



- 人工神经元模型**Neuron Model**

- 人工神经网络模型**Neural Networks Model**



Neural Network Structures

A Neural Network = Neurons + Connection

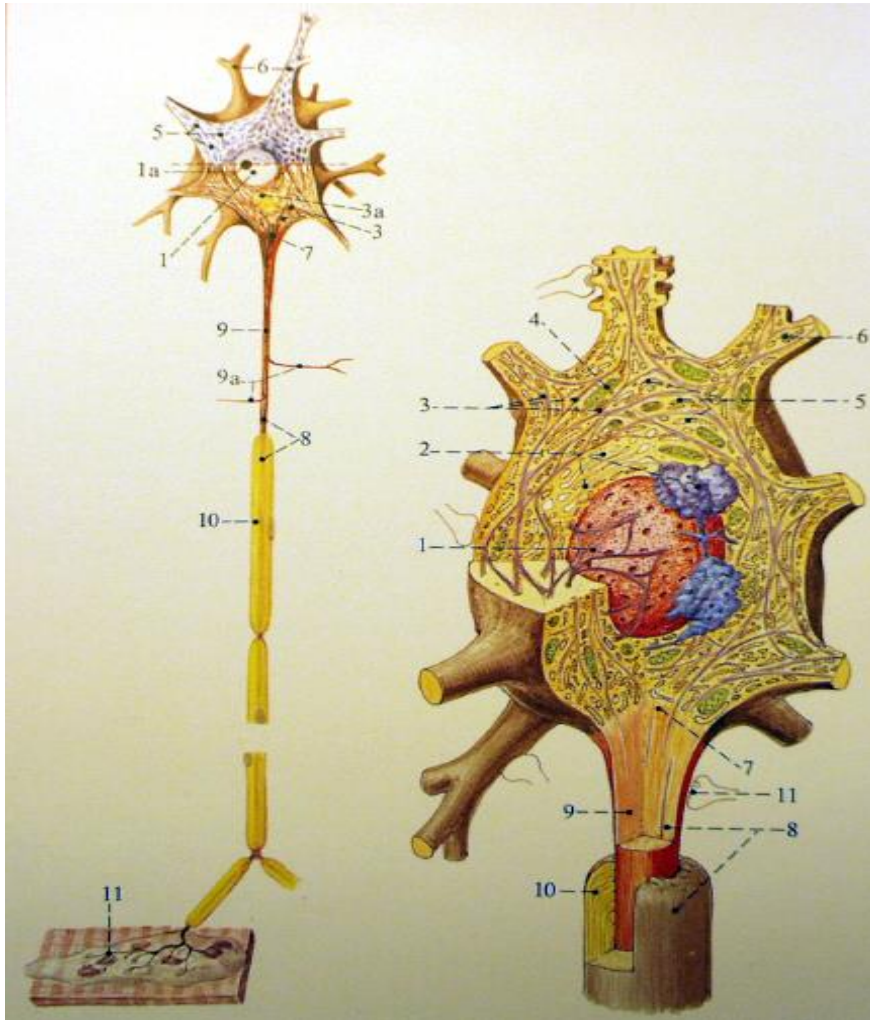




Neuron Model

- 生物神经元的基本功能
- 仿生模拟，构建数学模型
- 简单应用

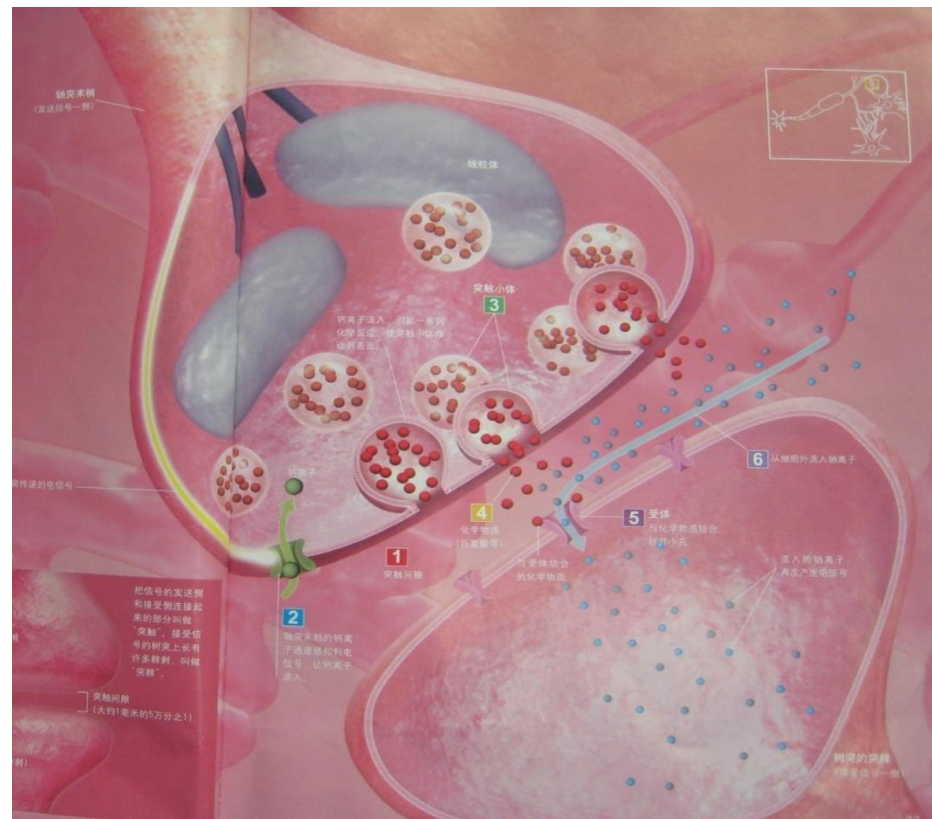
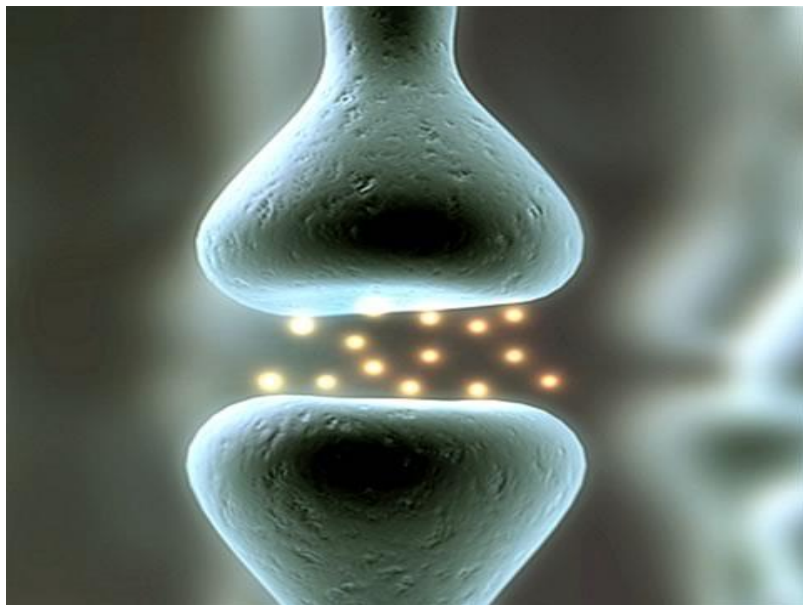
生物神经元(neuron)



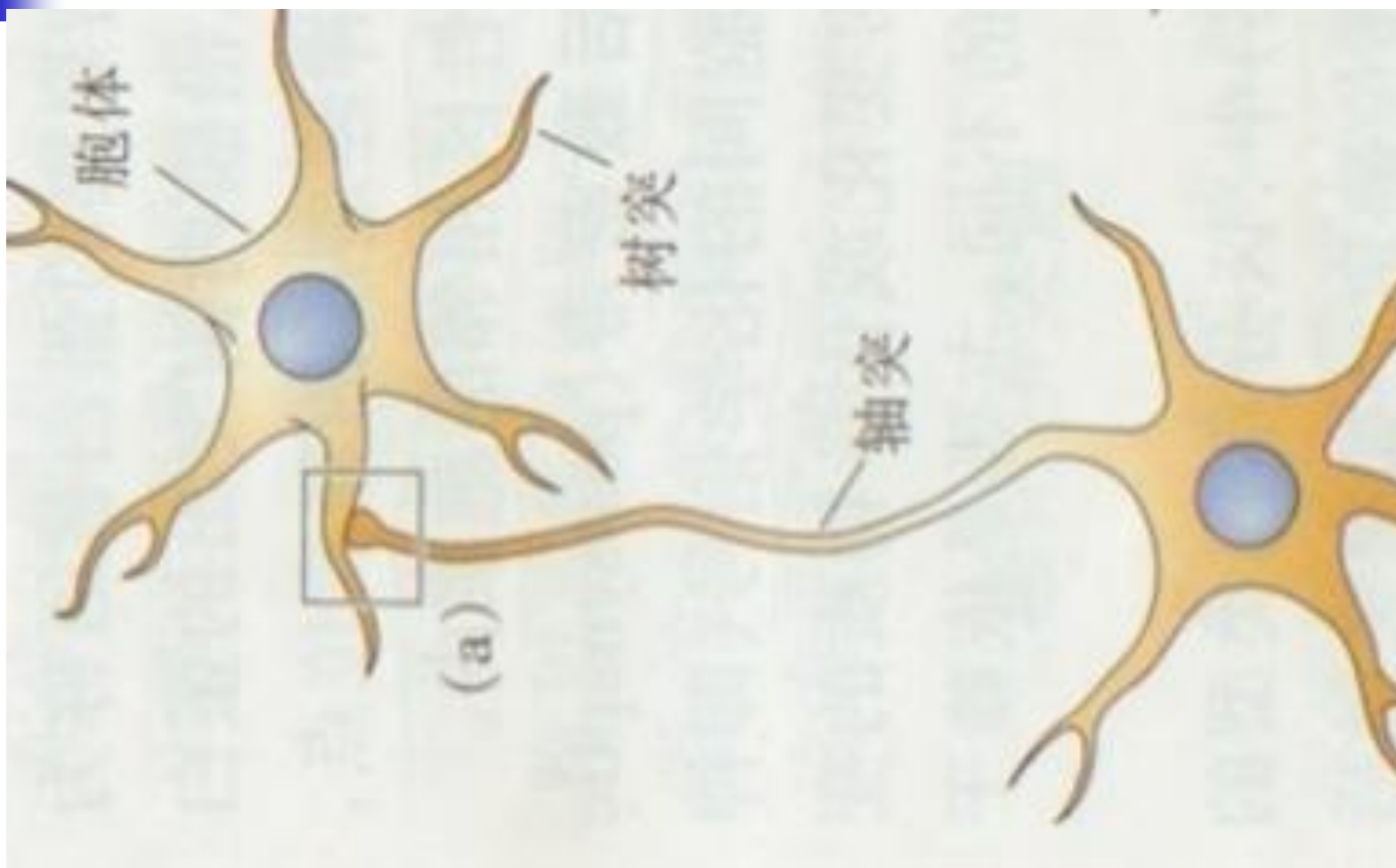
- **Soma:** 胞体
- **dendrite:** 树突
- **axon:** 轴突
- **synapse:** 突触
- **action potentials:** 动作电位= 神经冲动, 传导不衰减

突触(synapse)

轴突与其他神经元的连接点，进行电-化学-电信号转换。

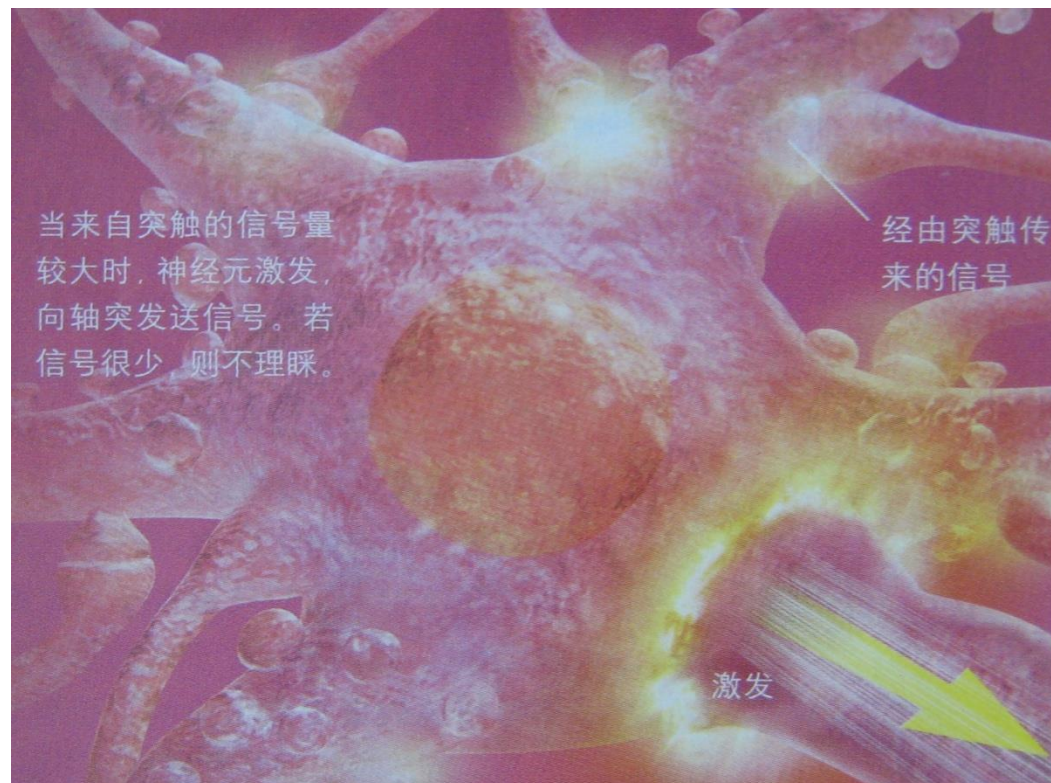


神经元之间的连接（突触传递）



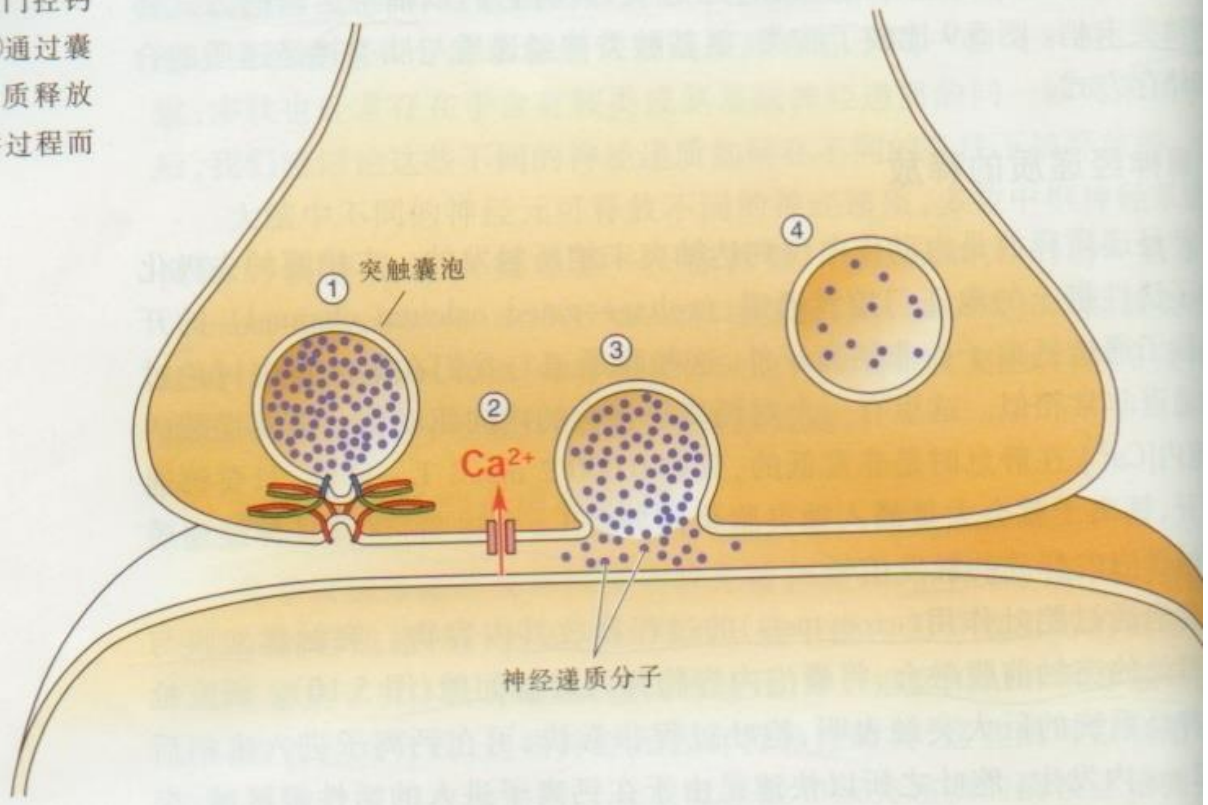
神经元之间的信号传递

- 将来自各个突触信号累加起来，到一定量激发，向轴突发送信号。



神经元之间的信号传递

神经递质通过胞吐释放。①内含神经递质的突触囊泡；②通过电压门控钙通道进入的 Ca^{2+} 触发囊泡反应；③通过囊泡膜与突触前膜融合而将神经递质释放到突触间隙；④囊泡最终通过胞吞过程而被循环利用。



