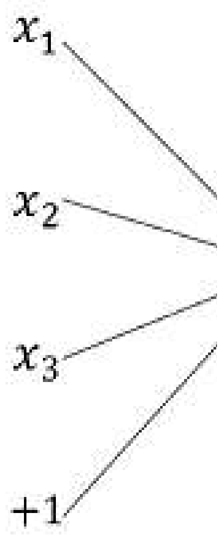
深度学习算法原理——神经网络的基本原理

一、神经网络

1、神经元概述

神经网络是由一个个的被称为“神经元”的基本单元构成，单个神经元的



对于上述的神经元，其输入为x_1，x_2，x_3以及截距$+1$，其输出为：

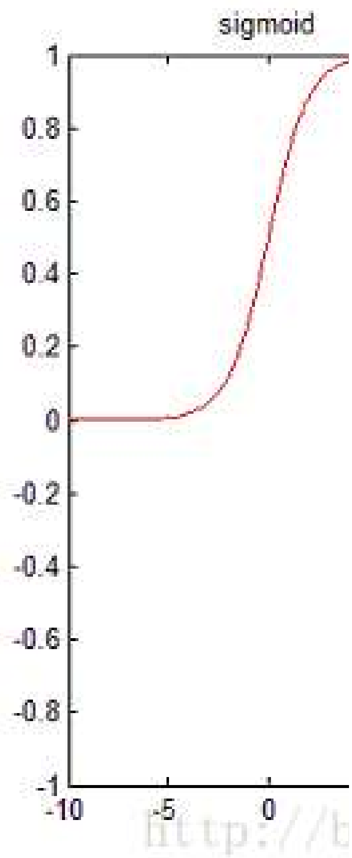
$$h_{\mathbf{W},b}\left(\mathbf{x}\right)=f$$

其中，$\mathbf{W}$表示的是向量，代表的是权重，函数f称为激活函数，通常激活

双曲正切函数的形式为：

$$f(z) =$$

以下分别是Sigmoid函数和tanh函数的图像，左边为Sigmoid函数的图像



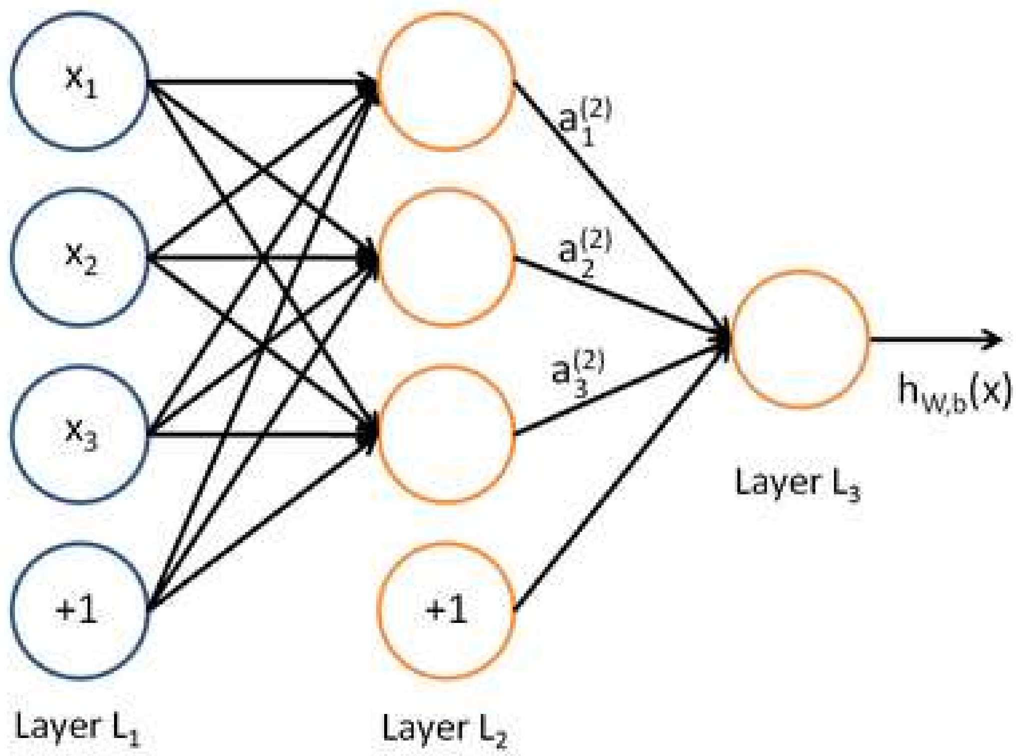
> Sigmoid函数的区间为$[0, 1]$，而tanh函数的区间为$[-1, 1]$。

