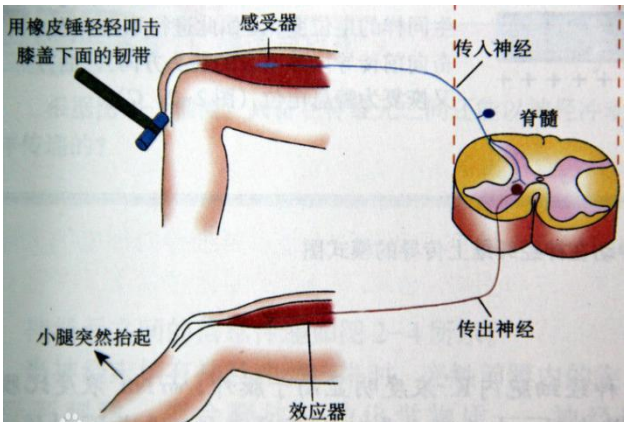
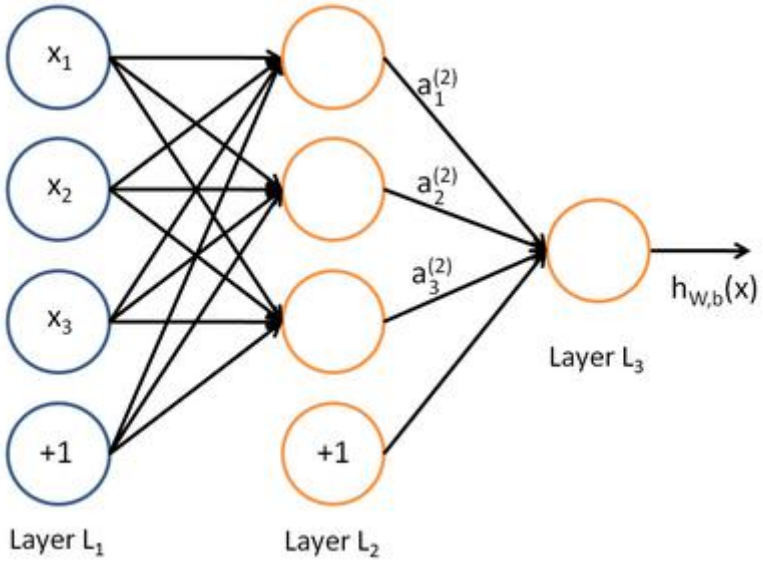
神经网络

目前，深度学习（Deep Learning，简称 DL）在算法领域可谓是大红大紫，现在不只是互联网、人工智能，生活中的各大领域都能反映出深度学习引领的巨大变革。要学习深度学习，那么首先要熟悉神经网络（Neural Networks，简称 NN）的一些基本概念。

当然，这里所说的神经网络不是生物学的神经网络，我们将其称之为人工神经网络（Artificial Neural Networks，简称 ANN）貌似更为合理。神经网络最早是人工智能领域的一种算法或者说是模型，目前神经网络已经发展成为一类多学科交叉的学科领域，它也随着深度学习取得的进展重新受到重视和推崇。



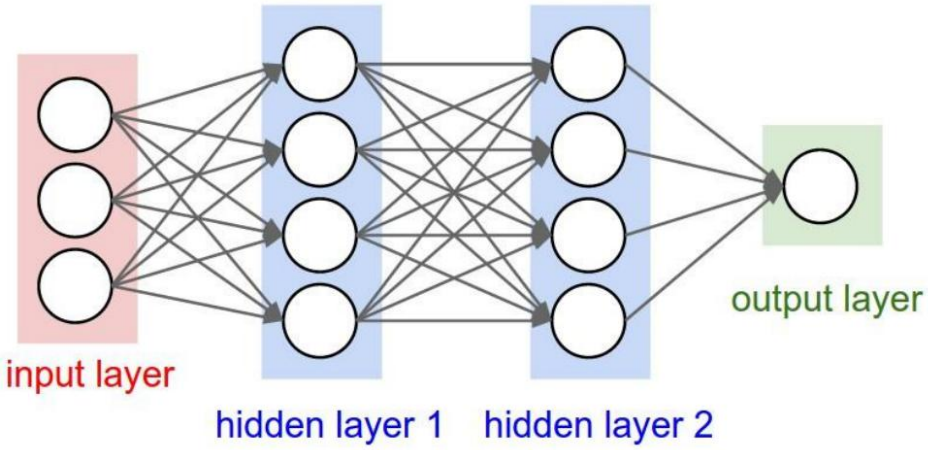
神经网络方面的研究很早就已出现，今天“神经网络”已是一个相当大的、多学科交叉的学科领域。神经网络中最基本的成分是神经元模型。



