

神经网络



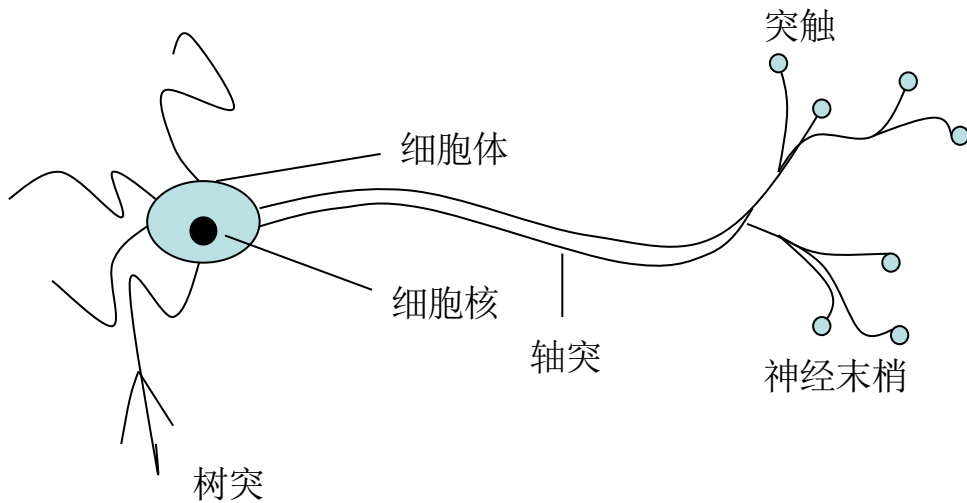
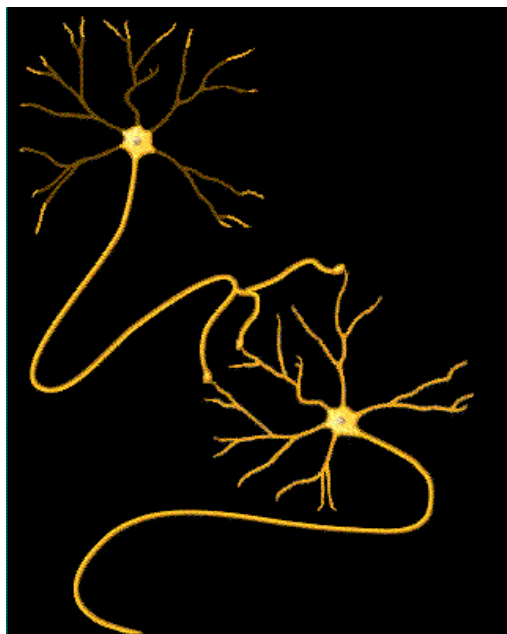
神经网络

- 生物大脑的神经网络，约由1000亿个神经元构成。若干个神经元细胞，经由视觉、触觉或是想像等刺激，会产生神经冲动，大脑中的每个功能就依赖于这些在神经元中不断传递与运行的电流与化学效应。



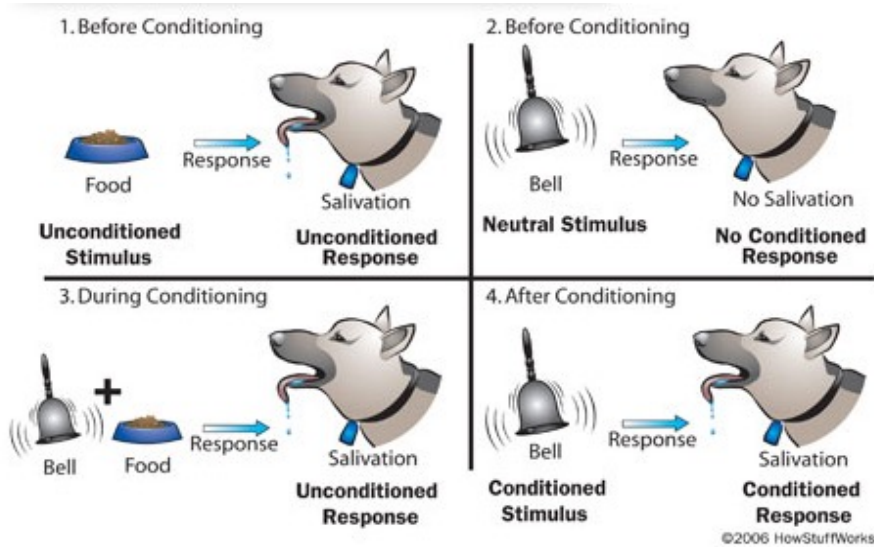
神经网络

- 神经元细胞由细胞体与突起两部分构成，突起包括树突和轴突两种。
- 细胞体起到联络和整合输入信息并传出信息的作用。
- 树突负责接受其他神经元传来的刺激，转化为神经冲动传给细胞体，轴突将神经冲动传导到其他神经元以及它支配的细胞上。