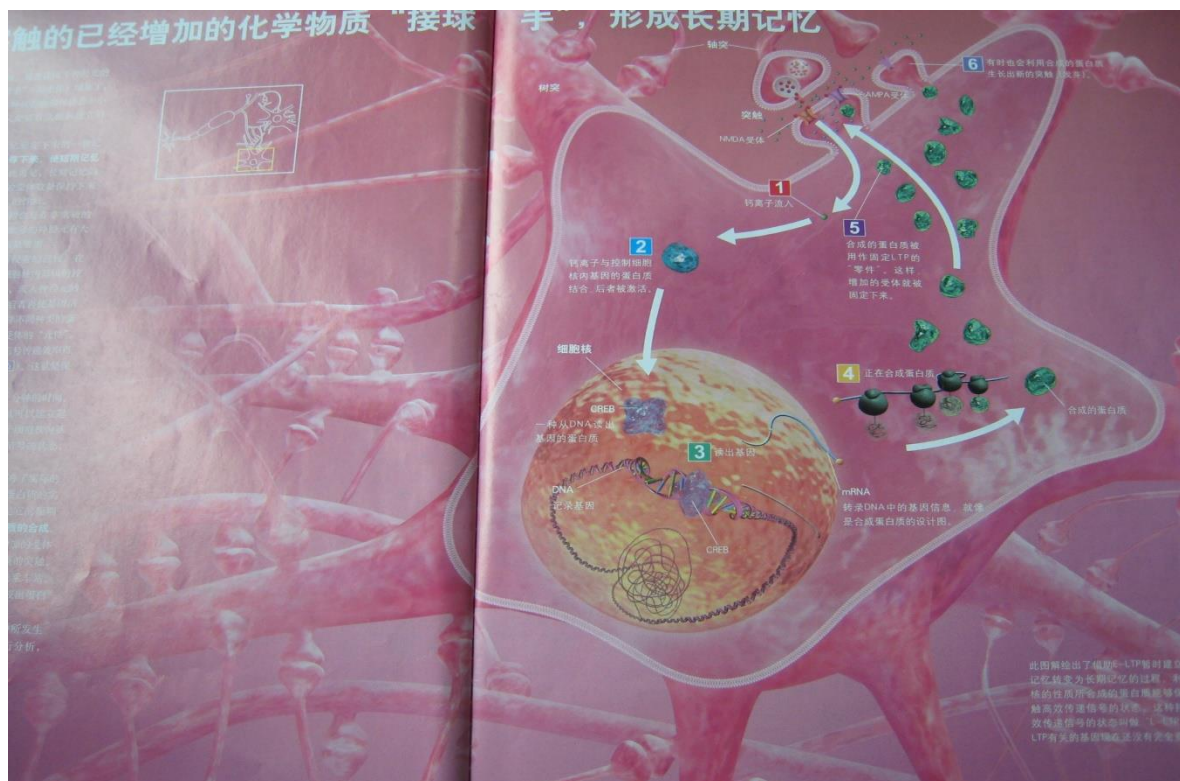
神经元之间的信号传递

- 受体数量增加，突触的信号传递效率提高



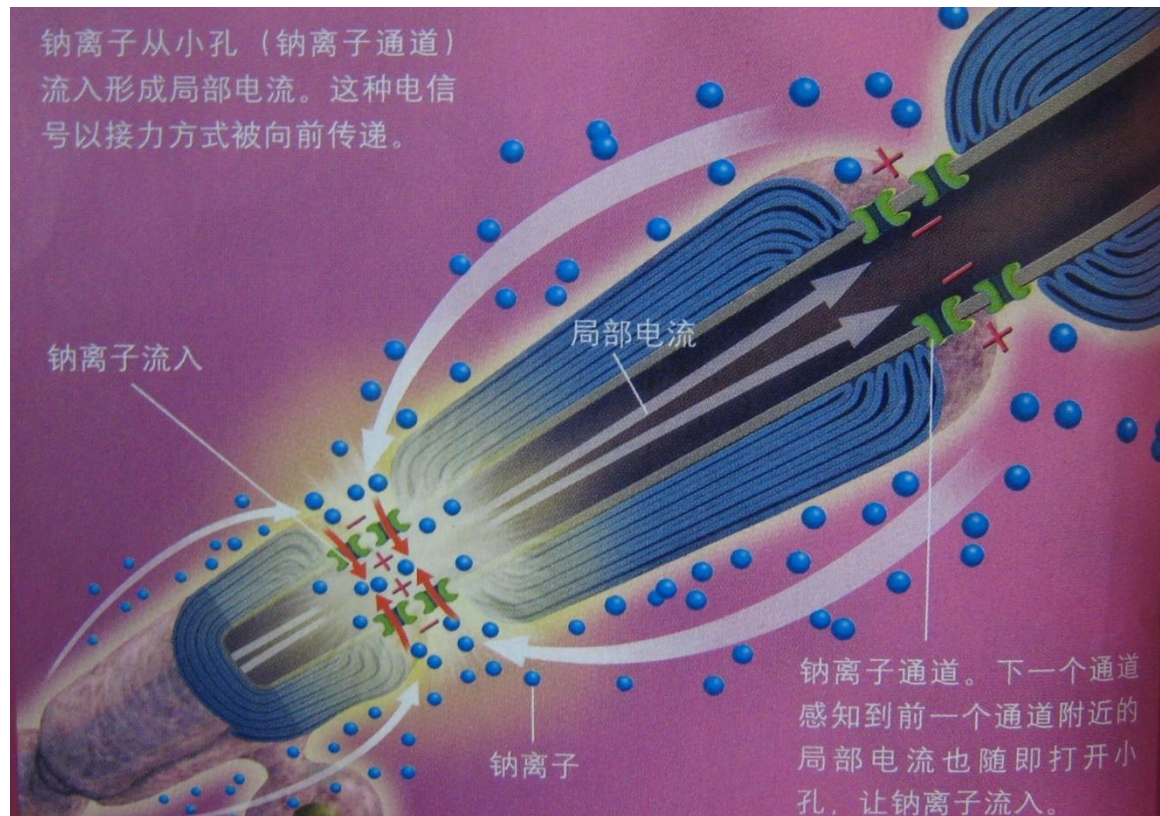
神经元之间的信号传递

- 合成出蛋白质，使信号高效传递的状态得到固定



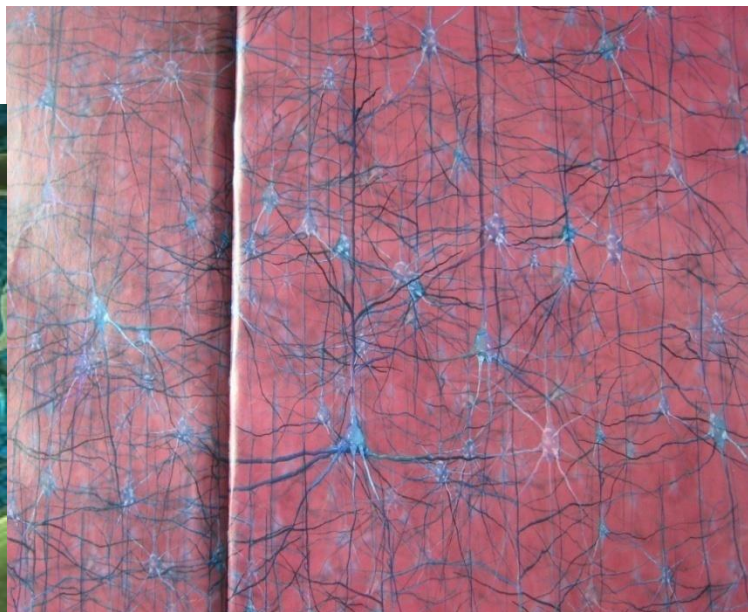
神经元之间的信号传递

- 沿着轴突传送的电信号被传递至成千上万个突触

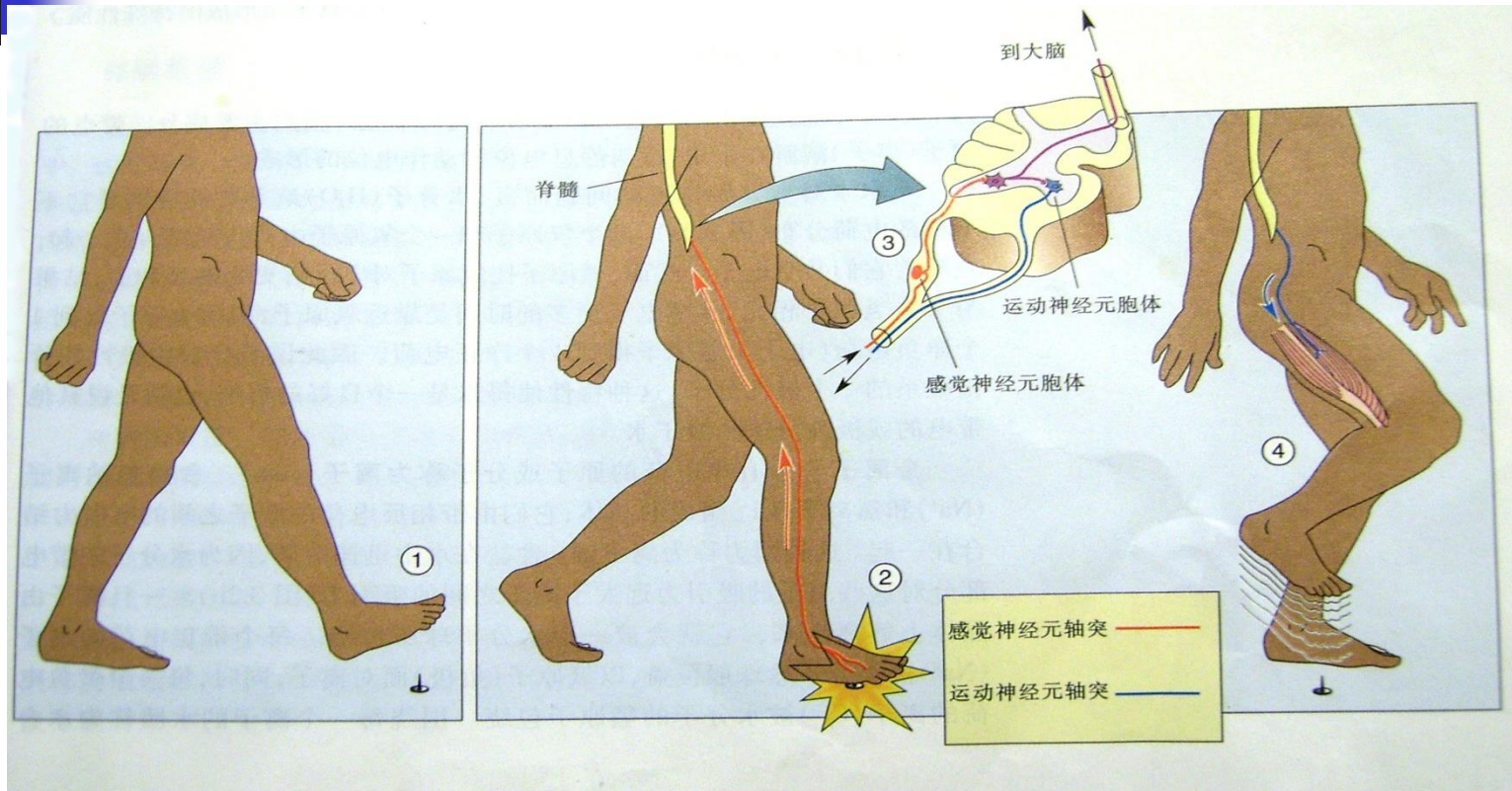


生物神经网络

- 无数的神经元通过突触连接构成神经网络



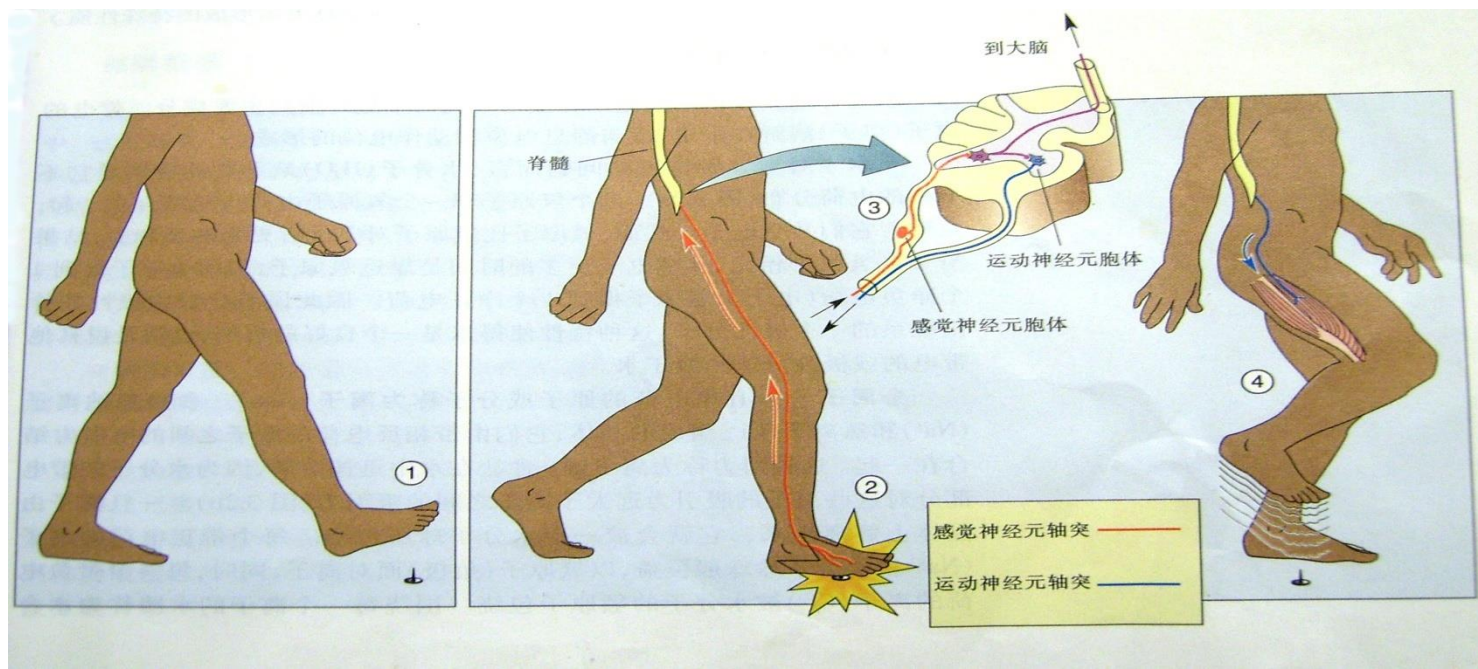
一个简单的反射现象



①踩上图钉 ②刺激产生冲动并传导 ③信息经中间神经元传给运动神经元、大脑 ④指令导致肌肉收缩

单个动作电位的产生

1. 图钉刺入皮肤
2. 皮肤中的神经纤维膜被牵张
3. 门控钠通道开放
4. 钠离子进入细胞内，使膜去极化
5. 膜电位去极化超过**阈值**（膜去极化到能引起动作电位的临界值），神经元膜上便产生动作电位





神经网络中的连接

神经元之间的连接是怎样产生的？



神经网络中的连接

- 建立连接





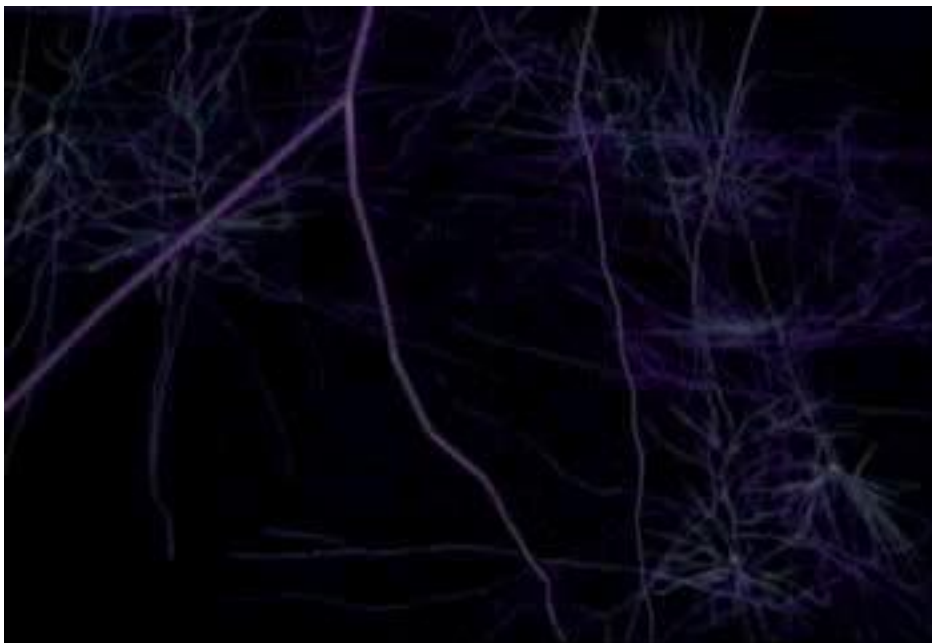
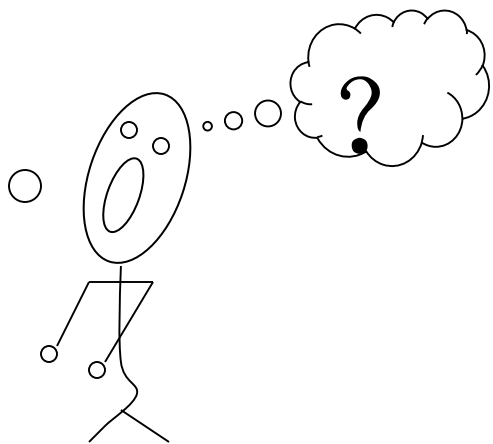
神经网络中的连接

- 生物神经网络中神经元的连接会减少吗？
 - 外因
 - 大脑自然删除



神经网络中的连接

•删除连接!!!



神经网络的特征

- 大脑有约 10^{11} （1000亿）个神经元；
- 每个神经元具有约 10^4 个连接；
- 大脑适于大规模并行计算。

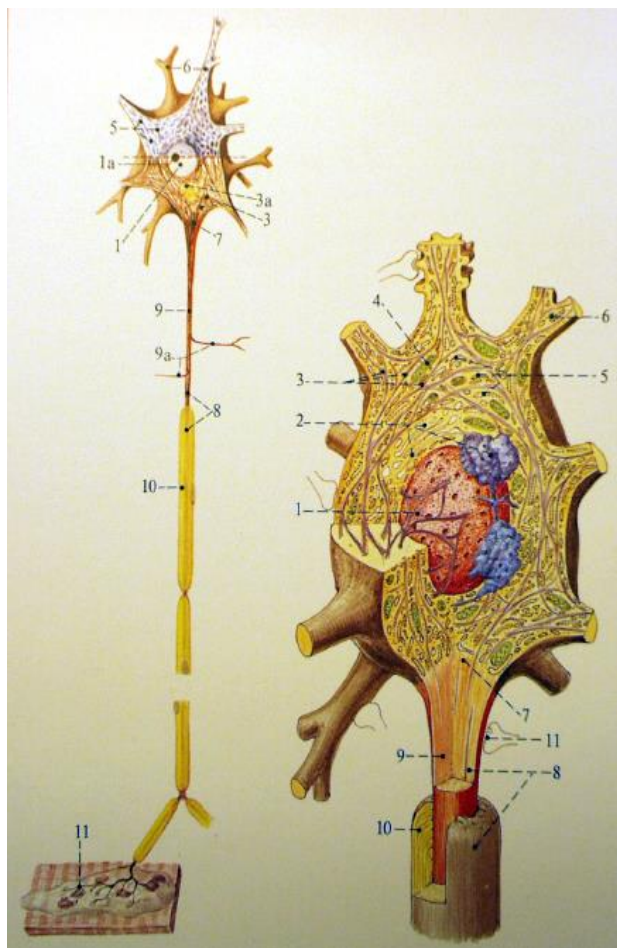




Neuron Model

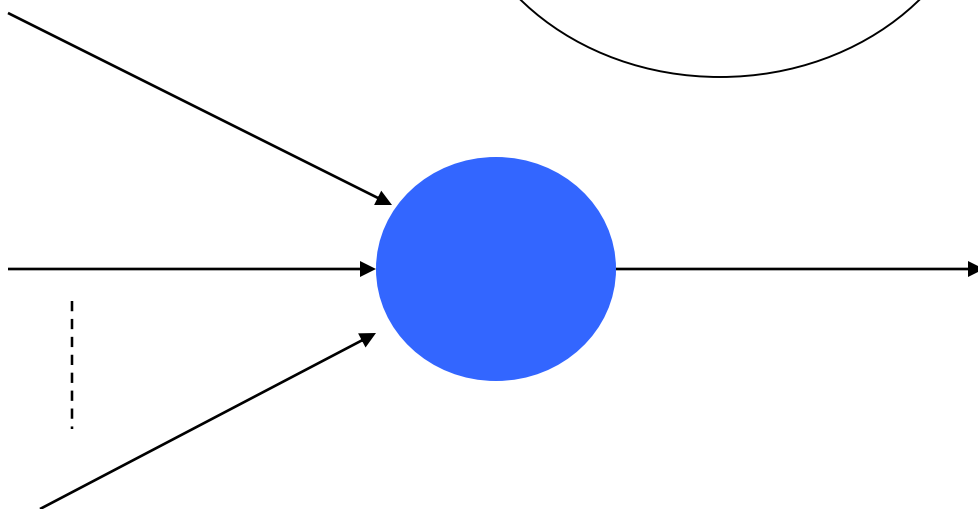
- 生物神经元的基本功能
- 仿生模拟，构建数学模型
- 简单应用

生物神经元拓扑结构

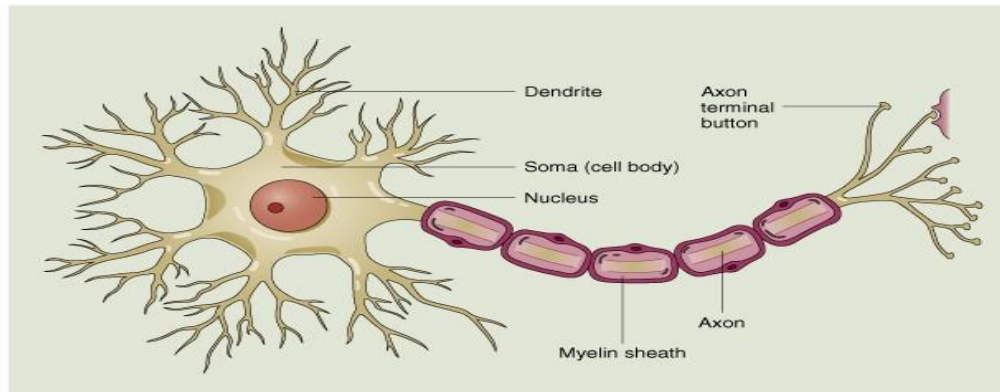


神经元特点

- 多输入
- 单输出



生物神经元

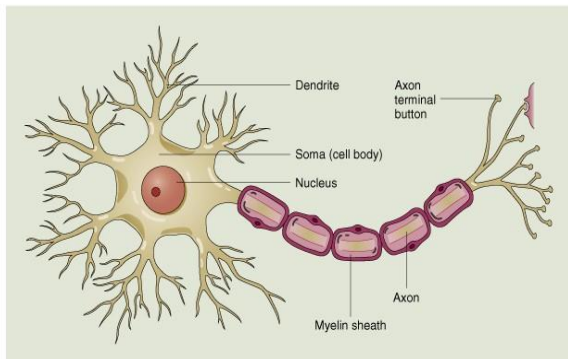


© 2000 John Wiley & Sons, Inc.

神经元特点

- 树突接收多个输入
- 胞体对输入信息叠加
- 当信息叠加到一定程度时产生脉冲
- 单输出

生物神经元



© 2000 John Wiley & Sons, Inc.

