第二讲 人工神经元

本讲主要内容

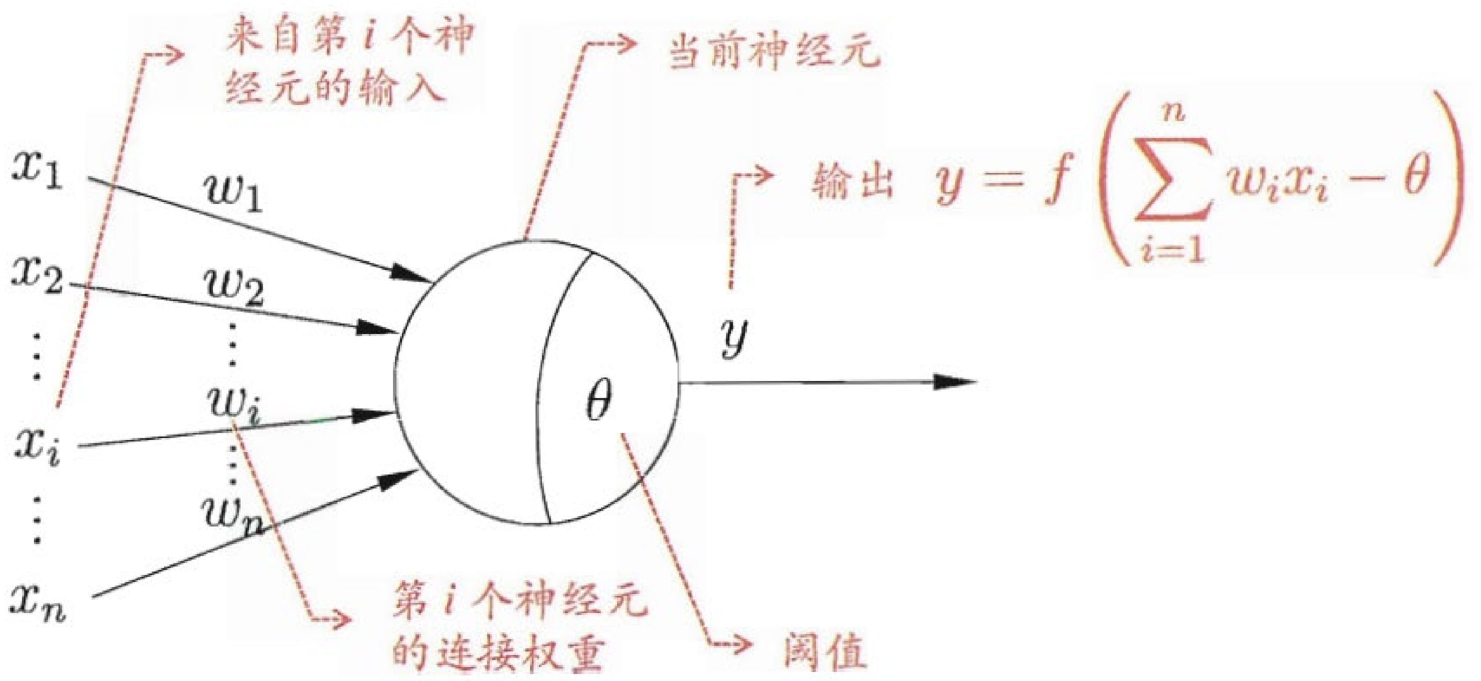
- MP神经元
- 线性模型
  - 线性回归
  - 线性划分
  - 线性模型的神经网络表示
- 感知器神经元
- FT神经元

一、MP神经元

MP模型

- 结合了神经生理学(McCulloch)和数理逻辑(Pitts)的研究描述了一个神经网络的逻辑运算
- 其神经元模型假定遵循有或无（all-or-none）规则
- 如果神经元数目足够多，适当设置突触连接，并且同步操作，原则上可以计算任何可计算函数
- 标志着神经网络和人工智能科学的诞生，神经网络计算模型的雏形

MP模型结构



- 输入： $u = w_1x_1 + w_2x_2 + \cdots + w_nx_n$
- 输出： $y = \begin{cases} 0 & u < 0 \\ 1 & u \geq= 0 \end{cases}$ (对应图中$\theta = 0$)

MP模型特征

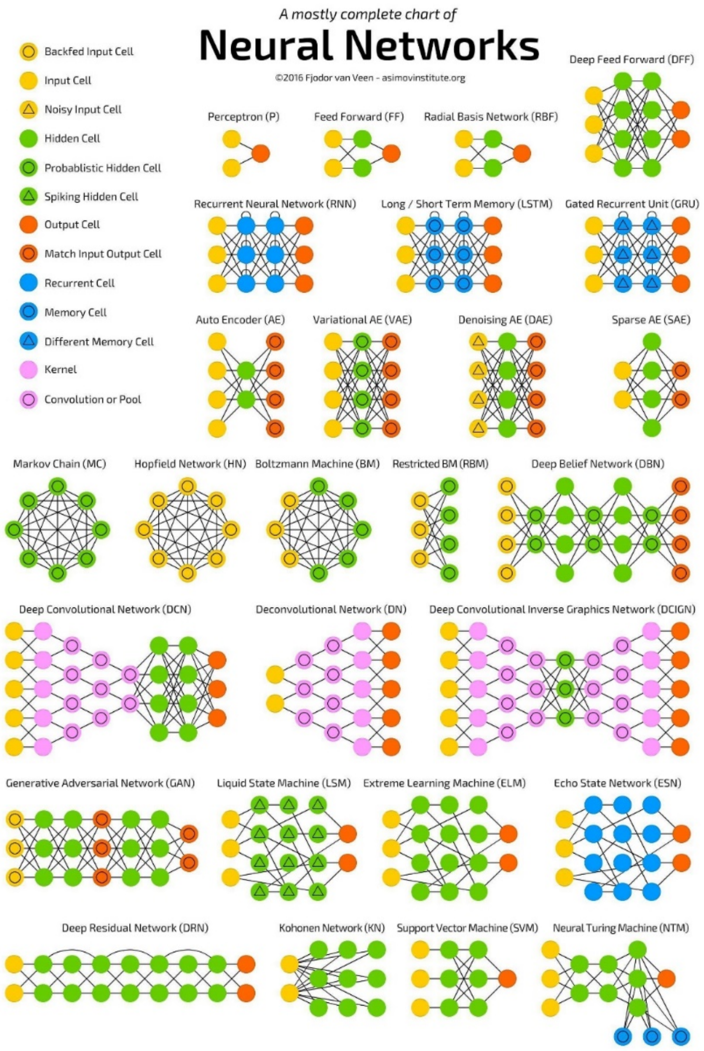
- 二值网络（binary：fire-1, or not fire-0）：自变量及其函数的值、向量分量的值只取0和1函数、向量。
- 由有方向的、带权重的路径联系
- 权为正：刺激；权为负：抑制
- 全或无：每个神经元有一个固定的阈值，如果输入大于阈值，就fires
- 绝对抑制：阈值被设为使得抑制为绝对的，即，非0的抑制输入将阻止该神经元兴奋。
- 花费一个时间单位使得信号通过一个连接

MP模型特征

- 固定权重和阈值（用户指定），每个神经元可提供一个简单的逻辑函数
- 输出变为兴奋（fire）状态：接收到k个或者更多的刺激输入，没有抑制输入
- 采用离散时间，能够使用MP模型来模拟有时间延迟下的物理现象
- 任意命题逻辑函数都可由一两层的MP模型计算
- 所有的命题逻辑函数都可以用MP AND逻辑门、MP OR逻辑门、MP NOT逻辑门予以表达和实现
- MP模型具有神经计算模型一般和普遍的特性，可表达一般人工神经网络的赋权联结和相对抑制

MP模型特征

- 之后大部分的神经元结构都采用MP神经元的多输入单输出模式。
- 多输入单输出的神经元通过不同的连接方式构成不同类型的神经网络。
- 神经元内部对数据进行简单的线性或非线性变换实现对数据的拟合。神经网络通过整合多个神经元实现对复杂任务的处理。

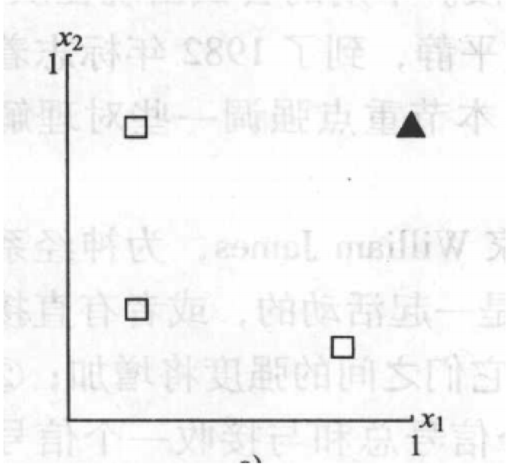


MP神经元示例：简单分类器

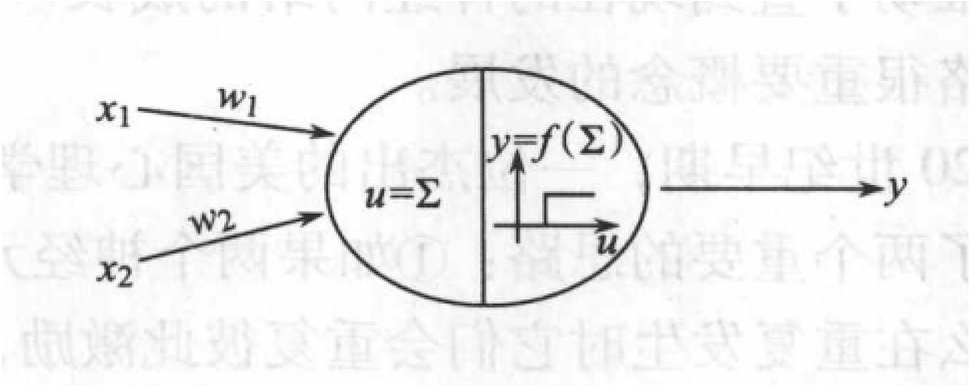
- 使用一个MP神经元实现简单分类器的功能

- 分类数据：

X_1	X_2	T
0.2	0.3	0
0.2	0.8	0
0.8	0.2	0
1.0	0.8	1



MP神经元示例：简单分类器



- 输入：x1和x2（2维输入）
- 输出：正确的类别
- 神经元实现:
- （1）通过w1,w2接收两个输入，在这里设定两个权值都为1.0
- （2）计算过程:

