零基础自学神经网络 BP 算法

荣耀学院

2022 年 06 月

序言

深度学习在围棋、图像识别、语音识别、游戏竞技、机器翻译、蛋白质结构预测、药物设计等领域取得令人瞩目的进展，在有些领域，能力已经超过了人类。

深度学习是生产力竞争的明星领域。理解深度学习是竞争的必需。

BP 算法是深度学习的核心。

BP 算法有一定难度，但并不是高不可攀。它的难，难在结构复杂，数学原理并不难，只需要理解函数微分求极值即可，这是微积分的入门内容。毫不夸张地说，找个小学高年级学生，教一点微积分，肯定可以推导出 BP 算法。成年人呢？正常的成年人不可能比小学生差，肯定也可以学会。

固如此，仍然需要一些技巧，这里将一一叙述。

本书致力于让所有人都能完全掌握 BP 算法。

目 录

第一章 微分求极值

大多数机器学习问题都可以抽象为一个求极值问题[1]（不是所有问题，比如 K 近邻分类就不是）。求极值，就是求极大值或极小值，两者本质上是一样的，求极小值的目标函数取个负号就是求极大值。

如果一个函数有极值，不论极大值还是极小值，那么它在极值点的一阶导数必然是 0，也叫零点。沿着函数自变量从小到大的方向，如果一阶导数由负值变为零，那么零点是函数的极小值点，如果一阶导数由正值变为零，那么零点是函数的极大值点。

以数值方式求解零点，任选一个点，这个点所在的一阶导数大概率不为零（其实为零也不要紧），然后用合适的步长移动，移动到一阶导数为 0 的地方。求函数极大值，自变量沿着一阶导数的方向移动，求函数极小值，沿着一阶导数的反方向移动。

以求 $y = (x-2)^2$ 的极小值为例，y 的一阶导数是 $y' = 2(x-2)$，迭代公式是

$$x = x - \eta * y' = x - 2\eta(x-2)$$

其中，η 是学习速率，英语发音 |eta|，是一个 $(0, 1)$ 区间的实数。学习速率不能太大，太大就是移动的步子大，容易跨过一阶导数零点，取 0.01、0.001 都是可以的，也可以在计算过程动态调整。

用 python 实现 $y = (x-2)^2$ 的极小值数值求解，初始值 $x = 10$，$\eta = 0.01$，迭代次数 1000 次，代码如下：

```
eta = 0.01
x = 10
iter_n = 1000

for i in range(iter_n):
    print('step', i, 'x=', x)
    x = x - eta * 2 * (x-2)
```

[1]极值不一定是最值，最值一定是极值，一个目标函数可能有多个极值，最大值或者最小值只有一个。

运行结果如下：

```
step 0 x= 10
step 1 x= 9.84
step 2 x= 9.6832
step 3 x= 9.529536
step 4 x= 9.37894528
step 5 x= 9.2313663744
...
step 331 x= 2.0099751868912272
...
step 999 x= 2.000000013738508
```