- 树突接收多个输入
- 胞体对输入信息叠加
- 当信息叠加到一定程度时产生脉冲
- 单输出

神经元功能部分划分

神经元输入

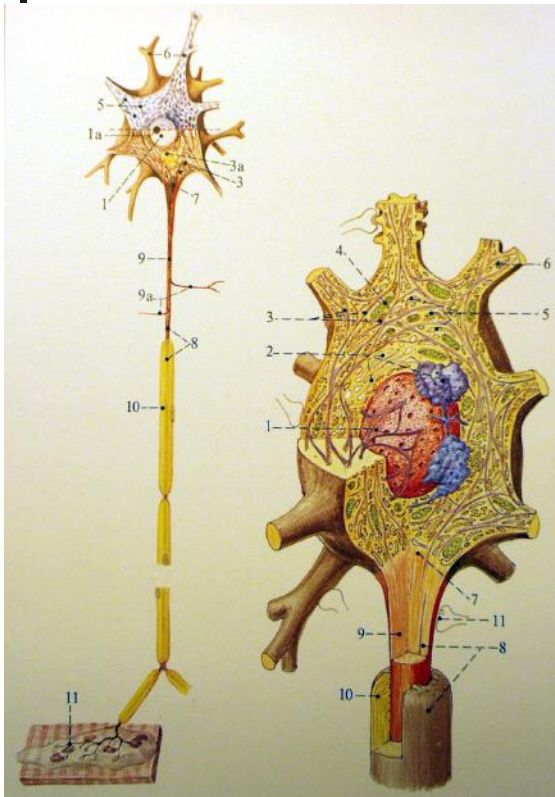


神经元胞体



神经元输出

生物神经元



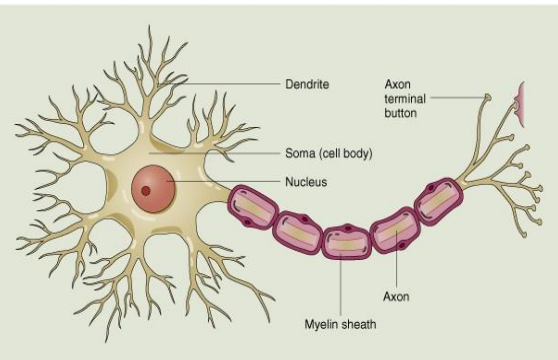
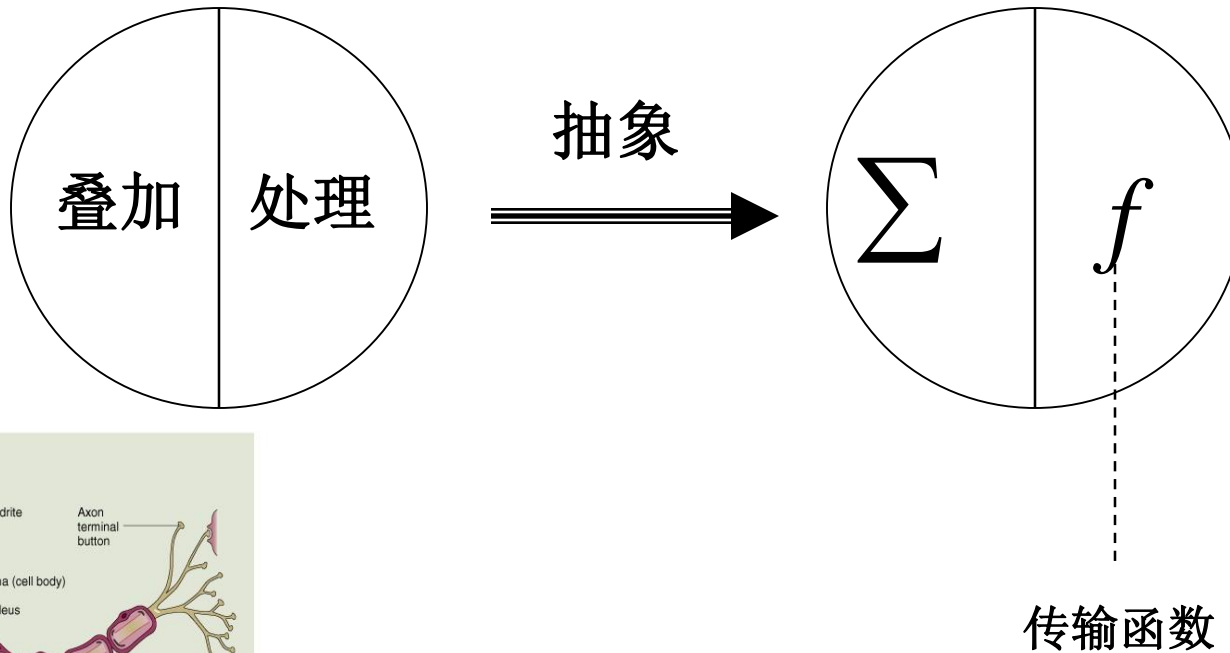
怎样抽象？



数学模型

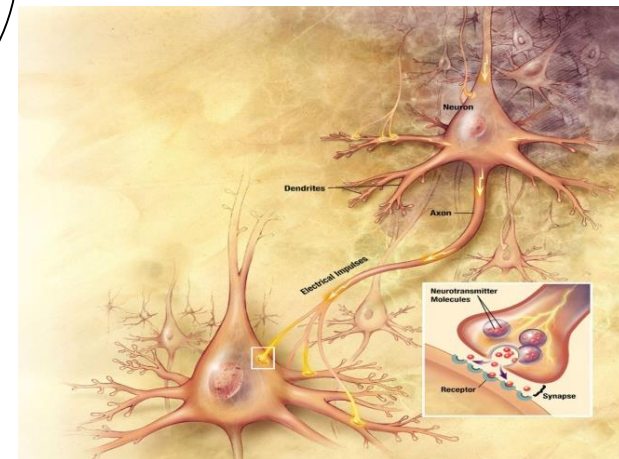
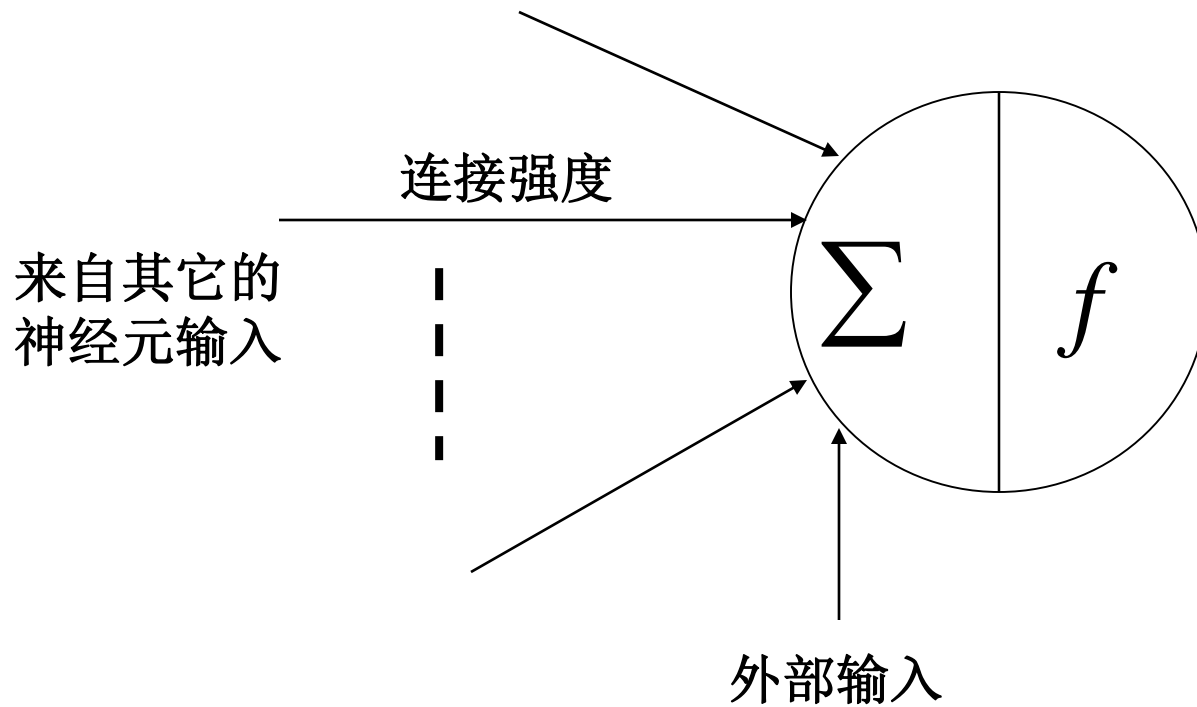
神经元数学模型

神经元胞体模型

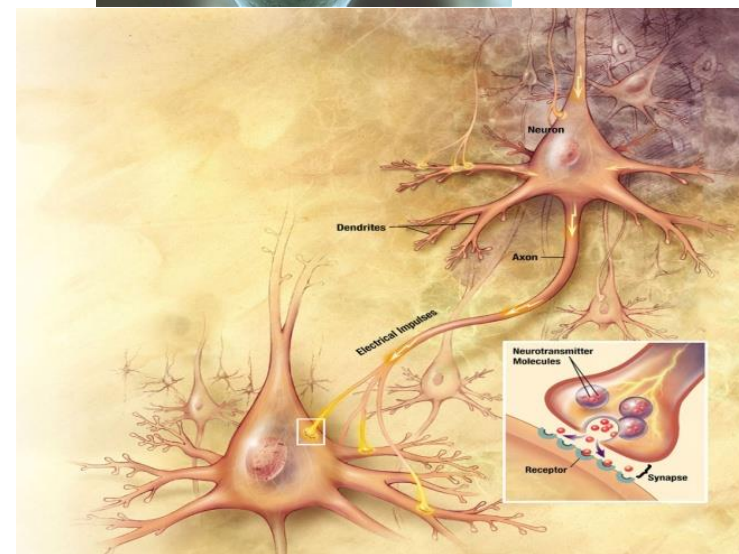
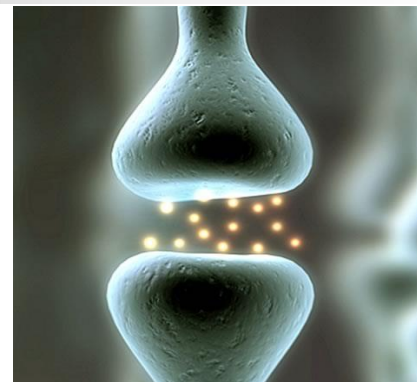
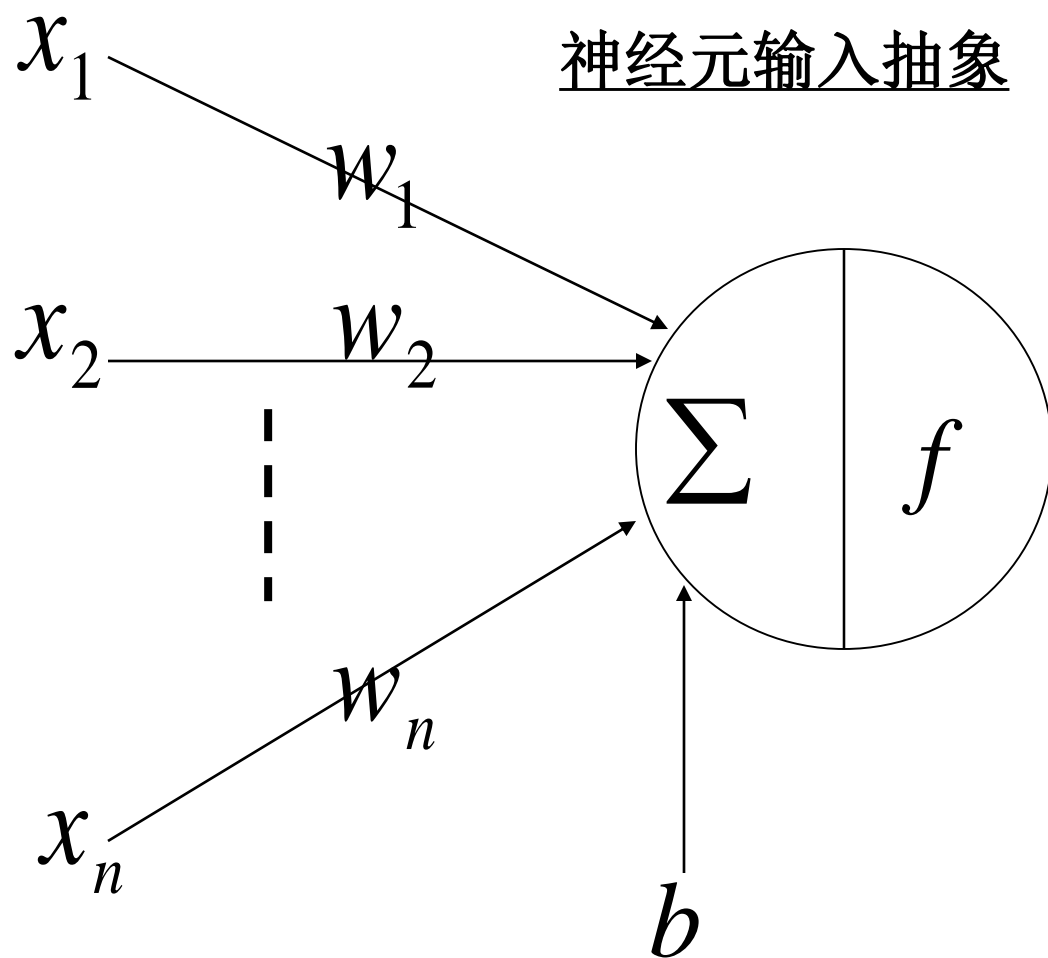


神经元数学模型

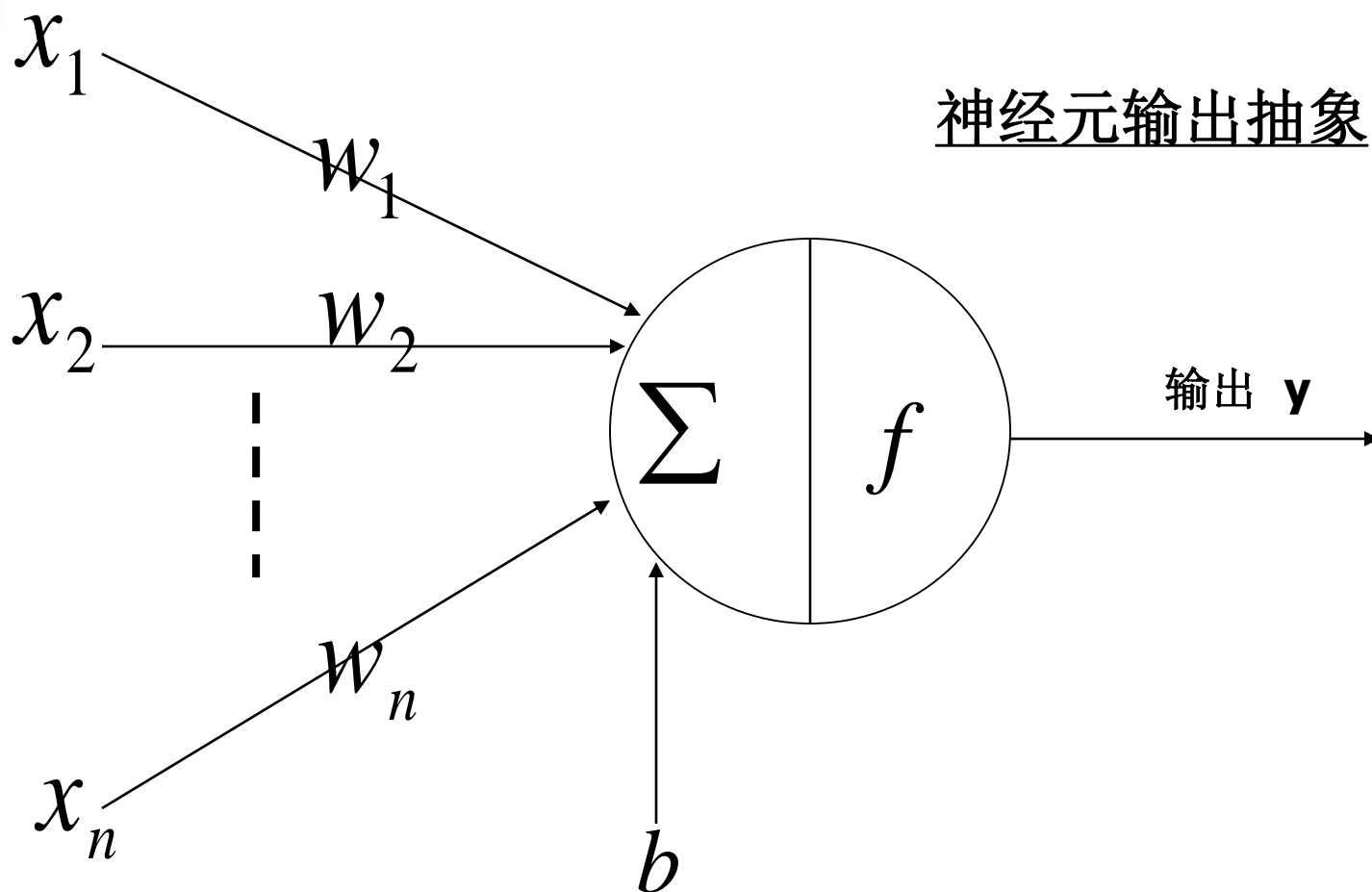
神经元输入抽象



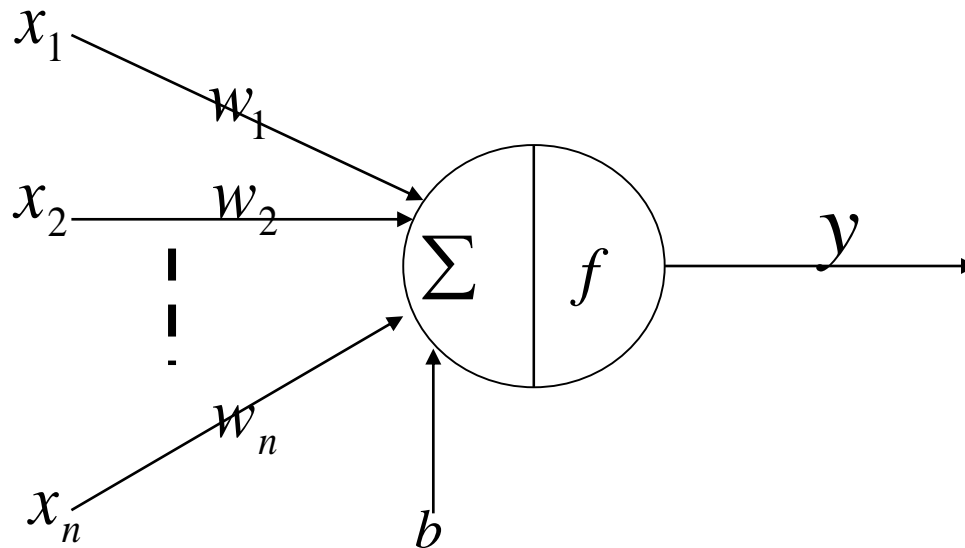
神经元数学模型



神经元数学模型

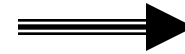
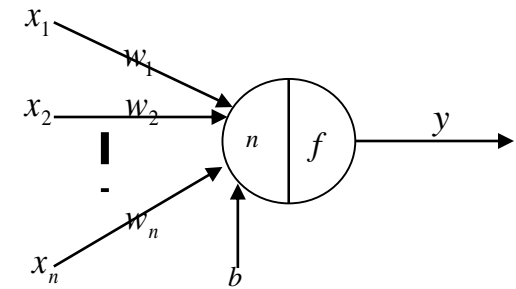
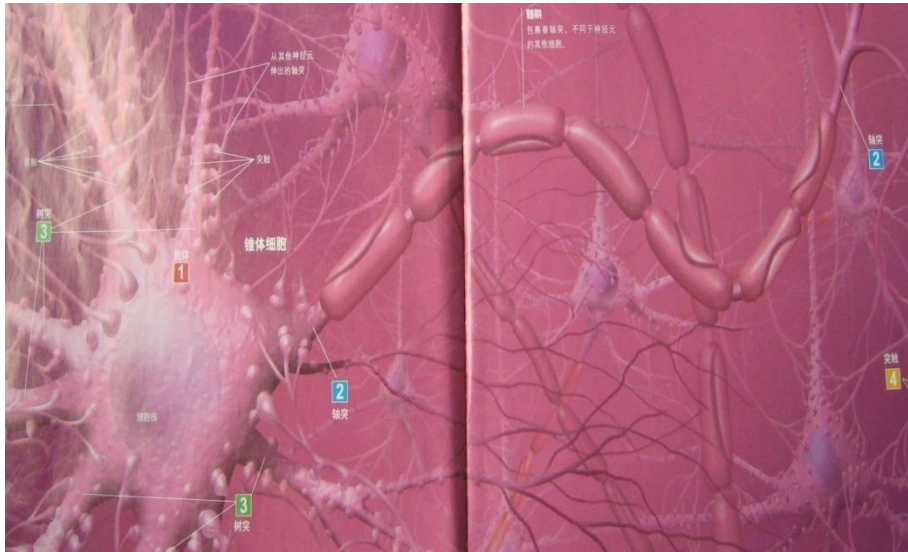


