

神经网络基本介绍&

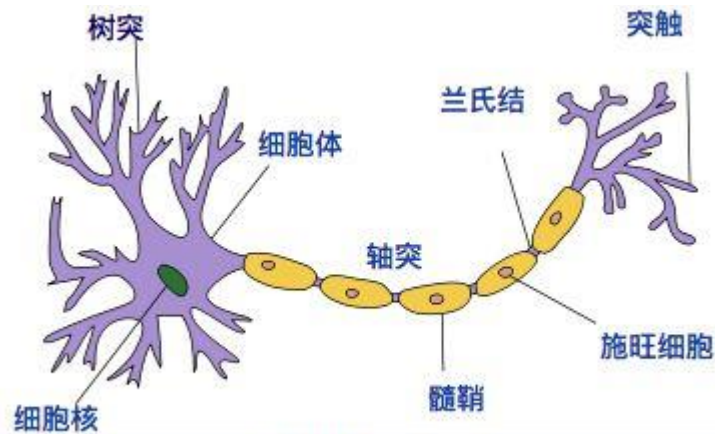
Homework challenge:应用神经网络处理多项选择模型

刘晶芳 (15320171151900) 余星月 (15320171151888)

一、神经元

(1) 为什么叫做神经元?

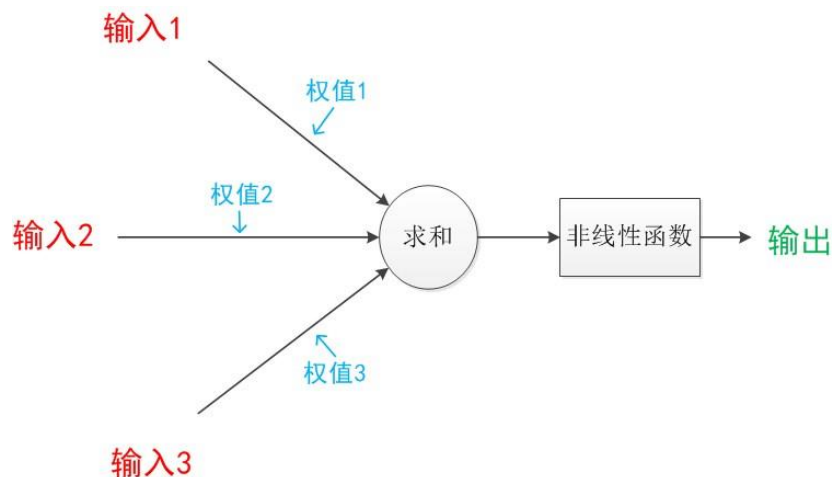
人体的一个神经元通常具有多个树突,主要用来接受传入信息;而轴突只有一条,轴突尾端有许多轴突末梢可以给其他多个神经元传递信息。轴突末梢跟其他神经元的树突产生连接,从而传递信号。这个连接的位置在生物学上叫做“突触”



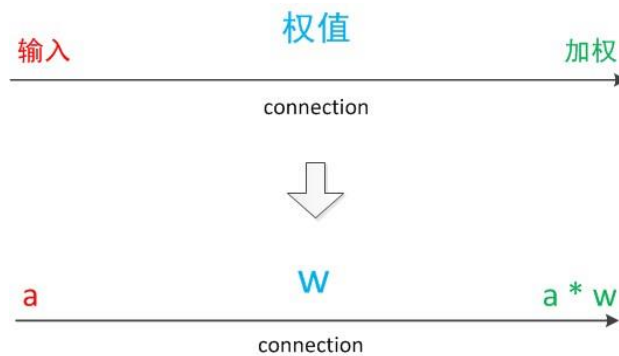
典型神经元的结构

而一个典型的神经元模型是一个包含输入,输出与计算功能的模型。输入可以类比为神经元的树突,而输出可以类比为神经元的轴突,计算则可以类比为细胞核。

(2) 神经元结构



这是一个典型的神经元模型: 包含有 3 个输入, 1 个输出, 以及 2 个计算功能。箭头线被称为连接, 连接是神经元中最重要的东西。每一个连接上都有一个权重。使用 a 来表示输入, 用 w 来表示权值。一个表示连接的有向箭头可以这样理解: 在初端, 传递的是 a , 端中间有加权参数 w , 经过这个加权后的信号会变成 $a*w$, 因此在连接的末端, 信号的大小就变成了 $a*w$ 。