y 在 $x = 2$ 的时候有全局最小值，运行到 331 步几乎非常接近了，也可以在 x 的变化很小的时候终止计算。

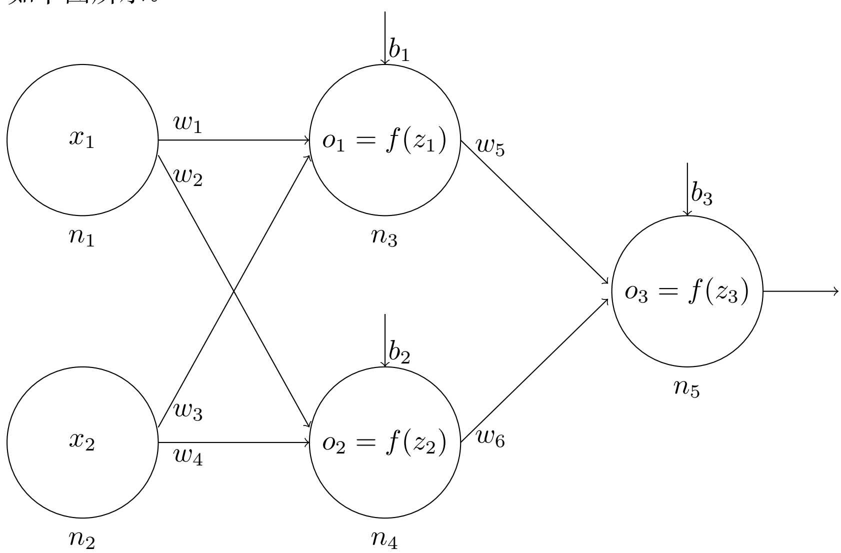
第二章　一个最简神经网络的 BP 算法推导

一个神经网络有多个神经元。神经元跟其他神经元的连接方式决定神经网络的类型。比如前馈神经网络，神经网络是多层的，每一层的神经元只跟前一层和后一层有连接关系。

推导 BP 算法，一定要从最简单的具体的神经网络开始，简单的好记，便于搞清楚所有细节，熟悉之后，再推导复杂的、通用的、抽象的神经网络，这叫最小原型原则。

用一个最简单的前馈神经网络演示 BP 算法推导过程。

这个神经网络只有三层：一个输入层，一个隐层，一个输出层。输入层有两个神经元，n_1 和 n_2，隐层有两个神经元，n_3 和 n_4，输出层有一个神经元，n_5，如下图所示。



其中，神经网络的参数有两类：权重，包括 w_1、w_2、w_3、w_4、w_5、w_6；偏

差，包括 b_1、b_2、b_3。x_1、x_2 是输入层的输入值，o_1、o_2 是隐层的输出值，o_3 是输出层的输出值。

假设一个样本，它有两个特征 $[x_1, x_2]$，目标值是 y。神经网络的学习过程，就是逐步调整权重和偏差，使得学习完成后，在神经网络输入是 $[x_1, x_2]$ 的时候，输出值尽可能接近 y。

从输入层到输出层，做一次前向传播，有如下结果：

$$z_1 = x_1w_1 + x_2w_3 + b_1$$

$$o_1 = f(z_1)$$

$$z_2 = x_1w_2 + x_2w_4 + b_2$$

$$o_2 = f(z_2)$$

$$z_3 = o_1w_5 + o_2w_6 + b_3$$

$$o_3 = f(z_3)$$

其中，$f(x)$ 是神经元的激活函数。比如，$f(x)$ 可以设置为 sigmod 函数，即：

$$f(x) = \frac{e^x}{1 + e^x}$$

此时，$f(x)$ 的一阶导数有一个特性：

$$f(x)^{'} = f(x)(1 - f(x))$$

这里不做证明了，很简单。

o_3 是神经网络的输出值，其值跟样本目标值 y 是不一样的，训练神经网络即是让它们之间的差异越来越小。用 Err 衡量它们的差异[1]：

$$Err = \frac{1}{2}(o_3 - y)^2$$

神经网络的训练，是一个求 Err 极小值的问题（对于该函数，求极小值等价于求最小值），也就是说，每一轮训练，都是计算 Err 对 w_1、w_2、w_3、w_4、w_5、b_1、b_2、b_3 的偏导，然后迭代更新。

根据函数微分求极值的规则，权重 w_5、w_6 和偏差 b_3 的迭代公式如下：

$$w_5 = w_5 - \eta\frac{\partial Err}{\partial w_5}$$

[1] Err 可以有多种形式，比如 $|o_3 - y|$，$(o_3 - y)^2$，视需求而定，这里设置为 $\frac{1}{2}(o_3 - y)^2$ 以便于求导求解。

$$w_6 = w_6 - \eta \frac{\partial Err}{\partial w_6}$$
$$b_3 = b_3 - \eta \frac{\partial Err}{\partial b_3}$$