神经网络

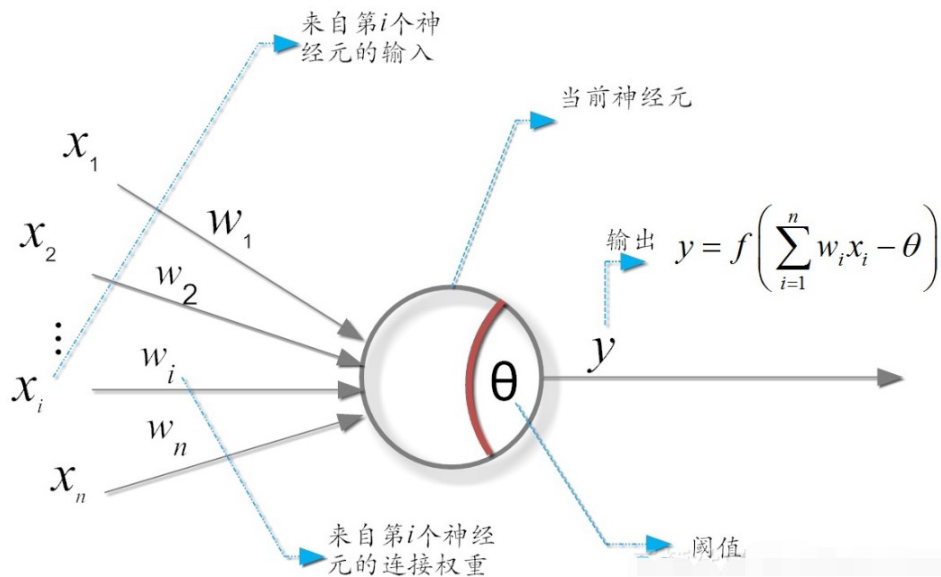
- 大脑的学习过程，就是通过神经元之间突触的形成、连接和强度的变化来实现的。
- 巴普洛夫关于神经反射的实验：通过后天学习和训练会形成反射。



建立前	无条件刺激 → 无条件反应 (食物) (唾液分泌)
	中性刺激 → 引起注意 (铃声) (无唾液分泌)
建立中 (多次重复)	中性刺激 (铃声) ↘ 无条件刺激 → 无条件反应 (食物) (唾液分泌)
建立后	条件刺激 → 条件反应 (铃声) (唾液分泌)

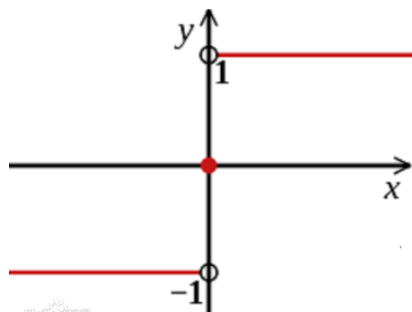
M-P神经元模型

下面就是最简单的一个单层感知机模型，即M-P模型。它是首个通过模仿神经元而成的模型，只有两层：输入层 x_i 和输出层 y 。在模型中，输入 x 相当于传入神经元的信号，权重 w 相当于传递突触的强度。将各个突触传来的信号量经过求和，经阈值进行调整，再通过激活函数 f 的处理后，即得到输出信号 y 。

