模式识别
（Pattern Recognition）

广东工业大学集成电路学院　邢延

第五讲 神经网络分类器

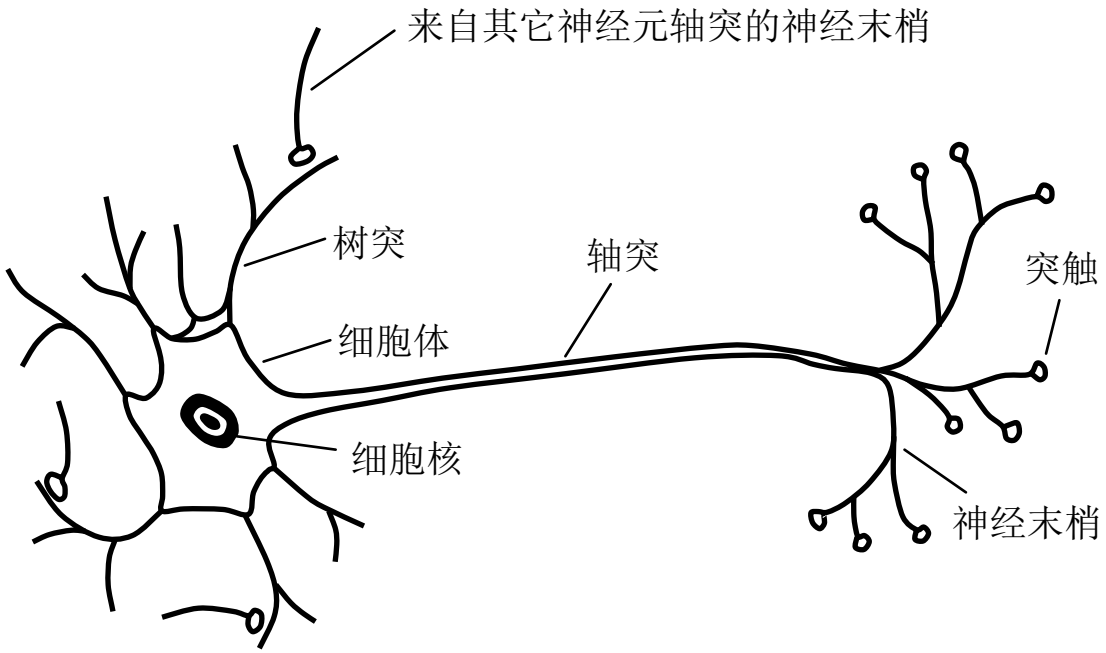
（05 Classifiers of Artificial Neural Networks）