其中:

$$\begin{aligned}
\frac{\partial Err}{\partial w_5} &= \frac{\partial(\frac{1}{2}(o_3 - y)^2)}{\partial w_5} \\
&= (o_3 - y)\frac{\partial(o_3 - y)}{\partial w_5} \\
&= (o_3 - y)\frac{\partial o_3}{\partial w_5} \\
&= (o_3 - y)\frac{\partial f(z_3)}{\partial w_5} \\
&= (o_3 - y)f^{'}(z_3)\frac{\partial(o_1 w_5 + o_2 w_6 + b_3)}{\partial w_5} \\
&= (o_3 - y)f^{'}(z_3)o_1
\end{aligned}$$

分析这个推导结果，可以推断一些有意思的“猜想”，是不是正确以后再证明:

1) 如果 $f(x)$ 是 sigmod 函数，那么它的值域是 $(0, 1)$，它的一阶导数 $f(x)^{'} = f(x)(1 - f(x))$ 的值域也是 $(0, 1)$，所以 $f^{'}(z_3)$ 和 o_1 都是正数，而且取值范围在 $(0, 1)$ 区域。

2) 如果 $o_3 > y$，也就是神经网络的输出值 o_3 比目标值 y 大，因为 $f^{'}(z_3$ 和 o_1 都是正数，那么必然有 $\frac{\partial Err}{\partial w_5} > 0$，根据迭代公式 $w_5 = w_5 - \eta \frac{\partial Err}{\partial w_5}$ 可知，迭代的结果让 w_5 变小了。也就是说，如果神经网络输出值比目标值大，把权重 w_5 调小。

3) 与上同理，如果 $o_3 < y$，神经网络的输出值 o_3 比目标值 y 小，因为 $f^{'}(z_3$ 和 o_1 都是正数，那么必然有 $\frac{\partial Err}{\partial w_5} < 0$，根据迭代公式 $w_5 = w_5 - \eta \frac{\partial Err}{\partial w_5}$ 可知，迭代的结果让 w_5 变大。也就是说，如果神经网络输出值比目标值小，把权重 w_5 调大。

4) 根据上两条可以合理地猜测一下，如果神经激励函数是 sigmod 函数，神经网络的所有权重，其迭代行为可能都象 w_5 一样：如果神经网络输出值比目标值大，所有权重都变小一些，如果神经网络输出值比目标值小，所有权重都变大一些。

5) 根据上三条可以合理猜测一下，当迭代到一定次数后，神经网络的权重的值会进入振荡，前一次迭代，导致神经网络输出值比目标值大，后一次迭

代，导致神经网络输出值比目标值小。至于具体多少次，是学习速率 η 决定的，如果 η 比较大，小训练次数就会进入振荡，如果 η 比较小，大训练次数才会进入振荡。动态调整 eta 肯定是合理的，比如一开始取大值，发现进入振荡，再把 η 调小，训练效果会更好。

6) 神经网络的输出值对迭代的影响，体现在推导结果的 $(o_3 - y)$ 上，因此所有参数的迭代公式必然都跟 $(o_3 - y)$ 相关，因此所有的参数推导结果都必然包含 $(o_3 - y)$。

7) 推导结果的 o_1，是跟 w_5 相关的，因此 w_5 只受 o_1 的影响，不受 o_2 的影响。

8) 可以合理推测，每个权重的迭代，受到三个影响：神经网络输出误差；跟它相连的前一个神经元的输出值；跟它相连的后一个神经元的激励函数的一阶导数。

9) 根据上三条，可以合理猜测 $\frac{\partial Err}{\partial w_6} = (o_3 - y)f^{'}(z_3)o_2$，对不对后面可以验证。

再推导 w_6 的迭代公式：

$$
\begin{aligned}
\frac{\partial Err}{\partial w_6} &= \frac{\partial(\frac{1}{2}(o_3 - y)^2)}{\partial w_6} \\
&= (o_3 - y)\frac{\partial(o_3 - y)}{\partial w_6} \\
&= (o_3 - y)\frac{\partial o_3}{\partial w_6} \\
&= (o_3 - y)\frac{\partial f(z_3)}{\partial w_6} \\
&= (o_3 - y)f^{'}(z_3)\frac{\partial(o_1 w_5 + o_2 w_6 + b_3)}{\partial w_6} \\
&= (o_3 - y)f^{'}(z_3)o_2
\end{aligned}
$$

