

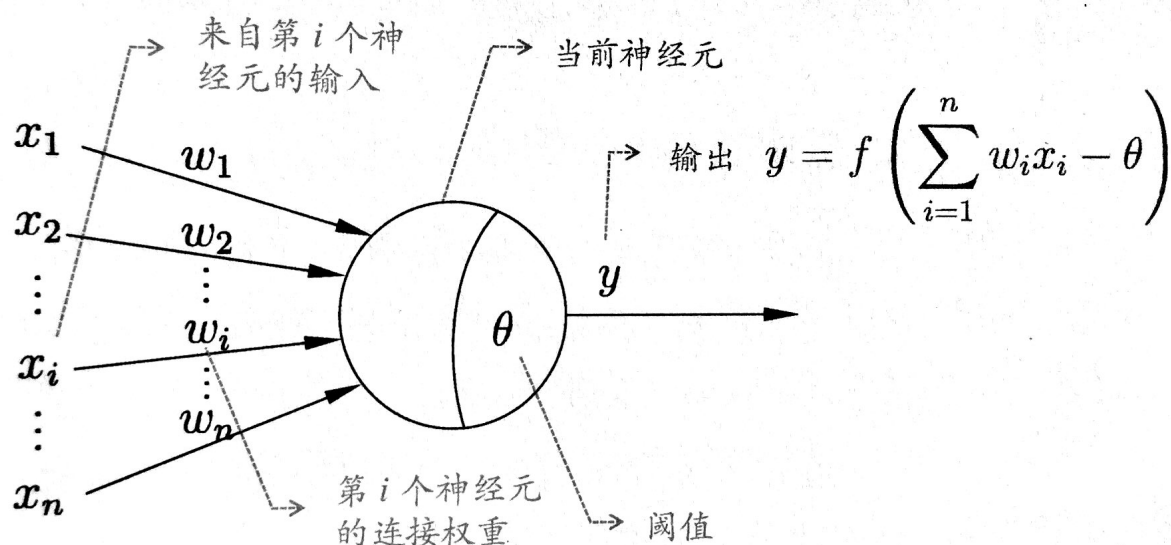
神经元模型

神经网络是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应

M-P神经元模型思路

将生物神经网络中的神经元抽象后得到经典的“M-P神经元模型”。在这个模型中，神经元接收来自n个其他神经元传递过来的输入信号，这些信号通过带权重的连接进行传递，神经元接收到总输入值，将与神经元的阈值进行比较，然后通过激活函数处理以产生神经元的输出