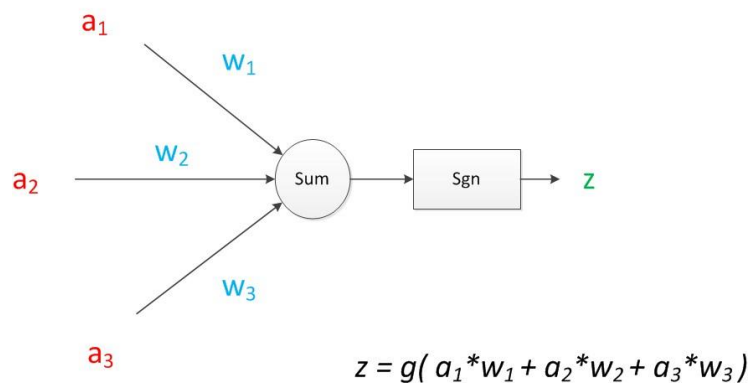
在神经元模型里，每个有向箭头表示的是值的加权传递。

二、单层神经网络

感知器是一种二元分类器，它是神经网络的基石。

设置输入向量： $a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$ ， a_0 是偏置单元 (bias unit)，相当于线性回归中的常

数项。在经过激活函数的计算后，感知器会输出一个大于 0 或小于 0 的数。箭头上的数字代表每个特征的权量 (weights)，相当于线性回归模型的参数。



我们有这样的模型

$$z = g(a_1 * w_1 + a_2 * w_2 + a_3 * w_3)$$

z 是在输入和权值的线性加权和叠加了一个函数 g 的值。在 MP 模型里，函数 g 是 sgn 函数，也就是取符号函数。这个函数当输入大于 0 时，输出 1，否则输出 0。这实际上是逻辑与或。这个函数 g 即为一个单层的感知器。

$$g(x) = \begin{cases} 1 : \text{if } a_1 * w_1 + a_2 * w_2 + a_3 * w_3 > 0 \\ 0 : \text{otherwise} \end{cases}$$

单层感知器无法来建立 (XOR)，这种情况下，就需要多层神经网络。

多层神经网络是多个感知器的组合，这些感知器以不同方式产生连接，并由不同的激活函数控制激活。

三、两层神经网络 (多层感知器)

发展

Marvin Minsky 在 1969 年出版的《Perception》书中用数学方法详细证明了单层

神经网络无法解决异或问题。而当神经网络增加一个计算层后，就可以解决上述问题，并且具有优秀的非线性分类的效果。但两层神经网络复杂计算量使其发展停滞不前。直到 1986 年，Hinton 和 Rumelhar 等人提出了反向传播 (Backpropagation, BP) 算法，才推动了对两层神经网络的研究。

基本结构

三层结构：输入层、中间层、输出层。其中，中间层和输出层都是计算层。现在，我们的权值矩阵增加到了两个，我们用上标来区分不同层次之间的变量。例如 $a_x^{(y)}$ 代表第 y 层的第 x 个节点。下图给出了 $a_1^{(2)}$ ， $a_2^{(2)}$ 的计算公式以及由此得到的最终输出方式 z 的值。

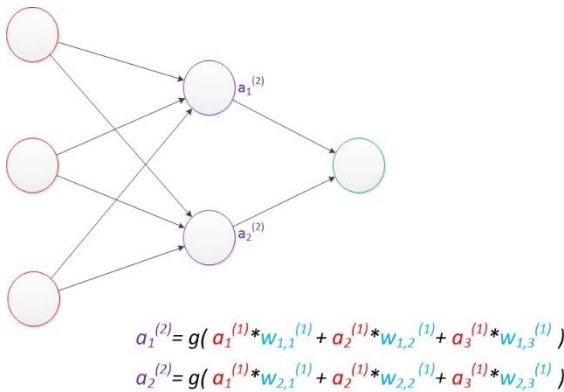


Figure 1 中间层计算

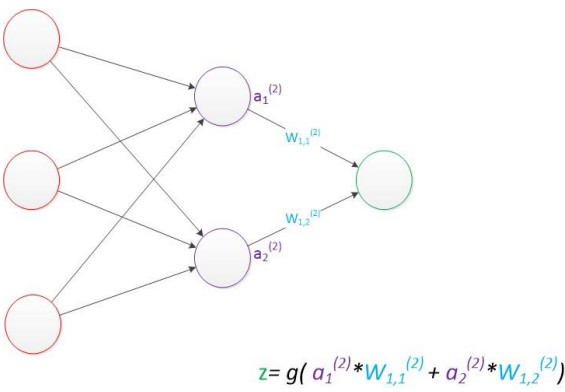


Figure 2 输出层计算

当 z 由值变为一个向量后，由前面的计算逻辑，我们将输出层增加一个值即可，这里我们可以用向量和矩阵来表示层次中的变量。 $a^{(1)}$ 、 $a^{(2)}$ ， z 表示网络中传输的向量数据， $w^{(1)}$ 、 $w^{(2)}$ 是神经网络的矩阵参数，计算公式如下：