推导结果符合前面的猜想。

$$\begin{aligned}
\frac{\partial Err}{\partial b_3} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial b_3} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial b_3} \\
&= (o_3-y)\frac{\partial o_3}{\partial b_3} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial b_3} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial b_3} \\
&= (o_3-y)f^{'}(z_3)
\end{aligned}$$

隐层的参数 w_1、w_2、w_3、w_4、b_1、b_2 的迭代公式如下：

$$w_1 = w_1 - \eta\frac{\partial Err}{\partial w_1}$$

$$w_2 = w_2 - \eta\frac{\partial Err}{\partial w_2}$$

$$w_3 = w_3 - \eta\frac{\partial Err}{\partial w_3}$$

$$w_4 = w_4 - \eta\frac{\partial Err}{\partial w_4}$$

$$b_1 = b_1 - \eta\frac{\partial Err}{\partial b_1}$$

$$b_2 = b_2 - \eta\frac{\partial Err}{\partial b_2}$$

其中:

$$
\begin{aligned}
\frac{\partial Err}{\partial w_1} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_1} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_1} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_1} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(z_3)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)(\frac{\partial(o_1w_5)}{\partial w_1}+\frac{\partial(o_2w_6)}{\partial w_1}+\frac{\partial b_3}{\partial w_1}) \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial o_1}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial f(z_1)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5x_1
\end{aligned}
$$

分析这个推导结果，可以推断一些有意思的“猜想”：

1)w_1 受 o_3-y 影响，符合前面的分析。

2)w_1 受跟它相连的前一层神经元输出影响，符合前面的分析。在这里，前一层是输入层，跟 w_1 相连的是 x_1。

3)w_1 受跟它相连的后一层神经元的一阶导数的影响，符合前面的分析。在这里，是 $f^{'}(z_1)$。

4)w_1 受 $f^{'}(z_3)$ 影响和 w_5 的影响，从神经网络的结构上看，可以合理猜测，从 w_1 到神经网络输出 o_3 的整条路径：x_1，$f(z_1)$，w_5，o_3，都对 w_1 产生影响，根据这条“路径”规则，也许可以写出其它权重的推导结果，是否正确后续再验证。

5) 根据上一条，可以合理猜测，$\frac{\partial Err}{\partial w_2}=(o_3-y)f^{'}(z_3)f^{'}(z_2)w_6x_1$

6)BP 算法的 back propagation，体现在求导结果的部分相似性：$\frac{\partial Err}{\partial w_1}$、$\frac{\partial Err}{\partial w_5}$ 和 $\frac{\partial Err}{\partial w_6}$，都有 $(o_3-y)f^{'}(z_3)$。因此，在计算的时候，可以从神经网络的后面向前计算，使用部分已经算好的结果，节省计算量。

7) 可以观察到，$f^{'}(z_3)$ 和 $f^{'}(z_1)$ 的值域都在 $(0,1)$ 上，因此它们相乘后，乘积比它们更小。可以合理猜测，如果神经网络的层数比较多，比如几十层或几百层，且激活函数都是 sigmod 函数，那么越往前计算，权重的更新量越小，因为几十或几百个 0 和 1 之间的数相乘，其结果必然趋近于零，导致“梯度消失”现象。如果对训练结果做预估，可有猜测，越靠近输出层，权重跟初始值相比变化越大，越靠近输入层，权重跟初始值相比变化越小。因此，设置太多的隐层是没有意义的，靠近输入层的隐层权重在训练时变化很小，徒然增加了计算量，但并没有什么用处。

$$
\begin{aligned}
\frac{\partial Err}{\partial w_2} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_2} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_2} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_2} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(z_3)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_2}+w_6\frac{\partial o_2}{\partial w_2}+\frac{\partial b_3}{\partial w_2}) \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial o_2}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial f(z_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(z_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(x_1w_2+x_2w_4+b_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6x_1
\end{aligned}
$$