图示如下



典型的激活函数

- 介跃函数

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

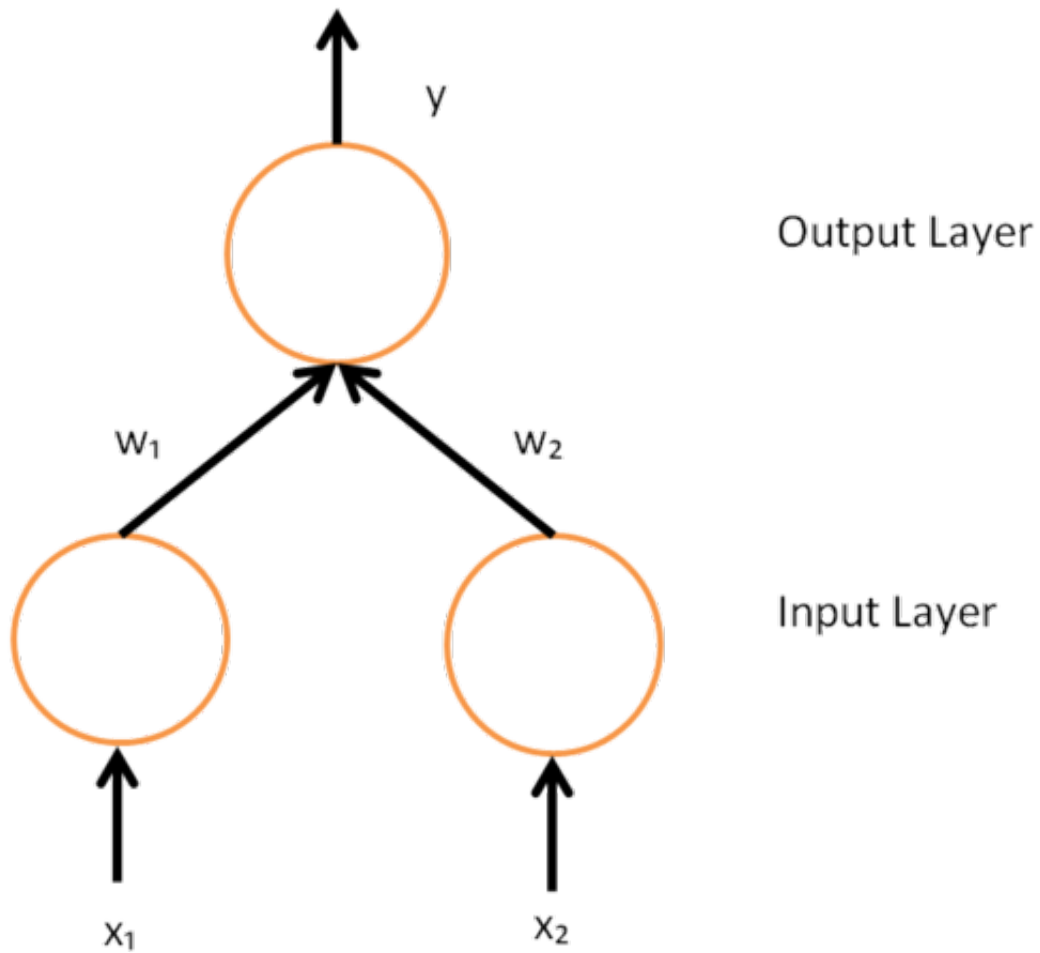
- Sigmoid 函数

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

感知机 (Perceptron)

感知机被视为一种最简单形式的前馈神经网络，是一种二元线性分类器。

Wikipedia: In machine learning, the **perceptron** is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input (represented by a vector of numbers) belongs to one class or another.



Definition: In the modern sense, the perceptron is an algorithm for learning a binary classifier: a function that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value):

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

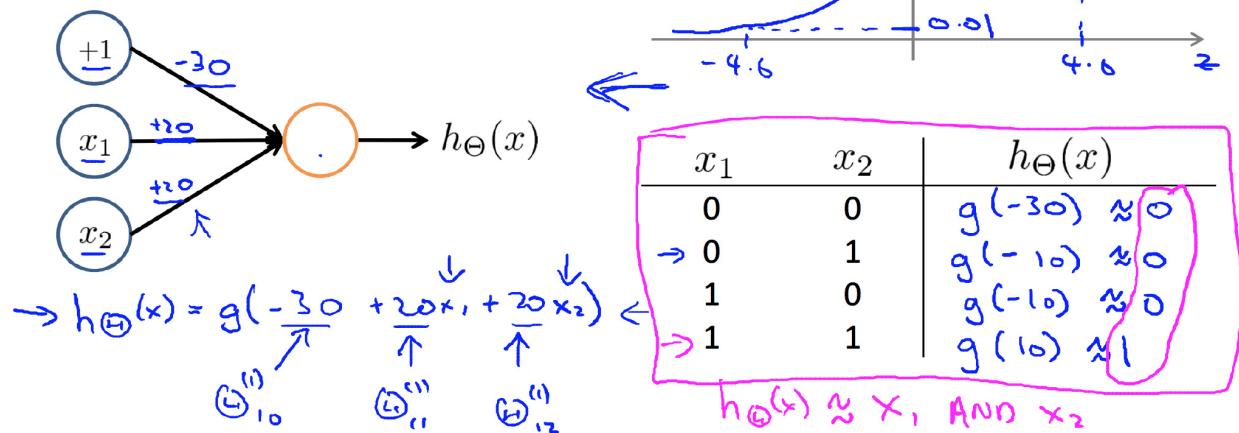
简单的说，其实就是一个超平面将样本空间一分为二

Example

Simple example: AND

→ $x_1, x_2 \in \{0, 1\}$

→ $y = x_1 \text{ AND } x_2$



神经网络模型

基本构成

常见的神经网络是由感知机构成的层级结构，每层神经元和下层神经元全互连，神经元之间不存在同层连接，也不存在跨层连接，这样的神经网络通常称为**多层前馈神经网络**。

