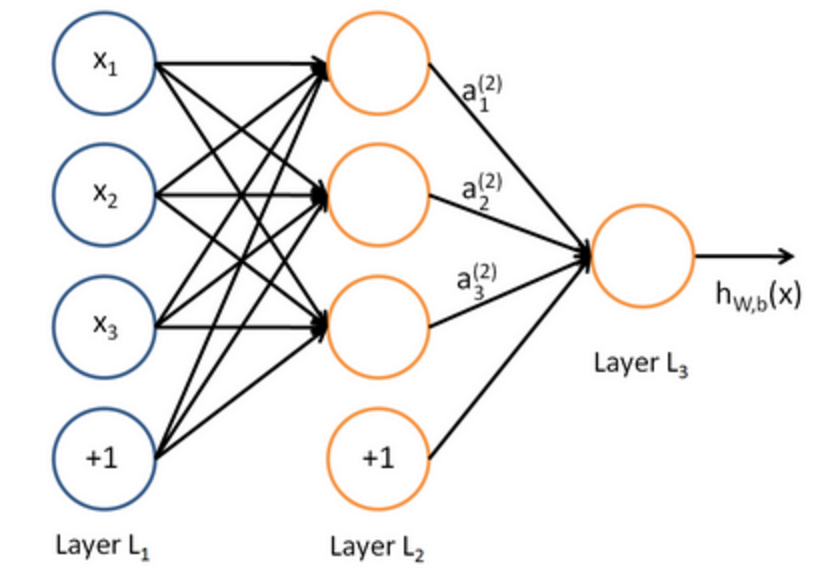
1. 神经网络模型：<http://ufldl.stanford.edu/wiki/index.php/%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>

layer1x为输入；

layer3为输出h(x), w:weight, b:bias h(x) = W\*X + B (线性或逻辑回归)；

layer2为hidden layer



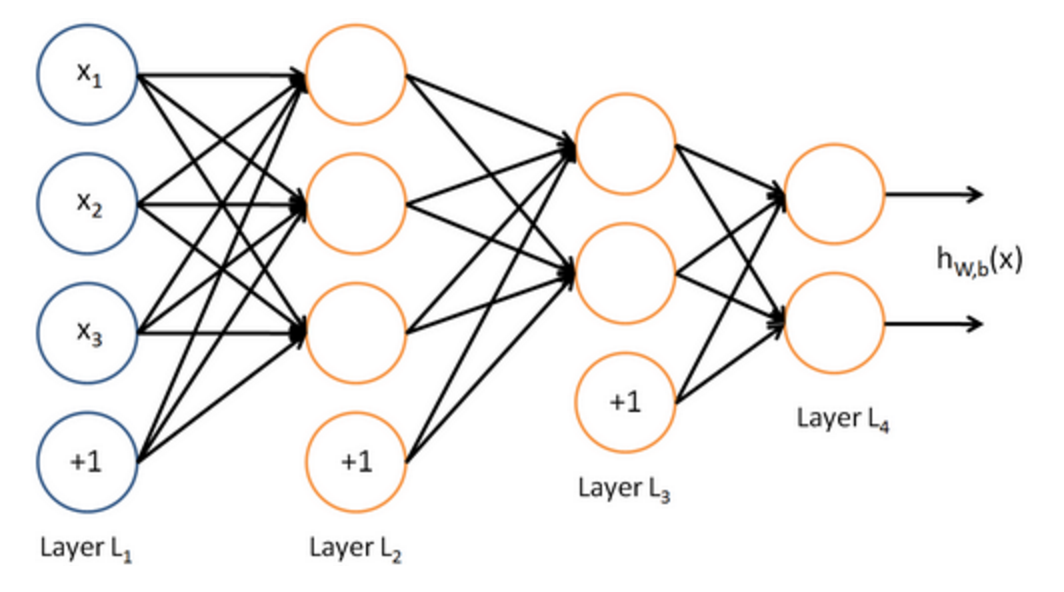
2. 每个节点的计算方式（Layer2和Layer3一样）

线性回归：Linear Regression

逻辑回归：Logistic Regression （函数包括：sigmoid(函数输出［0，1］), tanh(函数输出［－1，1］), softmax(函数输出［1，...，K］)）

