从神经网络的生物模型说起

我们知道人大脑信息的传递、对外界刺激产生反应都由神经元控制的，人脑就是由上百亿个的这样神经元构成。这些神经元之间并不孤立而且联系很密切，每个神经元平均与几千个神经元相连接，因此构成了人脑的神经网络。刺激在神经网络中的传播是遵循一定的规则的，一个神经元并非每次接到其他神经传递过来的刺激都产生反应。它首先会将与其相邻的神经元传来的刺激进行积累，到一定的时候产生自己的刺激将其传递给一些与它相邻的神经元。这样工作的百亿个的神经元构成了人脑对外界进行的反应。而人脑对外界刺激的学习的机制就是通过调节这些神经元之间联系以及其强度。当然，实际上以上说的是对人脑真正神经工作的一种简化的生物模型，利用这种简化的生物模型可以将它推广至机器学习中来，并把它描述成人工神经网络。BP神经网络就是其中的一种，来看看具体对神经元的分析。

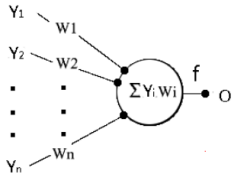


图1 神经网络中神经元示意图

神经元基本计算公式的理解.

神经元的积累的刺激是由其他神经元传递过来的刺激量和对应的权重之和，用 X_j 表示这种积累，Y_i 表示某个神经元传递过来的刺激量，W_i 表示链接某个神经元刺激的权重，得到公式：

$$X_j = (Y_1 * W_1)+(Y_2 * W_2)+\ldots+(Y_i * W_i)+\ldots+(Y_n * W_n) \Rightarrow X_j = \sum Y_i W_i$$

而当 X_j 完成积累后，完成积累的神经元本身对周围的一些神经元传播刺激，将其表示为 y_j 得到如下所示：

$$y_j = f(X_j)$$

一般选取Sigmod函数：$y_j = \frac{1}{1+e^{-x_j}}$

神经元根据积累后 X_j 的结果进行处理后，对外传递刺激 y_j 。用 f 函数映射来表示这种处理，将它称之为 **激活函数。**

BP神经网络的构成

分析完单个的神经元后，再来看看它们组成网络后的情形，用图形来说明是最直观的方法，如图2所示：

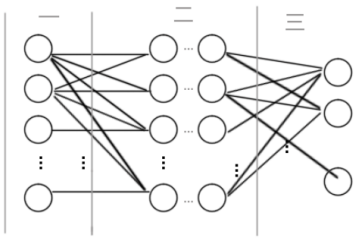


图2 BP神经网络示意图

第一区域的来说，它们相当于外界的刺激，是刺激的来源并且将刺激传递给神经元，因此把第一区域命名为**输入层。**第二区域，表示神经元相互之间传递刺激相当于人脑里面，因此把第二区命名为**隐藏层。**第三区域，表示神经元经过多层次相互传递后对外界的反应，因此把第三区域命名为**输出层。**

简单的描述就是，输入层将刺激传递给隐藏层，隐藏层通过神经元之间联系的强度（权重）和传递规则（激活函数）将刺激传到输出层，输出层整理隐藏层处理的后的刺激产生最终结果。若有正确的结果，那么将正确的结果和产生的结果进行比较，得到误差，再逆推对神经网中的链接权重进行反馈修正，从而来完成学习的过程。这就是BP神经网的反馈机制，也正是BP（Back Propagation）名字的来源：**运用向后反馈的学习机制，来修正神经网中的权重，最终达到输出正确结果的目的！**

BP神经网络的数学推导

从数学上对BP神经网络模型进行分析，本文第一部分神经网的生物模型中可以得到关于BP神经网络的第一个**公式(1)**：

$$x_j = \sum_i y_i w_{ji}$$

对于神经元本身的输出的激活函数，一般来说选取 **Sigmoid 函数**，那么可以得到第二个**公式（2）**：

$$y_j = \frac{1}{1+e^{-x_i}}$$

通过以上两个公式，可以分析出来BP神经网络中输出结果的计算过程。每个神经元收到刺激 y_i 然后加权积累（权重 W_{ji} ）完成后产生 x_j，再通过激活函数产生刺激 y_j，向下一层与它相连的神经元传递，依次类推最终输出结果。

我们再来分析如何利用向后反馈机制来修正神经元权重 W_{ji}，这一部分数学推导需要运用到多元微分的数学内容。要修正 W_{ji} 就需要得到误差量。具体来看，首先用 d_j 来表示真实的正确结果，并且设误差为 **E**，那么（y_j - d_j）对应的就是 **E** 对于 y_j 的微分增量，即 y_j 减去（y_j - d_j）后就能得到正确值，得到**公式（3）**：

$$\frac{\partial E}{\partial y_j} = y_j - d_j$$

然后，明确目标，需要知道的是对于权重 W_{ji} 的误差量是多少也就是 $\frac{\partial E}{\partial w_{ji}}$ **的值。**而由**公式（1）**中知道 W_{ji} 与 x_j 相关，那么可以推导出**公式（4）**：