内容提要

- 神经网络的基本概念
- 感知器
- BP神经网络分类器

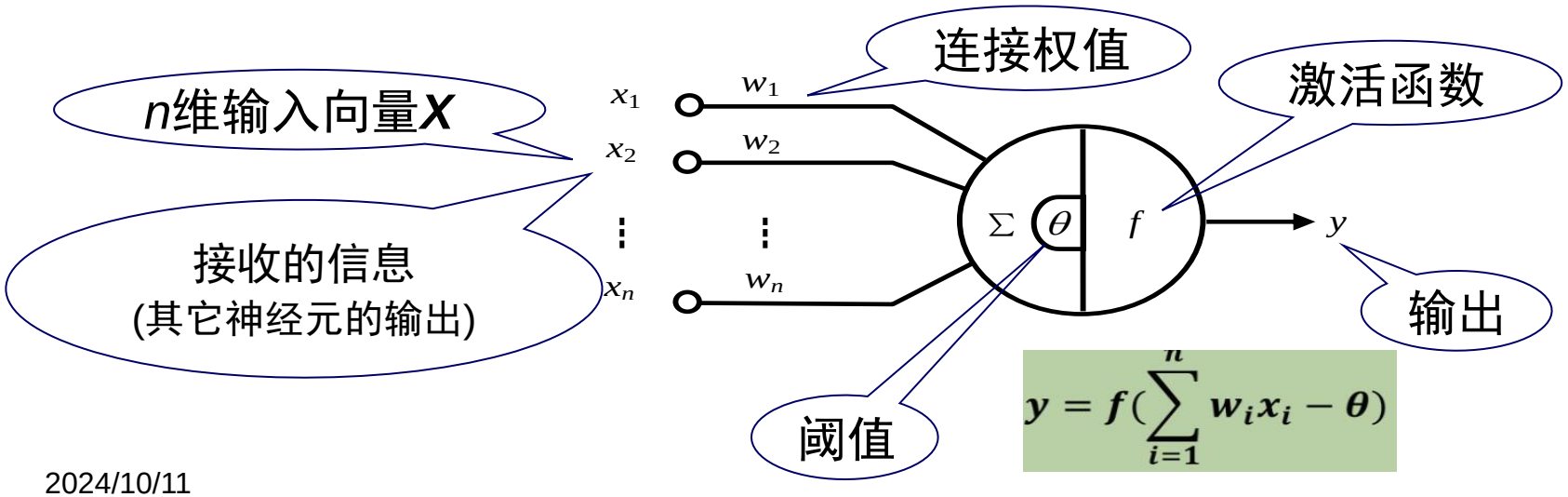
神经网络基本概念

- **生物神经元**
  - **结构**
    - **细胞体、树突、轴突和突触**
  - **兴奋和抑制两种状态**



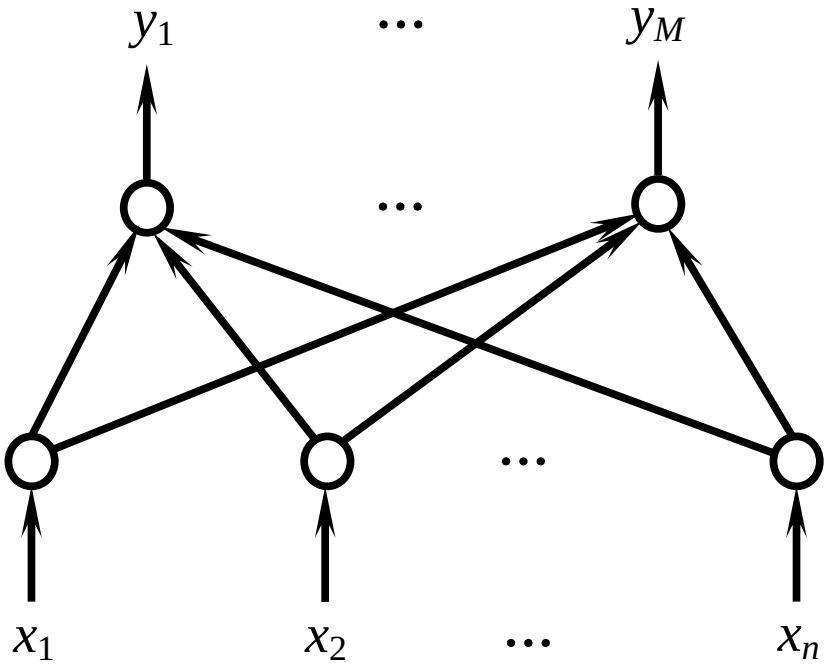
神经网络基本概念

- **人工神经元及神经网络**
  - **人工神经元**
    - **生物神经元的简化模拟**
  - **人工神经元间的互连**
    - **信息传递路径轴突-突触-树突的简化**
  - **连接的权值**
    - **两个互连的神经元之间相互作用的强弱**



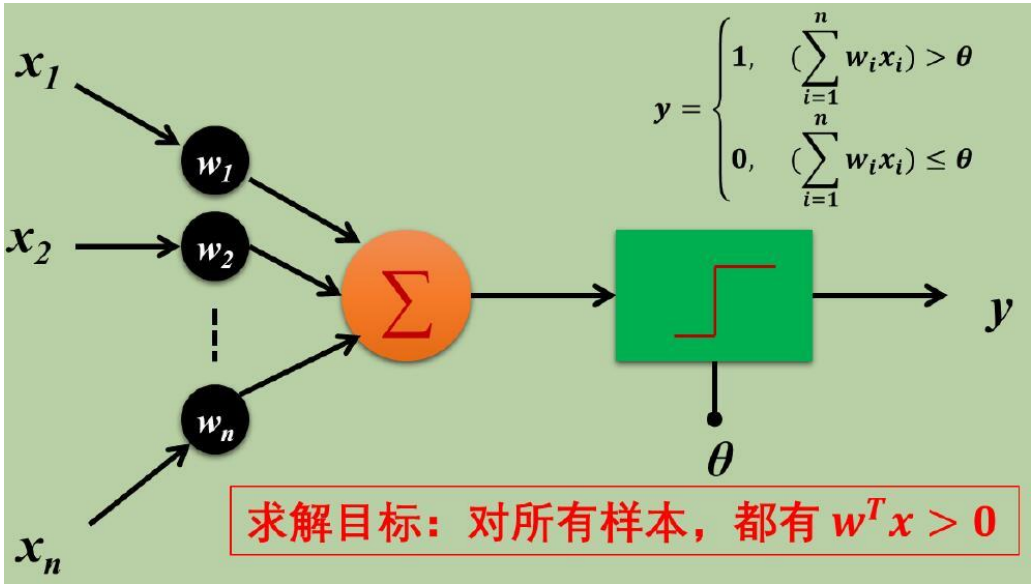
感知器（Perceptron）

- 起源
  - F．Rosenblatt于1957年提出
- 结构特点
  - 双层（输入层、输出层）
  - 两层单元之间为全互连；
  - 连接权值可调
  - 前馈网络
  - 输入神经元个数 = 特征数
  - 输出层神经元个数 = 类别数
    - ◆ 两类分类问题时输出层为一个神经元



感知器（Perceptron）

- **工作原理（以二类分类问题为例）**
  - **分类目标**
    - **能正确分类所有的样本**
  - **分类准则函数:**
    - $J = (-1) \times$ (所有错分样本的判别函数之和)
  - **如何求解**
    - **梯度下降法**



感知器（Perceptron）

- **工作原理**
  - **梯度下降法**
    - **从一个任意的初始权向量w_0出发，沿准则函数值下降最快的方向（即负梯度方向对权向量进行一步步修正，直到获得全局最优解为止。**
    - **即第$k+1$步是在第k步获得的权向量基础上进行递推得到的权向量,其中ρ是每一次递推的调整步长（即学习率）。**
    - **权向量递推公式：X_0为错分的样本集**

$$w(k+1) = w(k) + \rho(k+1)\sum_{x \in X_0} x$$

或者

$$w(k+1) = w(k) + \rho(k+1)x, x \in X_0$$

感知器（Perceptron）

- 算法1
  - 1）设定初始权向量：$w(0), k = 0$；
  - 2）对训练集的所有规范化增广特征向量进行分类，将分类错误的数据（即不满足$G_{ij}(x) = w^T x > 0$的数据）放入集合X_0中；
  - 3）修正权向量： $w(k+1) = w(k) + \rho(k+1) \sum_{x \in X_0} x$
  - 4）$k = k + 1$, 返回步骤2），直至所有数据都能被正确分类为止。