评价回归函数的好坏的标准是：Gradient Descent(梯度下降)的速度，越快越好。

3. Deep Neural Network（DNN）和传统的NN的区别：有多个层。



无论是NN还是DNN都是Feed-Forward NN（前馈型网络），图像识别中的CNN(卷积神经元网络)也是FFNN。由于数据量很多，所以feature（x1,x2, ... xn就是feature的向量）不能太多。如果feature太多，计算量会很大，需要降维（主成分分析：PCA）

4. FFNN无法处理语言识别，所以出现了RNN，具有反馈的NN，更进一步出现了LSTM。这篇文章写的很好：<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

5. 自主学习（self-encoding）：

以上都是有监督学习。在有监督学习中，训练样本是有类别标签的。现在假设我们只有一个没有带类别标签的训练样本集合

C:\Users\hxu\AppData\Local\YNote\data\xhsoldier@163.com\baa7cecb10fb45089ab9c189cfaf8400\ec377e100ed2.png

 ，其中

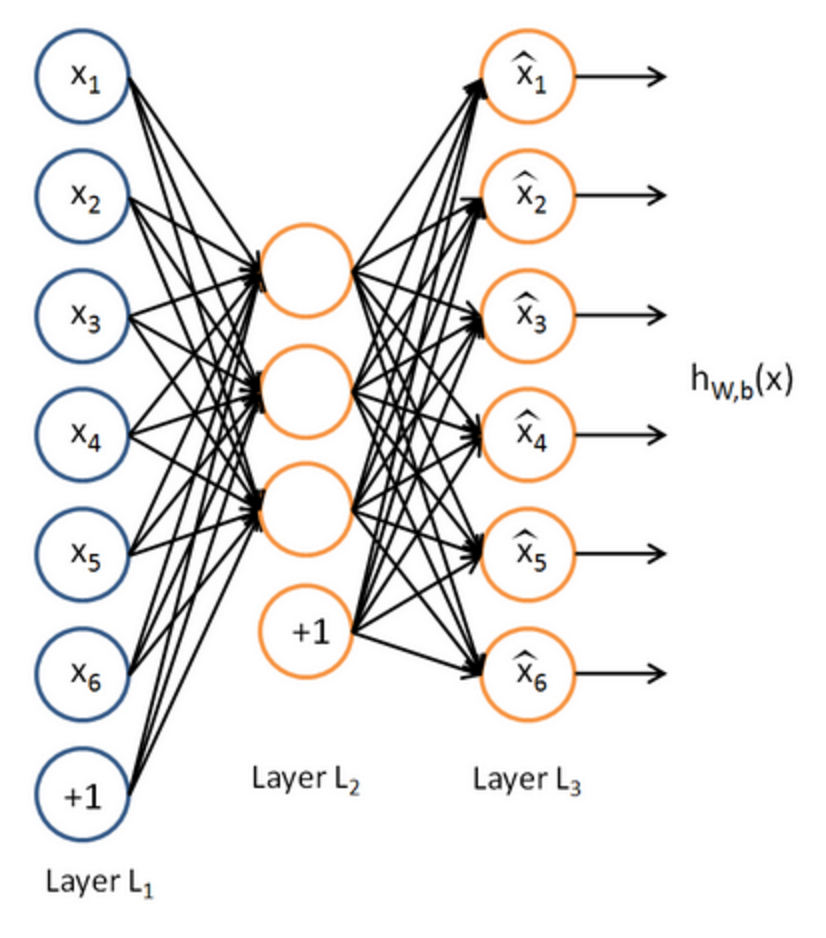
C:\Users\hxu\AppData\Local\YNote\data\xhsoldier@163.com\a98ca3975f674c7a956984fe0c98ac0c\a849465dd4a2.png

 。自编码神经网络是一种无监督学习算法，它使用了反向传播算法，并让目标值等于输入值，比如

C:\Users\hxu\AppData\Local\YNote\data\xhsoldier@163.com\1ecac013ce484861a990c75fe6714c7f\522f86679912.png

 。

恒等函数虽然看上去不太有学习的意义，但是当我们为自编码神经网络加入某些限制，比如限定隐藏神经元的数量，我们就可以从输入数据中发现一些有趣的结构。事实上，这一简单的自编码神经网络通常可以学习出一个跟主元分析（PCA）结果非常相似的输入数据的低维表示。



Lyaer2就成了feature，该网络学习了输入数据。

5. 谷歌的Tensor Flow封装了上诉的各种网络，主要支持python，部分支持C++： <https://www.tensorflow.org>

6. 斯坦福的吴恩达（现在在百度）的在线学习教程很好： <http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=DeepLearning>