上图中每个圆圈都是一个神经元，每条线表示神经元之间的连接。我们可以看到，上面的神经元被分成了多层，层与层之间的神经元有连接，而层内之间的神经元没有连接。

神经网络之 是什么

神经网络长什么样？ 什么鬼…

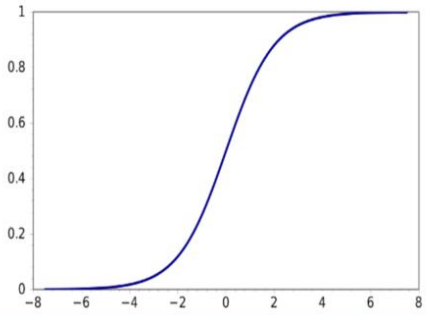


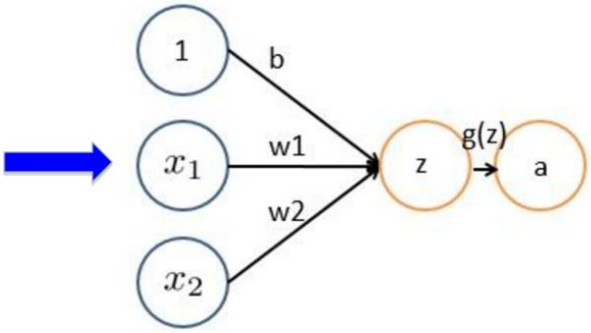
神经网络之 是什么

从逻辑回归到神经元『感知器』

$z = \theta_0 + \theta_1 X_1 + \theta_2 X_2$

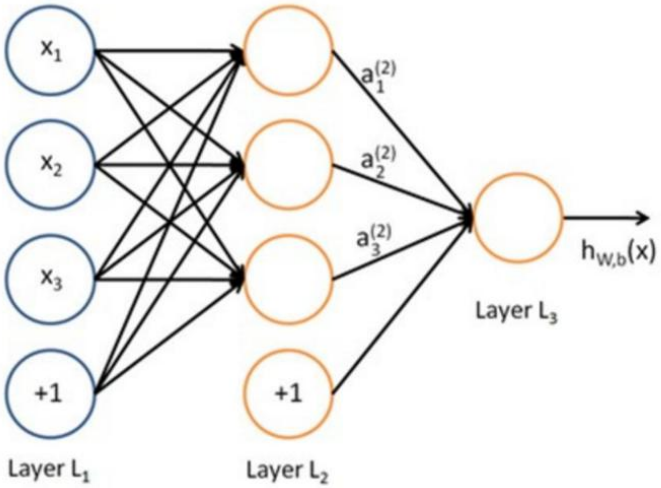
$a = g(z) = \frac{1}{1+e^{-z}}$





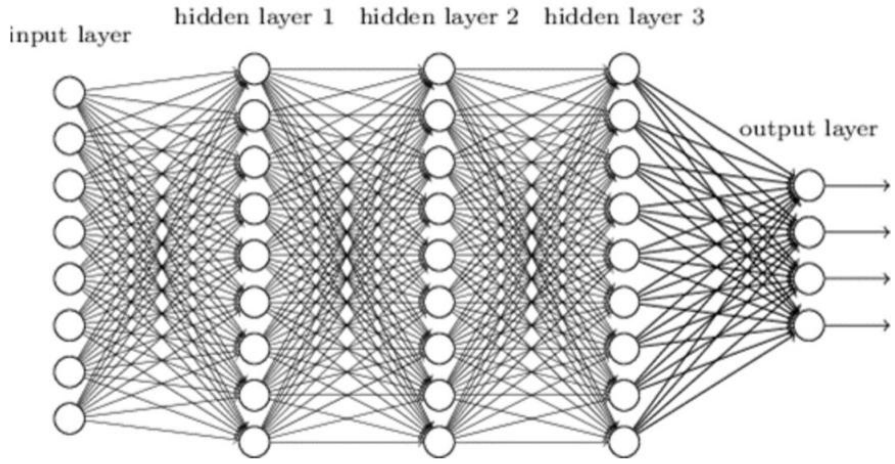
神经网络之 是什么

添加少量隐层 => 浅层神经网络



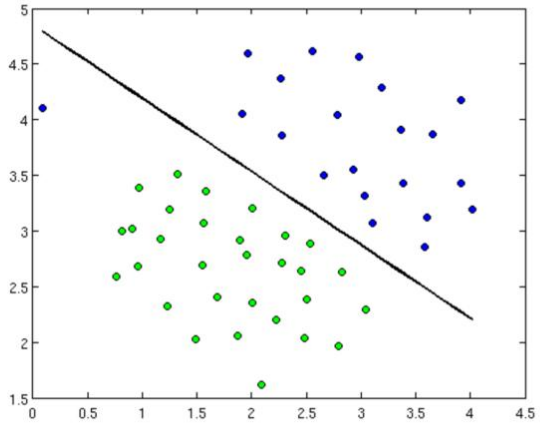
神经网络之 是什么

增多中间层 => 深度神经网络(DNN)



