http://www.cnblogs.com/jzhlin/archive/2012/07/28/bp.html

[BP神经网络-- 基本模型](http://www.cnblogs.com/jzhlin/archive/2012/07/28/bp.html)

     BP 神经网络中的 BP 为 Back  Propagation 的简写，最早它是由Rumelhart、McCelland等科学家于 1986 年提出来的，Rumelhart 并在Nature 上发表了一篇非常著名的文章 《Learning representations by back-propagating errors》 。随着时代的迁移，BP神经网络理论不断的得到改进、更新，现在无疑已成为了应用最为广泛的神经网络模型之一。让我们一起来探索下 BP神经网络最初的 基本模型和概念！

**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。

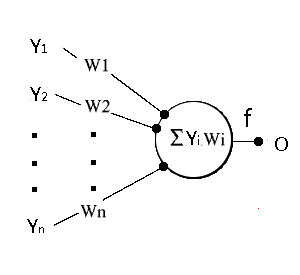


图1  神经网络中神经元示意图

         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 **Xj**表示这种积累，**Yi**表示某个神经元传递过来的刺激量，**Wi**表示链接某个神经元刺激的权重，得到公式：

**Xj = (y1 \* W1)+(y2 \* W2)+...+(yi \* Wi)+...+ (yn \* Wn)**

         而当 **Xj**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  **yj**  得到如下所示：

**yj = f（Xj）**

         神经元根据积累后 **Xj** 的结果进行处理后，对外传递刺激 **yj** 。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：

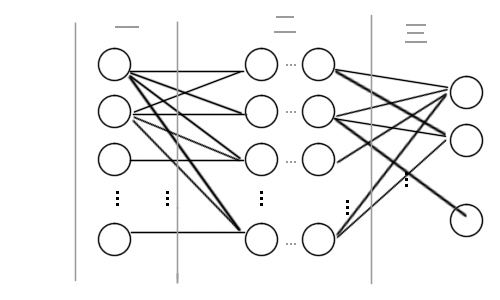


图2  BP神经网络示意图

         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)**：

BP神经网络公式

         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）**：

BP神经网络公式

         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 **yi**  然后加权积累（权重 **Wji** ）完成后产生 **xj**，再通过激活函数产生刺激 **yj**，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 **Wji**，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 **Wji**就需要得到误差量。具体来看，首先用 **dj**  来表示真实的正确结果，并且设误差为 **E** ，那么**（ yj- dj）**对应的就是 **E** 对于 **yj** 的微分增量，即 **yj** 减去**（ yj- dj）**后就能得到正确值，得到**公式（3）**：

BP神经网络公式

         然后，明确目标，需要知道的是对于权重 **Wji**的误差量是多少也**就是  BP神经网络推导量   的值**。而由**公式（1）**中知道 **Wji**与 **xj**相关， 那么可以推导出**公式（4）**：

BP神经网络公式

         需要求得**Wji**的误差量，转换为需要求  BP神经网络推导量  的值了。它的推导如下：

BP神经网络公式

         其中  BP神经网络推导量  的值，可以通过**公式（2）**求导得出来：

BP神经网络公式

         所以最终得到的的误差量的值为：

BP神经网络公式

         以上公式需要注意下标：最后一个是 **yi**，前面的都是 **yj**。推到这里可以算是完成了运用神经网络的输出值 **yj** 和正确值 **dj**  对最后一层隐藏层 **Wji** 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  BP神经网络推导量 （注意是 **yi**，**公式（3）**中是 **yj**），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：

BP神经网络公式

         这样所有的误差量的都可以 同理 推导完成！

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

BP神经网络公式

         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~

**参考文献**

**Learning representations by back-propagating errors   David E.Rumelhart\*,Geoffrey E.Hinton\*,Ronald J Williams\*  1986**

[**An introduction to neural networks**](http://www.ibm.com/developerworks/library/l-neural/index.html?S_TACT=105AGX52&S_CMP=cn-a-l)