  a.计算网络输入u, $u = w_1x_1 + w_2x_2 = x_1+x_2$

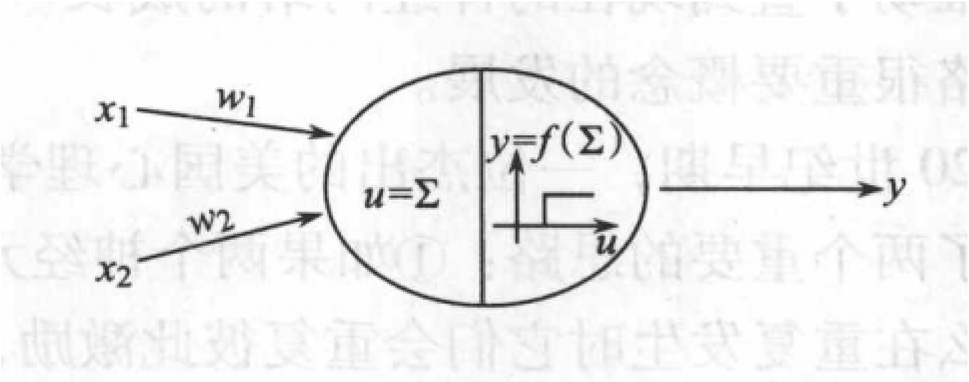
  b.使用阈值函数确定输出y，设定阈值为1.3

$$f(\Sigma) = y = \begin{cases} 0 & u < 1.3 \\ 1 & u \geqslant 1.3 \end{cases}$$

MP神经元示例：简单分类器

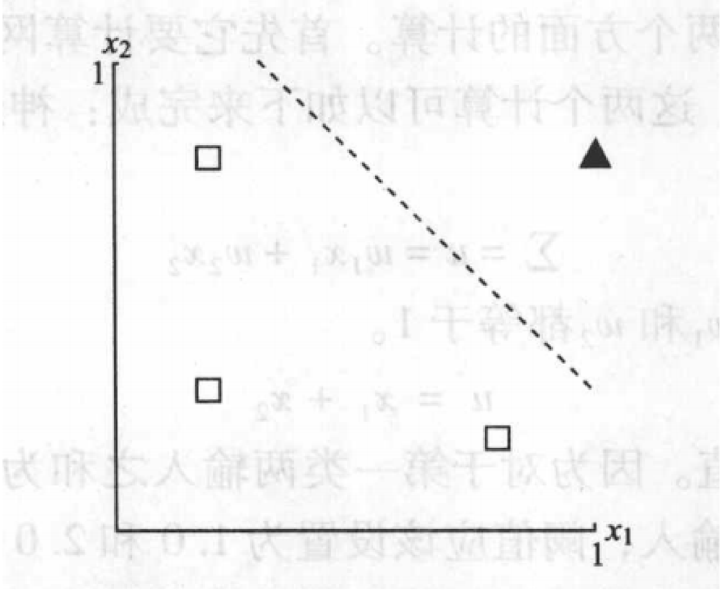
- 使用这个简单分类器，现在就能够检查它的性能。
- 对四个输入，神经元的输出结果如下表所示

(x1,x2)	u	T
(0.2,0.3)	0.5	0
(0.2,0.8)	1.0	0
(0.8,0.2)	1.0	0
(1.0,0.8)	1.8	1



MP神经元示例：简单分类器

- 神经元根据预先设置的U=1.3的阈值函数，可以对数据进行正确分类。
- 阈值函数的位置定义了两类，U=1.3决定一个分类界限可以表达为



$$u = x_1 + x_2 = 1.3$$

$$x_2 = 1.3 - x_1$$

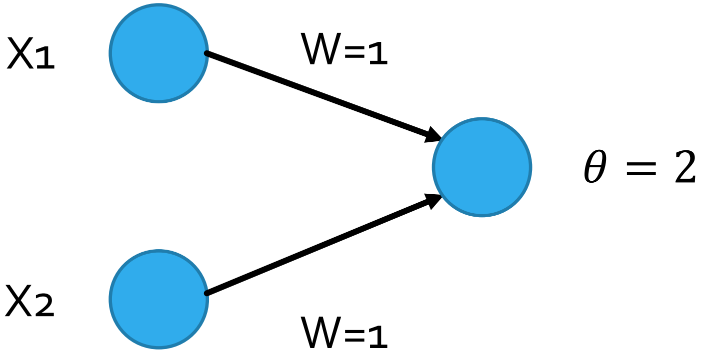
- 分类界限左下标记为0
- 分类界限右上标记为1

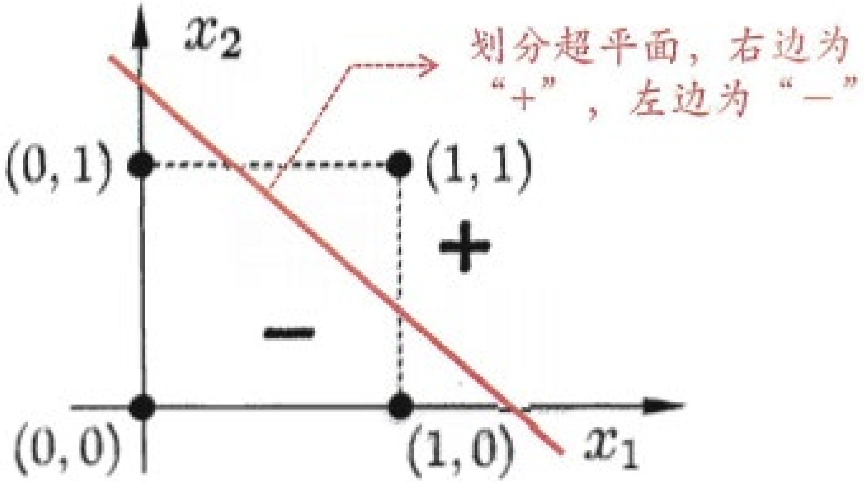
MP神经元分类器特点

- AND(与) x1 AND x2 -> y
  - W = 1, threshold = 2
- OR (或) x1 OR x2 -> y
  - W = 2， threshold = 2
- AND NOT （与非） x1 AND NOT x2 -> y
  - W = 2, p = 1, threshold = 2
- XOR (异或) x1 XOR x2 -> y
  - X1 XOR x2 <-> (x1 AND NOT x2) OR (x2 AND NOT x1)

“与”问题

- X1&X2





二、线性模型

这一节我们尝试如何用神经元去表示简单的**线性问题**。首先我们列举线性模型中经典的两种任务：**线性回归和线性分类**，然后我们用神经元模型来表示线性模型

股票预测问题

假设某只股票近二十天的股价记录如下:

日期	1	2	3	4	5	6	7	8	9	10
股价	1.0	0.5	1.9	2.5	2.6	3.2	3.4	4.4	4.3	5.8
日期	11	12	13	14	15	16	17	18	19	20
股价	5.6	6.3	7.0	7.0	7.2	9.0	8.6	8.7	9.6	10.4

如何通过日期来预测股票价格?

尝试建立一个描述日期与股价关系的方程。

构建模型

用x_i表示“输入”，也就是问题中的日期，也被称为**特征**(feature)；

用y_i表示“输出”，也就是问题中的股价，也被称为**目标**(target)。

如第三日的股价为1.9，即$x_3 = 3, y_3 = 1.9$。

一对输入输出(x_i, y_i)被称为一个**训练样本**(training sample)。

训练样本的集合$\{(x_1, y_1), \ldots, (x_n, y_n)\}$被称为**训练集**(training set)。

构建模型

用X表示输入空间，用Y表示输出空间。

这个问题中，$X = Y = \mathbb{R}$。

监督学习(supervised learning)即给定训练集，学习一个映射：

$$h:\quad X \longmapsto Y$$

使得$h(x)$是y的“好”预测。

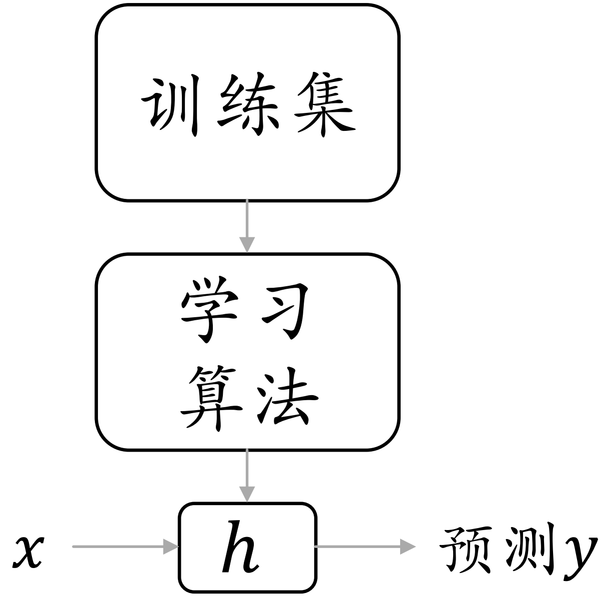
习惯上，方程h被称为一个**假设**(hypothesis)