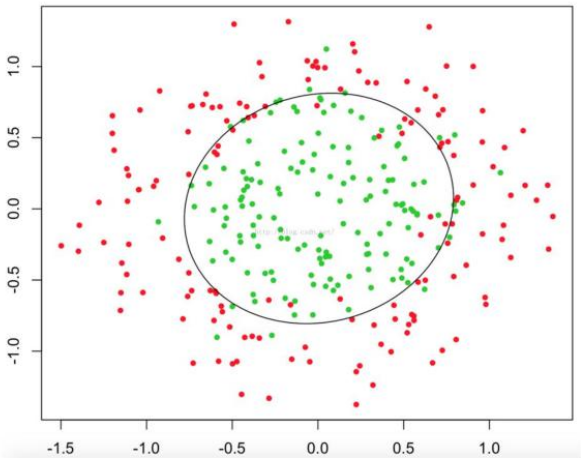
神经网络之 为什么

- 神经网络应用在分类问题中效果好
- LR或者linear SVM，线性分割



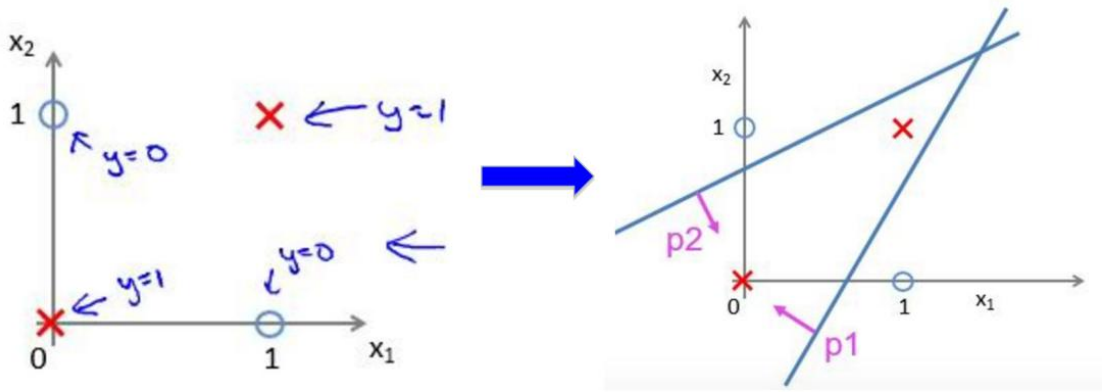
神经网络之 为什么

- LR和SVM对于非线性可分，怎么处理的？
- 为什么不用它们？



神经网络之 为什么

- 非线性可分，怎么办？

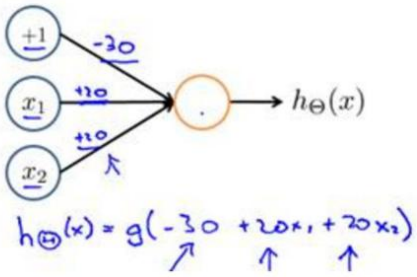