模型训练

神经网络的学习过程，就是根据训练数据来调整神经元之间的**连接权**以及每个功能神经元的**阈值**

误差逆传播算法 (Back Propagation of Error)

算法思想概览

Backward Propagation of Errors, often abbreviated as BackProp is one of the several ways in which an artificial neural network (ANN) can be trained. It is a **supervised training** scheme, which means, it learns from labeled training data (there is a supervisor, to guide its learning).

To put in simple terms, BackProp is like "learning from mistakes". The supervisor corrects the ANN whenever it makes mistakes.

An ANN consists of nodes in different layers; input layer, intermediate hidden layer(s) and the output layer. The connections between nodes of adjacent layers have "weights" associated with them. **The goal of learning is to assign correct weights for these edges.** Given an input vector, these weights determine what the output vector is.

In supervised learning, the training set is labeled. This means, for some given inputs, we know(label) the desired/expected output.

BackProp Algorithm:

Initially all the edge weights are randomly assigned. For every input in the training dataset, the ANN is activated and its output is observed. This output is compared with the desired output that we already know, and the error is "propagated" back to the previous layer. This error is noted and the weights are "adjusted" accordingly. This process is repeated until the output error is below a predetermined threshold.

Once the above algorithm terminates, we have a "learned" ANN which, we consider is ready to work with "new" inputs. This ANN is said to have learned from several examples (labeled data) and from its mistakes (error propagation).

摘自: [\(Quora\) How do you explain back propagation algorithm to a beginner in neural network?](#)

BP算法基于梯度下降策略, 以目标函数的负梯度方向对参数进行调整

梯度下降补充说明

梯度下降法也称最速下降法是以负梯度方向作为下降方向对极小化算法, 也是无约束最优化中最简单的方法

下面给出负梯度方向是下降最快的方向的证明过程

假定 $f(x)$ 在 x_k 附近连续可微, 于是有

$$f(x) = f(x_k) + g_k^T(x - x_k) + o(\|x - x_k\|) \text{ 其中 } g_k = \nabla f(x_k) \neq 0$$

记 $x - x_k = \alpha d_k$, α 是步长因子, d_k 为下降方向, 于是得到

$$f(x_k + \alpha d_k) = f(x_k) + \alpha g_k^T d_k + o(\|\alpha d_k\|)$$

$$f(x_k + \alpha d_k) - f(x_k) = \alpha g_k^T d_k + o(\|\alpha d_k\|)$$

想要下降最快, 也即最小化 $g_k^T d_k$ 利, 用 $C - S$ 可得

$$|g_k^T d| \leq \|g_k\| \cdot \|d_k\|$$

显然, 当 $d_k = -g_k$ 时 $g_k^T d_k$ 取到最小

BP算法流程

输入: 训练集合 $D = \{(x_k, y_k)\}_{k=1}^m$, 学习率 η

过程:

1. 在(0,1)范围内随机初始化网络中所有权值和阈值
2. repeat

3. for all $(x_k, y_k) \in D$ do
4. 计算当前样本的输出 \hat{y}_k
5. 计算输出层神经元的梯度项
6. 计算隐层神经元的梯度项
7. 更新连接的权值和阈值
8. end for
9. until 达到停止条件

输出：由连接权和阈值确定的多层前馈神经网络

BP算法执行的操作

先将输入示例提供给输入层神经元，然后逐层将信号前传，直到产生输出层的结果；然后计算输出层的误差，再将误差逆向传播至隐层神经元，最后根据隐层神经元的误差来对连接权和阈值进行调整