神经元数学模型



$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

神经元数学模型

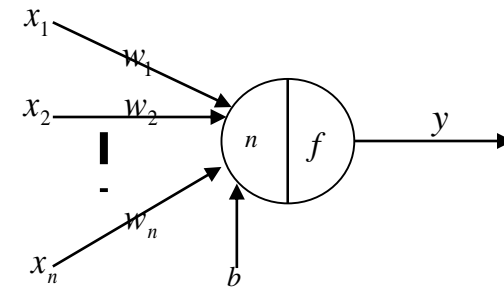
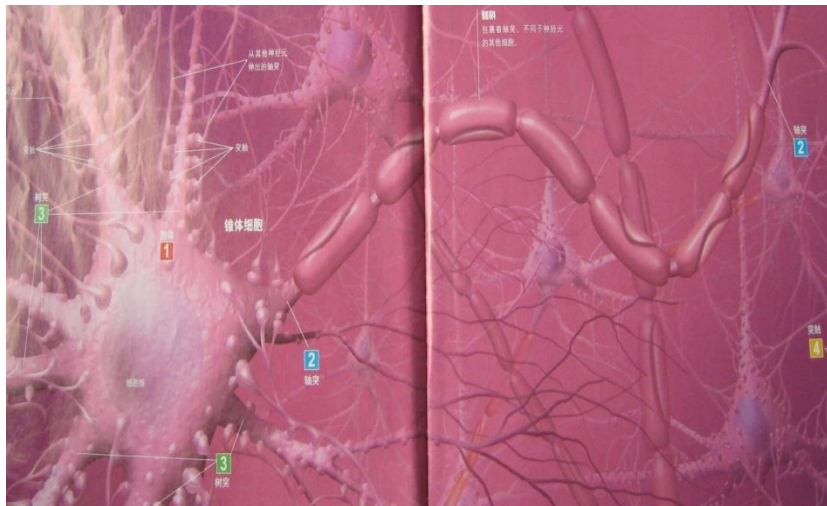


$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

内部输入

$$n = \sum_{i=1}^n w_i x_i + b$$

神经元数学模型

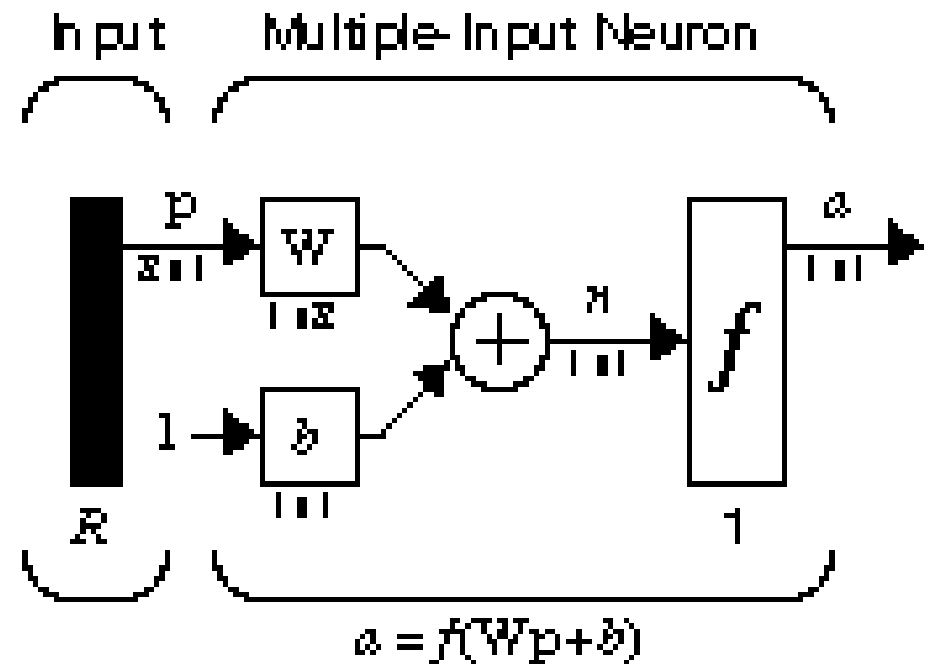
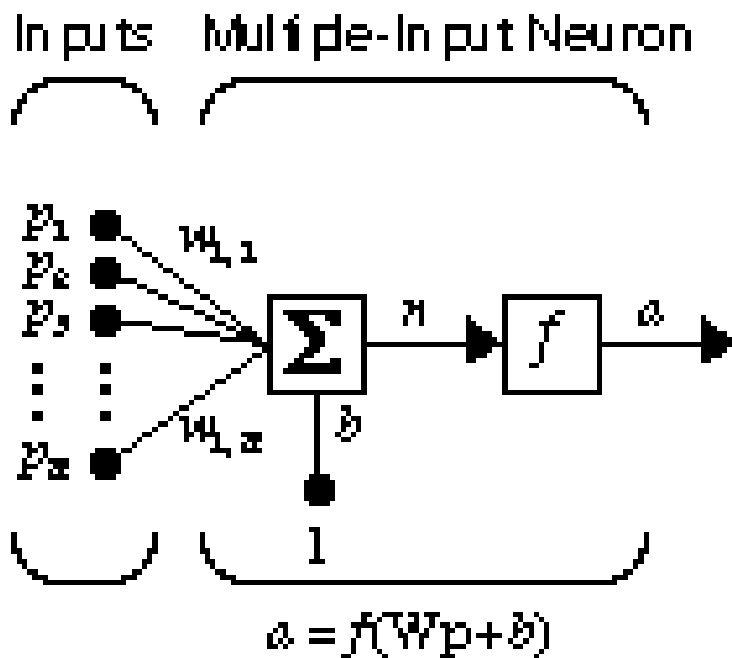


$$y = f(Wx + b)$$

内部输入

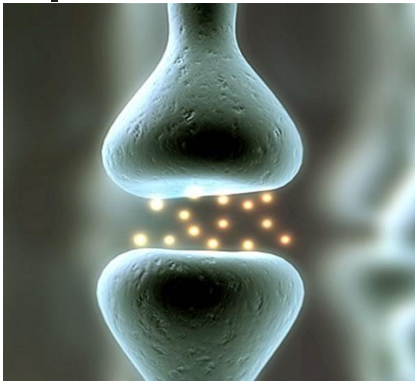
$$n = Wx + b$$

神经元数学模型

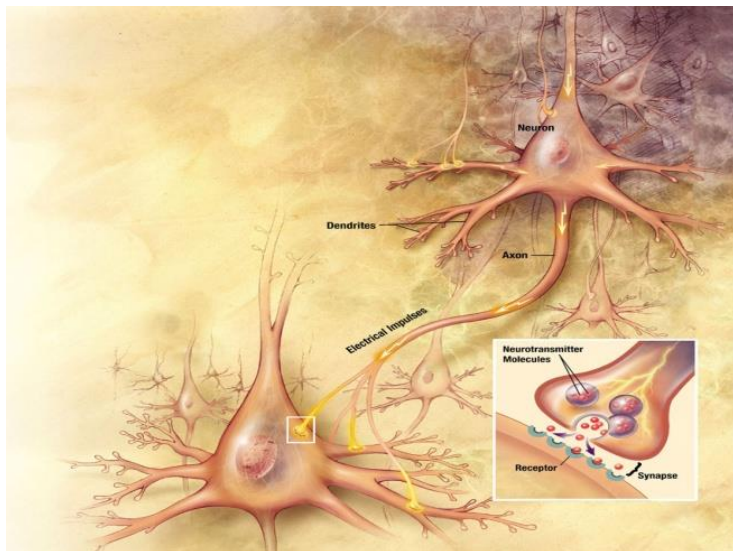


Abbreviated Notation

神经元之间的连接权



$w_{ij} \begin{cases} > 0, & \text{激励连接} \\ < 0, & \text{抑制连接} \end{cases}$

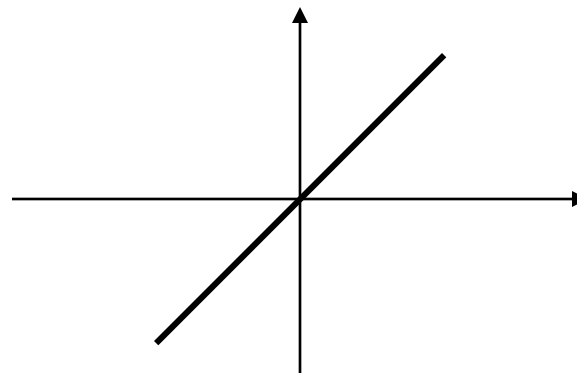
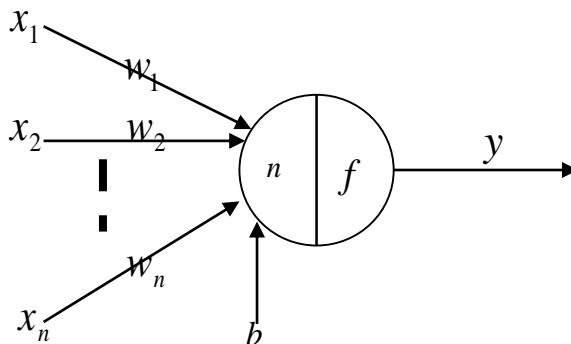


$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

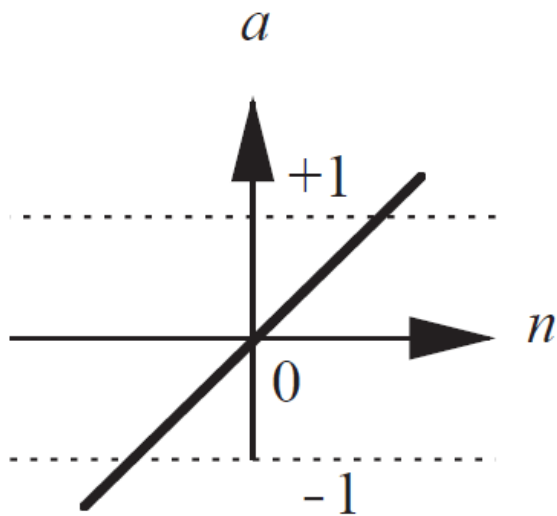
线性传输函数(purelin)

$$f(x) = x$$

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

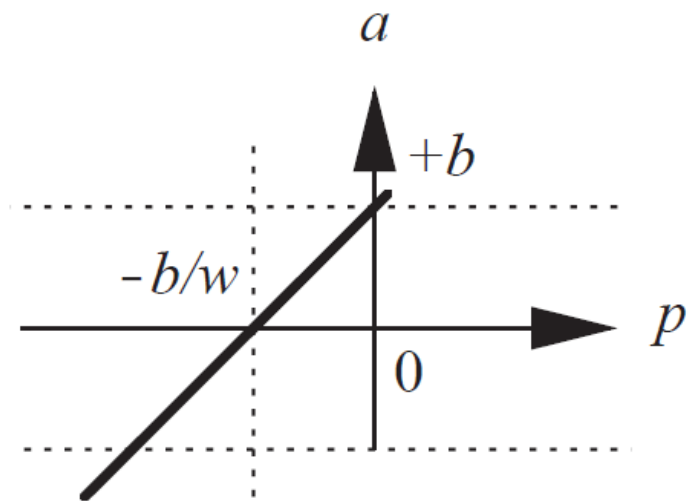


线性传输函数



$$a = \text{purelin}(n)$$

Linear Transfer Function

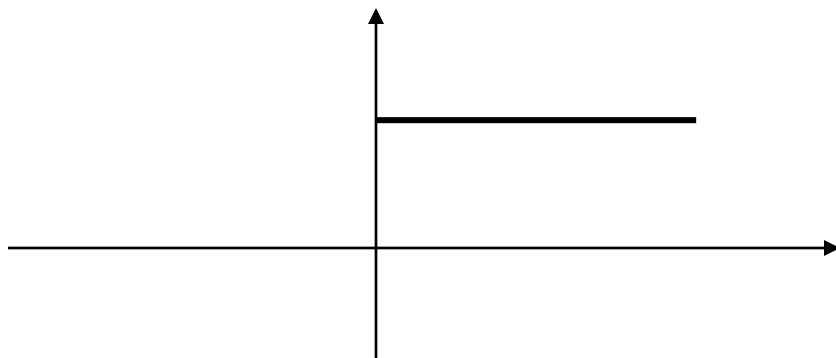
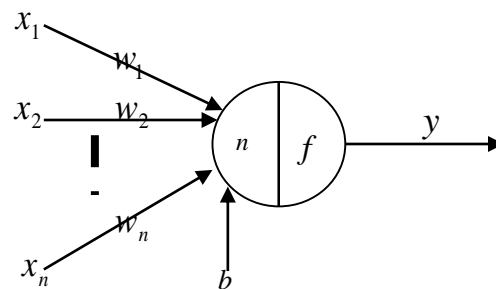


$$a = \text{purelin}(wp + b)$$

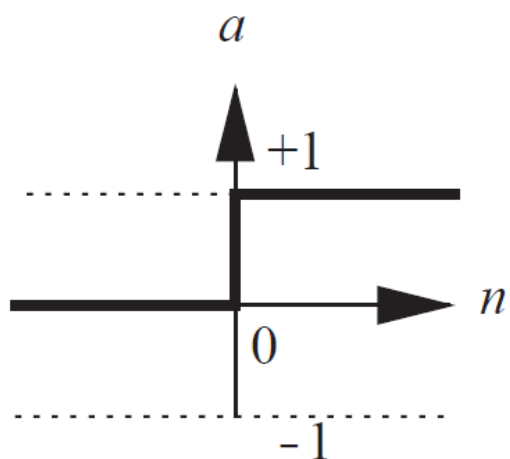
Single-Input *purelin* Neuron

Hardlim传输函数（硬极限函数）

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

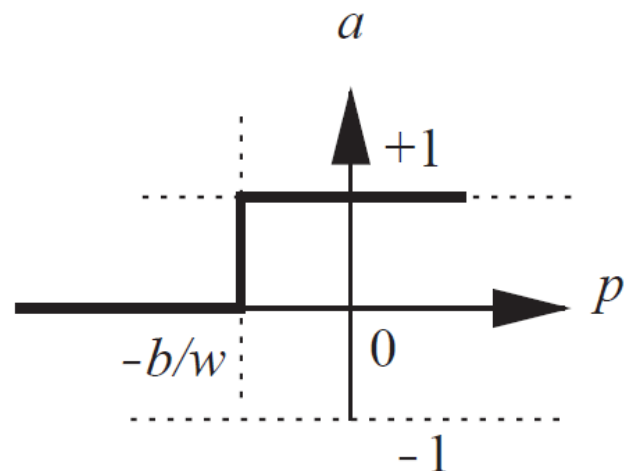


Hardlim传输函数



$$a = \text{hardlim}(n)$$

Hard Limit Transfer Function

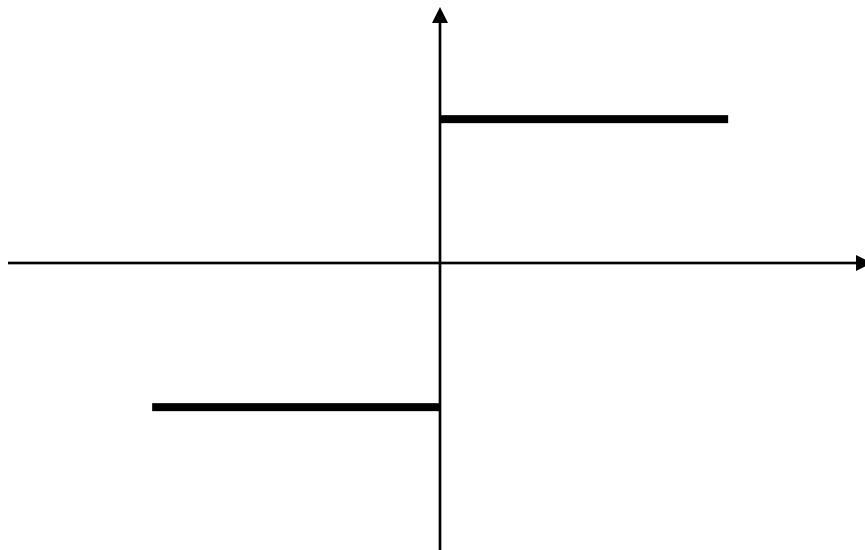
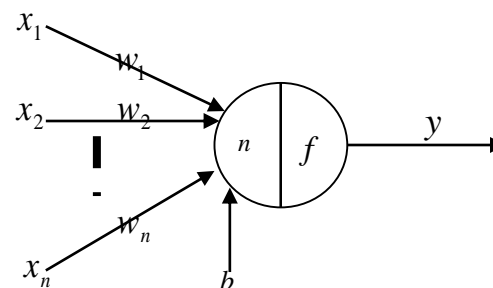


$$a = \text{hardlim}(wp + b)$$

Single-Input hardlim Neuron

Hardlims传输函数（对称硬极限函数）

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

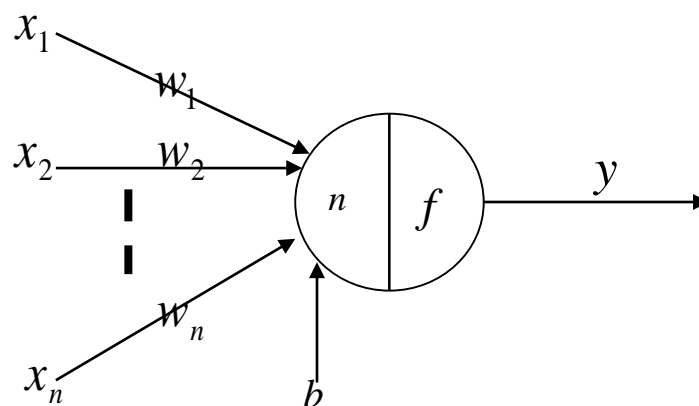




Sigmoid传输函数（对数-S形）

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

$$n = \sum_{i=1}^n w_i x_i + b$$

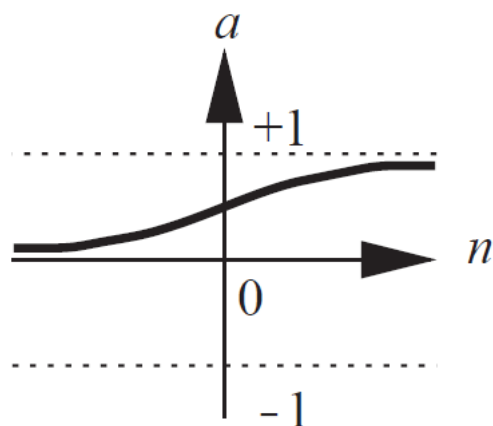


$$e \approx 2.7183$$

Sigmoid传输函数（对数-S形）

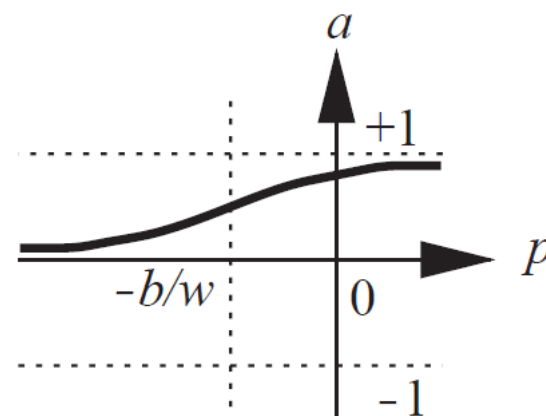
$$f(n) = \frac{1}{1 + e^{-n}}$$

$$n = \sum_{i=1}^n w_i x_i + b$$



$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function

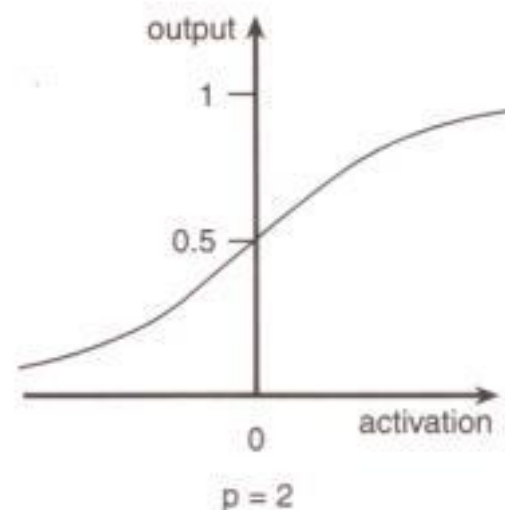
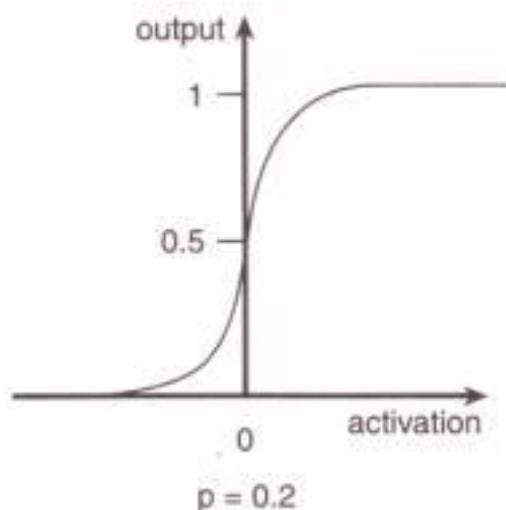


$$a = \text{logsig}(wp + b)$$

Single-Input *logsig* Neuron

Sigmoid传输函数（对数-S形）

$$f(n) = \frac{1}{1 + e^{-n/p}}$$



- **p**: 控制曲线形状变化快慢或陡峭程度，常设为**1**。**p**值越大，曲线越平坦，反之曲线越陡峭。**p**值足够小，就近似于**Hardlim**函数。



其它传输函数

- **tansig** 正切S型传递函数;
- **satlin** 饱和线性传递函数;
- **satlins** 对称饱和线性传递函数;
- **radbas** 径向基传递函数;
- **dist** 计算矢量间的距离;
- **compet** 自组织映射传递函数;
- **dpurelin** 线性传递函数的导数;
- **dtansig** 正切S型传递函数的导数;
- **dlogsig** 对数S型传递函数的导数

.....