线性回归和分类

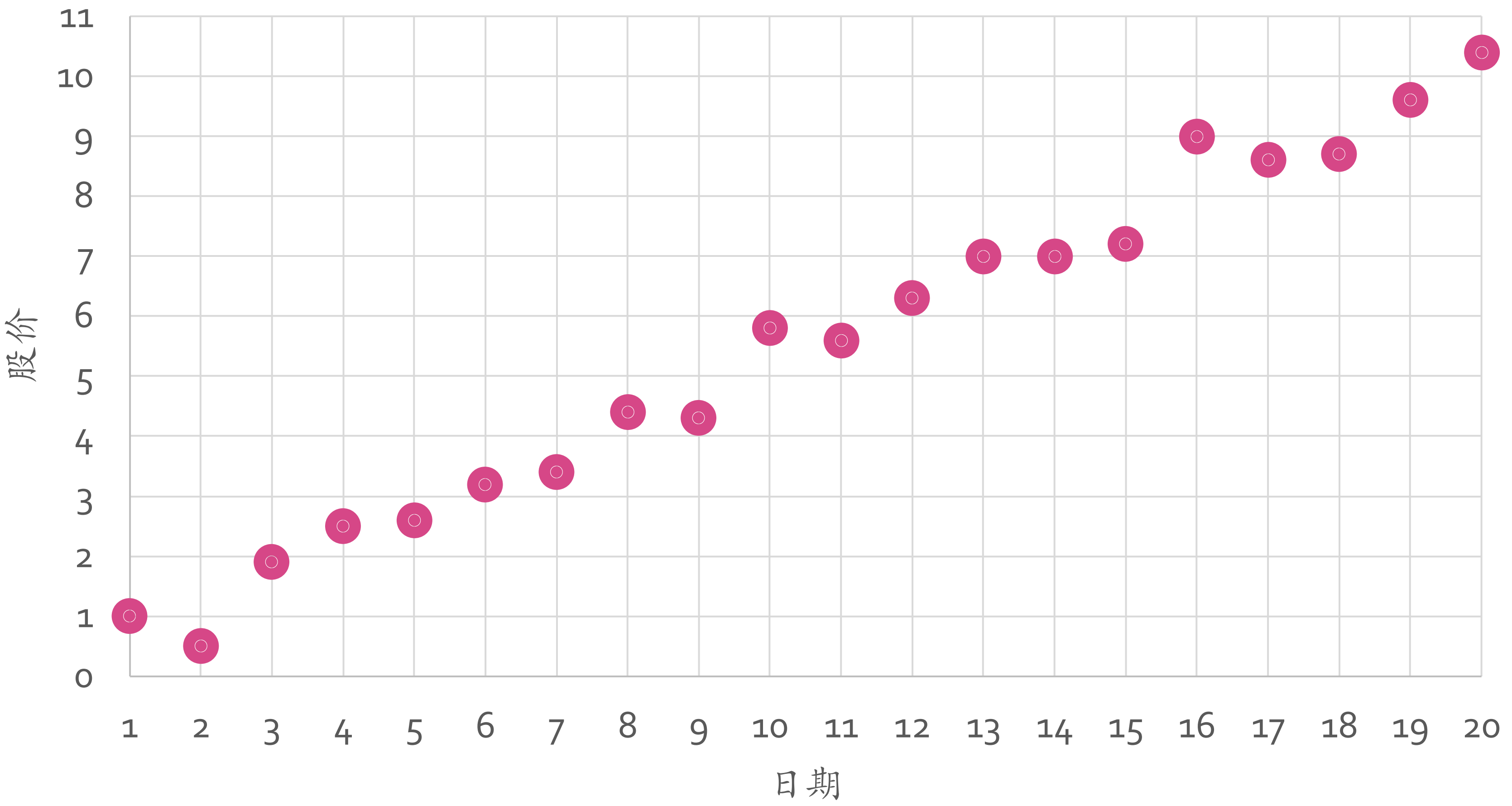
整个过程如右图所示：

如y可以取连续值，例如问题中的股价，则这个问题被称为**回归**问题(regression)；

如果y取有限数量的离散值（如预测股票涨或者跌），这个问题被称为**分类**问题(classification)。



线性回归



线性回归

可以看出来股价和日期非常接近线性关系，不妨假设 $h(x) = wx + b$。

这样线性关系的回归问题被称为**线性回归**(linear regression)。

其中w, b为所需求得的**参数**(parameter)。

现在，给定了训练集，为了求得参数还需要一个标准来评价预测$h(x)$是否足够“好”。

线性回归

线性回归的目标是要得到一条尽可能表示数据分布的线性函数。可以用均方误差表示这一目标：

训练集$\{(x_1,y_1),\dots,(x_n,y_n)\}$大小为$n$，如下公式

$$E=\frac{1}{n}\sum_{i=1}^{n}[h(x_i)-y_i]^2$$

被称为假设h的**均方误差**(mean squared error)。均方误差越小越“好”。

常用最小二乘法最小化均方误差。

最小二乘法

这样，问题就被化为求

$$\underset{w,b}{\operatorname{argmin}} \ E$$

$$= \underset{w,b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (wx_i + b - y_i)^2$$

要求使得E最小的w, b，可以用E对它们分别求偏导。

最小二乘法

$$\frac{\partial E}{\partial w} = \frac{2}{n}\left(w\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n}(b - y_i)x_i\right) \quad （1）$$

$$\frac{\partial E}{\partial b} = 2\left(nb + \sum_{i=1}^{n}(wx_i - y_i)\right) \quad （2）$$

令两式为零可求得w与b的**闭式解**(closed-form solution)。

最小二乘法

对(2)式 $$nb + \sum_{i=1}^{n}(wx_i - y_i) = 0$$

$$\Rightarrow b = \frac{\sum_{i=1}^{n}(y_i - wx_i)}{n}$$

对(1)式 $$w\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n}(b - y_i)x_i = 0$$

$$\Rightarrow w\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n}\left(\frac{\sum_{j=1}^{n}(y_j - wx_j)}{n} - y_i\right)x_i = 0$$

最小二乘法

$$\Rightarrow w\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_j\right)x_i\right] = \sum_{i=1}^{n}\left(y_i - \frac{\sum_{j=1}^{n} y_j}{n}\right)x_i$$

$$\Rightarrow w = \frac{\sum_{i=1}^{n} x_i\left(y_i - \frac{\sum_{j=1}^{n} y_j}{n}\right)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}$$

其中$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$，$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$为$x, y$的均值。

最小二乘法

$$w = \frac{\sum_{i=1}^{n} x_i \left(y_i - \frac{\sum_{j=1}^{n} y_j}{n}\right)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}$$

$$b = \frac{\sum_{i=1}^{n}(y_i - w x_i)}{n} = \bar{y} - w\bar{x}$$

其中$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$，$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$为$x, y$的均值。