$$g(w^{(1)} * a^{(1)}) = a^{(2)}$$

$$g(w^{(2)} * a^{(2)}) = z$$

事实上，在考虑神经网络是要考虑 bias unit，这些默认存在的节点存在于除输出层外的每个层次中。这里，我们将 bias unit 相关参数设置为向量 b 。下图给出了上述两种形式的两层神经网络图：

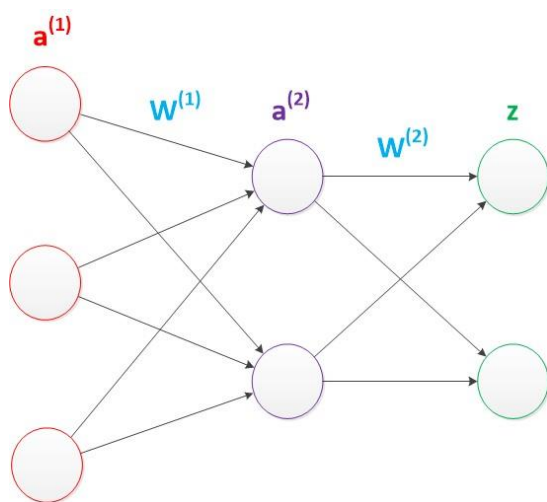


Figure 3 两层神经网络

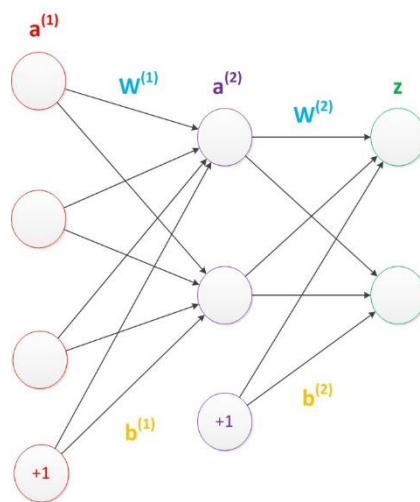


Figure 4 两层神经网络（含偏置节点）

评价

在当时，由于困扰神经网络的异或问题被解决，神经网络被应用到语音识别、图像识别、自动驾驶等多个领域。多领域的应用是研究两层神经网络的价值体现。但与此同时，神经网络仍存在若干问题未被解决：

- 1) 尽管采用 BP 算法，但神经网络训练耗时太久。
- 2) 局部最优解问题无法优化。
- 3) 隐藏层节点需要调参，使用不方便。

这些问题由于 90 年代中期，SVM 算法诞生，打败了神经网络算法，成为主流。

四、多层神经网络（深度学习）

发展

神经网络在度过了 10 年瓶颈期之后，迎来了第二春。Hinton 在提出 BP 算法 30 年后，重新定义了神经网络，也推动了神经网络研究的又一次热潮。2006 年，Hinton 在《Science》和相关期刊发表论文，提出了“深度信念网络”概念。与传统的 training 不同，有一个 pre-training 过程，这使得神经网络中的全连接找到一个最接近最优解的值。之后，运用 fine-tuning 技术对整个网络进行优化。他为多层神经网络相关学习方法赋予了新名字，“深度学习”。在此之后，深度神经网络发展呈燎原之势。

基本结构

在两层神经网络的输出层后面，继续添加层次。原来的输出层成为中间层，新增加的层次成为新的输出层。计算公式类似于两层神经网络，从最外层开始，算出所有单元值后，在进一步计算更深层次。公式如下：

$$\begin{aligned} g(w^{(1)} * a^{(1)}) &= a^{(2)} \\ g(w^{(2)} * a^{(2)}) &= a^{(3)} \\ g(w^{(3)} * a^{(3)}) &= z \end{aligned}$$