Neuron Model

- **生物神经元的基本功能**
- **仿生模拟，构建数学模型**
- 简单应用



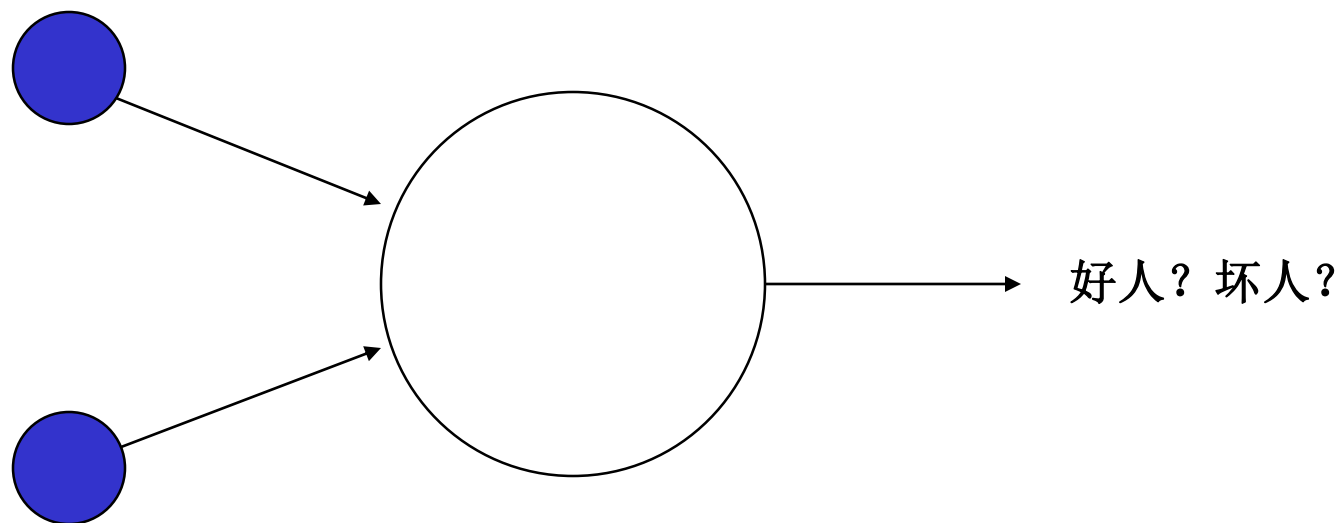
一个幼稚的例子

问题：设计一个神经元识别好人坏人

- 好人特征：眼睛慈善，面带笑容
- 坏人特征：目露凶光，面无表情

一个幼稚的例子

- 问题：设计一个神经元识别好人坏人
 - 好人特征：眼光慈善，面带笑容
 - 坏人特征：目露凶光，面无表情





一个幼稚的例子

- 问题：设计一个神经元识别好人坏人
 - 好人特征：眼光慈善，面带笑容
 - 坏人特征：目露凶光，面无表情

眼光

面部表情

眼光慈善： **1**

面带笑容： **1**

目露凶光： **-1**

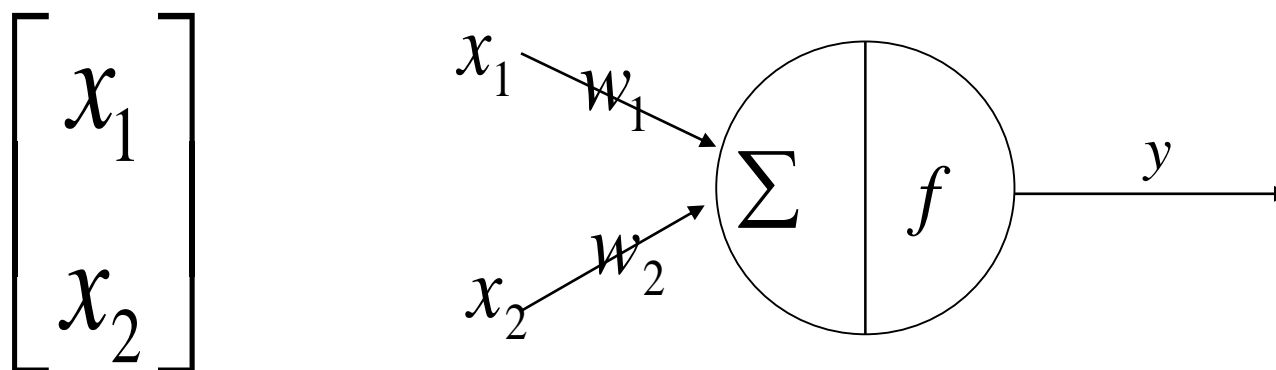
面无表情： **-1**

好人： **1**

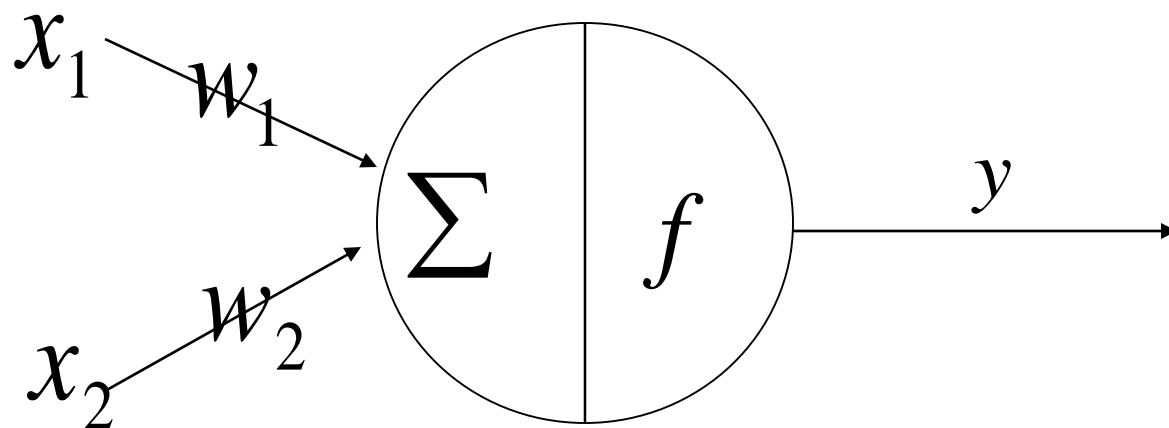
坏人： **-1**

一个例子

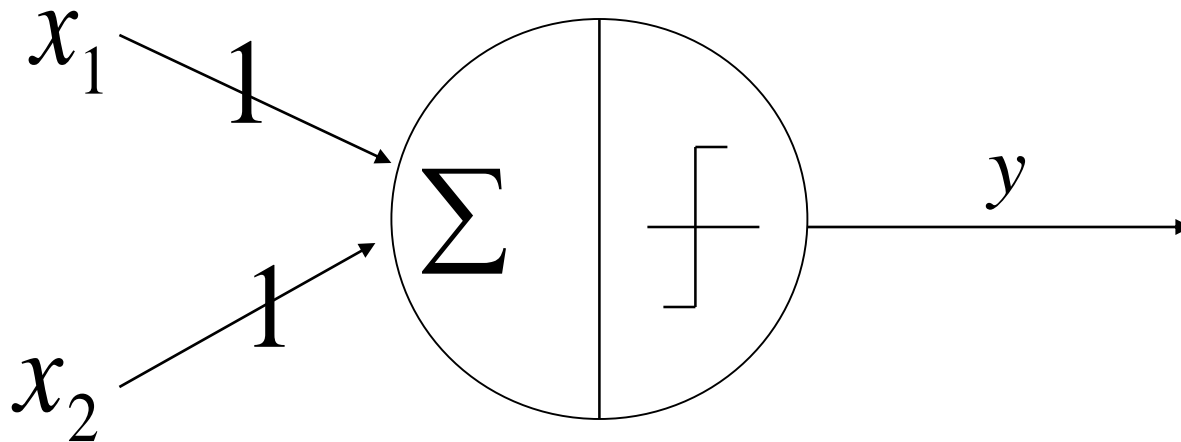
- 问题：设计一个神经元识别好人坏人
 - 好人特征：眼光慈善，面带笑容
 - 坏人特征：目露凶光，面无表情



神经元数学模型



神经元数学模型

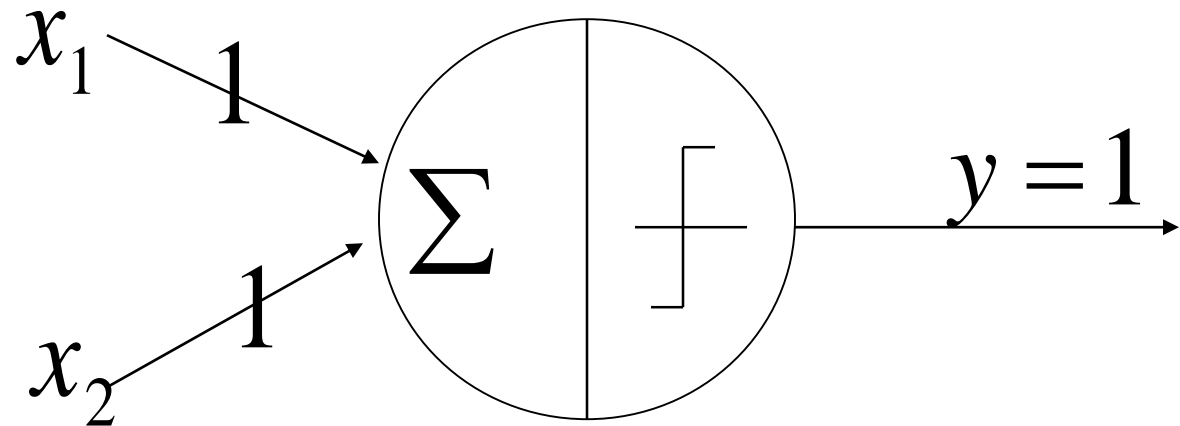


$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

神经元数学模型

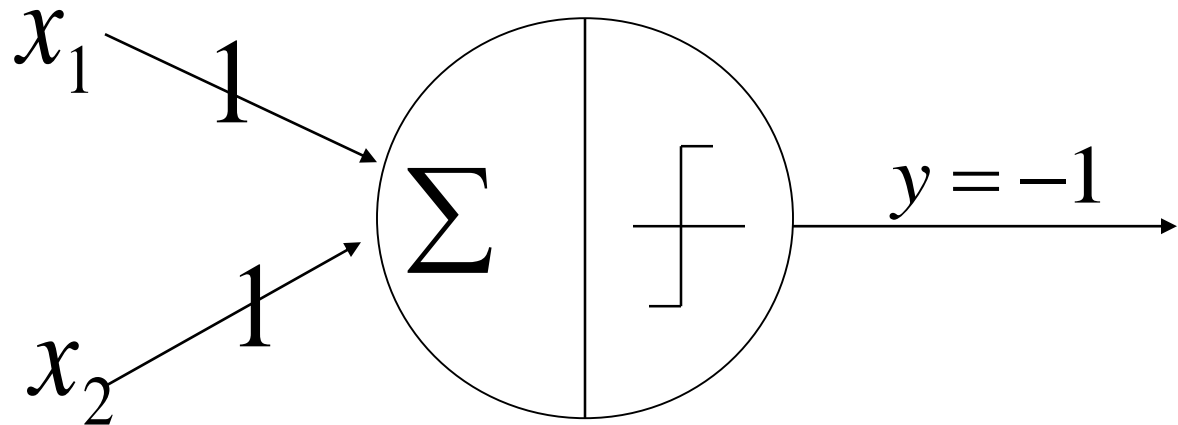
$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