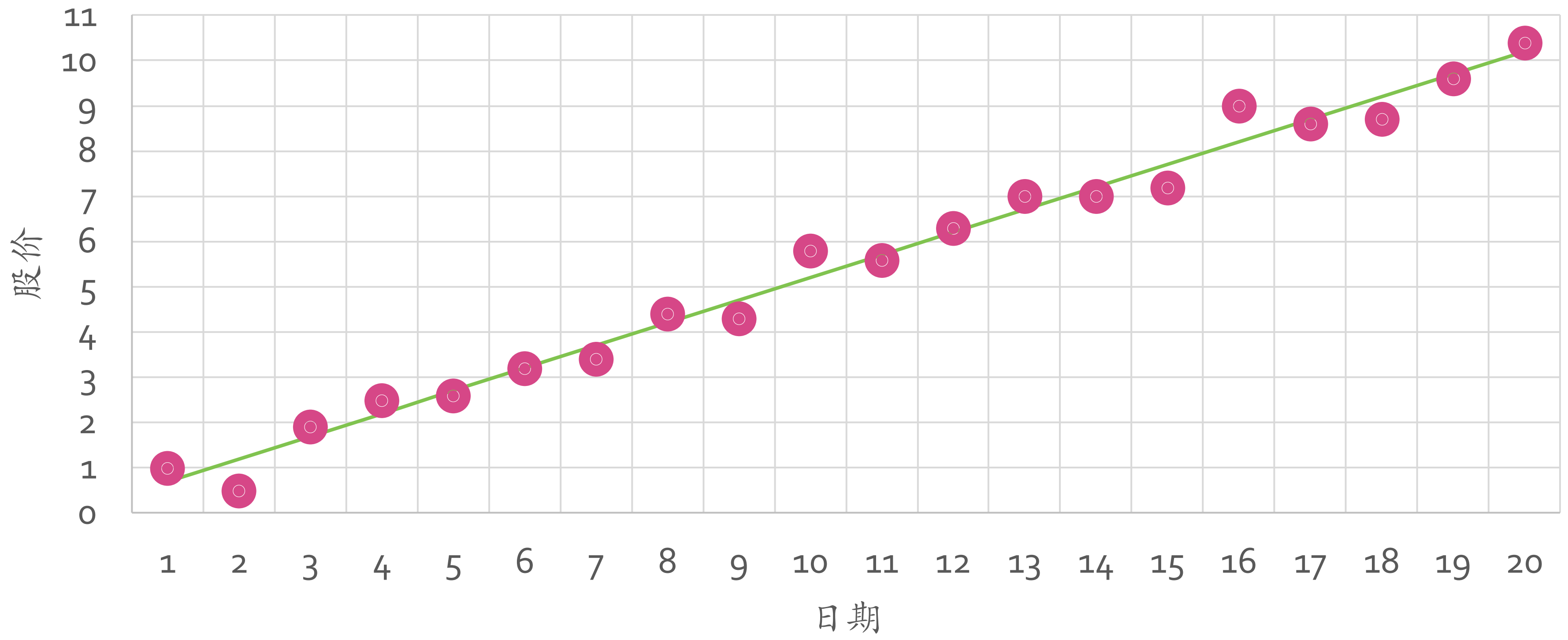
此外也可以化成

$$w = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

最小二乘法

将数据代入，可以求得$w = 0.5, b = 0.2$。

作直线$y = 0.5x + 0.2$与股票数据进行对比：



多元线性回归

之前的问题中用于预测股票价格的输入为日期，可以认为特征的维度为1。

而有些时候每个样本有不止一个输入，也即有多个维度的特征，设特征的维度为d。

将其写作向量形式$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$。

此时的线性回归可以写成：

$$
\begin{aligned}
h(\boldsymbol{x}_i) &= w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} + b \\
&= \boldsymbol{w}^T \boldsymbol{x}_i
\end{aligned}
$$

其中$\boldsymbol{w} = (w_1, w_2, \dots, w_d)^T$

多元线性回归

此时，对于训练集$\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$，尝试习得

$$h(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i + b$$

使得

$$h(\boldsymbol{x}_i) \simeq y_i, i = 1,2, \dots, n$$

则称为**多元线性回归**(multivariate linear regression)。

所求的$\boldsymbol{w}$又被称为特征的**权重**(weight)，b则称为**偏置**(bias)。

多元线性回归

类似地，使用最小二乘法来求解。

为书写方便，令

$$\hat{\boldsymbol{w}} = (\boldsymbol{w}; b)$$

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T & 1 \\ \boldsymbol{x}_2^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_n^T & 1 \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} & 1 \end{pmatrix}$$

$$\boldsymbol{y} = (y_1, y_2, \dots, y_n)^T$$

也就是将偏置并入权重，并令其对应的输入恒为1，并将整个数据集合并为矩阵形式。

多元线性回归

此时的均方误差为

$$E=\frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{w}}\|^2$$

$\|\cdot\|$为欧几里得范数。

问题可以写成求

$$\begin{aligned}&\underset{\widehat{\boldsymbol{w}}}{\operatorname{argmin}}\ E\\&=\underset{\widehat{\boldsymbol{w}}}{\operatorname{argmin}}(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{w}})^T\,(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{w}})\end{aligned}$$

多元线性回归

同样求导得到

$$\frac{\partial E}{\partial \hat{\boldsymbol{w}}} = 2\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$

令其为零可求得闭式解。

当$\boldsymbol{X}^T\boldsymbol{X}$满秩或正定时，解

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

若$\boldsymbol{X}^T\boldsymbol{X}$不满秩，例如当$d \geq n$时令$E$最小的结果不唯一，这里不作深入讨论。

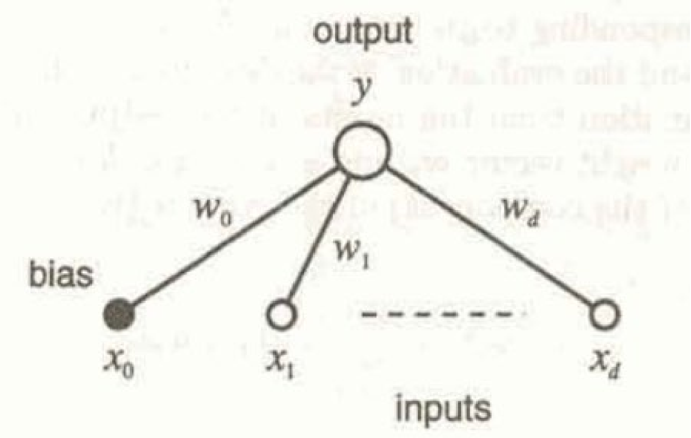
神经元表示

线性回归的线性方程可用如下神经元表示（输入神经元只负责传递数据）

在神经元内只对输入进行加权求和，x_0和w_0表示线性方程中常数项。

在线性回归问题中，输出y取一常数，一般取为0

上述神经元可用于股票问题线性回归问题。



对数线性回归

上述线性回归所得到的神经元的输出范围为整个实数域，而现实中股票价格显然是不会出现负数的。并且，股票价值的增减也常用相对比例来表示。此时，可以将回归的目标定为y的其它衍生物，如对数值。

按照之前线性回归的形式，假设可写成

$$\ln[h(x)] = \boldsymbol{w}^T\boldsymbol{x} + b$$

$$h(x) = e^{\boldsymbol{w}^T\boldsymbol{x}+b}$$

此时便称作对数线性回归。

广义线性回归

更一般的，考虑函数$g(\cdot)$，令

$$h(x) = g^{-1}(\boldsymbol{w}^T\boldsymbol{x} + b)$$

在函数 g 单调可微的情况下，都可以用最小二乘法求解，此时便称作广义线性回归，$g(\cdot)$ 称作联系函数。

显然，对数线性回归是$g(\cdot) = \ln(\cdot)$时的特例。

分类问题（多维输入）

这里列出了一些水果的宽和高的数据:

水果	宽度	高度	水果	宽度	高度	水果	宽度	高度	水果	宽度	高度
苹果	8.4	7.3	苹果	7.6	7.3	苹果	7.6	7.9	柠檬	6.0	7.5
苹果	8.0	6.8	苹果	7.7	7.1	柠檬	7.2	10.3	柠檬	5.9	8.0
苹果	7.4	7.2	苹果	7.6	7.5	柠檬	7.3	10.5	柠檬	6.0	8.4
苹果	7.1	7.8	苹果	7.5	7.6	柠檬	7.2	9.2	柠檬	6.1	8.5
苹果	7.4	7.0	苹果	7.5	7.1	柠檬	7.3	10.2	柠檬	6.3	7.7
苹果	6.9	7.3	苹果	7.4	7.2	柠檬	7.3	9.7	柠檬	5.9	8.1
苹果	7.1	7.6	苹果	7.5	7.5	柠檬	7.3	10.1	柠檬	6.5	8.5
苹果	7.0	7.1	苹果	7.4	7.4	柠檬	5.8	8.7	柠檬	6.1	8.1
苹果	7.3	7.7	苹果	7.3	7.1	柠檬	6.0	8.2			

尝试利用这些数据完成一个分类问题

分类问题

在这个问题中，特征的维度为2：长度和宽度，不妨令样本特征$x_i=(x_{i1},x_{i2})$，其中x_{i1}代表长度，x_{i2}代表宽度；目标是离散的，我们不妨设苹果为0，柠檬为1。学习的任务则可以表示为求

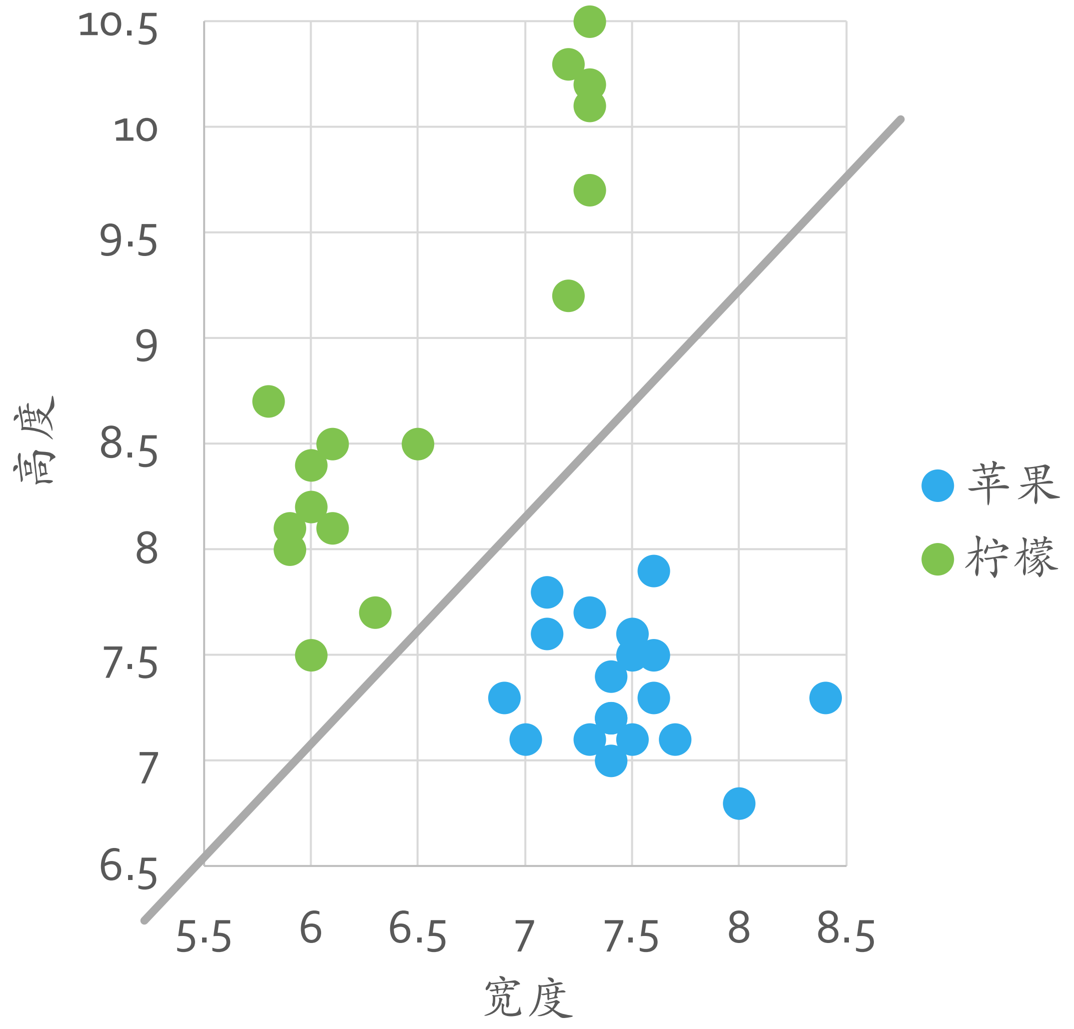
$$h(\boldsymbol{x}_i)\simeq y_i=\begin{cases}0 & ,当\boldsymbol{x}_i属于苹果时\\ 1 & ,当\boldsymbol{x}_i属于柠檬时\end{cases}$$