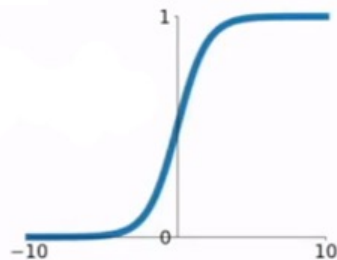


常用激活函数

$$\text{sign}(x) = \begin{cases} +1, & x > 0 \\ -1, & x < 0 \end{cases}$$

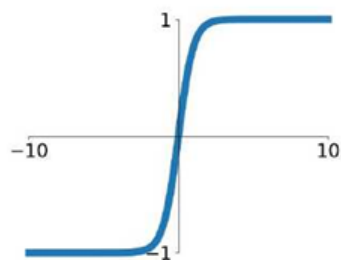


$$S(x) = \frac{1}{1 + e^{-x}}$$



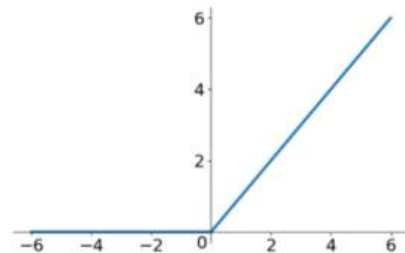
Sigmoid

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



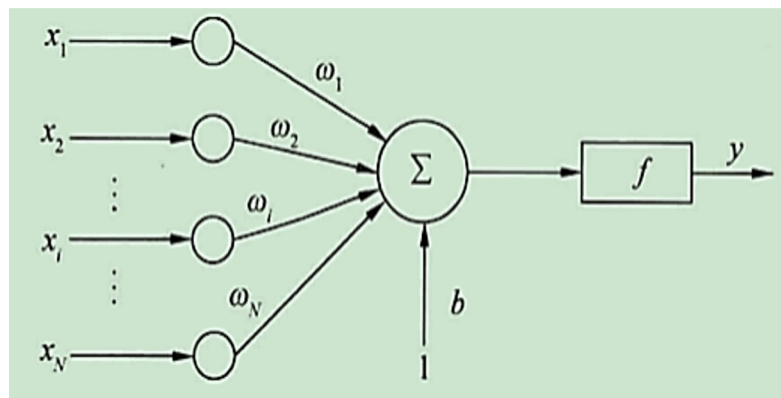
tanh

$$f(x) = \max(0, x)$$



ReLu

误差函数



$$\hat{y} = f\left(\sum_{i=1}^N \omega_i x_i + b\right)$$

误差计算公式：

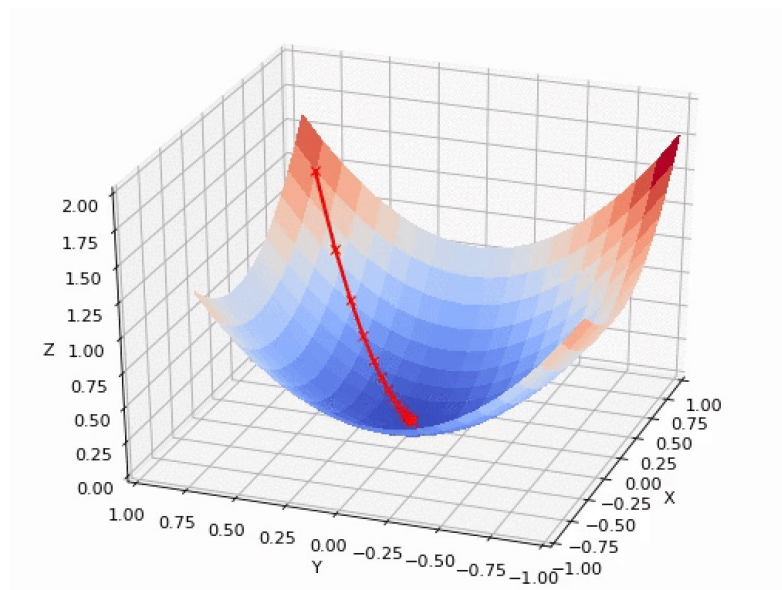
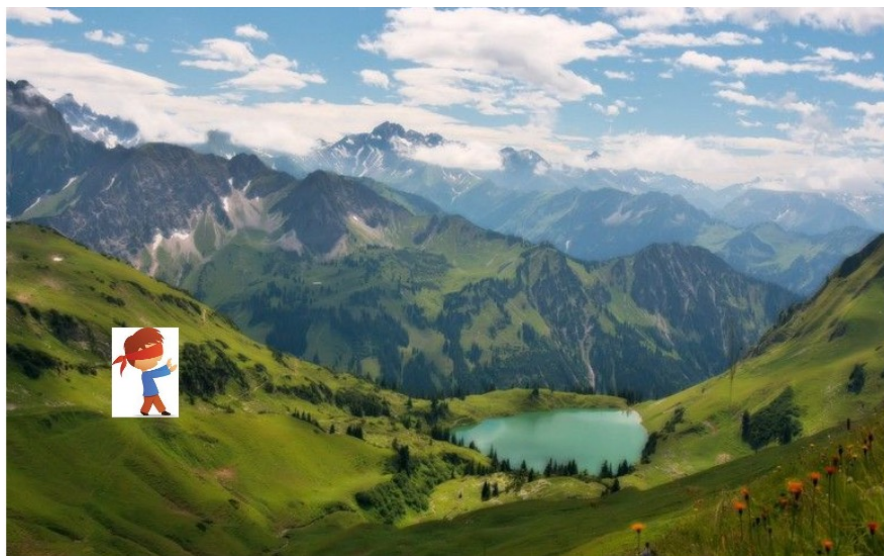
$$E = \frac{1}{2} \sum_{j=1}^N (\hat{y}_j - y_j)^2$$

网络训练目标：

更新权值和阈值，使得误差 E 最小

梯度下降法

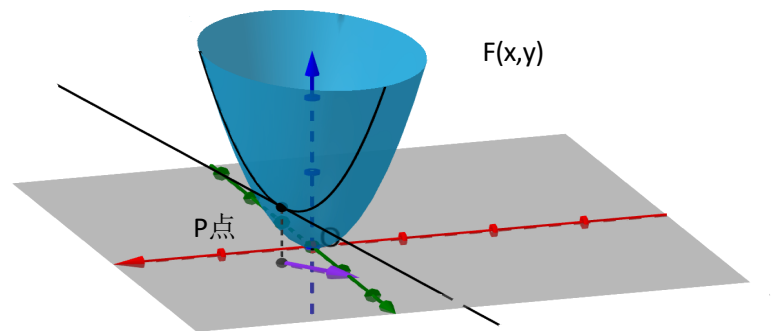
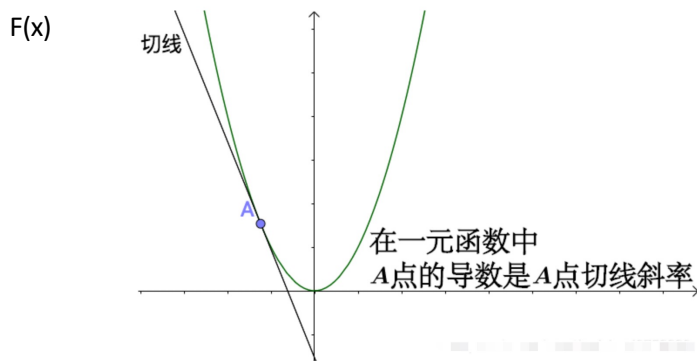
怎么下山：假如你在一座山上，但是你必须到达山谷中最低点的湖泊，你该怎么办？



梯度下降法

- 梯度下降法又称最速下降法，在神经网络常用于求解损失函数最小化问题。它的基本原理是：选取适当的初值 $x(0)$ ，沿着**负梯度**方向不断迭代，更新 x 的值，进行目标函数的极小化，直到收敛。
- 一元函数导数为切线斜率。斜率越大，切线越陡峭，变化越快。导数是标量。
- 方向导数是衡量函数沿着某一方向变化的快慢的物理量。
- 梯度表示某一函数在该点处的方向导数沿着该方向取得最大值，即函数在该点处沿着该方向变化最快，变化率最大。梯度是一个矢量。