神经元数学模型

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix}$$



$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

神经网络结构



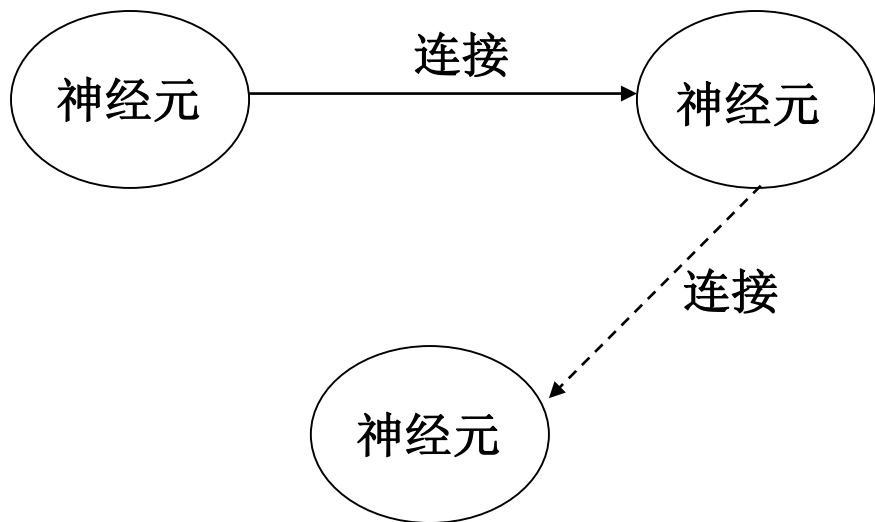


神经网络

神经网络=神经元+连接

神经网络分类

- 无反馈网络：前馈神经网络
- 有反馈网络：递归神经网络





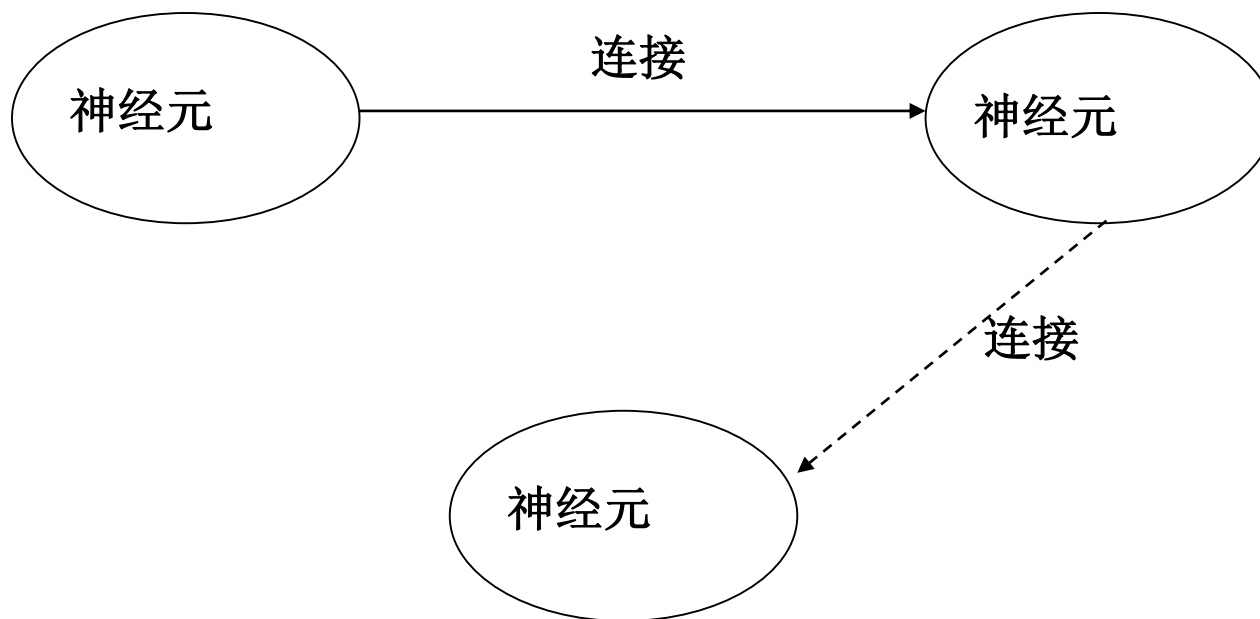
神经网络分类

- **Feed Forward NNs**前馈神经网络:无反馈网络
- **Recurrent NNs**递归神经网络:有反馈网络

前馈神经网络

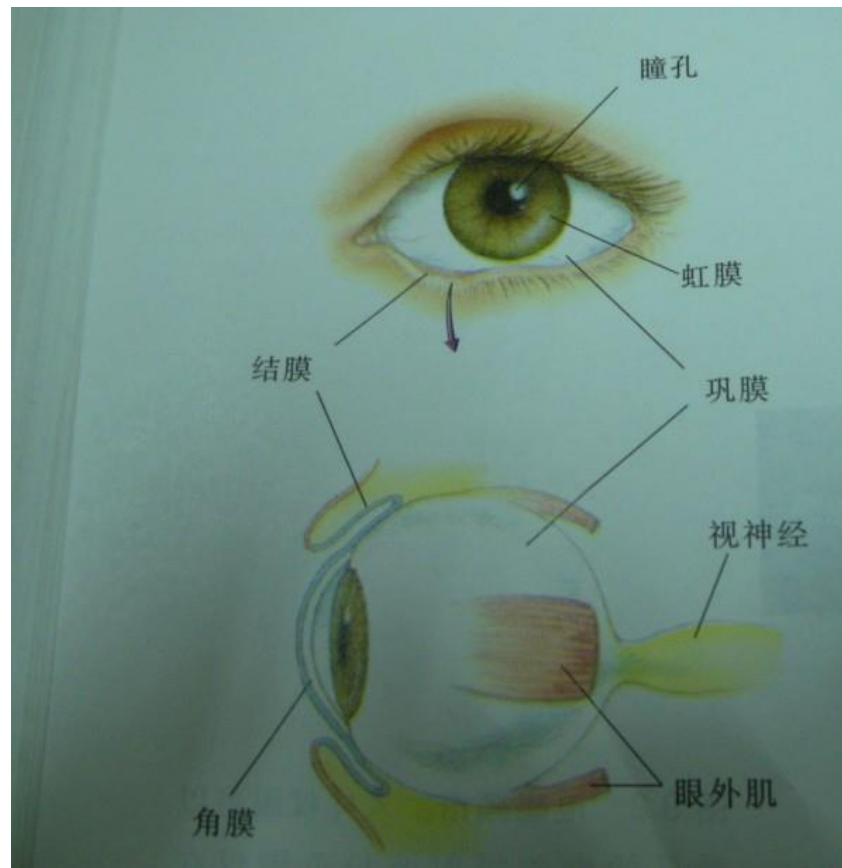
——无反馈的神经网络

- 源于生物视觉系统



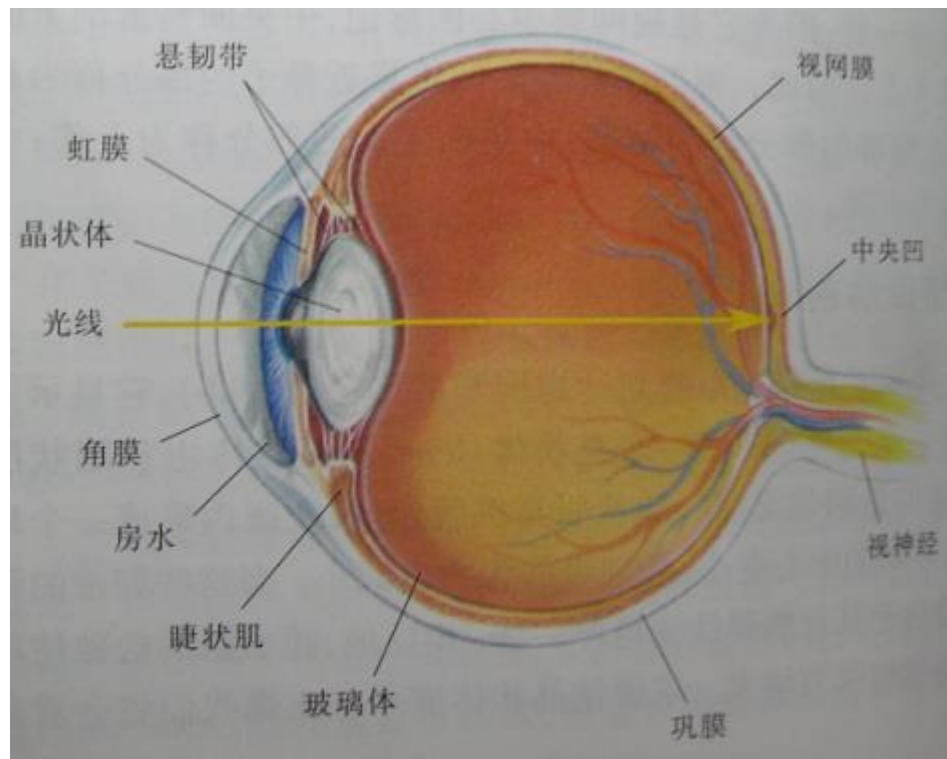
视觉系统

- 眼睛的正面解剖

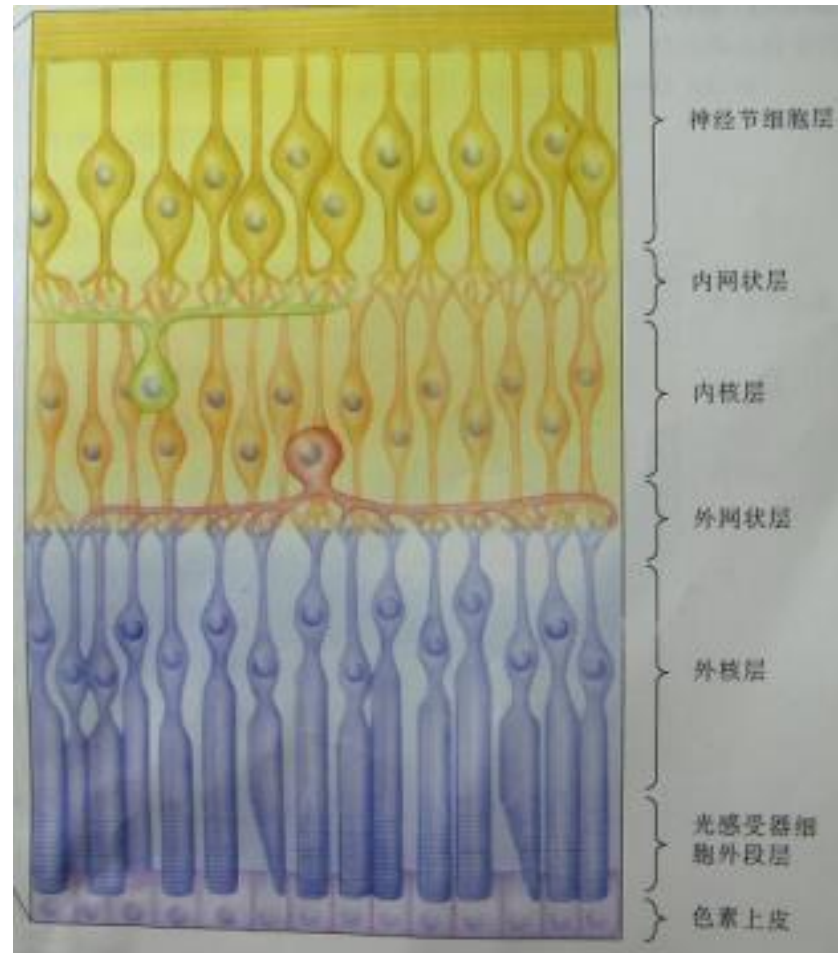


视觉系统

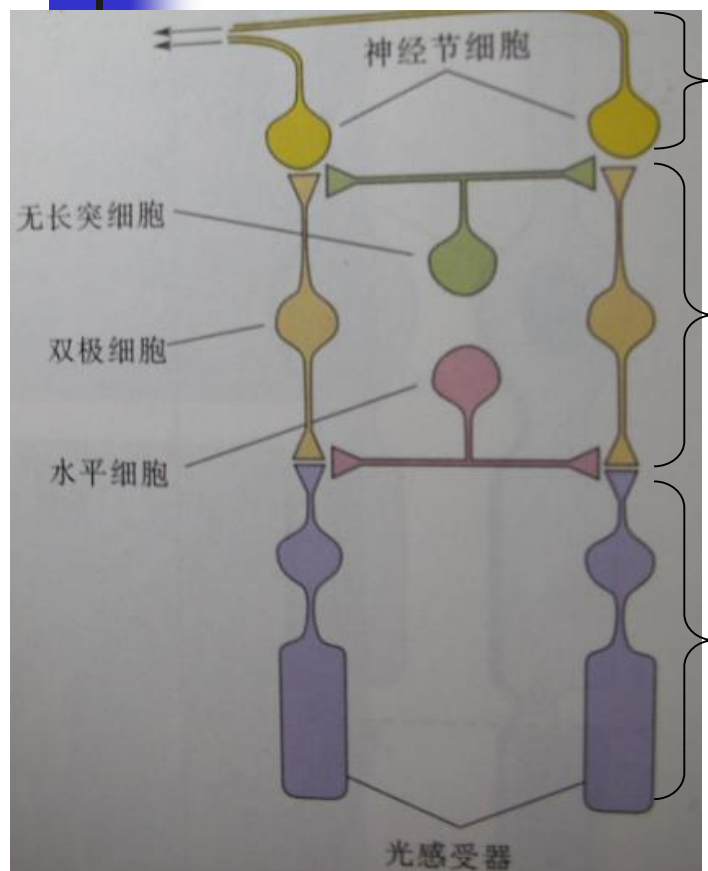
- 眼睛的横切面解剖图



视网膜的分层结构

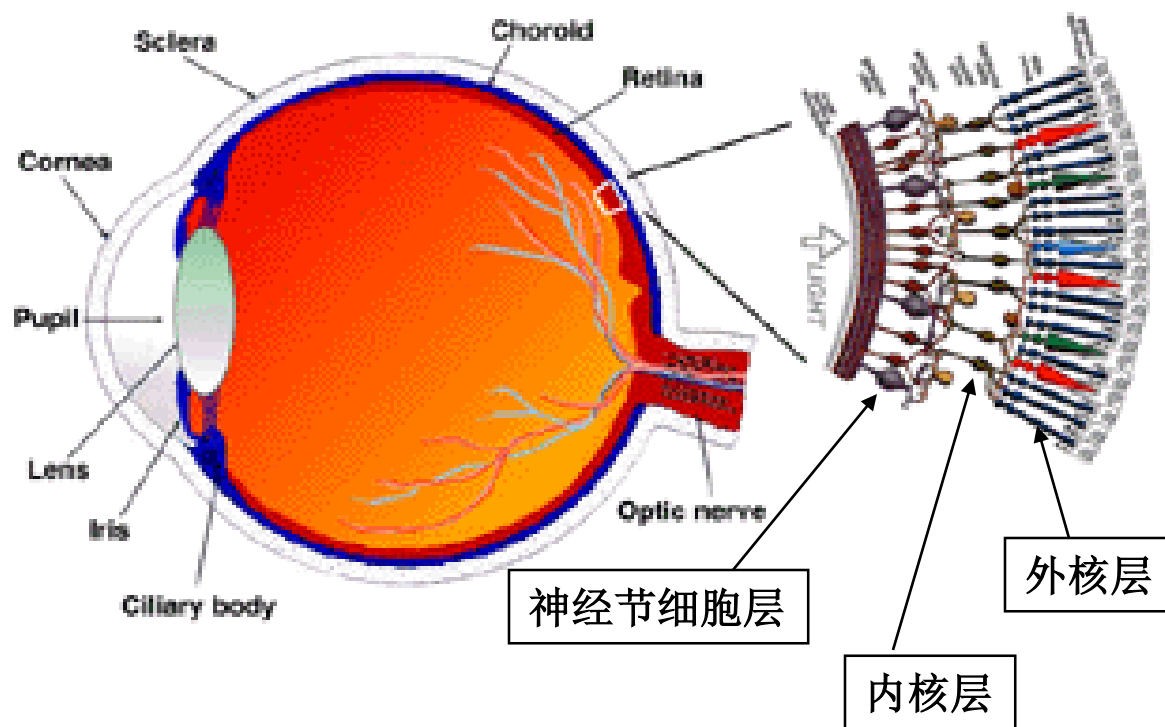


视网膜信息处理的基本系统

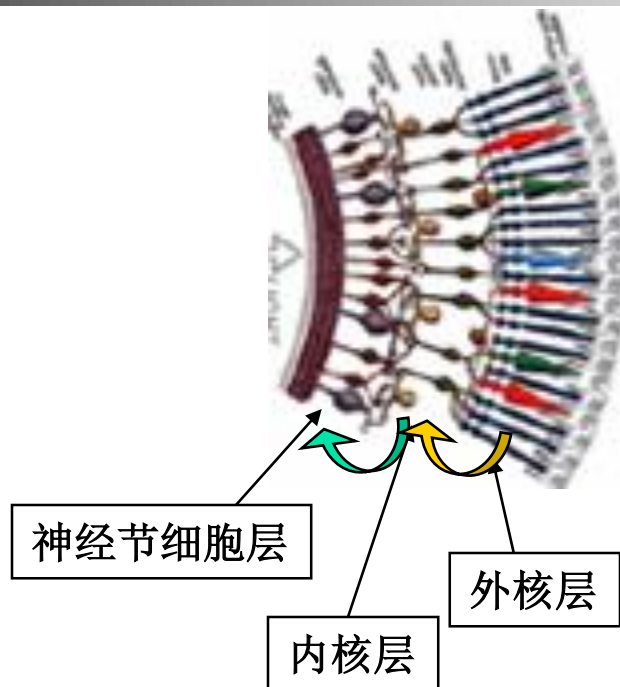


- 视网膜分3层神经细胞（自下而上）：
外层、中间层、最后层
- 光信息自光感受器经双极细胞传至神经节细胞，神经节细胞的轴突汇聚成视神经离开眼球。
- 水平细胞和无长突细胞通过侧向联系调节双极细胞和神经节细胞的反应。

