# Neural Networks: Representation

Di Zhao, 2016-4-5

zhaodi01@mail.ustc.edu.cn

## 1 Non-linear Hypotheses

考虑一个较复杂的分类问题，例如有100个特征。那么如果要包含二次方项，例如$x_1^2, x_1x_2$ 等，需要计算$O(n^2)$个特征约5000项；如果要计算三次方项，则需要计算$O(n^3)$个特征，约130000项。例如计算机视觉问题，按照每个像素的亮度作为特征，则$50 \times 50$分辨率就有2500个特征。

=> 使用非线性假设

## 2 Neurons and the Brain

Neural Networks: Origins - Algorithms that try to mimic the brain. Widely used in 80s and 90s; popularity diminished in late 90s.

Now: state-of-the-art technique for many applications.

neuro-rewiring experiments: 切断听觉和触觉皮质与相应器官的连接后，与视觉器官相连，则这些皮质最终会学会处理视觉信号。

同一块脑组织可学会处理听觉、视觉、触觉信号=> 同一个算法也许可以处理这些不同任务

生物学的神经元（Figure 1）

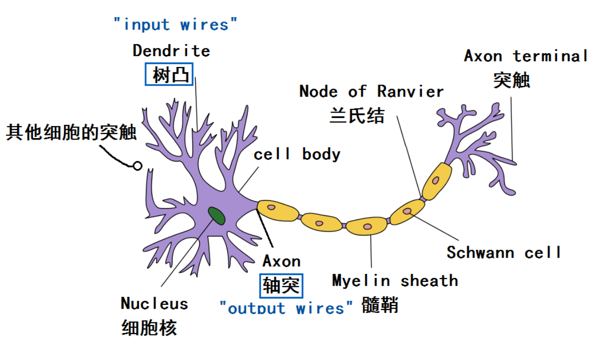


Figure 1: Neuron

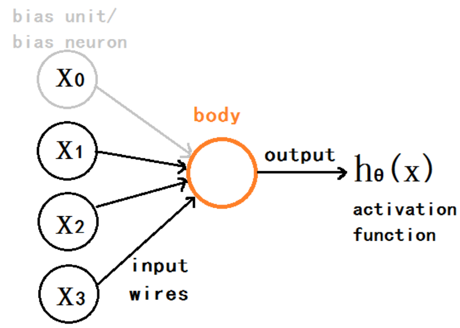


Figure 2: Neuron model: logistic unit

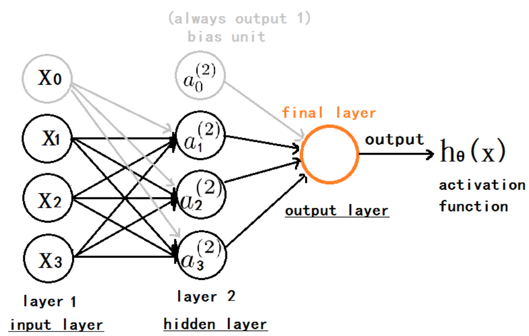


Figure 3: Neuron Network

# 3 Model Representation

在计算机实现上，将一个神经元建模为一个逻辑单元（Figure 2）.

其中，$x = [x_0\ x_1\ x_2\ x_3]^T$, $\theta = [\theta_0\ \theta_1\ \theta_2\ \theta_3]^T$。parameter $\theta$ 在神经网络中也称为"weights".

神经网络就是将单个的神经元紧密联系在一起（Figure 3）.

记号：

$a_i^{(j)}$ = "activation" of unit $i$ in layer $j$

$\Theta^{(j)}$ = matrix of weights controlling function mapping from layer $j$ to layer $j+1$

则：

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$

$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

If network has $s_j$ units in layer $j$, $s_{j+1}$ units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$.

因此，对于Figure 3中的例子，有3个输入单元、3个hidden units，$\Theta^{(1)} \in \mathbb{R}^{3\times 4}$.

## 3.1 Forward propagation - vectorized

令 $a^{(1)} = x = [x_0\ x_1\ x_2\ x_3]^T$, 令

$$z^{(2)} = \Theta^{(1)}x = \Theta^{(1)}a^{(1)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} = \begin{bmatrix} \Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3 \\ \Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3 \\ \Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3 \end{bmatrix}$$

（上标表示相关的layer）。

则$a^{(2)} = g(z^{(2)}) \in \mathbb{R}^3$.($g$作用于矩阵每个元素)

Add $a_0^{(2)} = 1$ (则$a^{(2)} \in \mathbb{R}^4$):

于是$z^{(3)} = \Theta^{(2)}a^{(2)}$,
$h_\Theta(x) = a^{(3)} = g(z^{(3)})$.
这一过程称为forward propagation。

## 3.2 Neural network learning its own features

从Figure 3中可以看到，神经网络和逻辑回归比较相似。区别是逻辑回归中使用原有的特征作为输入；而神经网络使用训练结果前一层的作为下一层的输入 => learn its own features. 通过这种方式可以学习到一些更复杂有趣的特征，得到更好的假设。

network *architecture*: 神经元之间如何连接。(分为几层、每层几个节点)