



(图片来自Machine Learning: An Algorithmic Perspective, 2nd Edition, Figure 4.2)

假定两个输入节点 A 和 B 的二进制输入分别为 1 和 0，则根据图中的权重系数可以计算出神经元 C 的输入为 0.5，而神经元 D 的输入为 0。在由 C 和 D 构成的隐藏层中，由于 C 的输入大于 0，因而符号函数使其输出为 1；由于 D 的输入等于 0，符号函数则使其输出为 0。在输出节点的神经元 E 上，各路输入线性组合的结果为 0.5，因而 E 的输出，也是神经网络整体的输出，为 1，与两个输入的异或相等。在此基础上可以进一步证明，这个神经网络的运算规则就是异或操作的运算规则。

多层感知器 (multilayer perceptron) 包含一个或多个在输入节点和输出节点之间的隐藏层 (hidden layer)，除了输入节点外，每个节点都是使用非线性激活函数的神经元。而在不同层之间，多层感知器具有全连接性，即任意层中的每个神经元都与它前一层中的所有神经元或者节点相连接，连接的强度由网络中的权重系数决定。多层感知器是一类前馈人工神经网络 (feedforward neural network)。网络中每一层神经元的输出都指向输出方向，也就是向前馈送到下一层，直到获得整个网络的输出为止。

多层感知器的训练包括以下步骤：首先确定给定输入和当前权重下的输出，再将输出和真实值相减得到误差函数，最后根据误差函数更新权重。在训练过程中，虽然信号的流向是输出方向，但计算出的误差函数和信号传播的方向相反，也就是向输入方向传播的，正因如此，这种学习方式得名**反向传播 (backpropagation)**。**反向传播算法通过求解误差函数关于每个权重系数的偏导数，以此使误差最小化来训练整个网络。**

在反向传播算法中，首先要明确误差函数的形式。当多层感知器具有多个输出时，每个分类结果 y_j 与真实结果 d_j 之间都会存在误差。在单层感知器中，误差直接被定义为两者之间的差值。但在多个输出的情形下，如果第一个输出神经元的误差大于零，第二个输出神经元的误差小于零，这两部分误差就可能部分甚至完全抵消，造成分类结果准确无误的假象。

如何避免这个问题呢？

为了避免这个问题，**在反向传播算法中，每个输出神经元的误差都被写成平方项的形式，整个神经网络的误差则是所有输出神经元的误差之和。**如此一来，误差函数就以二次型的形式体现，也就避免了符号的影响。

明确定义了误差函数后，就要想方设法让它取得最小值。影响误差函数的因素无外乎三个：输入信号、传递函数和权重系数。输入信号是完全不依赖于神经网络的外部信号，无法更改；传递函数在网络设计过程中已经确定，同样无法更改；所以在算法执行的过程中，能够更新的就只有权重系数了。

既然提到了传递函数，就有必要对多层感知器中的传递函数加以说明。单层感知器中使用的符号函数有一个缺点：它是不连续的函数，因而不能在不连续点上求解微分。为了解决这个弊端，**多层感知器采用对数几率函数作为传递函数。**还记得对数几率函数在哪里出现过吗？没错，在机器学习中的逻辑回归算法当中。你可以回忆一下对数几率函数的表达式。

根据最优化理论，**求解误差函数的最小值就要找到误差函数的梯度，再根据梯度调整权重系数，使误差函数最小化。**对误差函数的求解从输出节点开始，通过神经网络逆向传播，直到回溯到输入节点。这背后的原理在于权重系数的变化对输出的影响方式并非直接修改，而是通过隐藏层逐渐扩散。这样环环相扣的作用方式体现在数学上就是**求导的链式法则**。

链式法则是个非常有用的数学工具，它的思想是求解从权重系数到误差函数这个链条上每一环的作用，再将每一环的作用相乘，得到的就是链条整体的效果。利用链式法则求出梯度后，再以目标的负梯度方向对权重系数进行调整，以逐渐逼近误差函数的最小值。

将平方误差函数、对数几率函数、求导链式法则三大法宝放在一起，就可以召唤出反向传播算法的流程：

1. 初始化网络中所有权重系数和阈值；

2. 在前向计算中，将训练样本送入输入节点，在输出节点得到训练结果，再以平方误差形式计算训练输出和真实输出之间的误差函数 $E = \frac{1}{2} \sum_j (d_j - y_j)^2$;
3. 在反向计算中，计算神经网络的局域梯度 $\frac{\partial E}{\partial w_{hi}}$ ，并根据局域梯度和学习率 η 从输出层到隐藏层对权重系数进行逐层更新 $\Delta w_{hi} = -\eta \frac{\partial E}{\partial w_{hi}}$;
4. 利用新样本训练多层感知器，迭代进行前向计算和反向计算，直到满足停止准则。

抛开冗杂的数学符号和运算不论，反向传播算法的原理其实并没有那么复杂。如果你能在头脑中勾勒出包含单个隐藏层的多层感知器结构，那不妨直观想象一下当某个输入上出现细微的扰动会导致什么结果？