同样先作图。

分类问题

可以看出，通过宽度和高度的对比可以区分出苹果和柠檬。

用一条直线便可将代表苹果和柠檬的点分开，这个问题便可称为**线性划分**(linear division)。



线性划分

平面上的一条直线可以表示为

$$w_1 x_{i1} + w_2 x_{i2} + w_0 = \boldsymbol{w}^T \boldsymbol{x}_i + w_0 = 0$$

其中$\boldsymbol{w} = (w_1, w_2)^T$。

对于直线上方的区域，有 $\boldsymbol{w}^T \boldsymbol{x}_i + w_0 > 0$ ；对于直线下方的区域，有$\boldsymbol{w}^T \boldsymbol{x}_i + w_0 < 0$。

这样可以取

$$h(\boldsymbol{x}_i) = f\left(\boldsymbol{w}^T \boldsymbol{x}_i + w_0\right)$$

其中

$$f(u) = \begin{cases} 0 & ,u < 0 \\ 1 & ,u \geq 0 \end{cases}$$

也被称作**单位阶跃函数**(unit-step function)

线性二分类问题的神经网络表示

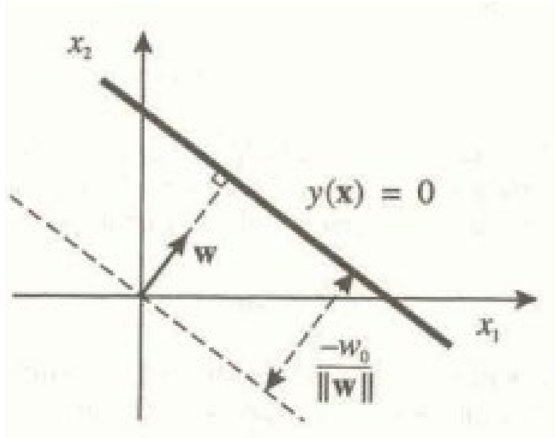
线性 决策边界$w_1x_{i1}+w_2x_{i2}+w_0=\boldsymbol{w}^T\boldsymbol{x}_i+w_0=0$ 与向量 $\boldsymbol{w}$ 正交。

其中$\boldsymbol{w}=(w_1,w_2)^T$。

对于直线上方的区域，有$\boldsymbol{w}^T\boldsymbol{x}_i+b>0$；对于直线下方的区域，有$\boldsymbol{w}^T\boldsymbol{x}_i+b<0$。

这样可以取

$$h(\boldsymbol{x}_i)=f\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)$$



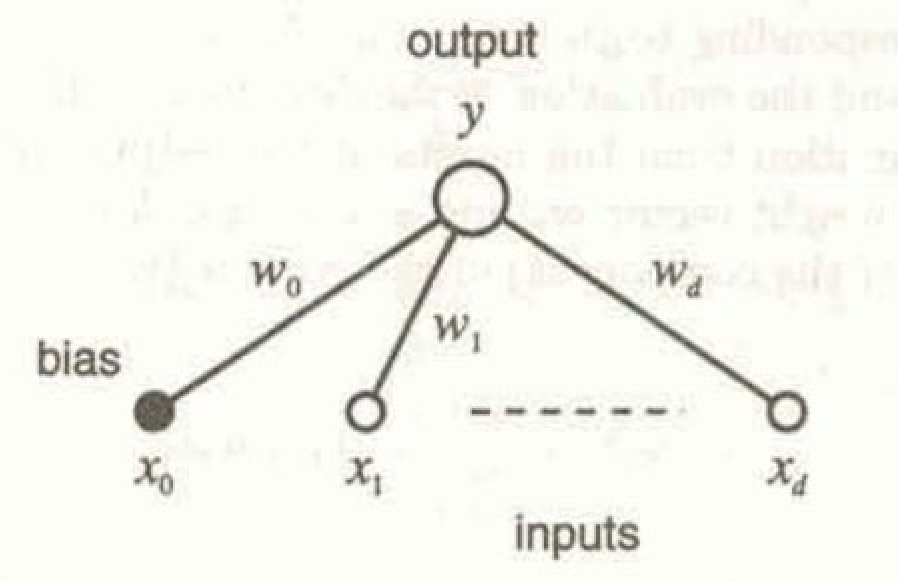
其中

$$f(u)=\begin{cases}0 & ,u<0\\ 1 & ,u\geq 0\end{cases}$$

也被称作**单位阶跃函数**(unit-step function)

神经元表示

线性分类需要更复杂的神经元来解决，两分类问题可用一个神经元来表示，但不同于线性回归，神经元中对数据加权求和后需要在进行一步非线性操作，即用**单位阶跃函数对**求和结果进行映射。



输出y也不再是一个常数，两分类中用0,1表示不同的类别。

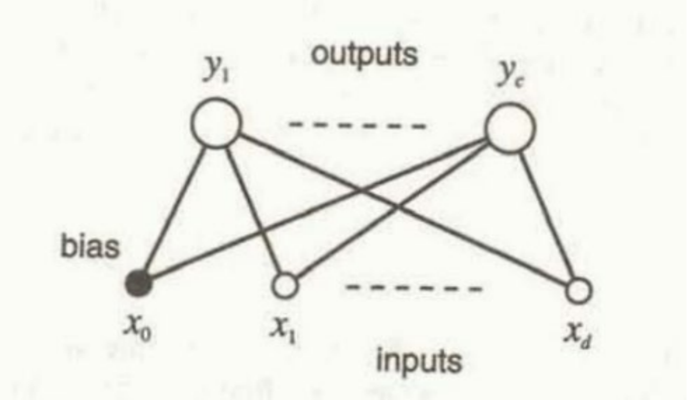
思考：如果是多分类问题该如何解决?

- 输出神经元可以直接输出一个向量，作为类别的one-hot编码

- 多个输出神经元

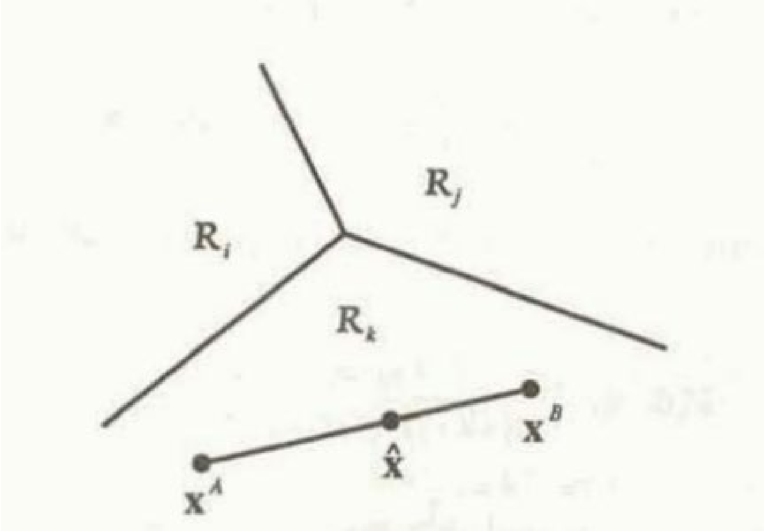
神经元表示

多分类情况下需要用更多的神经元来表示：



多个分类神经元表示的决策边界：

点A和点B同属于k区域（同一类），他们构建的线性关系，即A到B线段上的所有点都应属于k区域，这要求区域k是连续和凸形



线性可分

在之前的问题中，由$\boldsymbol{w}^T\boldsymbol{x}=0$决定的超平面将不同的类别分开，这样将不同类别分开的界限便被称为**决策边界**(decision boundary)。