感知器（Perceptron）

- **算法1说明**
  - **规范化增广特征向量**

感知器（Perceptron）

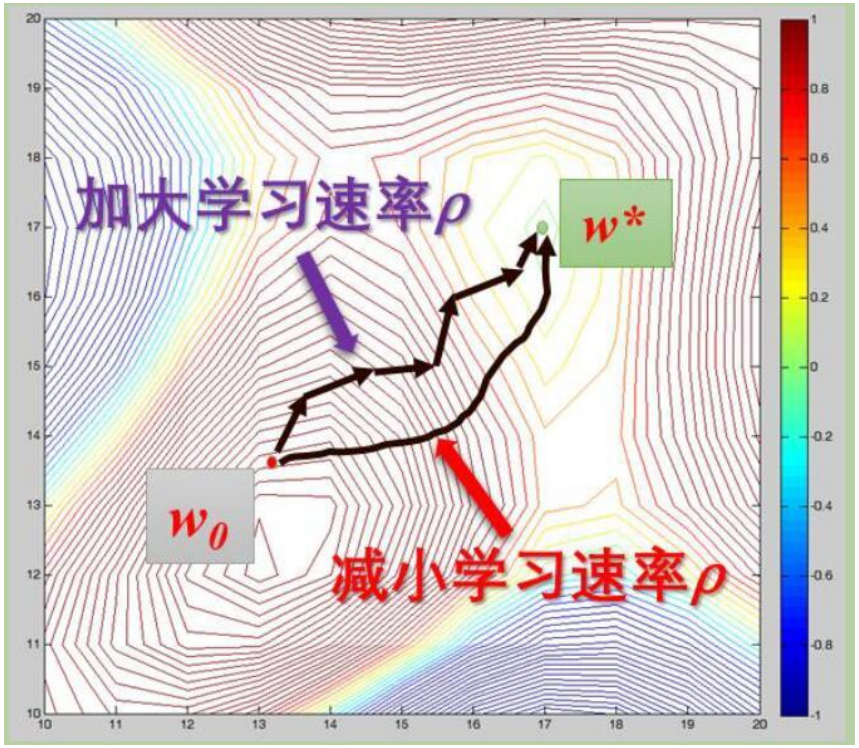
- **算法2**
  - **1）设定初始权向量：$w(0), k = 0$；**
  - **2）从训练数据中顺序抽取一个样本，将其规范化增广特征向量x进行分类；**
  - **3）若分类正确（即$G_{ij}(x) = w^T x > 0$），返回到步骤2），抽取下一个样本；**
  - **4）若分类错误（即不满足$G_{ij}(x) = w^T x > 0$），修正权向量：**

$$w(k+1) = w(k) + \rho(k+1)x$$

  - **5）返回步骤2），抽取下一个样本，直至所有数据都能被正确分类为止。**

感知器（Perceptron）

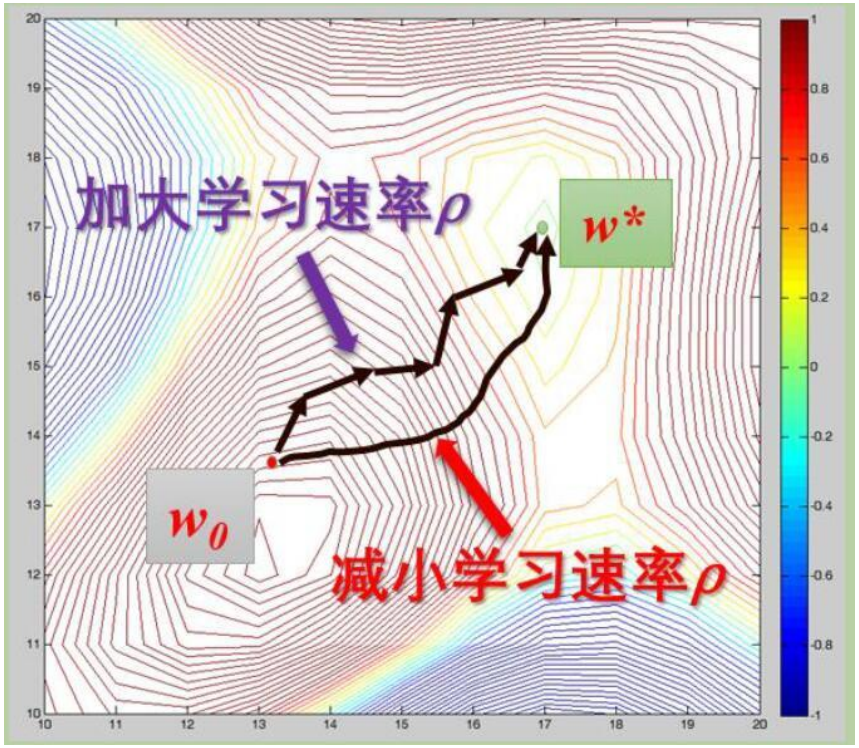
- **算法的学习速率讨论**



- ρ的取值越大，求解的速度越快，但求解路径越不光滑，求解的精度越差，容易过冲甚至振荡；
- ρ的取值越小，求解的速度越慢，但求解精度越高。
- 使用梯度下降法来训练感知器，其求解精度和速度之间是存在矛盾的。