$$|\text{grad}f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$



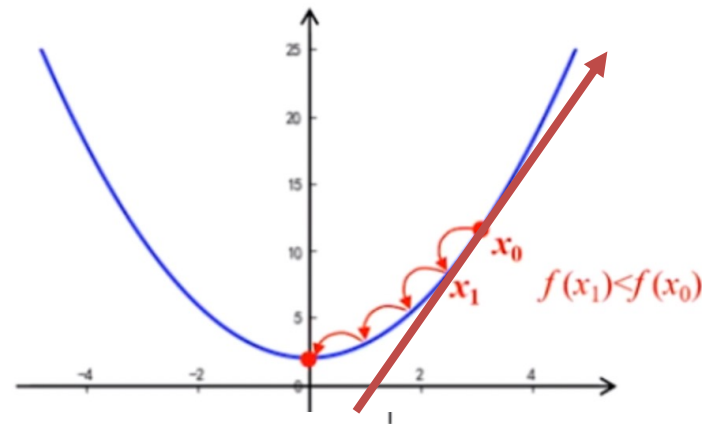
梯度下降法

使用梯度下降法求解. $f(x) = x^2 + 2$ 最小值：

- 选取适当的初值 $x(0)$
- 求梯度: $\nabla = 2x$
- 更新 x 的值, 其中 γ 为步长：

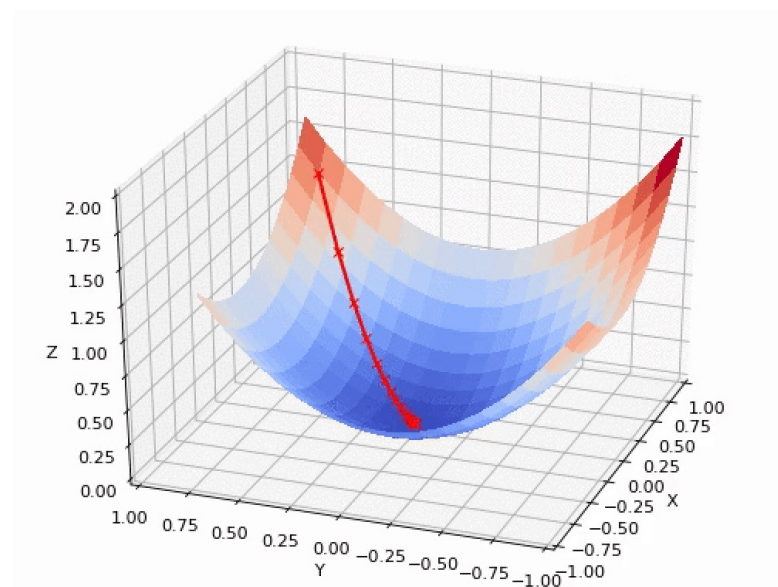
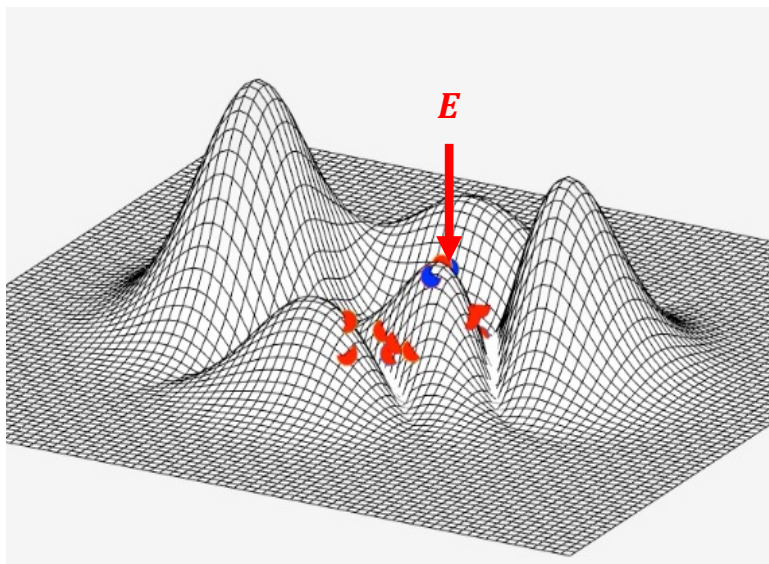
$$x \leftarrow x - \gamma \cdot \nabla$$

- 不断迭代, 进行目标函数的极小化, 直到最后收敛时 即使 x 值的变化, $f(x)$ 基本不变。



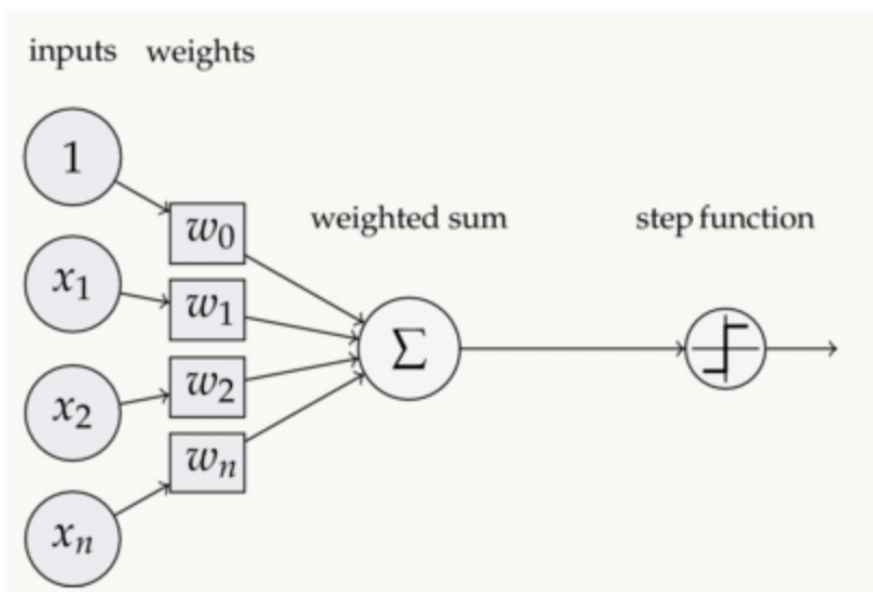
梯度下降法

$$E = \frac{1}{2} \sum_{j=1}^N \left(f \left(\sum_{i=1}^N \omega_i x_i + b \right) - y_j \right)^2$$