由于层与层之间的连接方式是全连接，因而单个输入的微小变化会传递到所有的隐藏神经元，每个隐藏神经元都会感受到来自输入的波动。由于隐藏神经元的输入是输入信号的线性组合，因而输入端的扰动体现在隐藏神经元上就经过了一重权重系数的放大。经过权重系数放大后的扰动又被送入非线性的传递函数里，传递函数输入端的扰动导致的输出端的改变就要再乘以一个传递函数的导数作为放大因子。

上面的这个过程发生在单个的隐藏神经元上，而在隐藏层中每个神经元上发生的都是同样的故事，所以整个网络的输出变化就等于所有隐藏神经元上由“权重系数 + 传递函数”计算出的输出变化的总和。不难看出，误差函数对单个输入的偏导数就是“权重系数 + 传递函数”的联合作用。

看到这儿你可能会问，说的这些都是输入信号的扰动对输出结果的影响方式，这和反向传播有什么关系呢？当然有啦！当输入信号不变时，两个导数之间的关系就完全由权重系数的变化决定。将上面过程中输入信号和权重系数的角色做个调换，就可以得出误差函数对权重系数的偏导数就是“输入信号 + 传递函数”的联合作用。而这，恰恰就是链式法则的原理。

多层感知器的核心结构就是隐藏层，之所以被称为隐藏层是因为这些神经元并不属于网络的输入或输出。在多层神经网络中，隐藏神经元的作用在于**特征检测**。随着学习过程的不断进行，隐藏神经元将训练数据变换到新的特征空间之上，并逐渐识别出训练数据的突出特征。

在解决实际问题时，多层感知器的设计要考虑一些工程因素。假设单隐层的多层感知器有 L 个输入节点、 M 个隐藏节点和 N 个输出节点，那这个网络的权重系数总数就是 $(L + 1) \times M + (M + 1) \times N$ 。这些权重的取值都需要由反向传播算法确定，而反向

传播算法又由训练数据的错误来驱动。因而用于训练的数据越多，多层感知器的学习效果也就越好。一个经验法则是训练样本数目应该是权重系数数目的 10 倍，这显然对计算能力提出了较高的要求。从这个角度看，明斯基对感知器的批评是站得住脚的。

抛开训练数据量不论，隐藏层和隐藏神经元的数目也是网络设计中需要考虑的问题。在数学上可以证明，单个隐藏层就能够实现凸区域，双隐藏层更是可以实现任意形状的凸区域，也就能够解决任何复杂的分类问题。

在隐藏层数目不变的前提下，区域的复杂程度还能够随着隐藏神经元数目的增加而提升。在数学上同样可以证明，只要隐藏神经元的数目足够多，一个隐藏层就能使多层感知器以任意精度逼近任意复杂度的连续函数。通常情况下，多层感知器不会选择两个以上的隐藏层，因为层数越多，要追踪哪些权重正在被更新就越困难，又不会带来性能的提升。

多层感知器的训练需要多次遍历整个数据集，因而迭代次数就成为另一个重要的问题。预先设定迭代次数无法保证训练效果，预先设定误差阈值则可能导致算法无法终止。因而常用的办法是：**一旦误差函数停止减小，就终止学习算法。**

同其他机器学习方法一样，多层感知器也面临过拟合的风险。模型的泛化能力可以通过验证集来监督，也就能够在一定程度上避免过拟合的发生。当训练集的误差下降但验证集的误差上升时让训练立即停止，这就是所谓“早停”的过拟合抑制策略。当然，正则化方法也可以应用在多层感知器的训练中。

今天我和你分享了多层感知器和反向传播的基本原理，关于反向传播具体的数学细节你可以参考相关文献，其要点如下：

在感知器的输入层和输出层之间添加隐藏层，就可以得到多层感知器；

多层感知器是一类前馈神经网络，采用的是反向传播的学习方式；

反向传播算法要根据误差函数的梯度来调整权重系数，需要应用求导的链式法则；

单个隐藏层就能使多层感知器以任意精度逼近任意复杂度的连续函数。

反向传播算法是在 1986 年由乔弗雷·辛顿提出的，可今天，作为反向传播之父的辛顿却要大义灭亲。辛顿的观点是没有目标函数就无法进行反向传播，而如果数据没有标签自然就没有目标函数了。因此，要实现无监督学习就必须告别反向传播算法。那么应该如何看待辛顿的观点呢？

人工神经网络 | 多层感知器要点

1. 在感知器的输入层和输出层之间添加隐藏层，就可以得到多层感知器；
2. 多层感知器是一类前馈神经网络，采用的是反向传播的学习方式；
3. 反向传播算法要根据误差函数的梯度来调整权重系数，需要应用求导的链式法则；
4. 单个隐藏层就能使多层感知器以任意精度逼近任意复杂度的连续函数。



人工智能基础课

通俗易懂的人工智能入门课

王天一

工学博士，副教授



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 17 神经网络 | 一个青年才俊的意外死亡：神经元与感知器

下一篇 19 神经网络 | 各人自扫门前雪：径向基函数神经网络

精选留言 (2)

写留言



思考问题的...

2018-02-18

4

一个学习的感受，如果没有相关基础多说内容看一遍是没有什么感觉的。必须看完之后再去看相关的资料和解释，然后再回过头来看文章，才会有有所感悟。

展开



jimi

2018-02-08

2

要是配上点图就更会明白了

展开