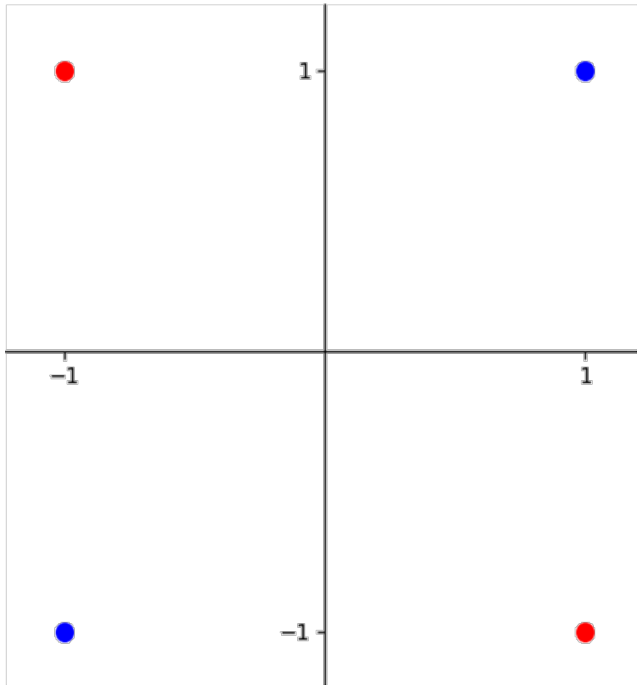
而能够由线性的决策边界进行划分的分类问题则被称为**线性可分**(linear separable)。

线性不可分

而现实中的很多分类问题是线性不可分的，如XOR问题。

如果令+1为真，−1为假，则XOR问题如图所示。

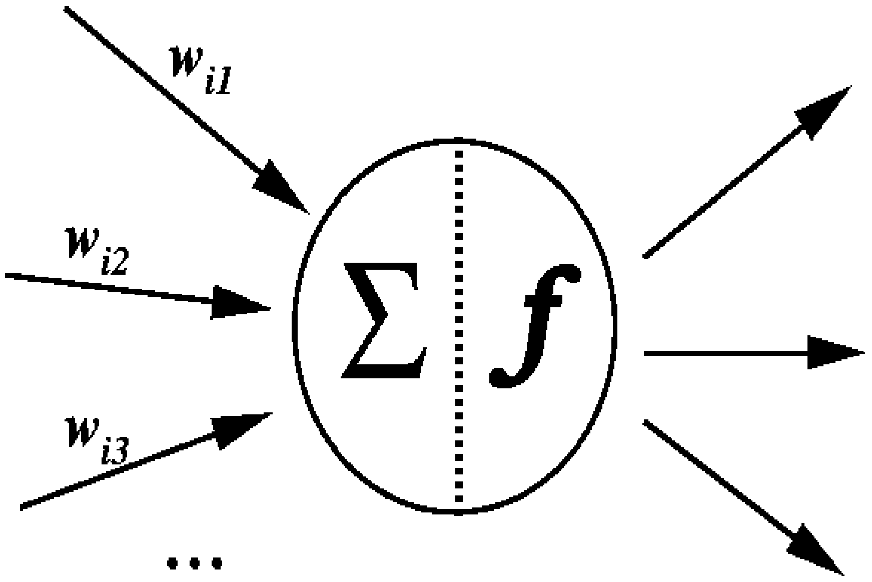
其中红色为真类，蓝色为假类。显然，无法找到一条直线划分两个类别。



三、感知器神经元

神经网络的最基本单元：感知器神经元

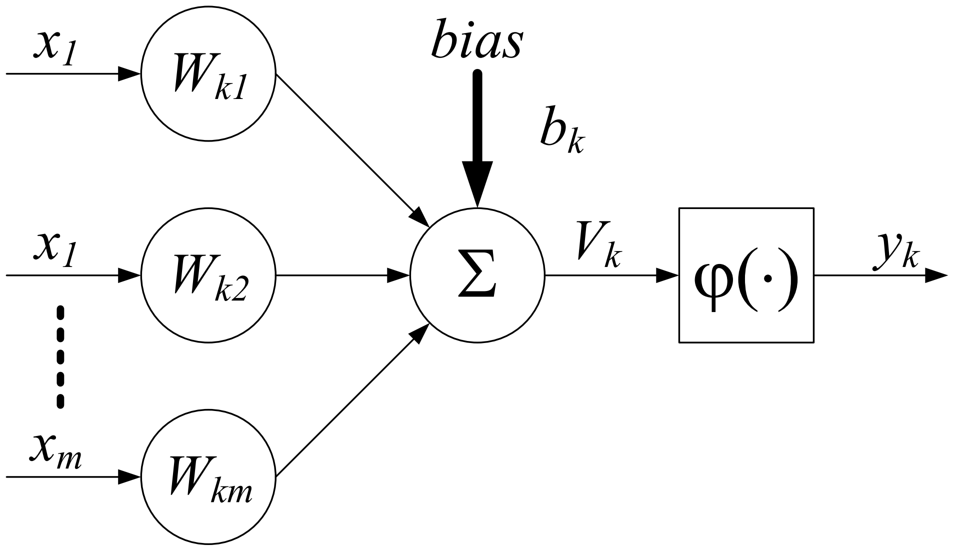
- 线性模型的例子引出最流行的神经元结构：感知器神经元
- 每个连接都有相应的权重，在典型的神经网络中，权重和输入信号相乘
- 每个神经元对其接受到的输入（输入信号的加权和）应用激活函数（activation function）来决定其输出
- 感知器神经元从样本数据中学习，权值在学习过程中变化

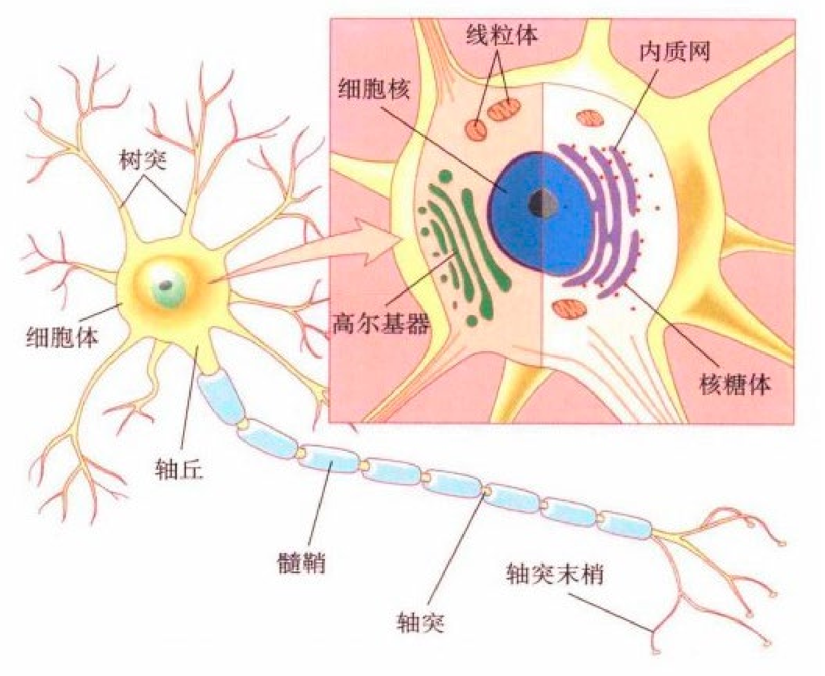


$$y_i = f(net_i)$$

$$y_i = f\left(\sum_j w_{ij}\, y_j\right)$$

神经元





$$net_k = u_k = \sum_{j=1}^{m} w_{kj} x_j \quad \Rightarrow \quad V_k = \sum_{j=0}^{m} w_{kj} x_j \text{ with} \begin{cases} \mathrm{w}_{k0} = \mathrm{b}_k \\ \mathrm{x}_0 = 1 \end{cases}$$

$$O_k = y_k = \varphi(u_k + b_k) \qquad y_k = \varphi(v_k)$$

神经元解析

- “多输入——单输出”的结构特征
  - 众多的树突为神经信号提供输入通道
  - 单一的轴突为神经信号提供输出通道
- “整合——激发”的功能特征
  - 整合输入信号：时间整合与空间整合功能
  - 激发神经冲动：“全或无”式的兴奋与抑制功能
- 记忆和学习
  - 联结权重的存储与更新
- 人工神经元就是一个具有记忆功能的输入权值化的“多输入—单输出”“整合—激发”装置。
  - 是一个信息处理单元，一个自动机
  - 可以概括各类人工神经元