预测误差

阈值 b 可看作一个固定输入为1.0的“哑节点”(虚拟节点), 所对应的连接权重 ω_0 , 这样权重与阈值的学习就可统一为权重的学习。



$$E = \frac{1}{2} \sum_{j=1}^{N+1} \left(f \left(\sum_{i=0}^N \omega_i x_i + b \right) y_j - y_j \right)^2$$

$$\nabla \omega_{(j)} = \eta (\hat{y}_j - y) x_{(j)}$$

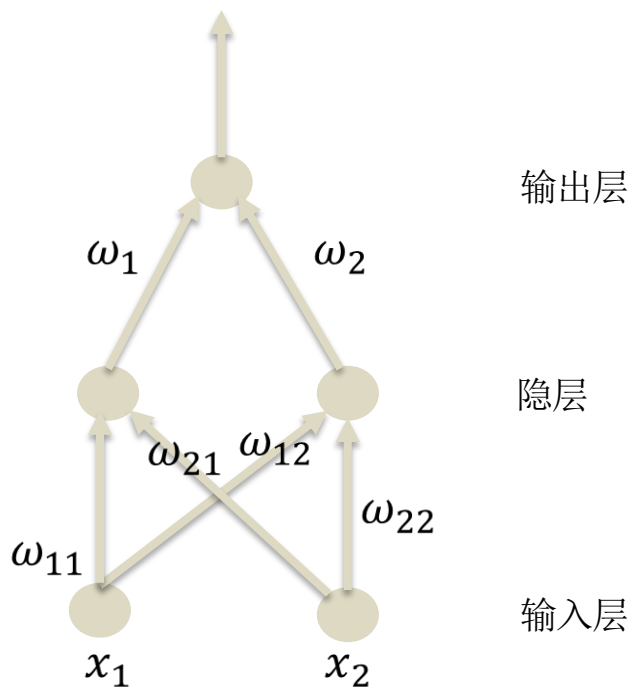
$$\omega' = \omega - \eta \nabla \omega$$

感知器模型

- 给定训练数据集，权重 $\omega_i (i = 1, 2, \dots, n)$ 以及阈值 b 可通过学习得到。
- 阈值 θ 可看作一个固定输入为-1.0的“哑节点”（虚拟节点），所对应的连接权重 ω_0 ，这样权重与阈值的学习就可统一为权重的学习。
- 感知机的学习规则：
 - 对训练样本 (x, \hat{y}) ，若当前感知机的输出为 y 。
 - 计算误差 E 和权重调整量： $\Delta \omega_i = \eta(\hat{y} - y)x_i$ ，其中， $\eta \in (0, 1)$ 称为学习率(learning rate)。
 - 更新权重： $\omega_i \leftarrow \omega_i - \Delta \omega_i$
 - 不断迭代训练，直到 $\hat{y} = y$ ，感知机停止权重调整。

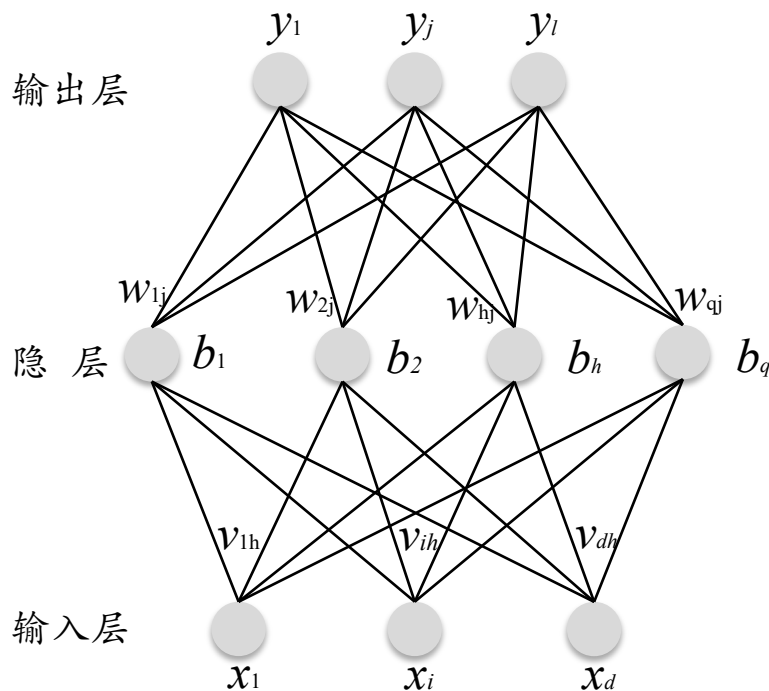
多层网络

- 要解决非线性问题，需要考虑使用多层功能神经元。输入层与输出层之间的一层神经元，被称为隐层 (hidden layer)。隐层与输出层神经元都是拥有激活函数的功能神经元。