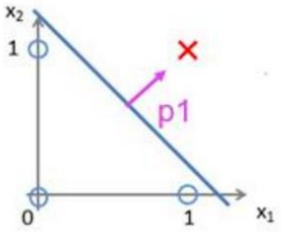
神经网络之 为什么

- 神经元完成『逻辑与』

$x_1, x_2 \in \{0, 1\}$

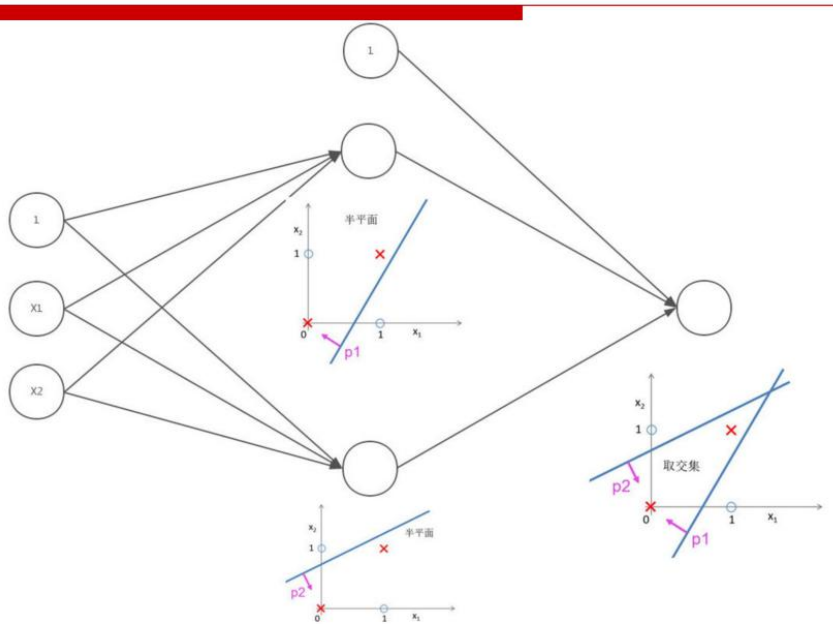
$y = x_1$ AND x_2





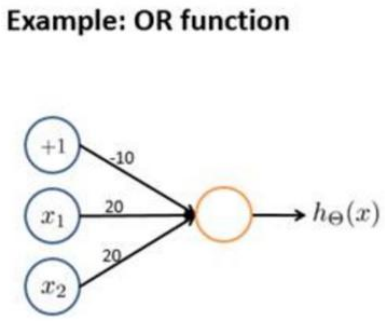
x_1	x_2	$h_\Theta(x)$
0	0	0
0	1	0
1	0	0
1	1	1