结构、功能、记忆和学习的方式形成不同的神经元

激活函数

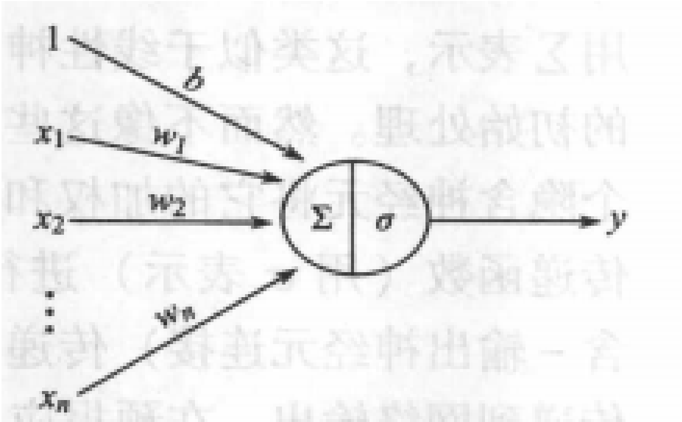
功能：将神经元的加权输入线性/非线性转换成一个输出的激活函数

神经元的输出：

σ：激活函数

b：与偏差输入相关的权值

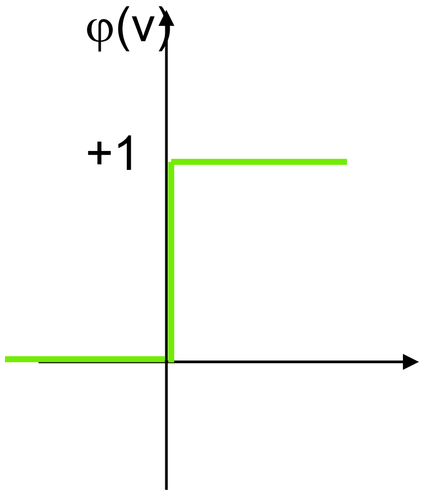
w_j:表示与第j个输入相关的权值



$$\sigma\left(\sum_{j=1}^{n} w_j x_j + \mathrm{b}\right)$$

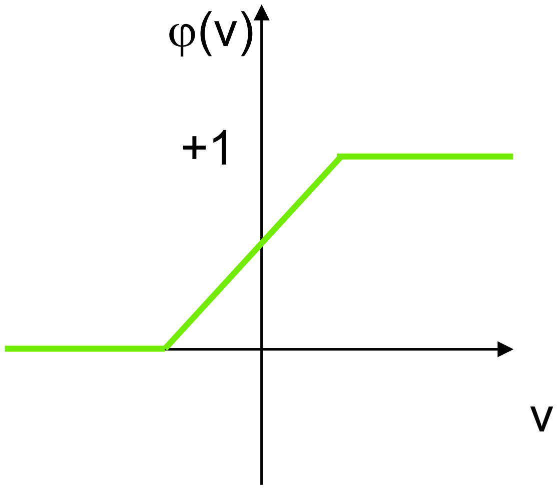
- Identity function (同一函数)
  - $f(x)=x$, for all x
- Threshold function (阈值函数)
  - $f(x)=1$, if x>=阈值
  - $f(x)=-1$, if x<阈值
- Piecewise linear function(分段线性函数)
  - $f(x)=1$, if x>=阈值
  - $f(x)=x$, if -阈值<x<阈值
  - $f(x)=-1$, if x<=-阈值

Activation function



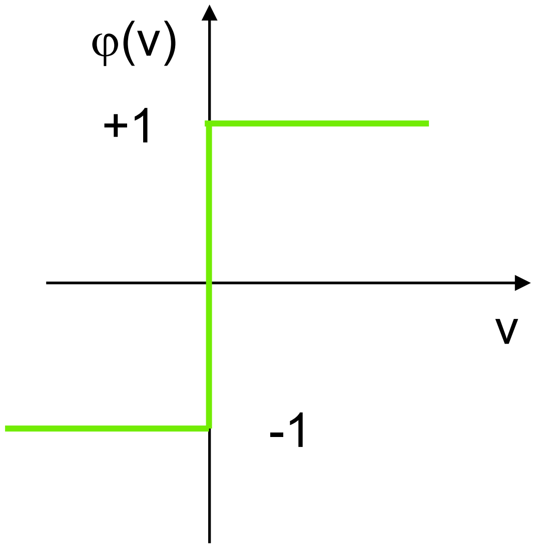
Threshold Function

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$



Piecewise-linear Function

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq \frac{1}{2} \\ v + \frac{1}{2} & \text{if } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{if } v < -\frac{1}{2} \end{cases}$$



Signum Function

思考：以上阶跃式激活函数存在什么问题?

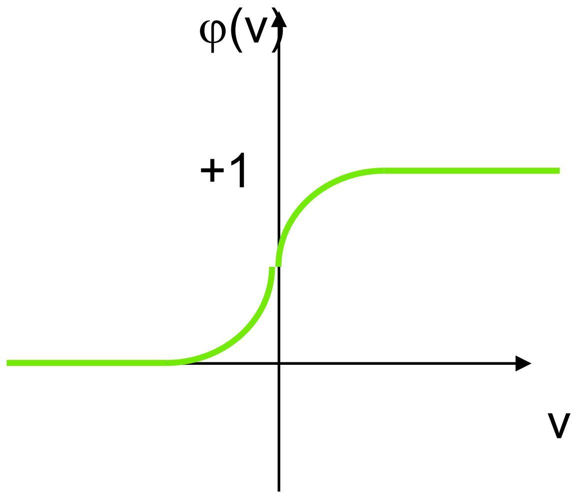
非线性激活函数

非线性激活函数：保持在上限和下限之间的非线性的连续函数

（1）非线性：函数的输出随输入做非线性的变化，

（2）连续函数：函数中没有顶点或者中断，可以从始至终进行微分

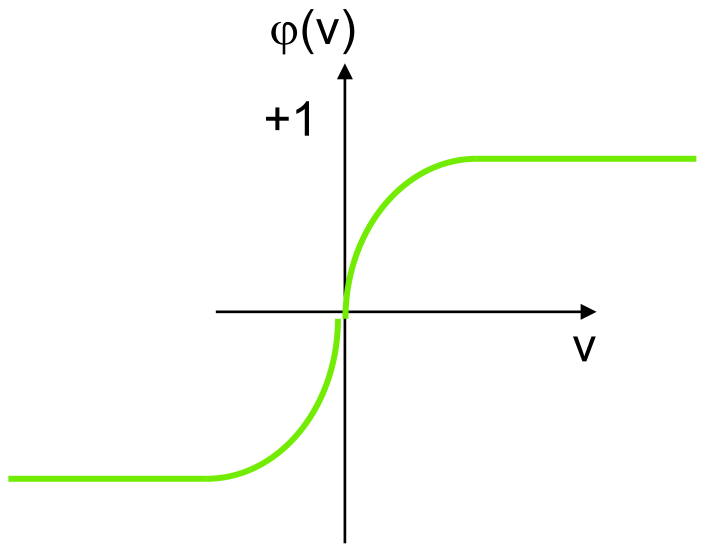
S型激活函数



Sigmoid (Uni-polar) Function
(differentiable)

