \cdot f(x_j) \right\rangle$$

特征映射法,解决非线性判别分类问题

- 核函数：$K(x_i \cdot x)$
  - ✓如果能确定某种函数$K(x_i \cdot x)$的确是x_i和x这两个样本数据某种映射的内积，就可用它来设计支持向量机，而不必知道对应哪一个函数 $f(*)$

$$L_D = \sum_i a_i - \frac{1}{2}\sum a_i a_j y_i y_j \left\langle f(x_i) \cdot f(x_j) \right\rangle$$

$$L_D = \sum_i a_i - \frac{1}{2}\sum a_i a_j y_i y_j K(x_i \cdot x_j)$$

分界面方程：$\sum_i^n a_i^* y_i K(x_i \cdot x) + w_0^* = 0$

分界面方程：$$\sum_{i}^{n} a_i^* y_i K(x_i \cdot x) + w_0^* = 0$$

- ✓这样一来，如果选择了一种函数 $K(a,b)$，其中**a**和**b**是原特征空间的两个数据点，那么只要这种函数是反映了特征映射后数据的内积，线性分类器的框架就可以用了。
- ✓选择合适的核函数 $K(a,b)$是设计中的重要问题之一。
- ✓常用的核函数：

$$K(x,x_i) = \left[\langle x \cdot x_i \rangle + 1\right]^q$$

$$K(x,x_i) = \exp\left(-\frac{|x-x_i|^2}{\sigma^2}\right)$$

$$K(x,x_i) = \tanh\left(v\langle x_i \cdot x \rangle + c\right)$$

- 支持向量机的基本思想可概括为：首先通过非线性变换将输入空间变换到一个高维空间，然后再这个新空间中求取最优线性分类面，而这种非线性变换是通过定义适当的内积函数来实现的。

- 核方法是支持向量机成功的关键技术，主要思想是用核函数替代内积，相当于用一个核诱导的非线性映射将原始数据映射到高维空间中进行线性操作。高维空间中的内积可通过核函数在低维空间中直接计算，从而计算量并没有随维数增加而增加很多。