感知器（Perceptron）

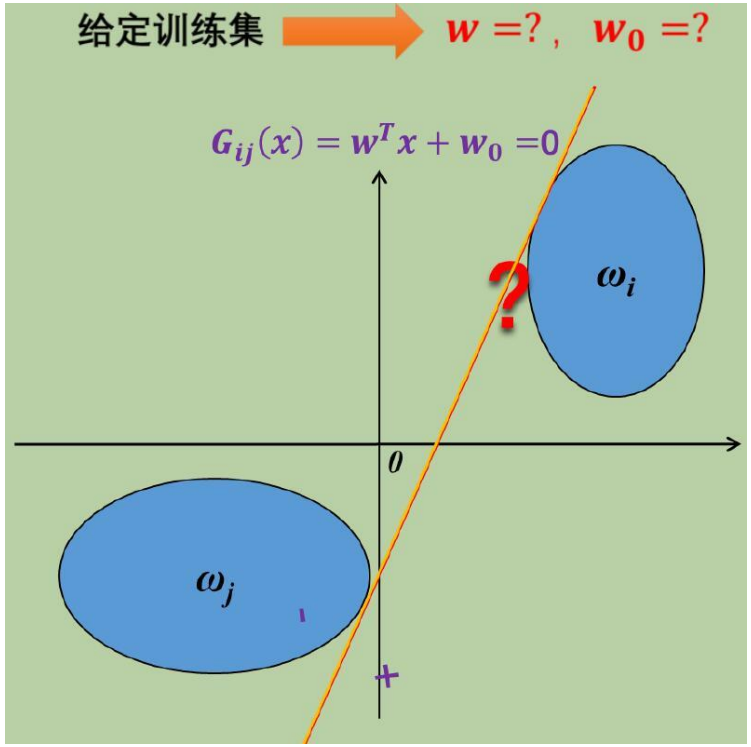
- **学习速率的设定**



- **学习率ρ的设定方法：**
- 固定值
- 变步长法
  - 按照某种规律逐步减少步长
  - 如：$\rho(k+1)=\gamma\frac{1}{k},\gamma>0$