前馈神经网络

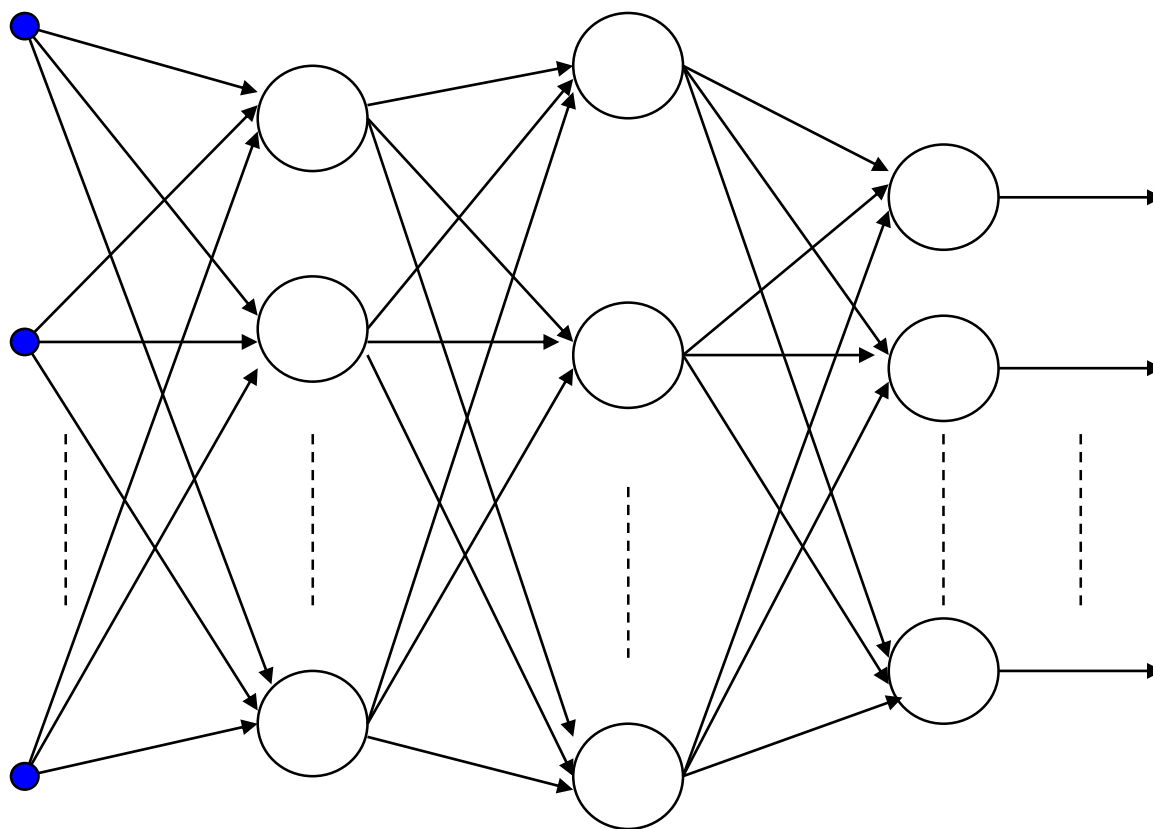


前馈神经网络

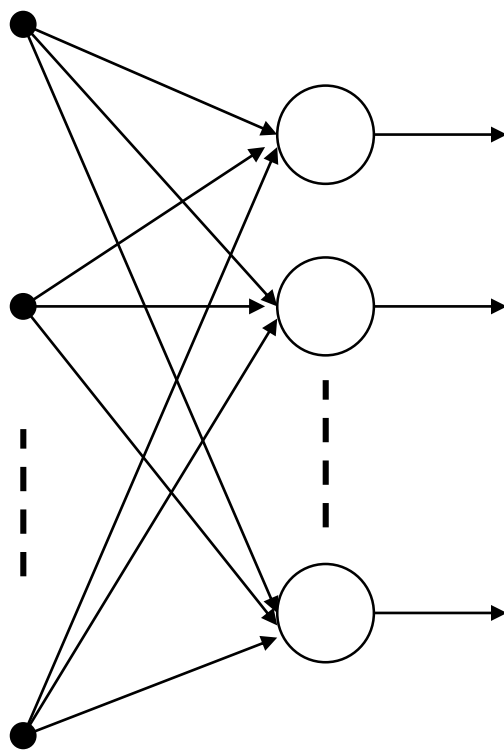


1. 三层神经网络：神经节细胞层-内核层-外核层
2. 每层神经元之间无连接
3. 前一层神经元计算完后传递给下一层神经元进行计算

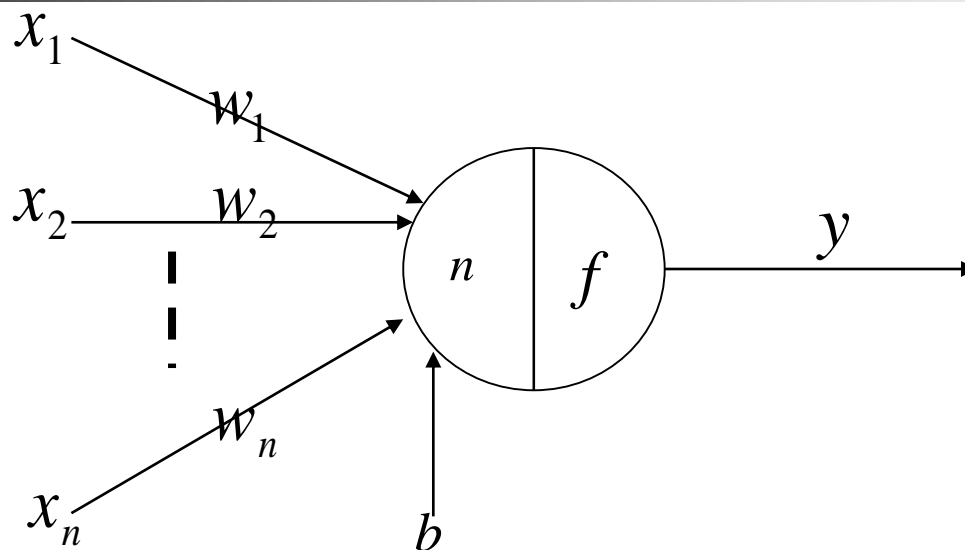
前馈网络拓扑结构



单层网络拓扑结构



神经元数学模型

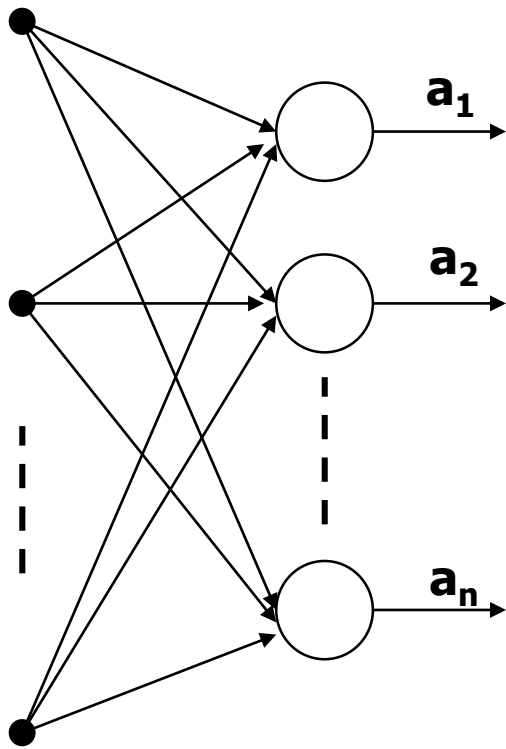


$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

内部输入

$$n = \sum_{i=1}^n w_i x_i + b$$

单层网络拓扑结构



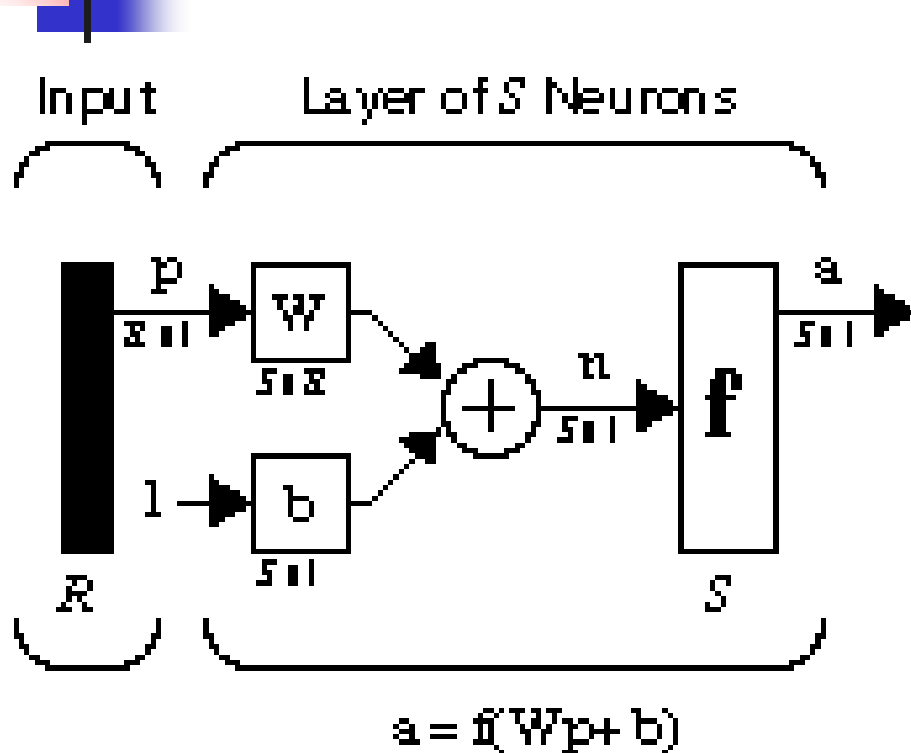
数学模型

$$a_i = f_i \left(\sum_{j=1}^n w_{ij} p_j + b_i \right)$$

$$n_i = \sum_{j=1}^n w_{ij} p_j + b_i$$

$$a = f(Wp + b)$$

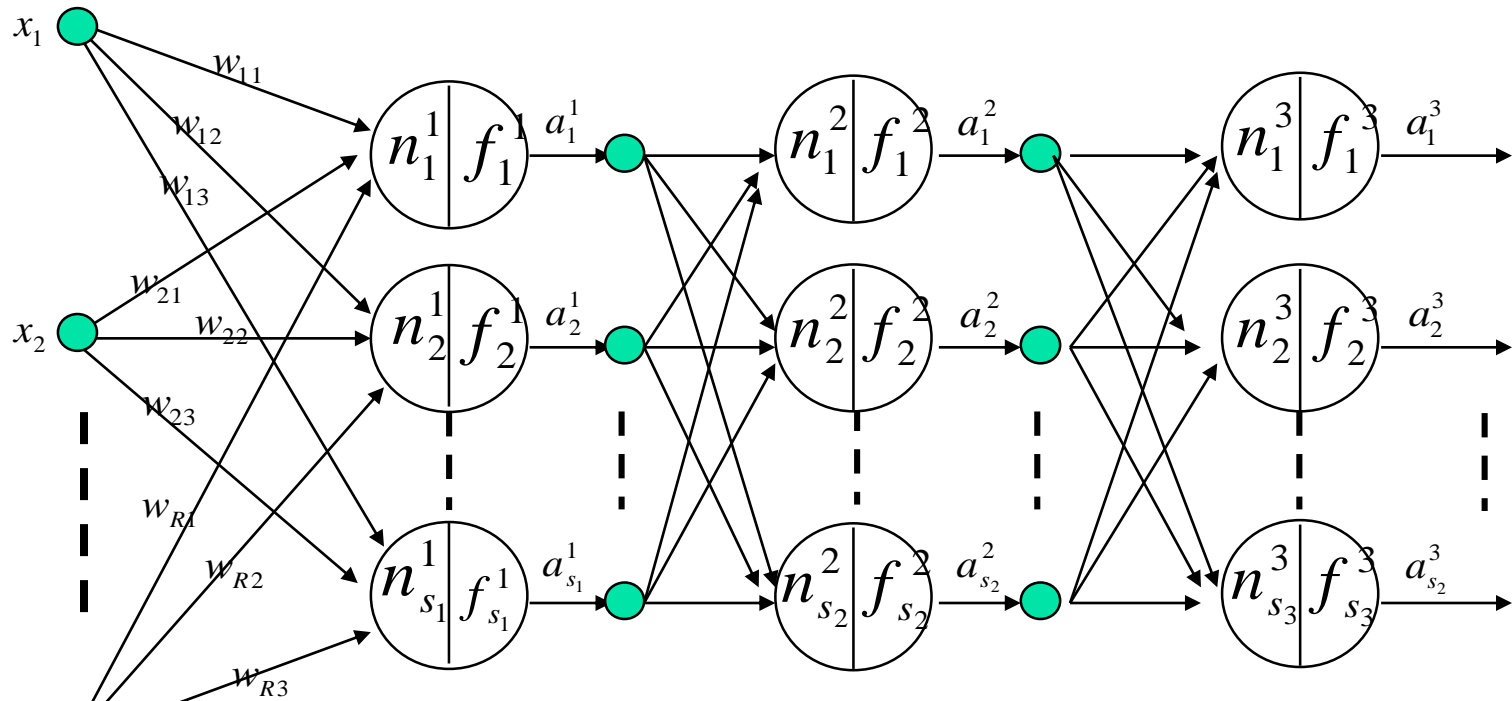
单层网络模型



$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

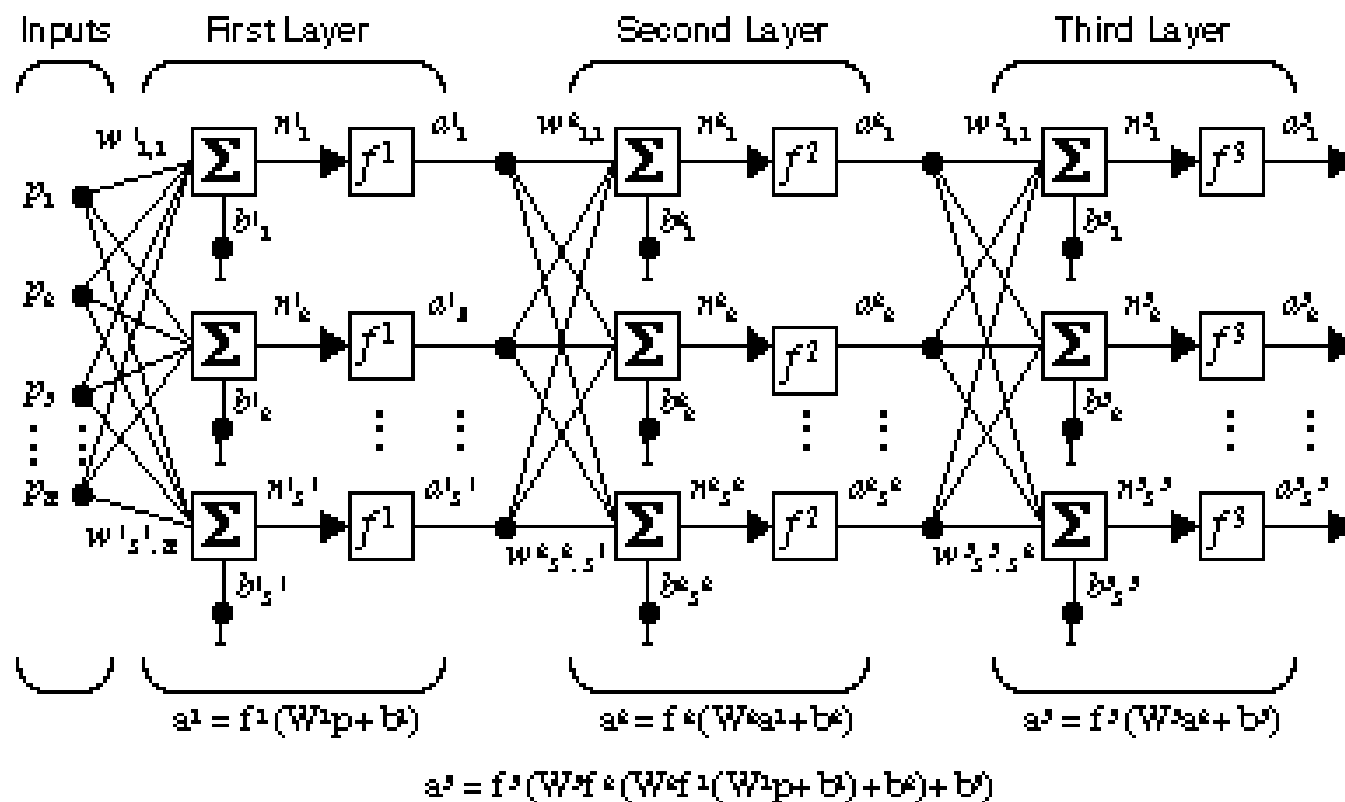
$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