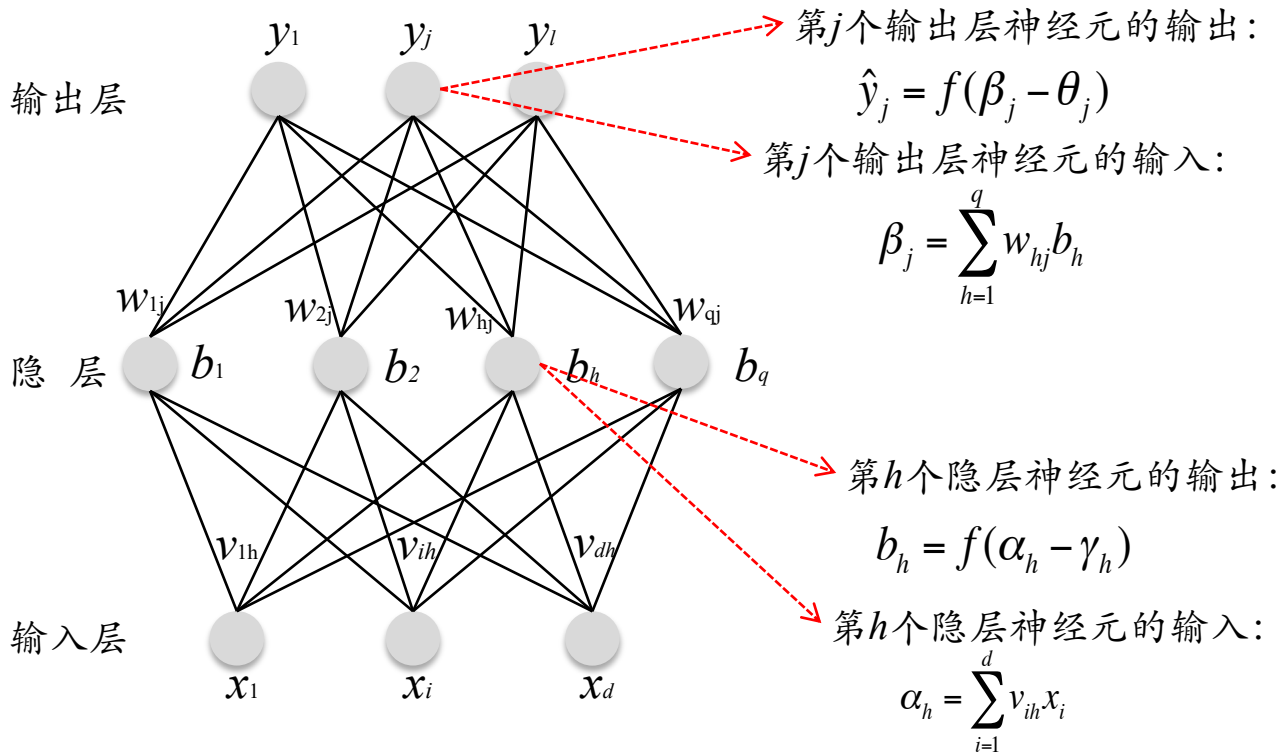


BP神经网络

- BP (error BackPropagation) 神经网络，指使用BP算法训练的多层前馈神经网络。BP算法又称误差逆传播算法或反向传播算法，是学习算法中最杰出的代表之一，也是迄今为止最成功的神经网络学习算法。现实中大多数的神经网络都是使用BP算法进行训练的，比如说深度学习概念下的卷积神经网络。



BP神经网络



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

$$\Delta w_{hj} = \eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j) b_h$$

$$\Delta \theta_j = -\eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j)$$

$$\Delta v_{ih} = \eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j x_i$$

$$\Delta \gamma_h = -\eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j$$

BP算法步骤

➤ **输入：**训练集 $D = \{(x_k, y_k)\}_{k=1}^m$ ；

学习率 η

➤ **过程：**

在(0,1)范围内随机初始化网络中所有连接权和阈值

repeat

for all $(x_k, y_k) \in D$ do

 根据网络输入和当前参数计算当前样本的输出 \hat{y}_k

 计算输出层神经元的梯度项 g_j

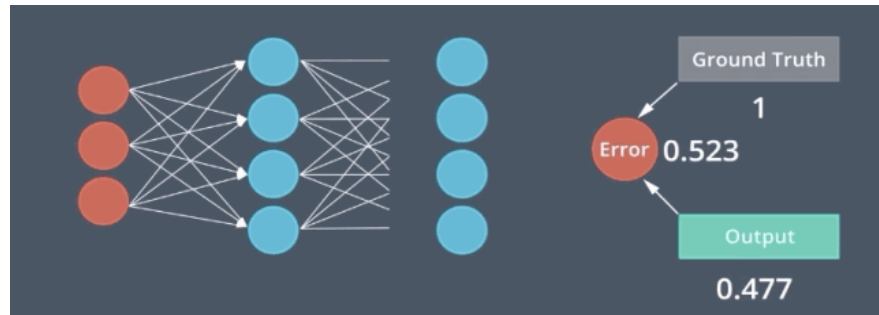
 计算隐层神经元的梯度项 e_h

 更新连接权 ω_{hj} , v_{ih} 与阈值 θ_j , γ_h

end for

Until 达到停止条件

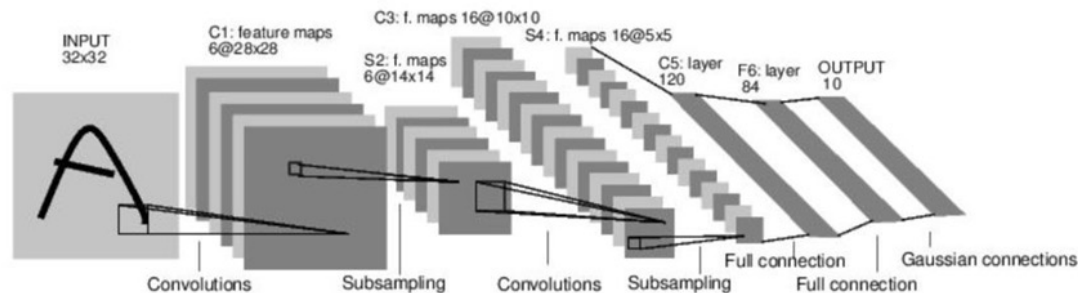
➤ **输出：**连接权与阈值确定的多层前馈神经网络



CNN卷积神经网络模型

CNN卷积神经网络模型

- 一个深度卷积神经网络通常包含输入层、多个卷积层、对应的非线性映射层和池化层。最后通过全连接层将二维的特征图连接成一个向量输入到最后的分类器，得到概率的输出，如下图：