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_j}\frac{\partial x_j}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} y_i$$

需要求得 W_{ji} 的误差量，转换为需要求 $\frac{\partial E}{\partial x_j}$ 的值了。它的推导如下：

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial x_j}$$

其中 $\frac{\partial y_j}{\partial x_j}$ 的值，可以通过**公式（2）**求导得出来：

$$\frac{\partial y_j}{\partial x_j} = y_j(1-y_j)$$

所以最终得到的的误差量的值为：

$$\frac{\partial E_j}{\partial w_{ji}} = y_j(1-y_j)(y_j - d_j)y_i$$

以上公式需要注意下标：最后一个是 y_i，前面的都是 y_j。推到这里可以算是完成了运用神经网络的输出值 y_j 和正确值 d_j 对最后一层隐藏层 W_{ji} 的修正，那么对其他隐藏层呢？ 接着往下看。

上面的推导过程由**公式（3）**开始，如果我们知道 $\frac{\partial E_j}{\partial y_i}$ （注意是 y_i，**公式（3）**中是 y_j），就可以 同理 推导求出其对应其他隐藏层需要修正的权重值误差量了。推导如下：

$$\frac{\partial E_j}{\partial y_i} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial x_j}\frac{\partial x_j}{\partial y_i} = (y_j - d_j)y_j(1-y_j)w_{ji}$$

这样所有的误差量的都可以 同理 推导完成！

最后一步修正 W_{ji}，就是加上下面变量了，设置一个 **l**（0 到 1 之间）学习率。

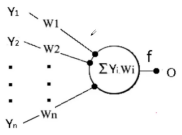
$$\Delta W = -l\frac{\partial E}{\partial w}$$

至此，BP神经网络反馈部分的数学推导算完成了，可以在自己的草稿纸上画画~

算法推导过程：

神经网络的实现

神经元基本计算公式



$$x_j = \sum_i y_i w_{ji}$$

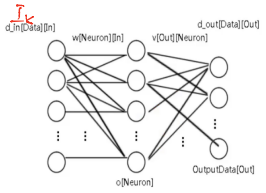
$$y_j = \frac{1}{1+e^{-x_j}}$$

$E=1/2\times\Sigma (y_j-d_j)^2$

均方误差

神经网络的实现

输出层和隐藏层权值的调整方法



$y_j = \sum_i o_i v_{ji} \Rightarrow \frac{\partial y_j}{\partial v_{ji}} = o_i, \frac{\partial y_j}{\partial o_i} = v_{ji}$

$\frac{\partial E}{\partial y_j} = y_j - d$

$\therefore \frac{\partial E}{\partial v_{ji}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial v_{ji}} = (y_j - d)o_i$

$\frac{\partial E}{\partial o_i} = \sum_j \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial o_i} = \sum_j (y_j - d)v_{ji}$

$x_i = \sum_k I_k w_{ki}$

$o_i = \frac{1}{1+e^{-x_i}}$

$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial o_i}\frac{\partial o_i}{\partial w_{ki}}$

$\frac{\partial o_i}{\partial w_{ki}} = \frac{\partial o_i}{\partial x_i}\frac{\partial x_i}{\partial w_{ki}} = o_i(1-o_i)I_k$

$\therefore \frac{\partial E}{\partial w_{ki}} = o_i(1-o_i)I_k\sum_j (y_j - d)v_{ji}$

I_k: d_in[Data][In]

y_j: OutPutData[Out]

d_j: d_out[Data][Out]

$(I_k \xrightarrow{W} x_i \xrightarrow[\text{激励函数}]{f} O_i \xrightarrow{V} y_j \downarrow E)$

$$\begin{cases} E = \frac{1}{2}\sum (y_j - d_j)^2 \\ y_j = O_i V_{ji} \end{cases}$$

$$\frac{\partial E}{\partial V_{ji}} = \frac{\partial E}{\partial y_j}\cdot\frac{\partial y_j}{\partial V_{ji}}$$

$$= (y - d_j)\cdot O_i$$

$$\frac{\partial E}{\partial W_{ki}} = \frac{\partial E}{\partial O_i}\frac{\partial O_i}{\partial W_{ki}}$$

$$= \left(\sum_j \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial O_i}\right)\cdot\left(\frac{\partial O_i}{\partial x_i}\cdot\frac{\partial x_i}{\partial W_{ki}}\right)$$

$$= \sum_j (y_j - d_j)V_{ji}\cdot O_i(1-O_i)\cdot I_k$$