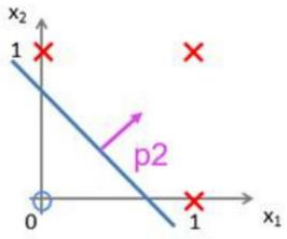
神经网络之 为什么



神经网络之 为什么

- 神经元完成『逻辑或』

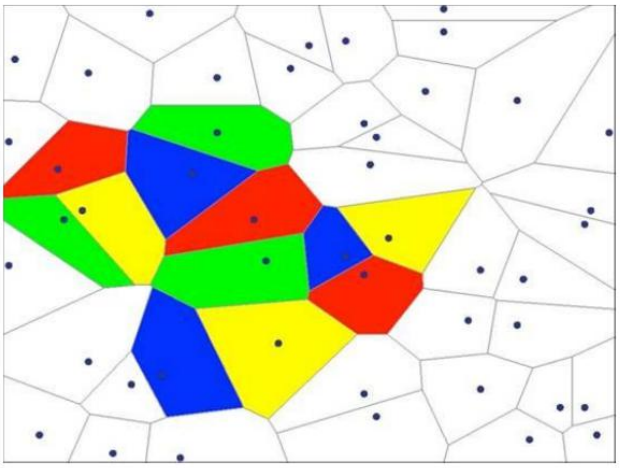




x_1	x_2	$h_\Theta(x)$
0	0	0
0	1	1
1	0	1
1	1	1

神经网络之 为什么

- 对线性分类器的『与』和『或』的组合
- 完美对平面样本点分布进行分类



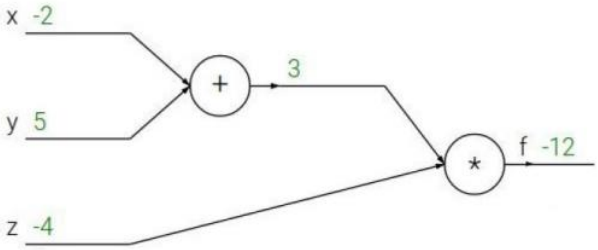
反向传播

$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

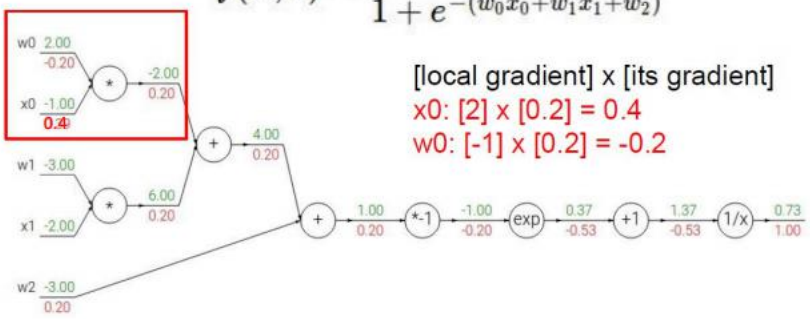
$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$

$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$



Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]
x0: [2] x [0.2] = 0.4
w0: [-1] x [0.2] = -0.2

$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

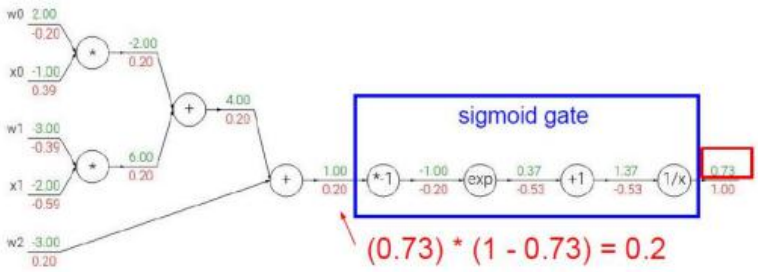
$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