若是使用sigmoid作为神经元的激活函数，则当神经元的输出为1时表示该
表示该神经元被激活，否则称为未被激活。

2、神经网络

2.1、神经网络的结构

神经网络是由很多的神经元联结而成的，一个简单的神经网络的结构如下



其中一个神经元的输出是另一个神经元的输入，$+1$项表示的是偏置项。
出层。

2.2、神经网络中的参数说明

在神经网络中，主要有如下的一些参数标识：

- 网络的层数n_1。在上述的神经网络中$n_l = 3$，将第l层记为L_l，则
- 网络权重和偏置$(\mathbf{W}, \mathbf{b}) = \left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\right)$，其中$W_{ij}^{(l}$
  $l+1$层的第i个神经元的偏置项。在上图中，$\mathbf{W}^{(1)} \in \Re^{3\times 3}$，$\mathbf{W}^{(2}$

2.3、神经网络的计算

在神经网络中，一个神经元的输出是另一个神经元的输入。假设$z_i^{(l)}$表示的
$l=1$时，$a_i^{(1)}=x_i$。根据上述的神经网络中的权重和偏置，就可以计算

对于上述的神经网络结构，有下述的计算：

$$z_1^{(2)} = W_{11}^{(1)}x$$

$$a_1^{(2)} = f\left(W_{11}^{(1)}\right.$$

$$z_2^{(2)} = W_{21}^{(1)}x$$

$$a_2^{(2)} = f\left(W_{21}^{(1)}\right.$$

$$z_3^{(2)} = W_{31}^{(1)}x$$

$$a_3^{(2)} = f\left(W_{31}^{(1)}\right.$$

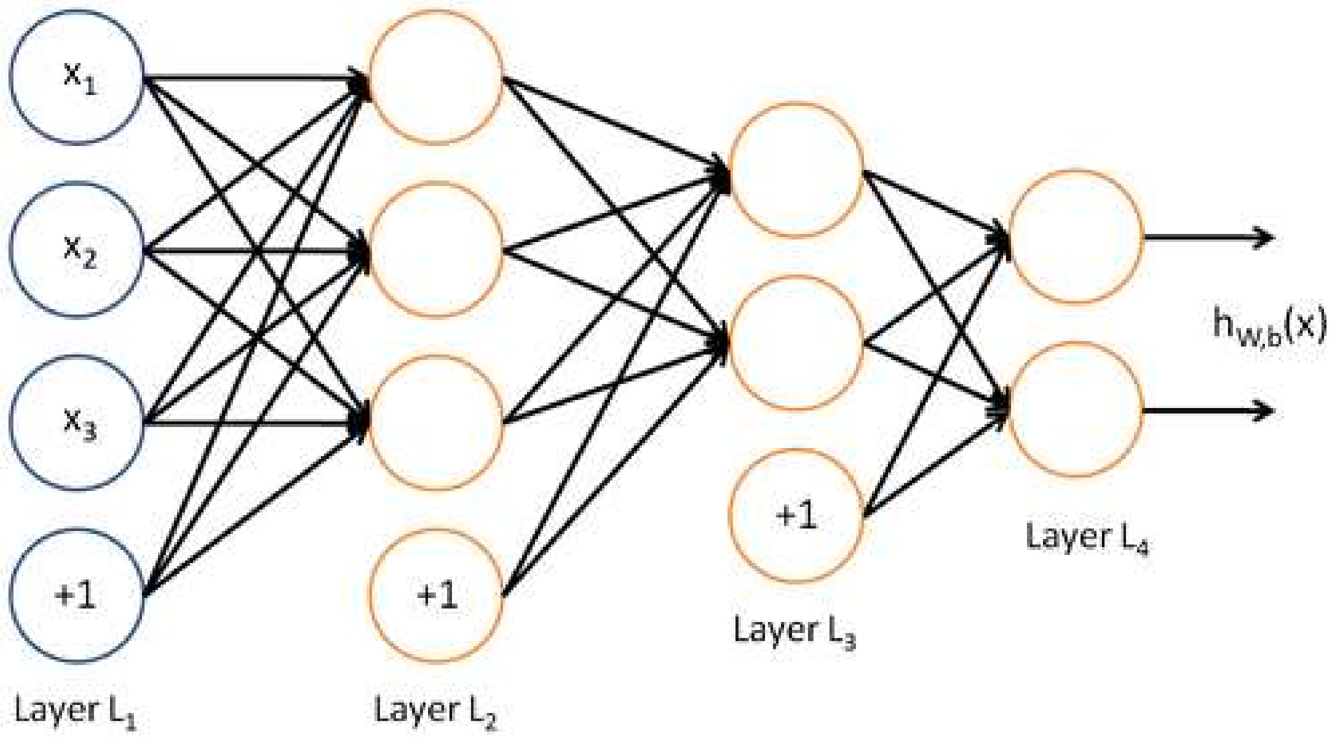
从而，上述神经网络结构的最终的输出结果为：

$$h_{\mathbf{W},\mathbf{b}}\left(\mathbf{x}\right) = f\left(W_{11}^{(2)}\right.$$