推导结果符合前面的猜想。

$$
\begin{aligned}
\frac{\partial Err}{\partial w_3} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_3} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_3} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_3} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_3}+w_6\frac{\partial o_2}{\partial w_3}+\frac{\partial b_3}{\partial w_3}) \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial o_1}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial f(z_1)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial z_1}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5x_2
\end{aligned}
$$

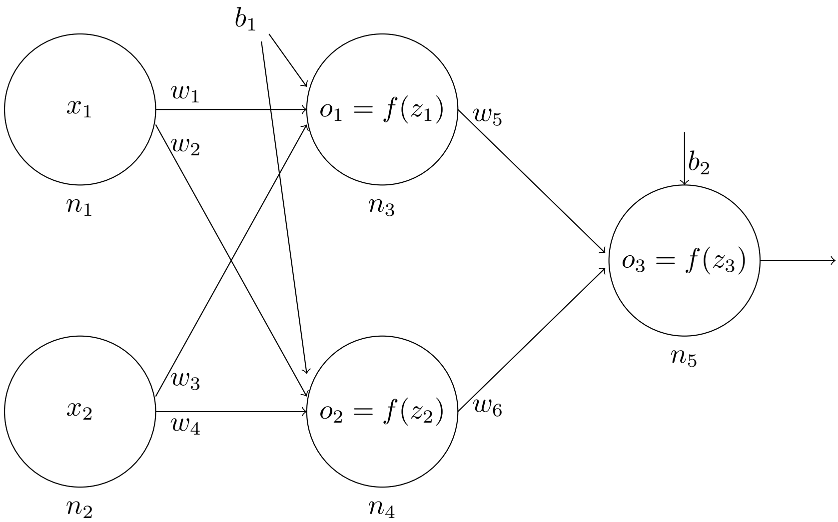
$$
\begin{aligned}
\frac{\partial Err}{\partial w_4} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_4} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_4} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_4} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_4}+w_6\frac{\partial o_2}{\partial w_4}+\frac{\partial b_3}{\partial w_4}) \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial o_2}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial f(z_2)}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial z_2}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(x_1w_2+x_2w_4+b)}{\partial w_4} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6x_2
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Err}{\partial b_1} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial b_1} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial b_1} \\
&= (o_3-y)\frac{\partial o_3}{\partial b_1} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)(\frac{\partial(o_1w_5)}{\partial b_1}+\frac{\partial(o_2w_6)}{\partial b_1}+\frac{\partial b_3}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial o_1}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial f(z_1)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial z_1}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Err}{\partial b_2} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial b_2} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial b_2} \\
&= (o_3-y)\frac{\partial o_3}{\partial b_2} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_3)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)(\frac{\partial(o_1w_5)}{\partial b_2}+\frac{\partial(o_2w_6)}{\partial b_2}+\frac{\partial b_3}{\partial b_2}) \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial o_2}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial f(z_2)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial z_2}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(x_1w_2+x_2w_4+b_2)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6
\end{aligned}
$$