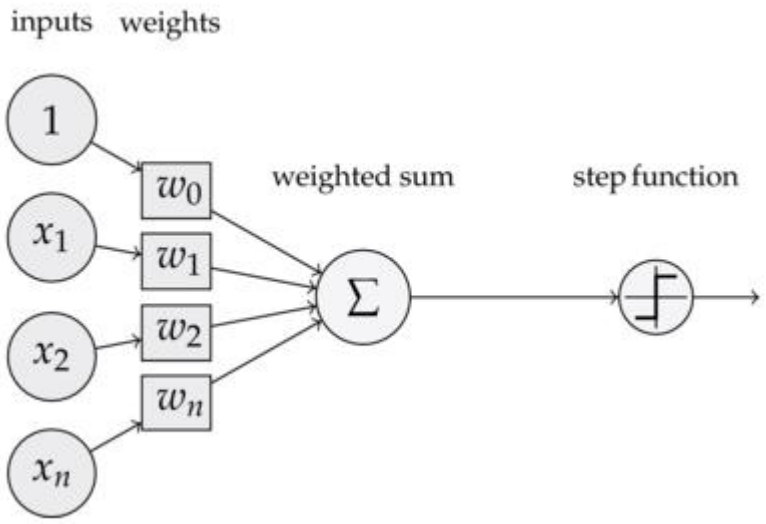
$$\sigma(x) = \frac{1}{1+e^{-x}}$$ sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



1 感知器

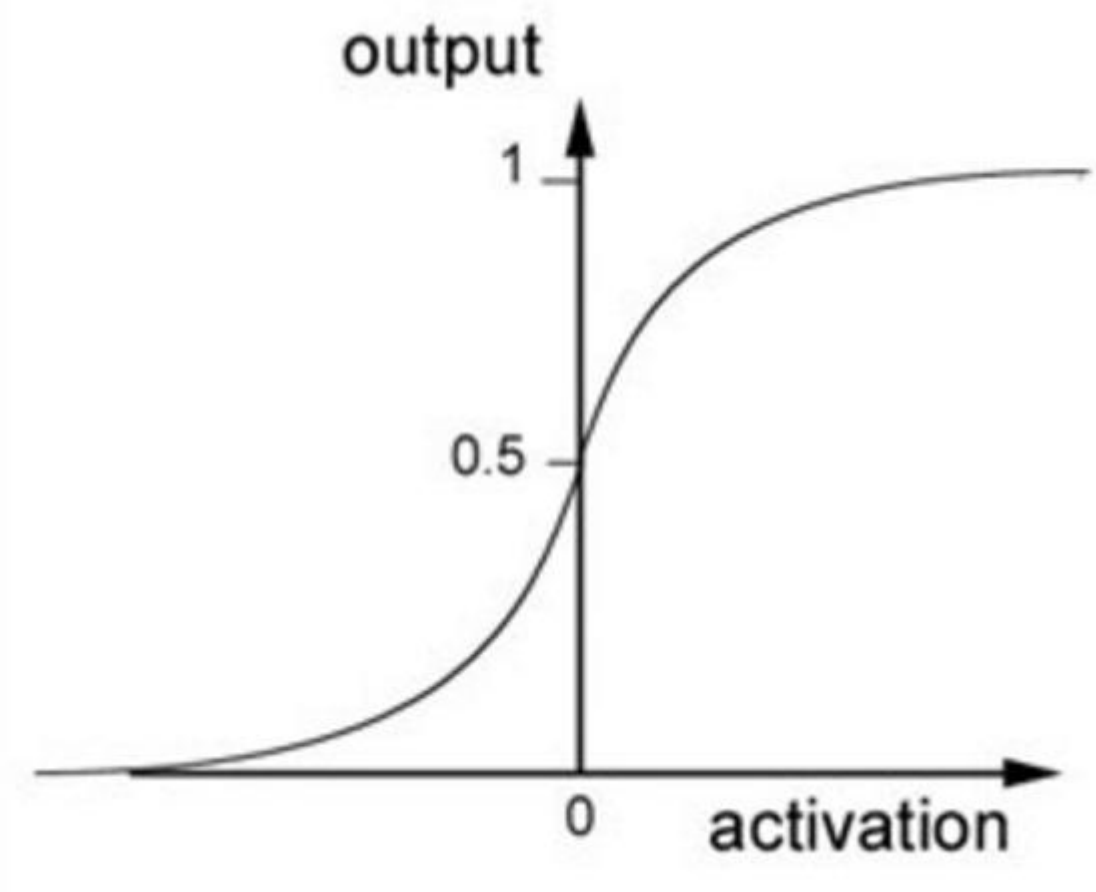
为了理解神经网络，我们应该先理解神经网络的组成单元——神经元。神经元也叫做感知器。还记得之前的线性回归模型中权重的作用吗？每一个输入值与对应权重的乘积之和得到的数据或通过激活函数来进行判别。下面我们看一下感知器：