前馈网络模型

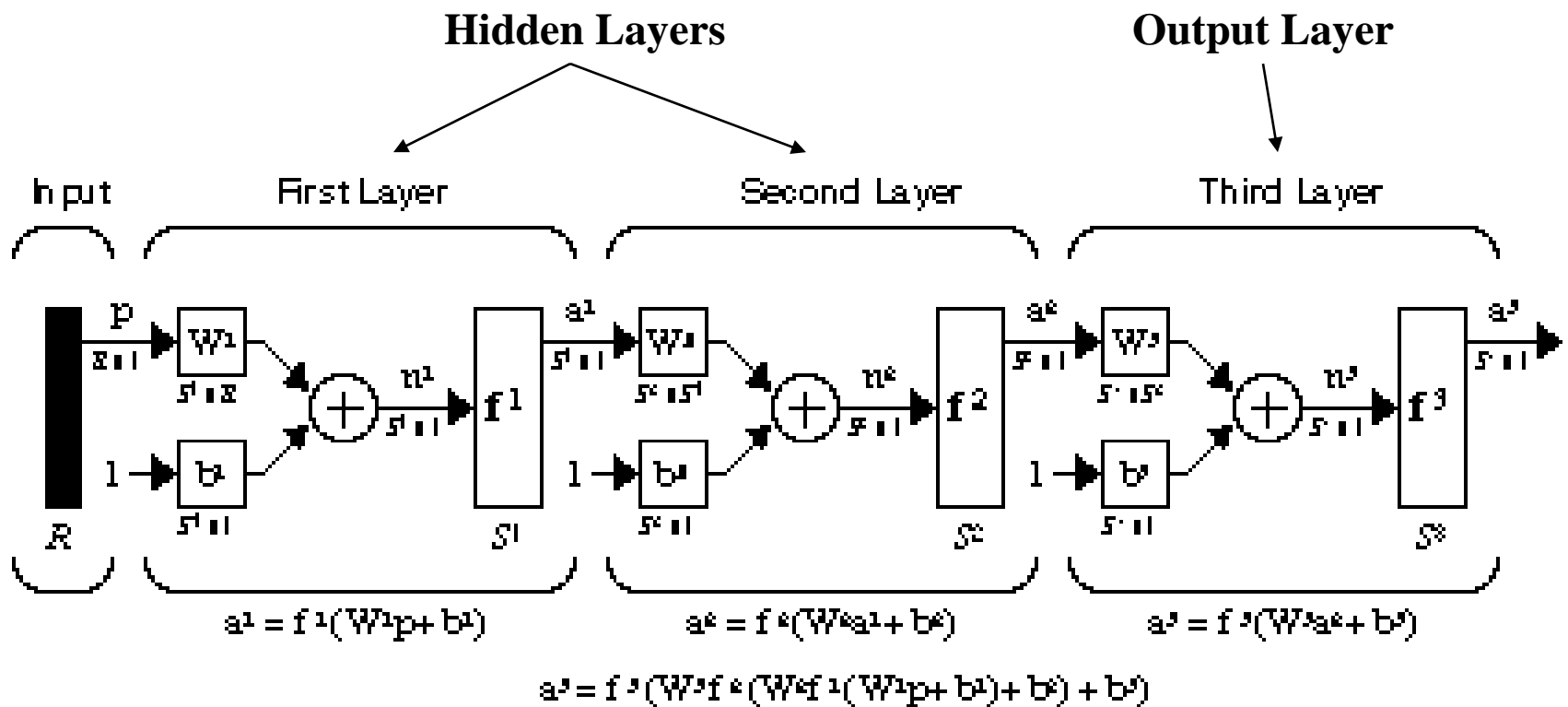


$$a = f(Wx)$$

前馈网络模型



多层网络模型





主要的前馈神经网络

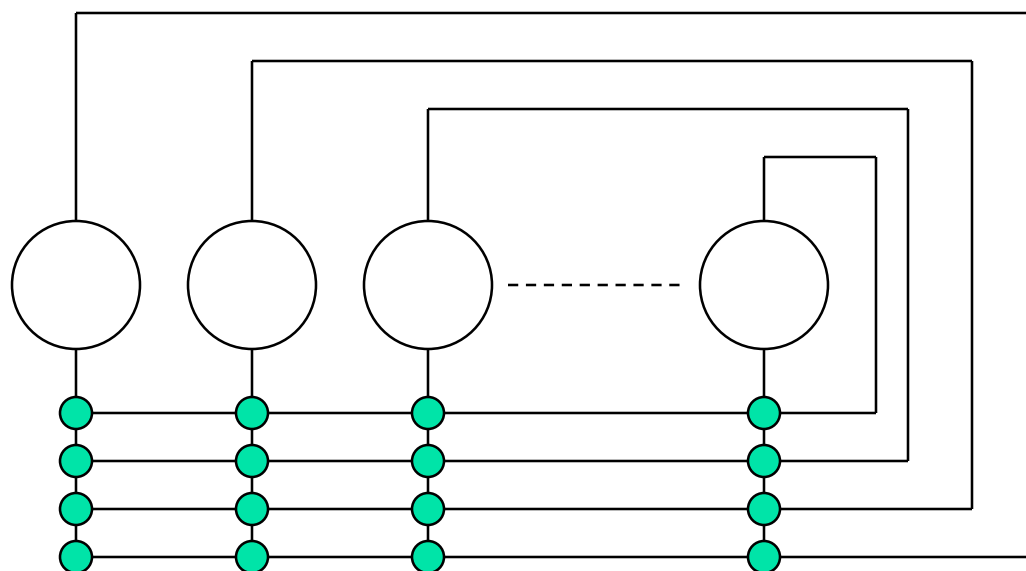
- 感知机
- 线性联想器
- 多层感知机
- BP网络
- RBF网络

递归神经网络

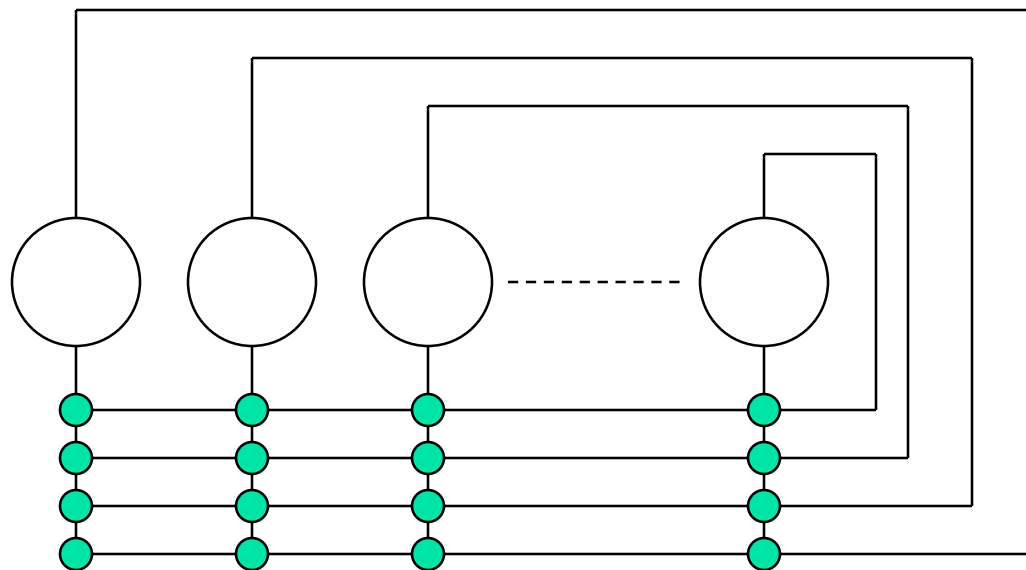
——有反馈的神经网络



递归神经网络拓扑结构

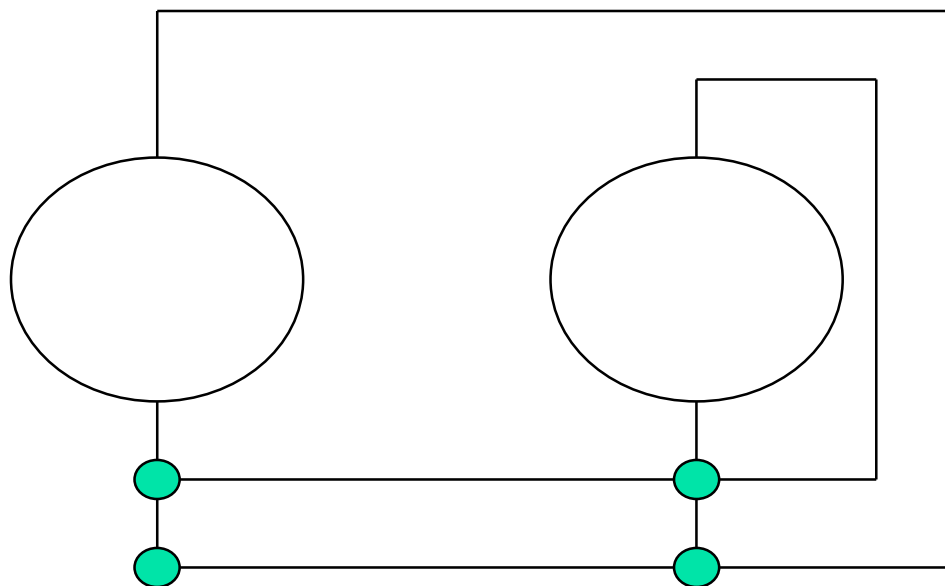


递归神经网络

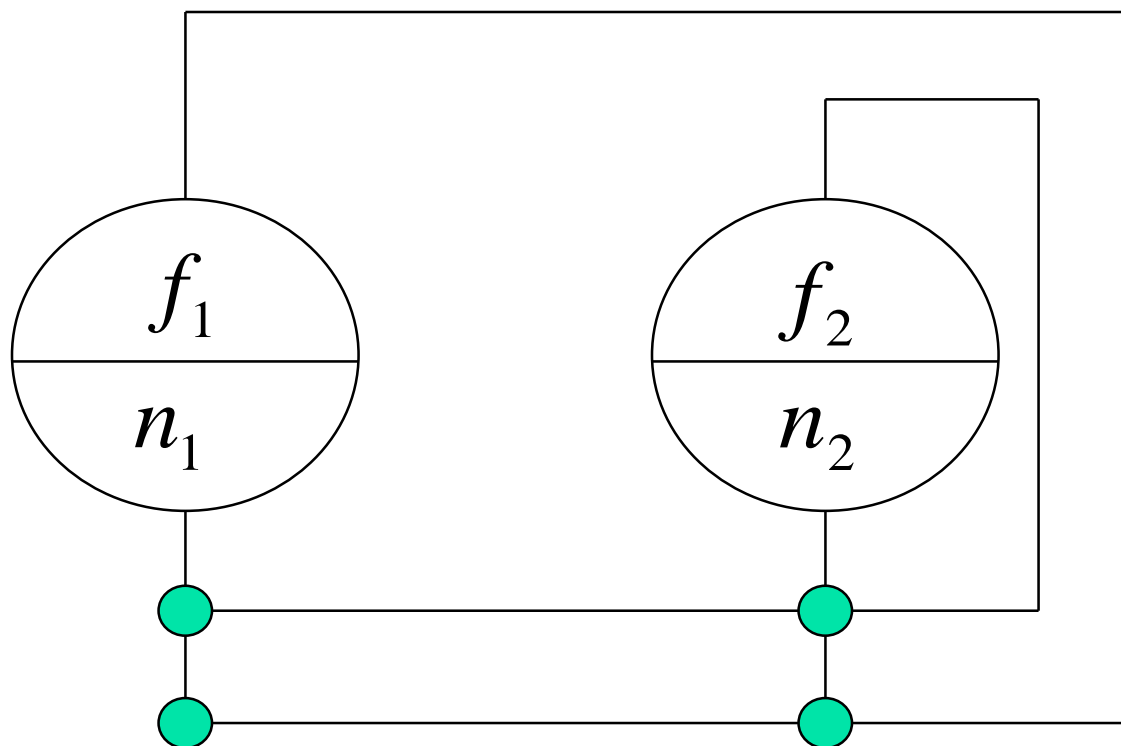


问题：怎样抽象出数学模型？

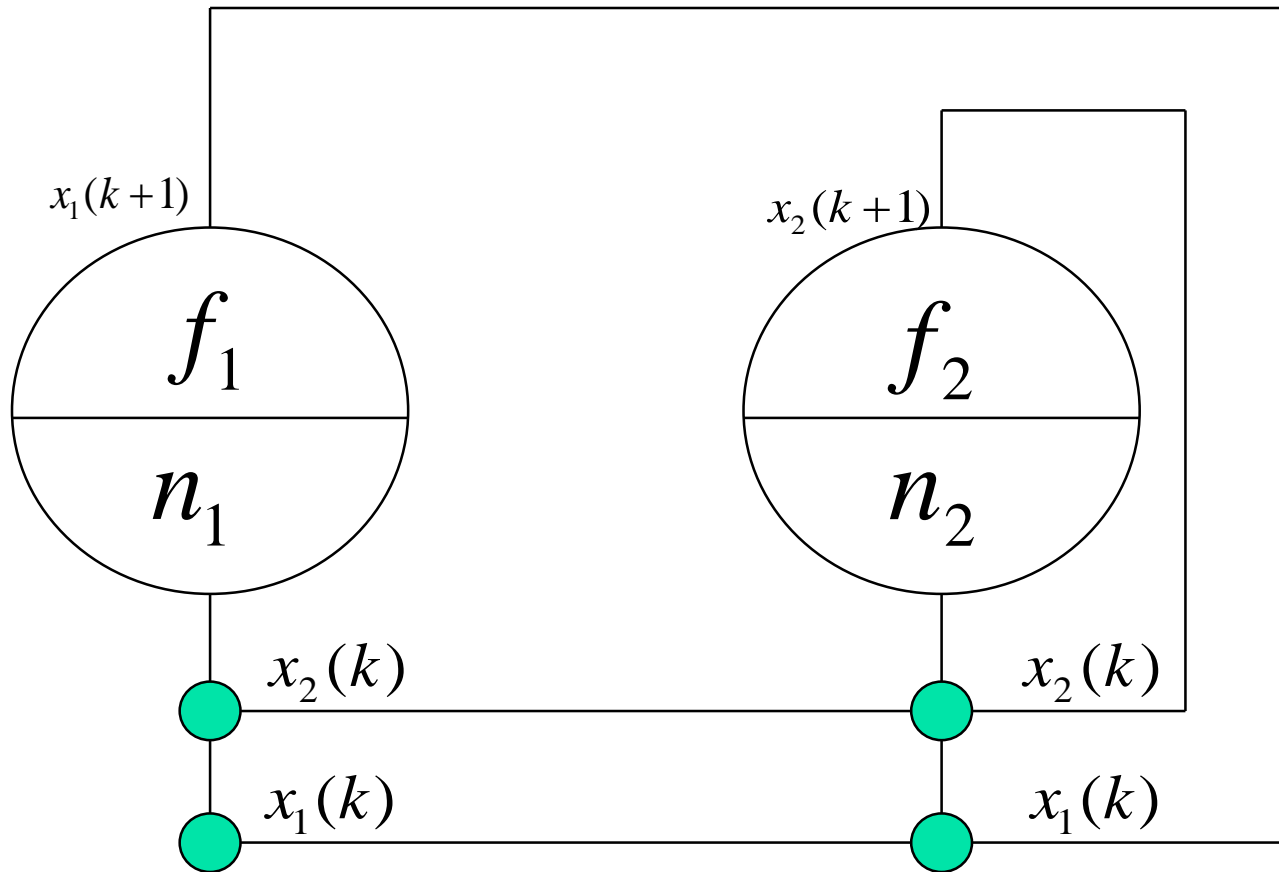
递归神经网络



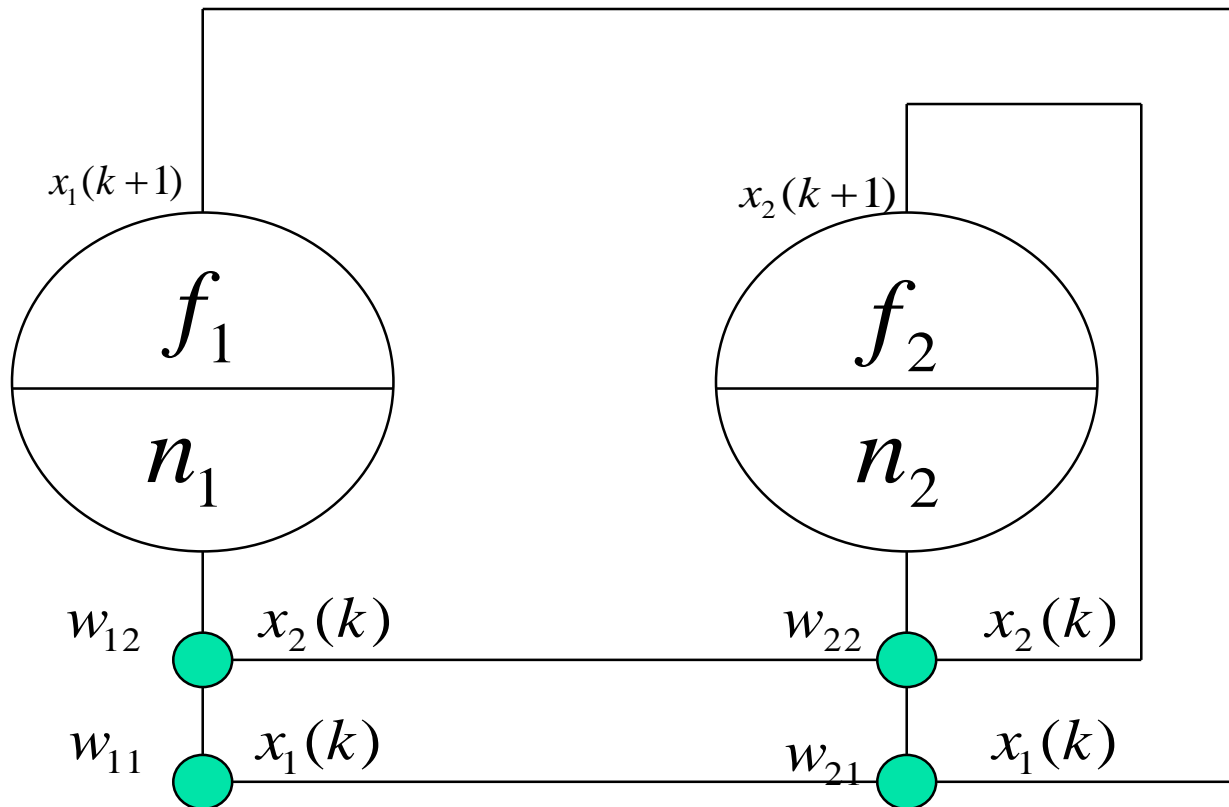
递归神经网络



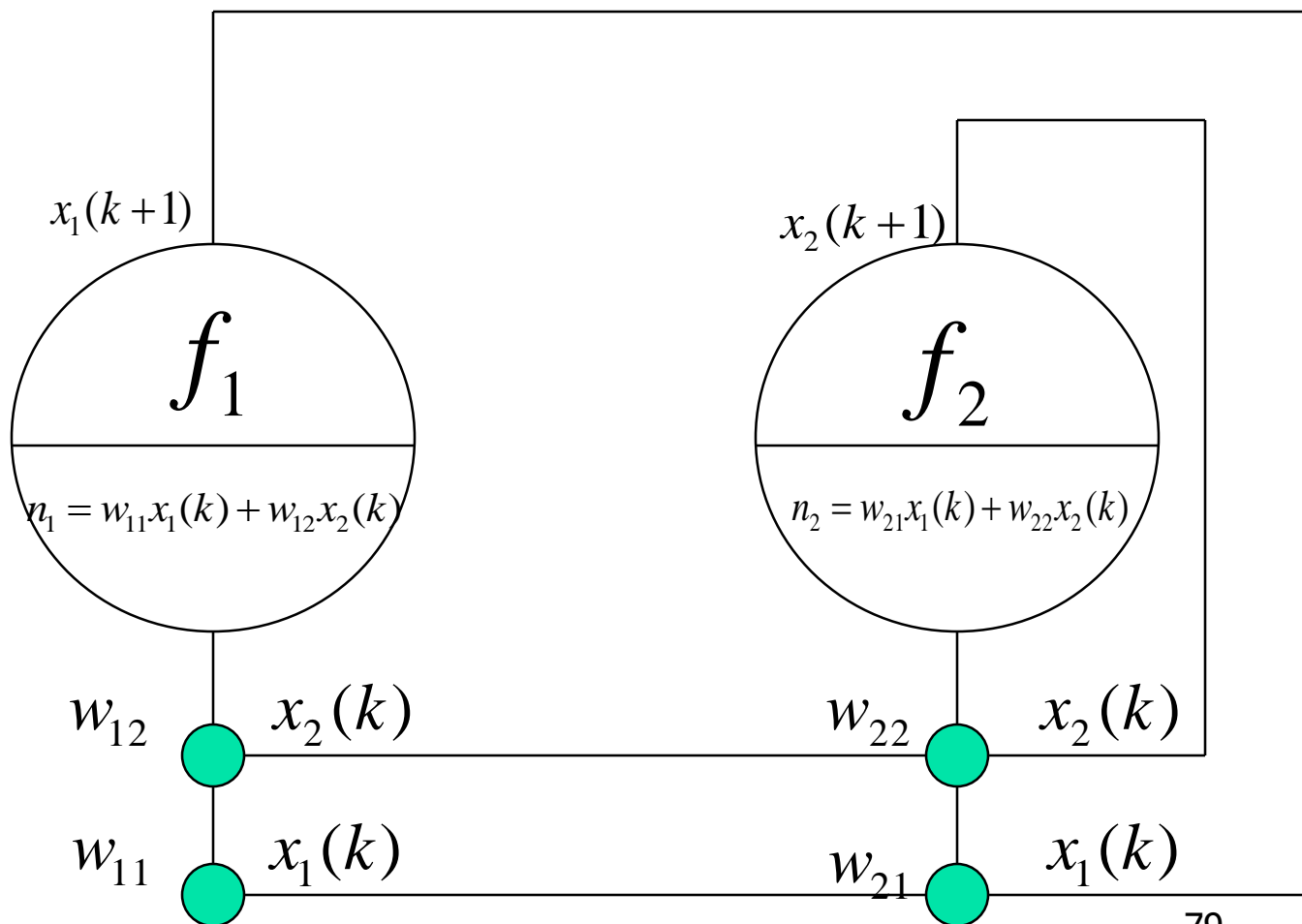
递归神经网络



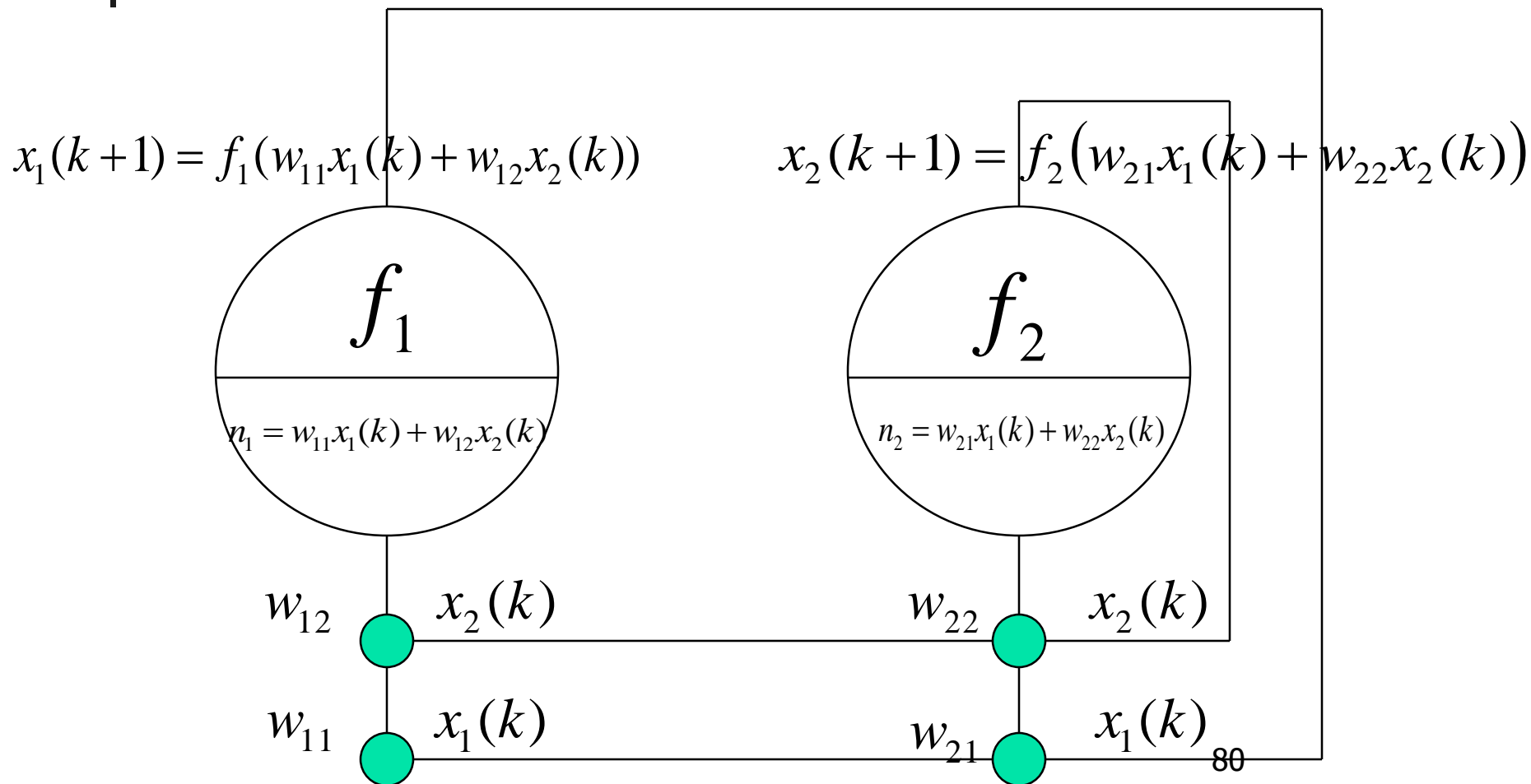
递归神经网络



递归神经网络



递归神经网络





递归神经网络

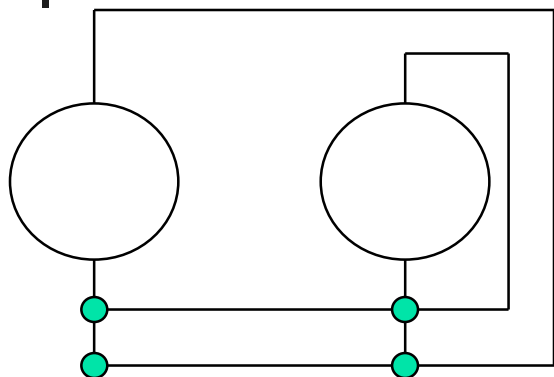
$$\begin{cases} x_1(k+1) = f_1(w_{11}x_1(k) + w_{12}x_2(k)) \\ x_2(k+1) = f_2(w_{21}x_1(k) + w_{22}x_2(k)) \end{cases}$$

向量表达式

$$\Downarrow$$
$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = f\left(\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix}\right)$$

$$\Downarrow$$
$$x(k+1) = f(wx(k))$$

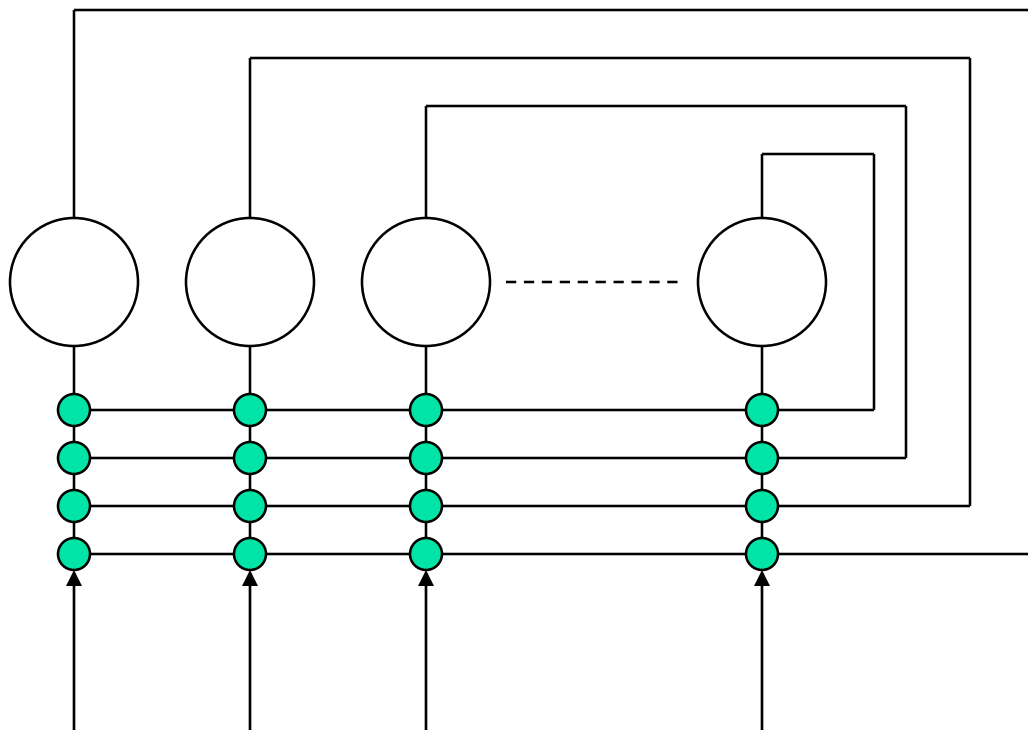
递归神经网络



数学模型

$$x(k+1) = f(wx(k))$$

递归神经网络

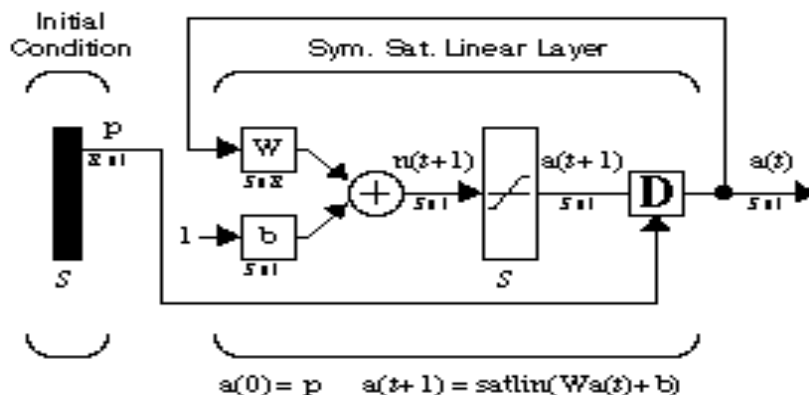


增加外部输入

数学表达式:

$$x(k+1) = f(wx(k) + b)$$

离散递归神经网络: Hopfield网络



利用输入向量对网络进行初始化，网络不断迭代，直到收敛。

$$a(0) = p \quad a(t+1) = \text{satlins}(Wa(t) + b)$$

这里，**satlins**是 $[-1, 1]$ 区间上的对称饱和线性传输函数。

- 当输入 >1 时，函数输出恒为 1 ；
- 当输入 <-1 时，函数输出恒为 -1

$$a(1) = \text{satlins}(Wa(0) + b) = \text{satlins}(Wp + b)$$

$$a(2) = \text{satlins}(Wa(1) + b)$$

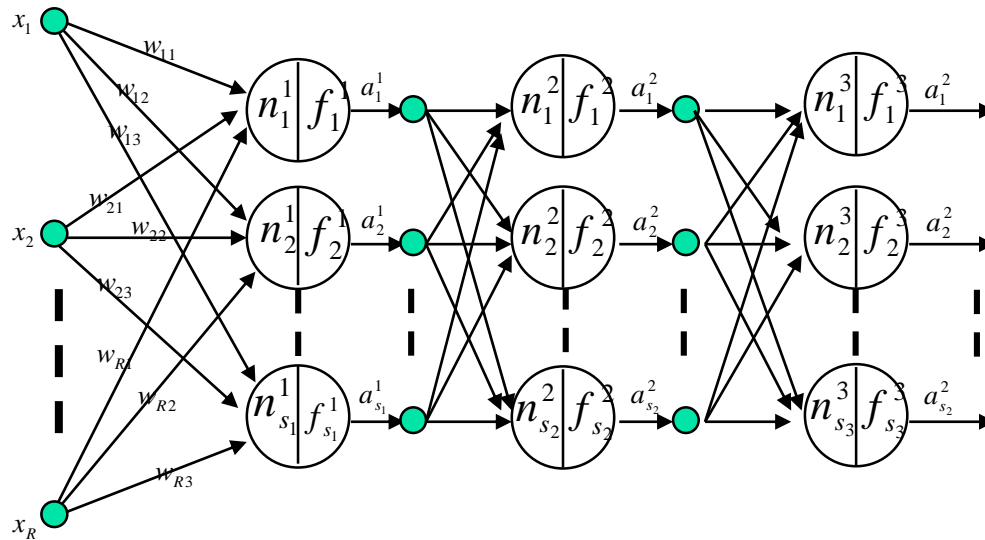


主要的递归神经网络

- Hopfield网络
- Boltzman机
- 双向联想存储器
-

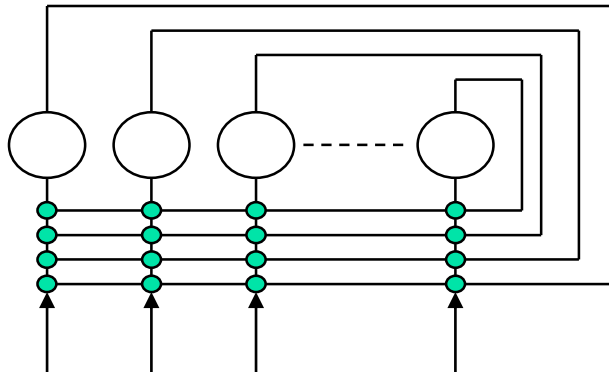
总结

FFNNs



$$a = f(Wx)$$

RNNs



数学表达式:

$$x(k+1) = f(wx(k) + b)$$