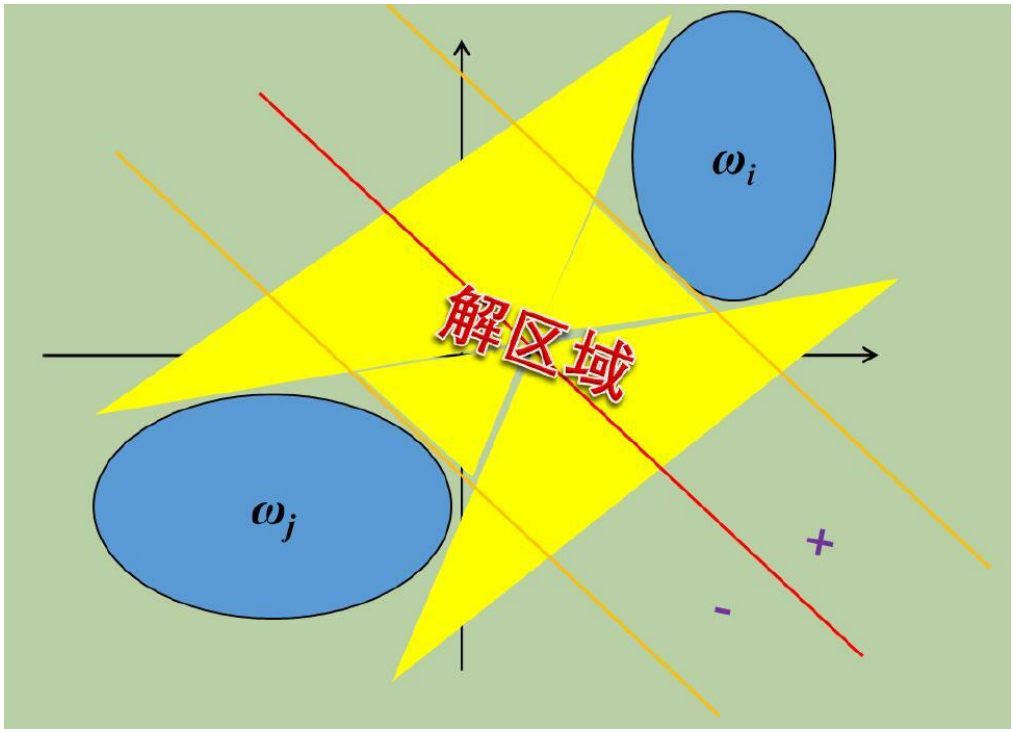
感知器（Perceptron）

- 算法的深入分析
  - 是线性分类器



感知器（Perceptron）

- 算法的深入分析
  - 存在多个最优解



感知器（Perceptron）

- 算法的深入分析
  - 首个采用误差反馈学习规则的算法
    - ◆误差反馈学习规则

$$w(k+1)=w(k)+\rho(k+1)r(y(x)-\hat{y}(x))x$$

其中，$y(x)$是样本x对应的真实输出，$\hat{y}(x)$是当前模型的输出，r是输出误差的一个函数（即学习信号），代表误差如何反馈影响权向量的调整。

- ◆感知器权向量递推公式

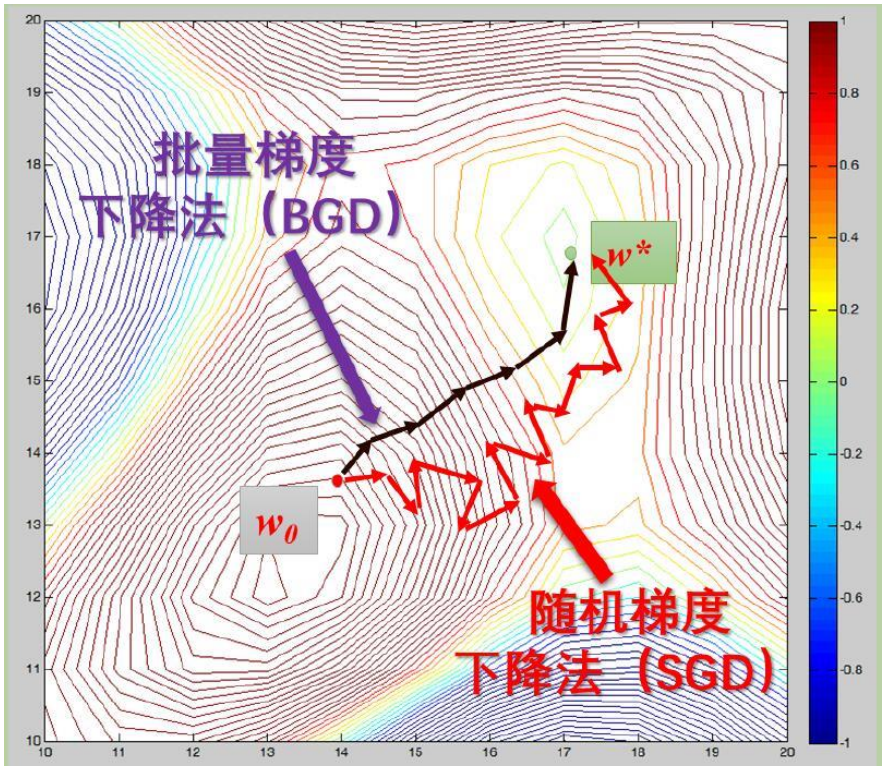
$$w(k+1)=w(k)+\rho(k+1)\sum_{x\in X_0}x$$