可以看到，一个感知器有如下组成部分：

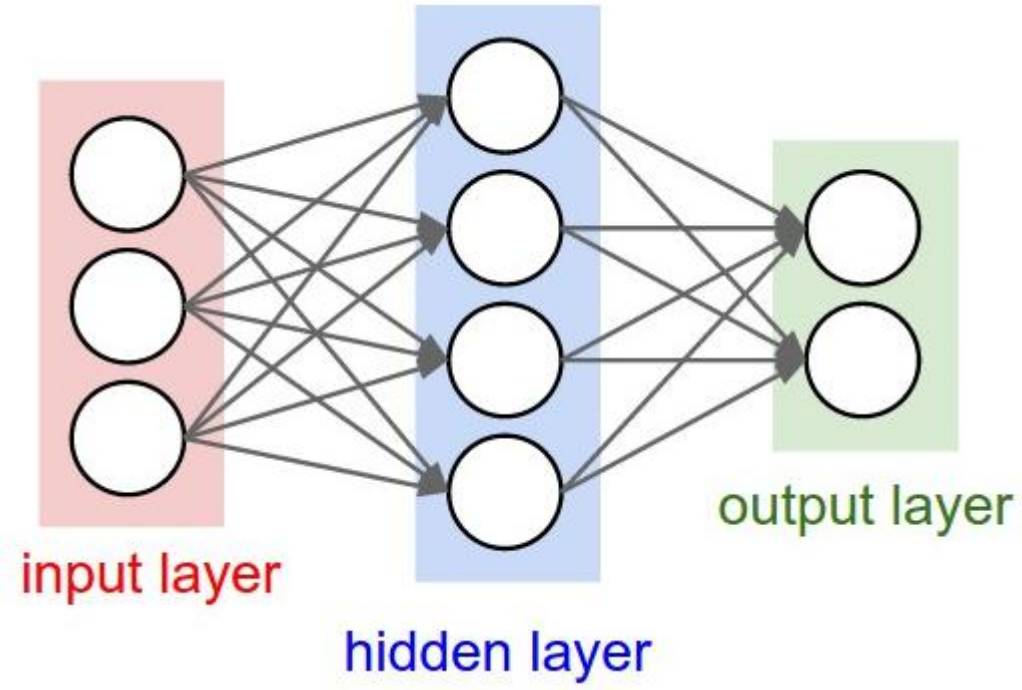
- 输入权值，一个感知器可以有多个输入 $x1 \quad ,x2 \quad ,x3 \quad ...xn$ 每个输入上有一个权值 $w\,i$
- 激活函数 ,感知器的激活函数有许多选择 ,以前用的是阶跃函数，$sigmoid\left(\frac{1}{1+e^{w*x}}\right)$，其中 z$z$ 为权重数据积之和
- 输出，$y=f(w*x+b)$