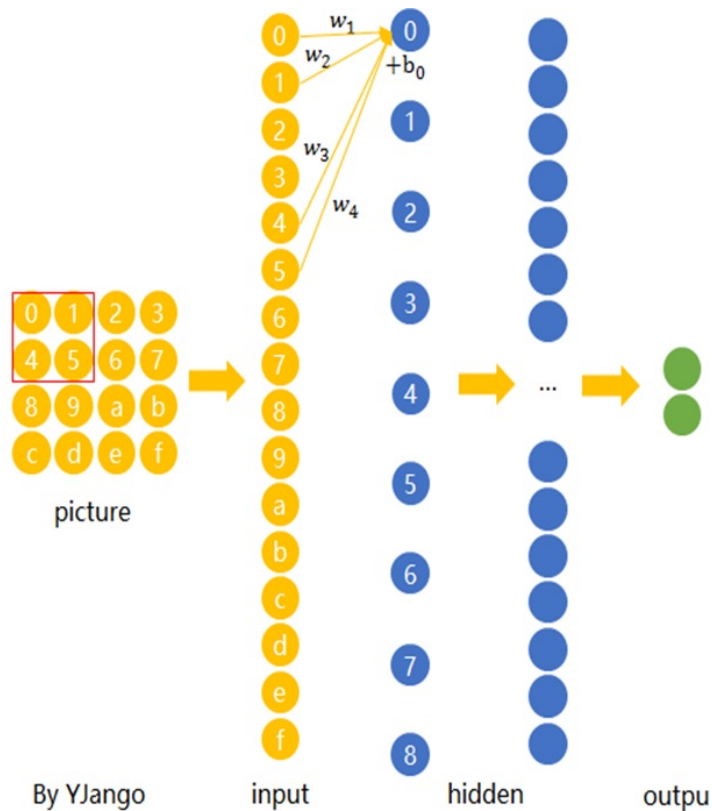


Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-CONV-FC]

CNN卷积神经网络模型

卷积神经网络

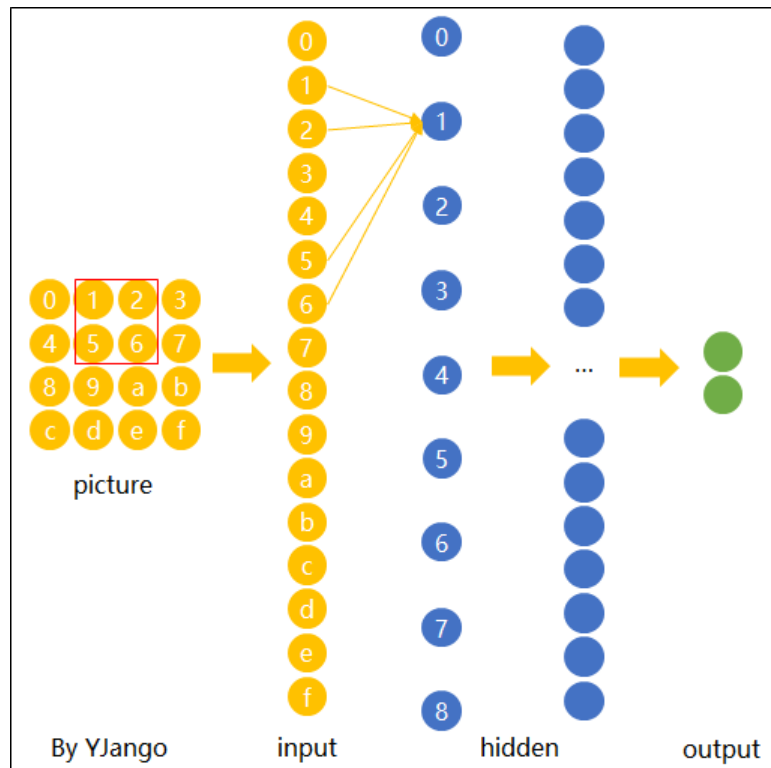
- 卷积层：先选择一个局部区域（卷积核），用这个局部区域去扫描整张图片。局部区域所圈起来的所有节点会被连接到下一层的一个节点上。



CNN卷积神经网络模型

卷积神经网络

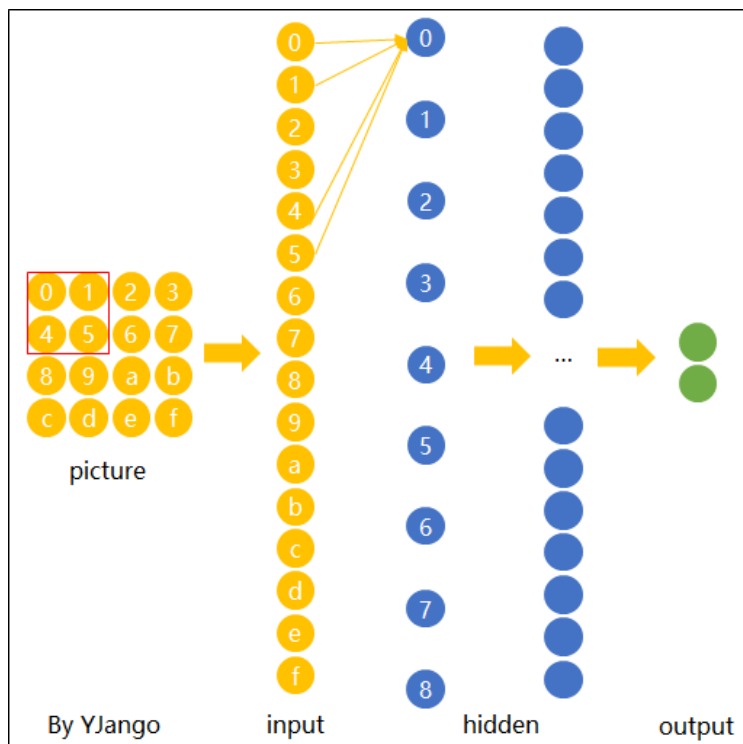
- 卷积层：先选择一个局部区域（卷积核），用这个局部区域去扫描整张图片。局部区域所圈起来的所有节点会被连接到下一层的一个节点上。



