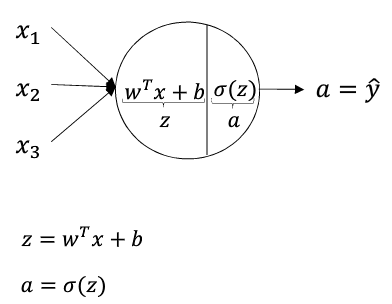
神经网络构建

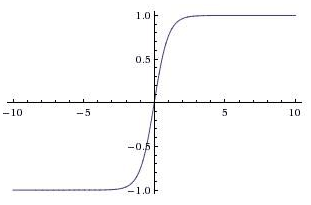
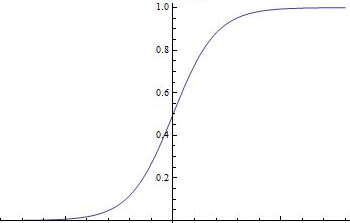
一、逻辑回归中的线性函数与激活函数

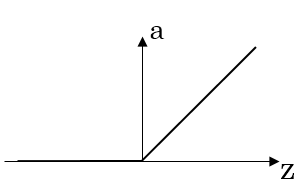
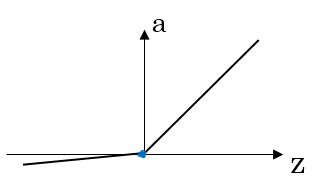
一个典型的逻辑回归可以看作是神经网络中上一层的每个神经元与当前层一个神经元的线性函数与激活函数构成（忽略维度）。



线性函数为简单，这里需要注意的是传统机器学习中样本X的矩阵形式按行为一个样本，而深度学习中按列为一个样本。

激活函数一般采用非线性激活函数，如果采用线性函数是没有意义的。常用的激活函数有sigmoid，tanh，relu，leaky relu函数，下面是几个激活函数图，在不同网络层选择合适的激活函数也是调节神经网络的重要参数（超参）。







这里需要关心的是线性函数与激活函数本身以及其导数，因为前向传播就是线性函数与激活函数的计算，反向传播就是反向求损失函数，激活函数与线性函数的导数。下面以一个简单的逻辑回归为例，损失函数为对数损失，逻辑回归的对数损失解释可以最大似然推导出来。



损失函数的导数为



Sigmoid函数导数：



Tanh函数导数：

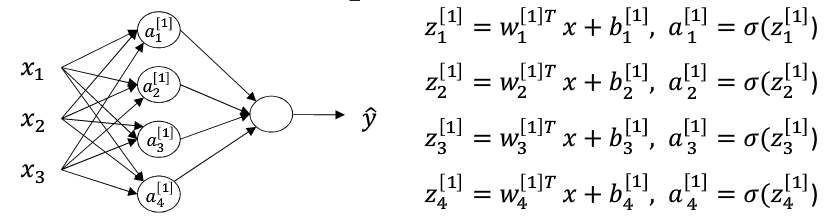


Relu函数导数：



2神经网络结构与前向反向传导

逻辑回归可以看做是一个没有隐藏层的神经网络，如果忽略输入特征，那么就只有输出层一个神经元。而神经网络是一个多层，每一层有多个神经元的网络结构。如果说逻辑回归是一个线性函数与激活函数的复合函数，那么神经网络就是多个这样的神经元的复合函数。在计算过程中，与逻辑回归不同的地方是前向传导需要计算L次，同样反向传播也是**损失函数**对参数求导计算L次。同时为了计算方便，后期在神经网络计算中统一采用矩阵形式。

前向传导：

每个神经元与上一层神经元的函数关系，都是一个线性函数与激活函数的复合函数。

矩阵表示形式如下：

![C:\Users\Administrator\AppData\Roaming\Tencent\Users\869233039\QQ\WinTemp\RichOle\(@377@3_]_[I7E](0[8I8J1.png](data:image/png;base64,)

矩阵表示每一层之间的关系，也是前向传导的计算公式,所以每一层前向传导过程可以分为两步，先对线性函数进行计算，然后对激活函数进行计算，上一层的激活值作为下一层的输入特征。

这里特别注意的是矩阵表示时，X，Y，W，b，Z，A的矩阵行列,其中X,Y,W,b是数据初始化的，Z，A是计算过程中产生的，最为常见的错误就是矩阵大小不匹配。令A0=X，那么可以看作第一层神经元个数为特征维度，那么有以上矩阵的大小为：



反向传播：

反向传播算法的核心就是一个链式求导，**损失函数**对每一层网络神经元中的参数求偏导，而且神经网络具有表达的形式循环性，使得可以使用循环来进行链式求导，当采用矩阵形式表示时，每一层求导都在一次for循环中完成。下面求两层网络的反向求导过程。

损失函数同样采用对数损失，输出层采用sigmoid，隐藏层采用relu激活函数，同样这里采用矩阵形式表示。

损失函数对预测值AL求偏导：



可见AL，Y都是一个行向量。

激活函数求导：



线性函数求导：



可见dZ与Al-1T乘积是n个样本在一个维度上的反映，所以Z对W求导前面有1/n，而b则是dZ按行求和乘1/n。



除输出层后每层网络重复求如下偏导：



所以每一层神经网络的反向求导都可以分成两步来做，第一步是对激活函数求导，第二步是对线性函数求导，当然输出层AL的导数需要单独根据损失函数定义来求导，而且每一层循环求导Al-1也只是为了联系上一层网络，所以直到输入层时，对A0（X）求导可有可无。

\*\*\*这里有个重点，采用链式求导是最直观的理解，但是很多时候我们并不需要分别计算A，Z的导数，dAdZ可以一步完成（当然与激活函数有关），加快计算效率，这里dAdZ简称残差。对于求导公式不必死记，只需要记住根据复合函数，进行链式求导。

反向传播求导的目的是为了实现梯度下降算法，即往负梯度方向求解最小值。所以参数更新表达式如下：



这里需要指出的是W，b是指整个网络中所有的W，b。

3加载数据

以deeplearning.ai教程中提供的数据集为例。

每一张图片数据存储都是64\*64\*3的格式，所以整个训练集是一个209\*64\*64\*3的矩阵。所以在加载数据之后，需要对原始的数据进行生成列为样本，行为多个样本的形式。将每一张64\*64\*3图片拉成一个列向量。

4网络初始化

神经网络中超参有，网路层数，每层神经元个数，激活函数的选择，学习速率，迭代次数，参数有W，b。

参数的初始化依赖于超参的定义， 定义好网络层数与每一层神经元个数，每一层统一使用的激活函数。可以对W，b进行初始化。

5算法设计流程

1 加载数据，数据预处理

2 参数初始化

3梯度下降

1）前向传导

2）计算损失

3）反向传播

4）更新参数

4预测