上述的步骤称为**前向传播**，指的是信号从输入层，经过每一个神经元，直

2.4、其他形式的神经网络模型

上述以单隐层神经网络为例介绍了神经网络的基本结构，在神经网络的结
输出单元的神经网络模型：



2.5、神经网络中参数的求解

对于上述神经网络模型，假设有m个训练样本$\left\{ \left(\mathbf{x}^{(1)}, y^{(1)} \right), \cdots, \left(\mathbf{x}^{(m)}\right.\right.$

$$J\left(\mathbf{W}, \mathbf{b};\right.$$

为了防止模型的过拟合，在损失函数中会加入正则项，即：

其中，$loss$表示的是损失函数，R表示的是正则项。则对于上述的含有m

$$J\left(\mathbf{W}, \mathbf{b}\right) = \left[\frac{1}{m} \sum_{i=1}^{m} J\left(\right.\right.$$

> 通常，偏置项并不放在正则化中，因为在正则化中放入偏置项只会对神经网络产

我们的目标是要求得参数$\mathbf{W}$和参数$\mathbf{b}$以使得损失函数$J\left(\mathbf{W},\mathbf{b}\right)$达到最小

> 参数的初始化有很多不同的策略，基本的是要在0附近的很小的邻域内取得随机

在随机初始化参数后，利用**前向传播**得到预测值$h_{\mathbf{W},\mathbf{b}}\left(\mathbf{x}\right)$，进而可以得
降对参数的调整如下：

$$W_{ij}^{(l)} =$$

$$b_i^{(l)} =$$

其中，α称为学习率，在计算参数的更新公式中，需要使用到**反向传播算**

而$\frac{\partial}{\partial W_{ij}^{(l)}}J\left(\mathbf{W},\mathbf{b}\right)$，$\frac{\partial}{\partial b_i^{(l)}}J\left(\mathbf{W},\mathbf{b}\right)$的具体形式如下：

$$\frac{\partial}{\partial W_{ij}^{(l)}}J\left(\mathbf{W},\mathbf{b}\right)=\left[\frac{1}{m}\right.$$

$$\frac{\partial}{\partial b_i^{(l)}}J\left(\mathbf{W},\mathbf{b}\right)=$$

反向传播算法的思路如下：对于给定的训练数据$\left(\mathbf{x},y\right)$，通过**前向传播算**
其“残差”，如第l层的神经元i的残差可以表示为$\delta_i^{(l)}$。该残差表示的是该
二是神经元为非输出神经元。这里假设$z_i^{(l)}$表示第l层上的第i个神经元的输

- 对于输出层n_l上的神经元i，其残差为：

$$\delta_i^{(n_l)}$$

$$= \frac{}{\partial}$$

$$= \frac{}{\partial z}$$

$$= \left(y_i\right.$$

$$= -$$

-对于非输出层，即对于$l = n_{l-1}, n_{l-2}, \cdots, 2$各层，第$l$层的残差的计算

$$\delta_i^{(n_l-1)}$$

$$= \frac{}{\partial z}$$

$$= \frac{\partial}{\partial z_i}$$

$$= \frac{1}{2} \sum$$

$$= \frac{1}{2} \sum_{j=1}^{s_{n_l}}$$

$$= \sum$$

$$= \sum_{j=1}^{s_{n_l}} \left(\delta_j^{(n_l)} \right.$$

$$= \sum_{j=1}^{s_{n_l}}$$

$$= \left(\sum_{j=}^{s_{n_l}} \right.$$

因此有：

$$\delta_i^{(l)} = \left(\sum_{j} \right.$$

对于神经网络中的权重和偏置的更新公式为：

$$\frac{\partial}{\partial W_{ij}^{(l)}}$$

$$\frac{\partial}{\partial b_i^{(l)}}$$

2.6、神经网络的学习过程

对于神经网络的学过程，大致分为如下的几步：

- 初始化参数，包括权重、偏置、网络层结构，激活函数等等
- 循环计算
    - 正向传播，计算误差
    - 反向传播，调整参数
- 返回最终的神经网络模型

参考文献

1、英文版：UFLDL Tutorial

2、中文版：UFLDL教程