或者

$$w(k+1)=w(k)+\rho(k+1)x, x\in X_0$$

感知器（Perceptron）

- 算法的深入分析
  - 梯度下降法1 – 批量梯度法（BGD）
  - 梯度下降法2 – 随机梯度法（SGD）



$$w(k+1) = w(k) + \rho(k+1)\sum_{x \in X_0} x$$

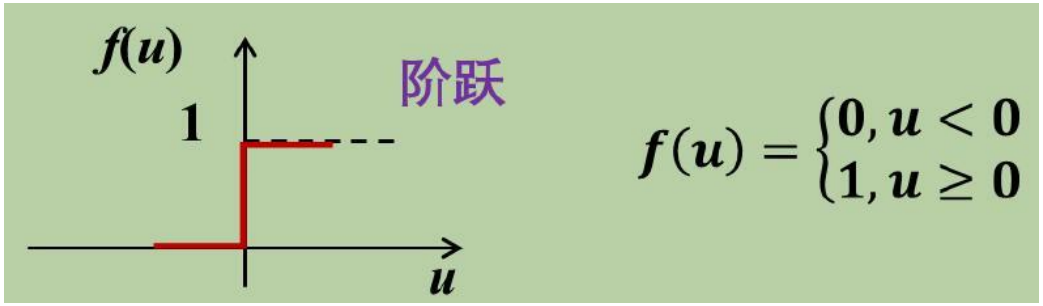
或者

$$w(k+1) = w(k) + \rho(k+1)x, x \in X_0$$

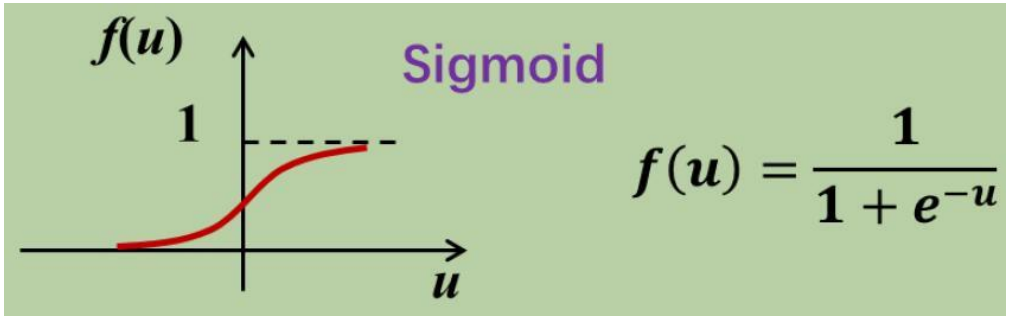
提问：与感知器的权向量递推公式比较哪个是BGD？哪个是SGD？

感知器（Perceptron）

- 算法的深入分析
  - 激活函数
    - ◆阶跃函数 -- 感知器常用

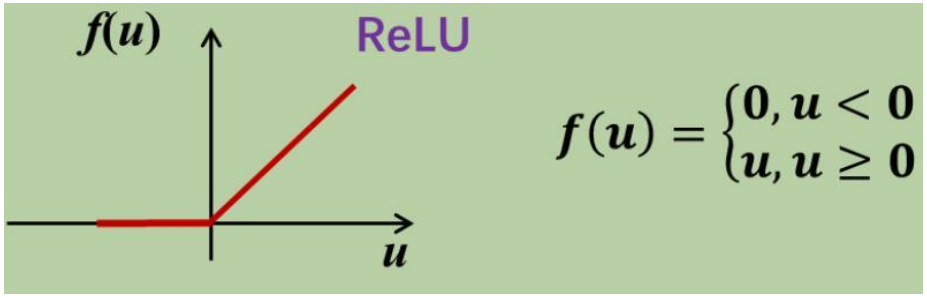


    - ◆Sigmoid函数 – BP网络常用

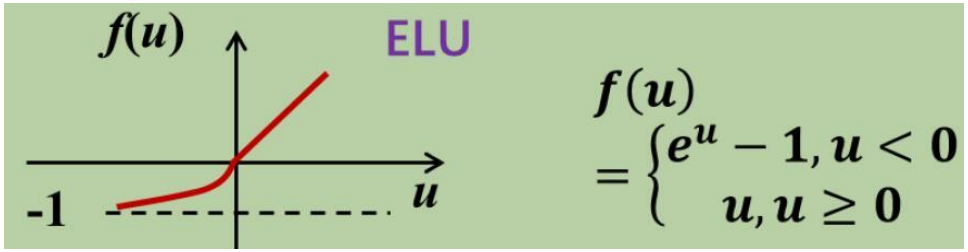


感知器（Perceptron）

- 算法的深入分析
  - 激活函数
    - ◆ ReLU函数– 深度神经网络常用

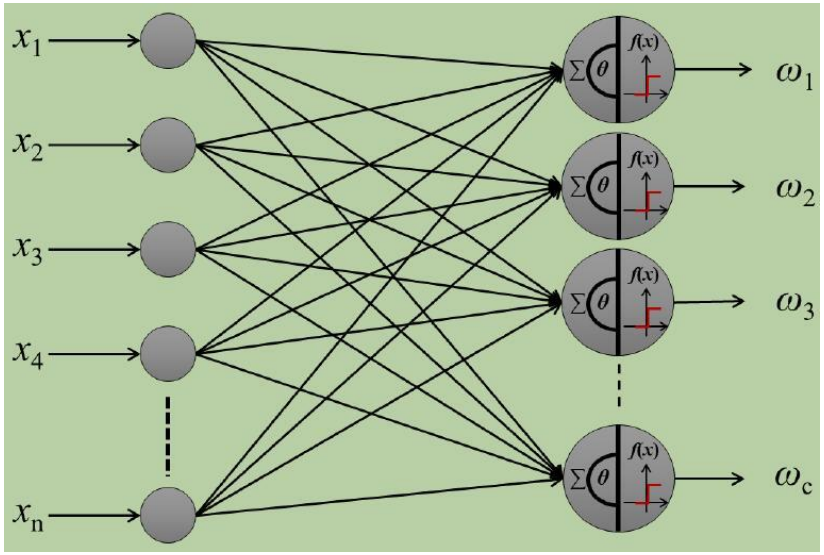


    - ◆ ELU函数 –深度神经网络常用



感知器（Perceptron）