$$\varphi(v) = \frac{1}{1+\exp(-av)}$$

a is slope parameter

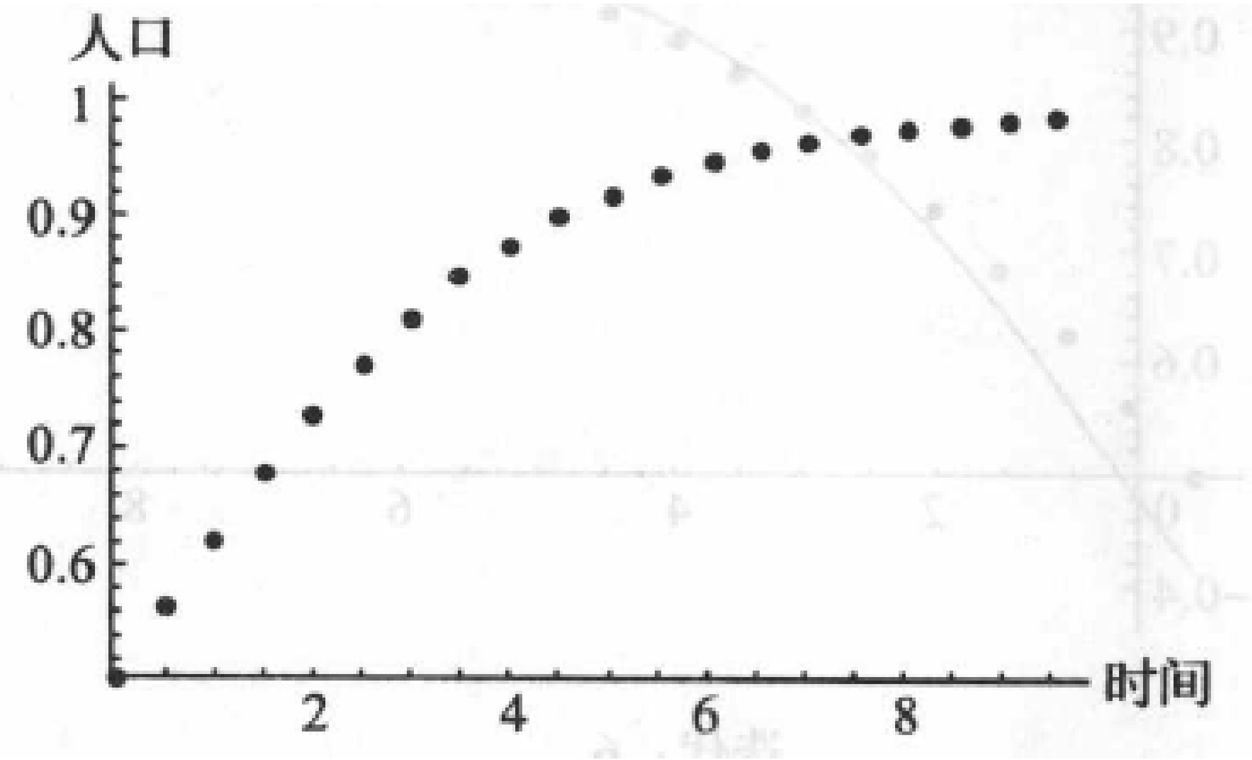


Hyperbolic tangent (Bi-polar)Function
（双曲正切）

$$\varphi(v) = \tanh(av) = \frac{1-\exp(-av)}{1+\exp(-av)}$$

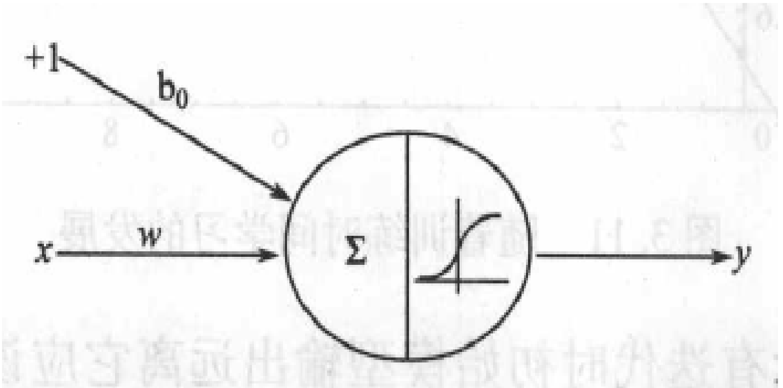
非线性函数实例

人口增长模型：人工增长模型常用指数函数建模。一个拥有对数函数的单一神经元可以利用一下图示数据训练人口增长模型



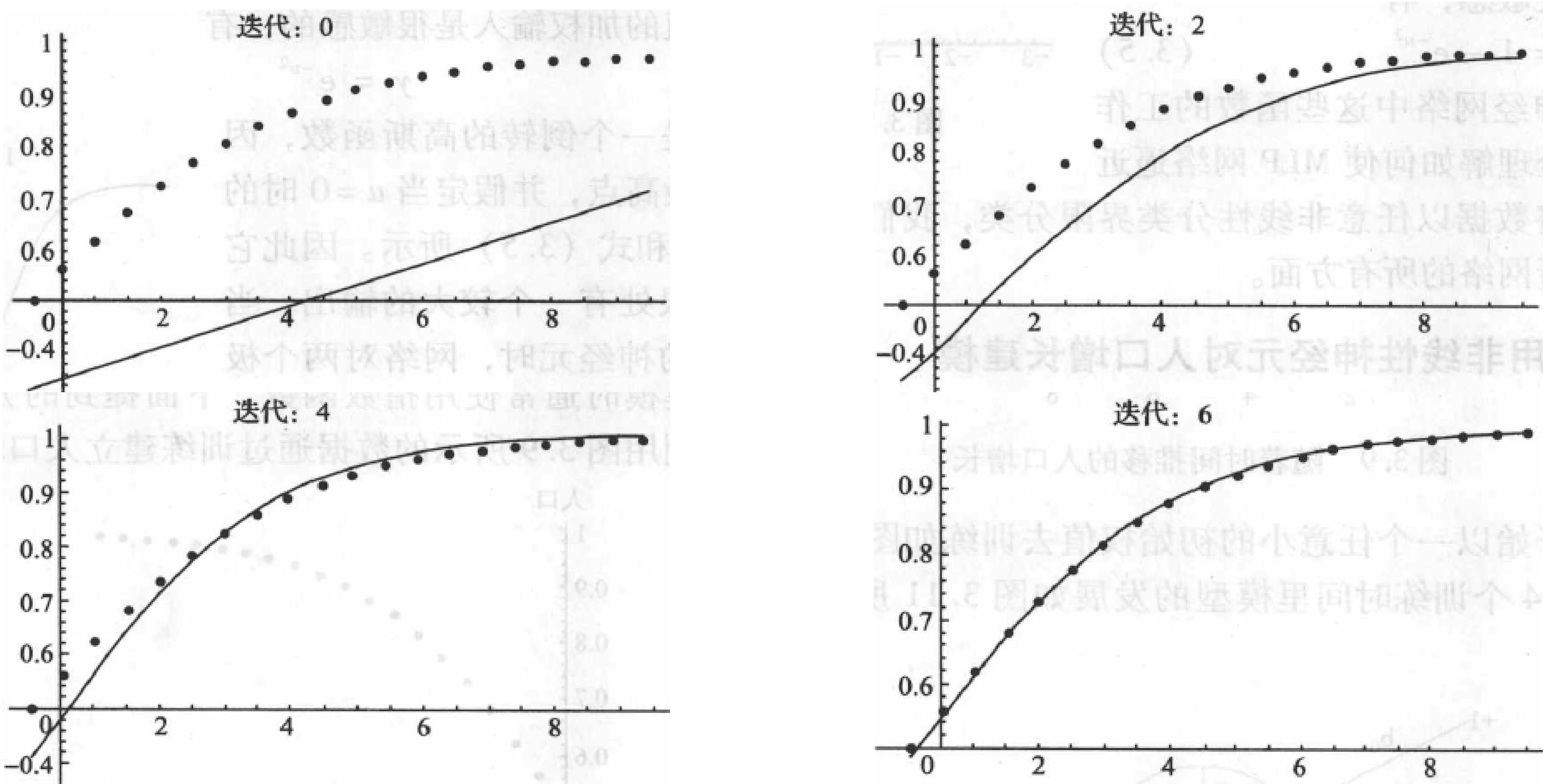
非线性函数实例

人口增长模型神经元:



非线性函数实例

通过一些学习方法（牛顿法，梯度下降法）拟合数据，神经元的学习大致过程



非线性函数实例

上述训练后的神经元输出模型：$y = \frac{1}{1+e^{-0.5x}}$，bias为-0.00002，w为0.5

一个具有对数传递函数的单一非线性神经元是具有能力建立简单的、非线性函数模型的。

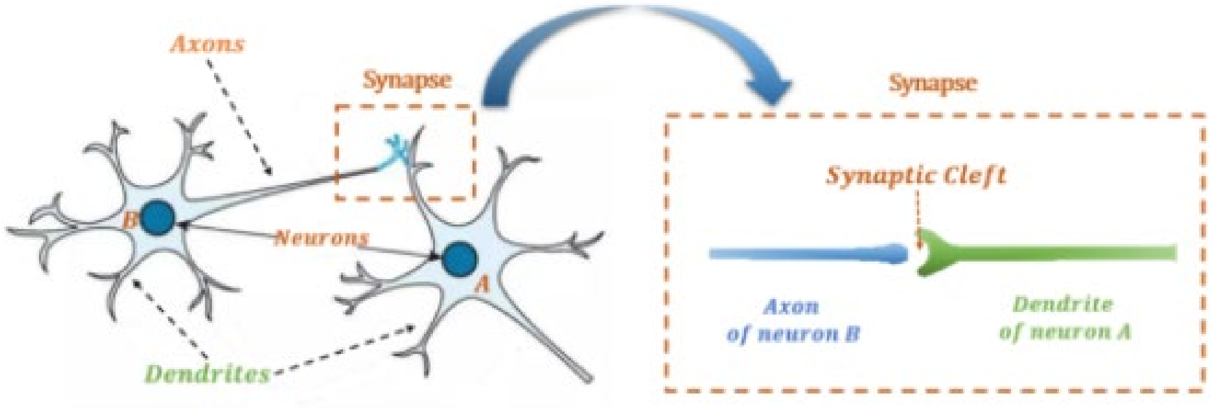
一个单一的神经元可以建立任何输出区域模型，这里输出是单调（连续的）增加或减少的。带有多个输出的单一神经元将产生一个多维数的非线性模型，输出形式为

$$y = \frac{1}{1 + e^{-(w_1x_1+w_2x_2+w_3x_3+\ldots+w_nx_n)}}$$

四、FT神经元

另外一种神经元：FT神经元

启发来源：神经元A接收到来自神经元B的刺激信号后的响应，不仅取决于神经元B的轴突传递强度，还依赖与神经元A的树突浓度，这与神经元A的记忆单元有关。

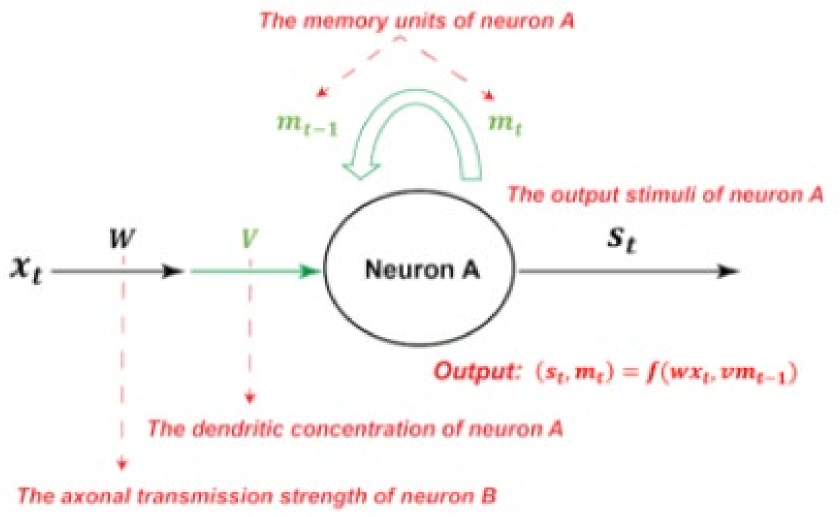


另外一种神经元：FT神经元

输入参数：

w表示轴突传递强度，

v表示树突浓度，与记忆强度有关



输出：

s_t:神经元生成的生物电/化学刺激信号，传递给下一个神经元

m_t:神经元的当前记忆强度，更新该神经元的记忆强度

$$(s_t, m_t) = f(x_t, m_{t-1}; w, v).$$

作业

- 试画出能够实现OR和NOT逻辑运算的感知器神经元模型，并尝试将其组合为实现XOR的感知器.
- 利用最小二乘法求解下列x,y关系的线性回归方程，并用单神经元表示
  - 需要指明各参数和函数类型

x	24	15	23	19	16	11	20	16	17	13
y	92	79	97	89	64	47	83	68	71	59

- 提交格式：姓名_学号_作业1.zip
- 提交邮箱：sjwl_2020_yjs@163.com（研究生）
  sjwl_2020_bks@163.com（本科生）