我们了解过 sigmoid 函数是这样,在之前的线性回归中它对于 **二类分类** 问题非常擅长。所以在后续的多分类问题中，我们会用到其它的激活函数



2 神经网络

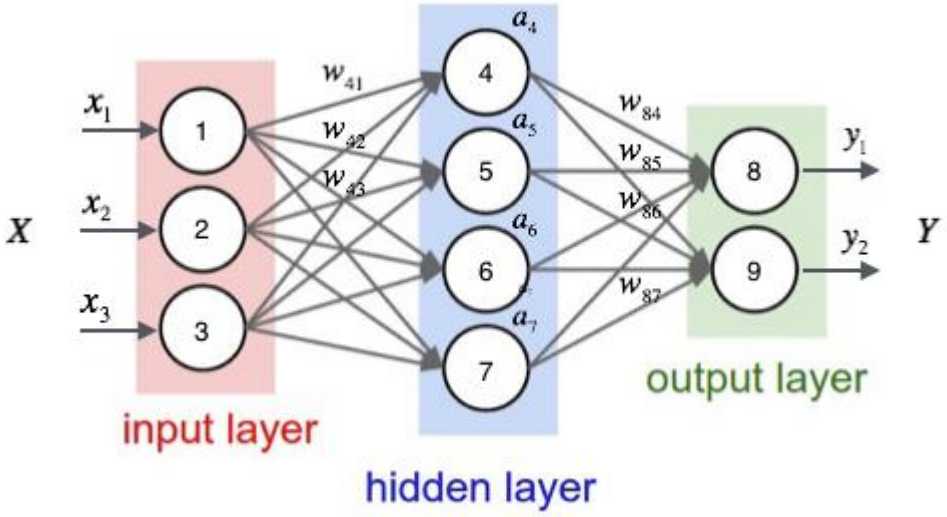
那么我们继续往后看，神经网络是啥？



神经网络其实就是按照一定规则连接起来的多个神经元。

- 输入向量的维度和输入层神经元个数相同
- 第 N 层的神经元与第 N-1 层的所有神经元连接，也叫 **全连接**
- 上图网络中最左边的层叫做输入层，负责接收输入数据；最右边的层叫输出层，可以有多个输出层。我们可以从这层获取神经网络输出数据。输入层和输出层之间的层叫做隐藏层，因为它们对于外部来说是不可见的。
- 而且同一层的神经元之间没有连接
- 并且每个连接都有一个权值，

那么我们以下面的例子来看一看，图上已经标注了各种输入、权重信息。



对于每一个样本来说，我们可以得到输入值 $x1 \quad ,x2 \quad ,x3 \quad$ ，,也就是节点 1，2，3 的输入值，那么对于隐层每一个神经元来说都对应有一个偏置项 b，它和权重一起才是一个完整的线性组合

$$a4=sigmoid\,(W_{41}*x1+W_{42}*x2+W_{43}*x3+W_{4b})$$

$$a5=sigmoid\,(W_{51}*x1+W_{52}*x2+W_{53}*x3+W_{5b})$$

$$a6=sigmoid\,(W_{61}*x1+W_{62}*x2+W_{63}*x3+W_{6b})$$

$$a7=sigmoid\,(W_{71}*x1+W_{72}*x2+W_{73}*x3+W_{7b})$$

这样得出隐层的输出，也就是输出层的输入值.

矩阵表示

$$\vec{a}=\begin{bmatrix}a_4\\a_5\\a_6\\a_7\end{bmatrix},\qquad W=\begin{bmatrix}\vec{w}_4\\\vec{w}_5\\\vec{w}_6\\\vec{w}_7\end{bmatrix}=\begin{bmatrix}w_{41},w_{42},w_{43},w_{4b}\\w_{51},w_{52},w_{53},w_{5b}\\w_{61},w_{62},w_{63},w_{6b}\\w_{71},w_{72},w_{73},w_{7b}\end{bmatrix},\qquad f(\begin{bmatrix}x_1\\x_2\\x_3\\.\\.\\.\end{bmatrix})=\begin{bmatrix}f(x_1)\\f(x_2)\\f(x_3)\\.\\.\\.\end{bmatrix}$$

同样，对于输出层来说我们已经得到了隐层的值，可以通过同样的操作得到输出层的值。那么重要的一点是，**分类问题的类别个数决定了你的输出层的神经元个数**