- 算法的深入分析
  - 感知器用于多类分类问题

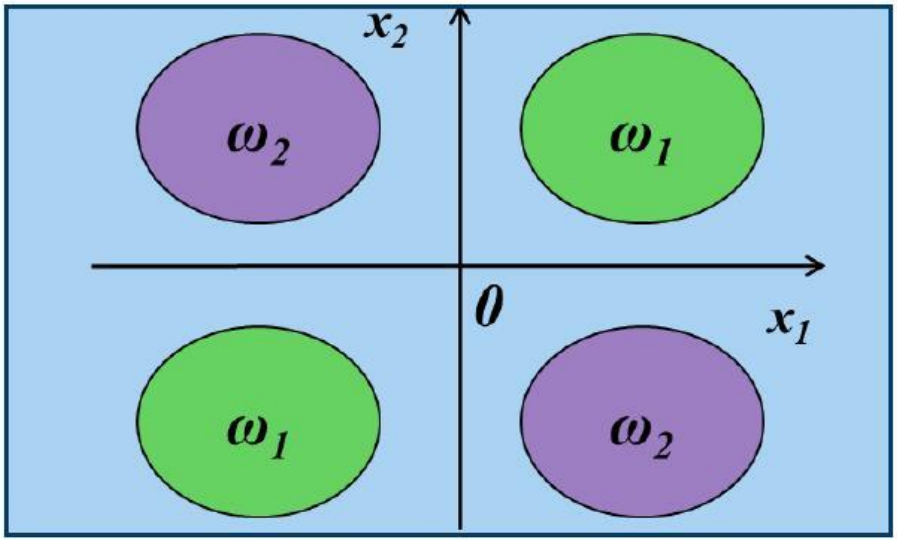


感知器（Perceptron）

- 算法的深入分析
  - 缺陷
    - 无法解决线性不可分问题！

BP神经网络分类器

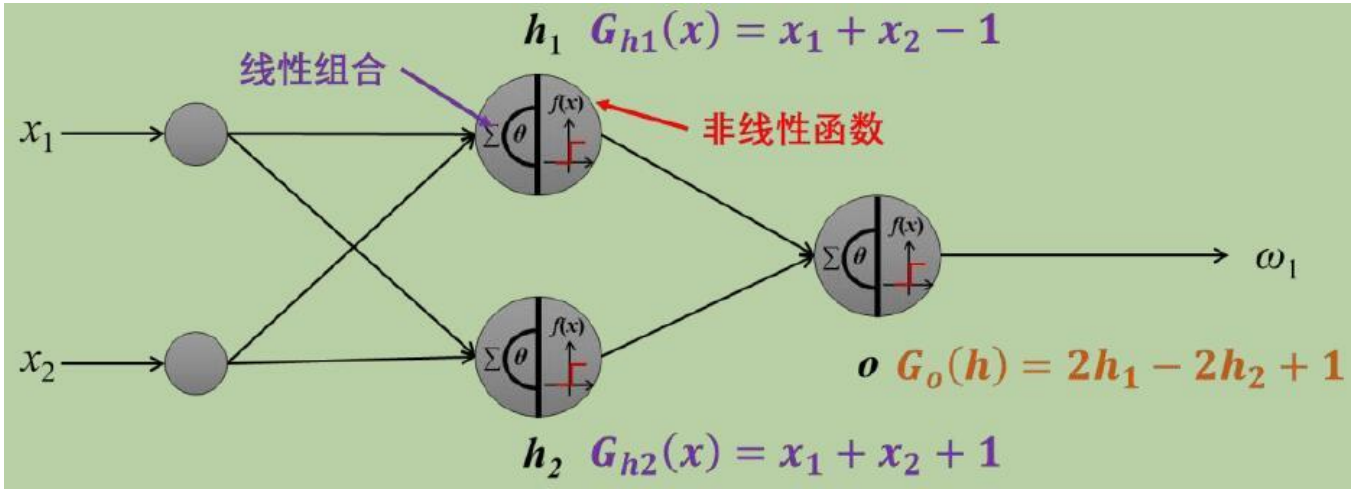
- 非线性的异或问题 vs 多层感知器

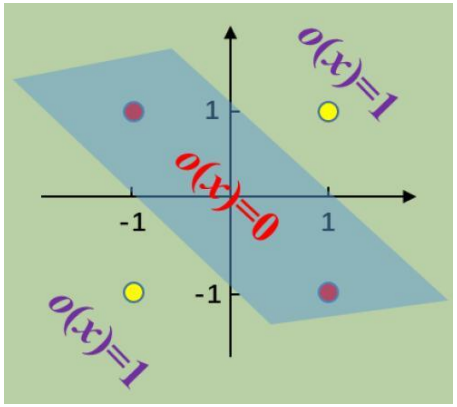


$$G(x) = x_1 x_2$$ ← 非线性判别函数

BP神经网络分类器

- 非线性的异或问题 vs 多层感知器

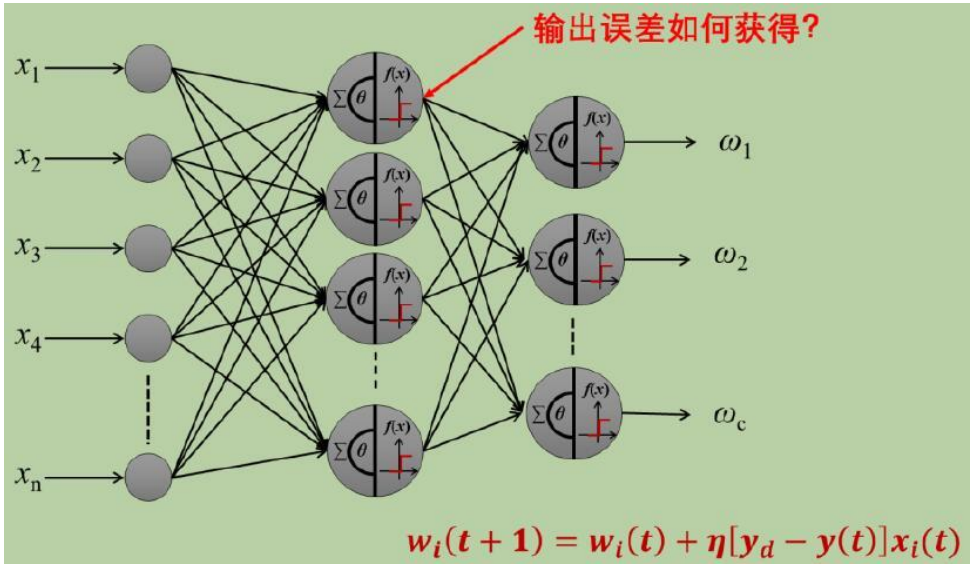




激活函数的非线性，使得具有多个层次的神经网络的输入输出之间可以形成复杂的非线性映射关系

BP神经网络分类器

- 多层感知器的难题
  - 如何训练
    - 怎么获得用于隐层神经元训练的输入和输出之间的

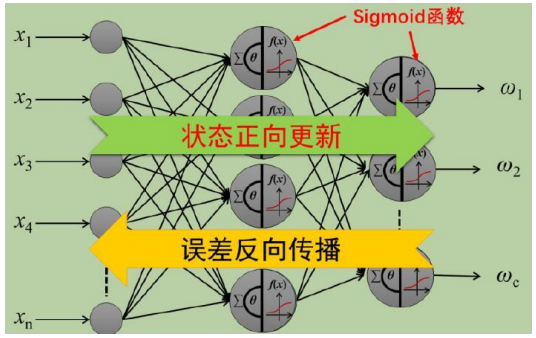


BP神经网络分类器

- 反向传播（Back Propagation,BP）网络
  - 一种有效的多层神经网络训练方案
  - 典型的前馈型网络
  - 相邻两层神经元间是全连接的
  - 输入层的神经元个数为样本特征向量的维数
  - 输出神经元的个数为类别数
  - 激活函数采用Sigmoid函数