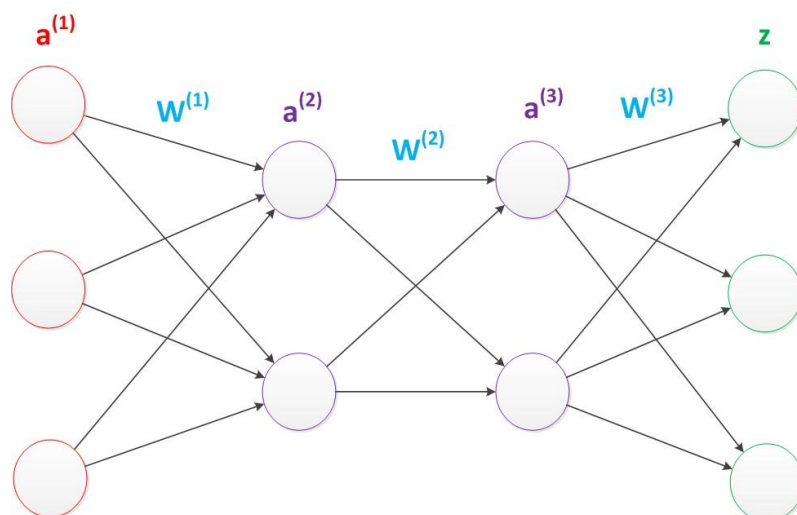


Figure 3 多层神经网络

从上图可以看出，层数越多、每层向量越多，都会使得神经网络的参数值增加。从而带来更好的表示能力，这是多层神经网络的一个重要性质。

这里我们来比较一下矮胖和高瘦的参数值多少。虽然参数数量仍然是 33，但却有 4 个中间层，是原来层数的接近两倍。这意味着一样的参数数量，可以用更深的层次去表达。在参数一致的情况下，我们也可以获得一个“更深”的网络。

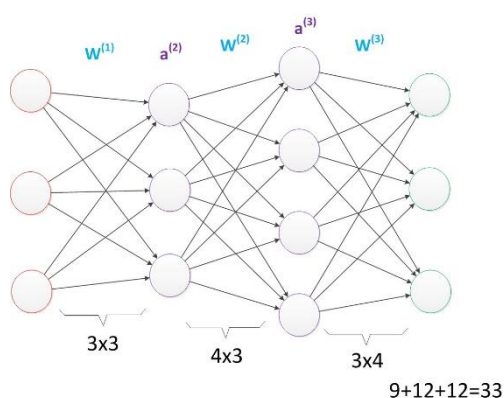


Figure 4 每层向量更多

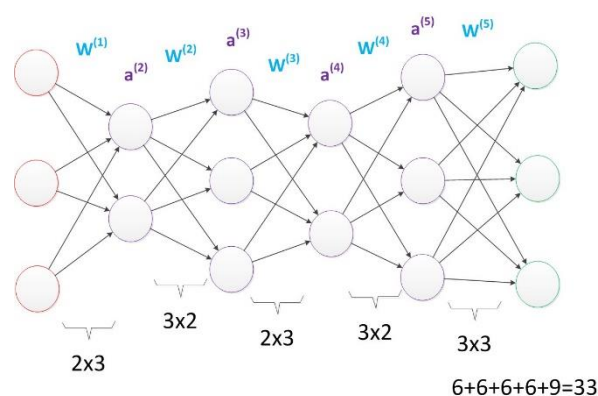


Figure 5 更深层次

评价

深度学习在图像、语音、自然语言处理等领域已经取得了大量关键性突破。例如，国际上 IBM, google 的 DNN 语音识别的研究，在自然语言处理等领域主要应用机器翻译以及语义挖掘方面。深度学习推动人工智能领域的发展。

但深度学习仍存在一些难题需要被攻破。首先，深度学习缺乏理论支撑，完整系统的理论支撑是大大增加该领域的发展深度。其次，对抗样本，例如两张几乎一样的图片，在图像识别中，通过深度学习模型给出了两种完全不一样的分类结果。

五、运用神经网络处理多项离散选择模型

(1) 数据来源：transport.txt

(2) 变量：被解释变量：出行方式

解释变量：收入的对数值、距离变量

(3) 结果及分析

分别采取了 logistic 回归、决策树、随机森林和神经网络四种方法对样本进行分析

Logistic 回归下 Training error rate 为 0.522

决策树、随机森林和神经网络方法下，将样本数据分为处理组和测试组，得到的误分类率分别为 0.496、0.53 和 0.482

与其他方法相比较，神经网络有效降低了错误率

参考文献来源：

Zhihu: 神经网络浅讲：从神经元到深度学习

<https://www.cnblogs.com/subconscious/p/5058741.html>

Neural networks

http://ufldl.stanford.edu/wiki/index.php/Neural_Networks

Andrew Ng neural networks

https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

中科院 史忠植 神经网络 讲义(sildes)