CNN卷积神经网络模型

卷积神经网络：

卷积过程：



CNN卷积神经网络模型

卷积神经网络

卷积后排列成2维结构

1	0	1
0	1	0
1	0	1

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	0	1	1	0

Image

4	3	

Convolved
Feature

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

4	3	4
2	4	3
2	3	4

Convolved
Feature

1	1	1	0	0
0 _{x1}	1 _{x0}	1 _{x1}	1	0
0 _{x0}	0 _{x1}	1 _{x0}	1	1
0 _{x1}	0 _{x0}	1 _{x1}	1	0
0	1	1	0	0

Image

4	3	4
2		

Convolved
Feature

CNN卷积神经网络模型

卷积神经网络

卷积后排列成2维结构

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

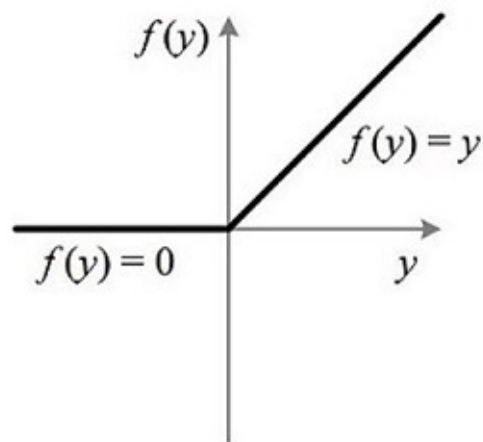
4		

Convolved
Feature

CNN卷积神经网络模型

卷积神经网络

非线性映射层



$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

CNN卷积神经网络模型

卷积神经网络

➤ 引入ReLU激活函数原因：

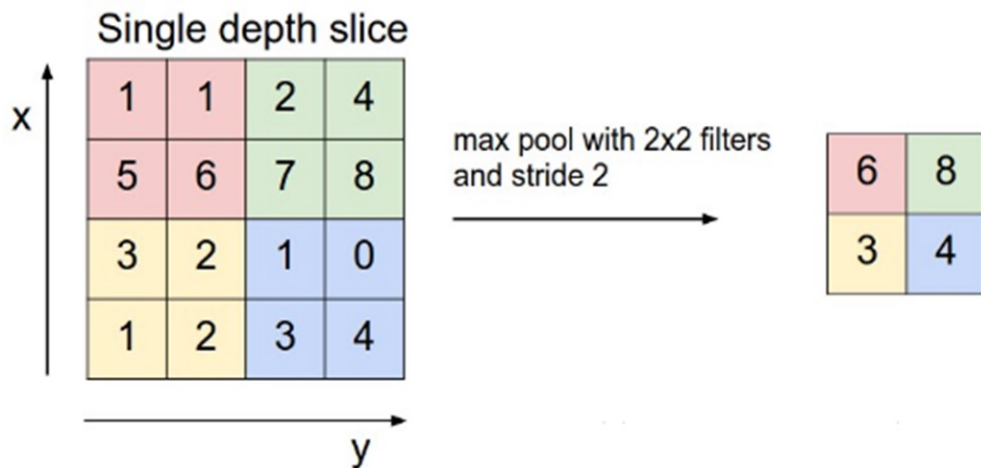
- 加入非线性因素，使网络有更好的表达能力，加强模型的学习能力；
- 算法简单，网络运算效率更高；
- 梯度是常数，不会出现梯度消失，使得模型收敛速度稳定；
- 收敛速度比sigmoid和tanh函数快。

CNN卷积神经网络模型

卷积神经网络

➤ 池化层

池化层的主要目的是降维

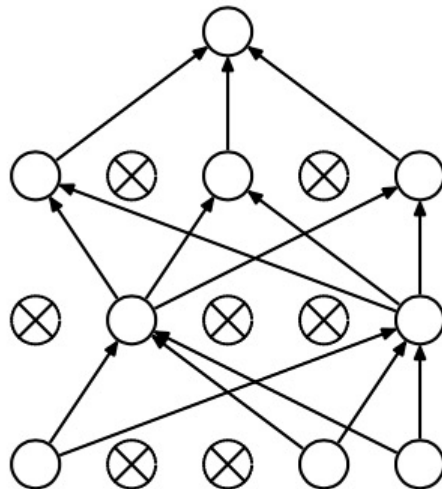
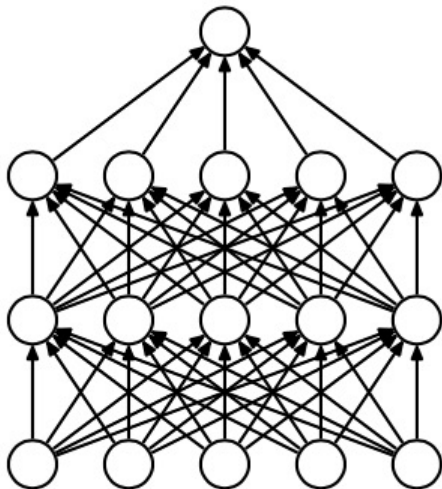


CNN卷积神经网络模型

卷积神经网络

➤ dropout层

训练过程中使部分神经元不参与“学习”。



CNN卷积神经网络模型

卷积神经网络

➤ 全连接层

- 当抓取到足以用来识别图片的特征后，接下来的就是如何进行分类。
- 全连接层（也叫前馈层）就可以用来将最后的输出映射到线性可分的空间。
- 卷积网络的最后会将末端得到一个长长的向量，并送入全连接层配合输出层进行分类。



Thank you!