BP神经网络分类器

- 训练过程
  - 1）状态正向更新
    - 输入样本 - >输入层 - >隐含层（处理） - >输出层
    - 注1：若输出层实际输出与期望输出（卷标）不符，则转入2）（误差反向传播过程）
  - 2）误差反向传播
    - 输出误差（某种形式） - >隐含层） - >输入层
    - 其主要目的是通过将输出误差反传，将误差分摊给各层所有单元，从而获得各层单元的误差信号，进而修正各单元的权值（是一个权值调整的过程）。



BP神经网络分类器

- **误差反向传播机理**
  - **设定输出误差的损失函数（单调递增）**
    - **例如：均方误差函数** $L = \frac{1}{2}(y_d^{(o)} - y^{(o)})^2$
  - **计算损失函数对各层权向量的梯度：**

$$\frac{\partial L}{\partial w^{(o)}} = \frac{\partial L}{\partial y^{(o)}} \cdot \frac{\partial y^{(o)}}{\partial w^{(o)}} = \frac{\partial L}{\partial y^{(o)}} \cdot f'(\cdot) y^{(h)}$$

$$\frac{\partial L}{\partial w^{(h)}} = \frac{\partial L}{\partial y^{(o)}} \cdot w^{(o)^T} f'(\cdot) \cdot f'(\cdot) x$$

  - **以梯度下降法去递推修正各层权向量的值：**

$$\delta^{(L-1)} = w^{(L)^T} \delta^{(L)} \cdot f'(\cdot)$$

  - **具体推导采用的是链式求导**
    - **推导过程及实例详见“BP算法推导及示例”ppt**

BP神经网络分类器

- **BP算法的深入分析**
  - **误差反向传播要求激活函数可导，用连续、光滑、有界的sigmoid 函数作为每个神经元的激活函数；**
  - **BP 网络是全连接网络，当有一定的深度时计算量非常大，运行效率不高；**
  - **隐层的层数和每一层神经元的数量可以任意设定；**
    - **如果隐层神经元数量太少，则系统输出所获得的调节量会太少，误差缩减得很慢；**
    - **如果隐层神经元数量太多，又容易产生过拟合现象。**

BP神经网络分类器

- **BP算法的深入分析**
  - **BP 算法有梯度下降法的通病；**
    - **网络初始参数难以优化设定，最优化求解时也容易陷入局部极小值，而无法求得全局最优解**
  - **BP 算法会产生“梯度消失”**
    - **激活函数sigmoid 函数导数的最大值仅为0.25，连乘后梯度消失非常快，网络的深度连5层都难以达到**
  - **BP 网络总是收敛很慢**
    - **sigmoid 函数的导数衰减很快，再加上是恒大于0 的，网络参数只能单向调节**
  - **属于浅层神经网络**

BP神经网络分类器

- **BP算法的深入分析**
  - **样本的划分**
    - **训练样本 - 权值更新**
    - **验证样本 - 更新迭代的终止依据**
    - **测试样本**

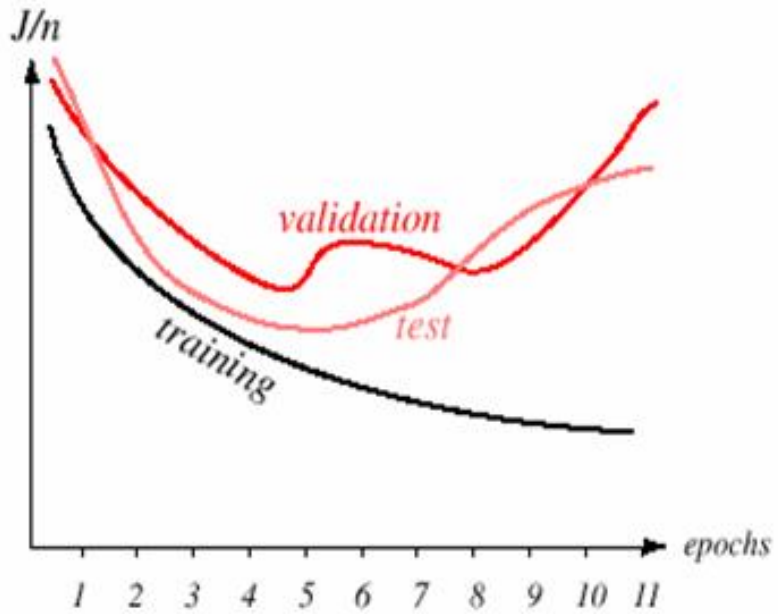


FIGURE 6.6. A learning curve shows the criterion function as a function of the amount of training, typically indicated by the number of epochs or presentations of the full training set. We plot the average error per pattern, that is, $1/n \sum_{p=1}^{n} J_p$. The validation error and the test or generalization error per pattern are virtually always higher than the training error. In some protocols, training is stopped at the first minimum of the validation set. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.