第三章　一个参数更少神经网络的 BP 算法推导

上一个神经网络，每个神经元都有一个 b 值。为减少参数，还可以设置成每层神经网络的神经元共享一个 b 值。如下：



假设一个样本有两个特征 $[x_1, x_2]$，样本标记是 y。那么，从输入层到输出层，做一次前向传播，有如下结果：

$$z_1 = x_1 w_1 + x_2 w_3 + b_1$$

$$o_1 = f(z_1)$$

$$z_2 = x_1 w_2 + x_2 w_4 + b_1$$

$$o_2 = f(z_2)$$

$$z_3 = o_1 w_5 + o_2 w_6 + b_2$$

$$o_3 = f(z_3)$$

其中，$f(x)$ 是神经元的激活函数。

输出层的参数 w_5、w_6、b_2 的迭代公式如下：

$$w_5 = w_5 - \eta\frac{\partial Err}{\partial w_5}$$

$$w_6 = w_6 - \eta\frac{\partial Err}{\partial w_6}$$

$$b_2 = b_2 - \eta\frac{\partial Err}{\partial b_2}$$

其中：

$$\begin{aligned}
\frac{\partial Err}{\partial w_5} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_5} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_5} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_5} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_5} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial w_5} \\
&= (o_3-y)f^{'}(z_3)o_1
\end{aligned}$$

$$\begin{aligned}
\frac{\partial Err}{\partial w_6} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_6} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_6} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_6} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_6} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial w_6} \\
&= (o_3-y)f^{'}(z_3)o_2
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial Err}{\partial b_2} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial b_2} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial b_2} \\
&= (o_3-y)\frac{\partial o_3}{\partial b_2} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial b_2} \\
&= (o_3-y)f^{'}(z_3)
\end{aligned}
$$

隐层的参数 w_1、w_2、w_3、w_4、b_1 的迭代公式如下:

$$w_1 = w_1 - \eta\frac{\partial Err}{\partial w_1}$$

$$w_2 = w_2 - \eta\frac{\partial Err}{\partial w_2}$$

$$w_3 = w_3 - \eta\frac{\partial Err}{\partial w_3}$$

$$w_4 = w_4 - \eta\frac{\partial Err}{\partial w_4}$$

$$b_1 = b_1 - \eta\frac{\partial Err}{\partial b_1}$$

其中:

$$
\begin{aligned}
\frac{\partial Err}{\partial w_1} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_1} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_1} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_1} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(z_3)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)(\frac{\partial(o_1w_5)}{\partial w_1}+\frac{\partial(o_2w_6)}{\partial w_1}+\frac{\partial b_2}{\partial w_1}) \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial o_1}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial f(z_1)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial w_1} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5x_1
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Err}{\partial w_2} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_2} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_2} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_2} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(z_3)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_2}+w_6\frac{\partial o_2}{\partial w_2}+\frac{\partial b_2}{\partial w_2}) \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial o_2}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)w_6\frac{\partial f(z_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(z_2)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(x_1w_2+x_2w_4+b_1)}{\partial w_2} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_2)w_6x_1
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Err}{\partial w_3} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial w_3} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial w_3} \\
&= (o_3-y)\frac{\partial o_3}{\partial w_3} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_3}+w_6\frac{\partial o_2}{\partial w_3}+\frac{\partial b_2}{\partial w_3}) \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial o_1}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)w_5\frac{\partial f(z_1)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial z_1}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial w_3} \\
&= (o_3-y)f^{'}(z_3)f^{'}(z_1)w_5x_2
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial Err}{\partial w_4} &= \frac{\partial(\frac{1}{2}(o_3 - y)^2)}{\partial w_4} \\
&= (o_3 - y)\frac{\partial(o_3 - y)}{\partial w_4} \\
&= (o_3 - y)\frac{\partial o_3}{\partial w_4} \\
&= (o_3 - y)\frac{\partial f(z_3)}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)\frac{\partial z_3}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)\frac{\partial(o_1 w_5 + o_2 w_6 + b_2)}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial w_4} + w_6\frac{\partial o_2}{\partial w_4} + \frac{\partial b_2}{\partial w_4}) \\
&= (o_3 - y)f^{'}(z_3)w_6\frac{\partial o_2}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)w_6\frac{\partial f(z_2)}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial z_2}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)f^{'}(z_2)w_6\frac{\partial(x_1 w_2 + x_2 w_4 + b_1)}{\partial w_4} \\
&= (o_3 - y)f^{'}(z_3)f^{'}(z_2)w_6 x_2
\end{aligned}$$

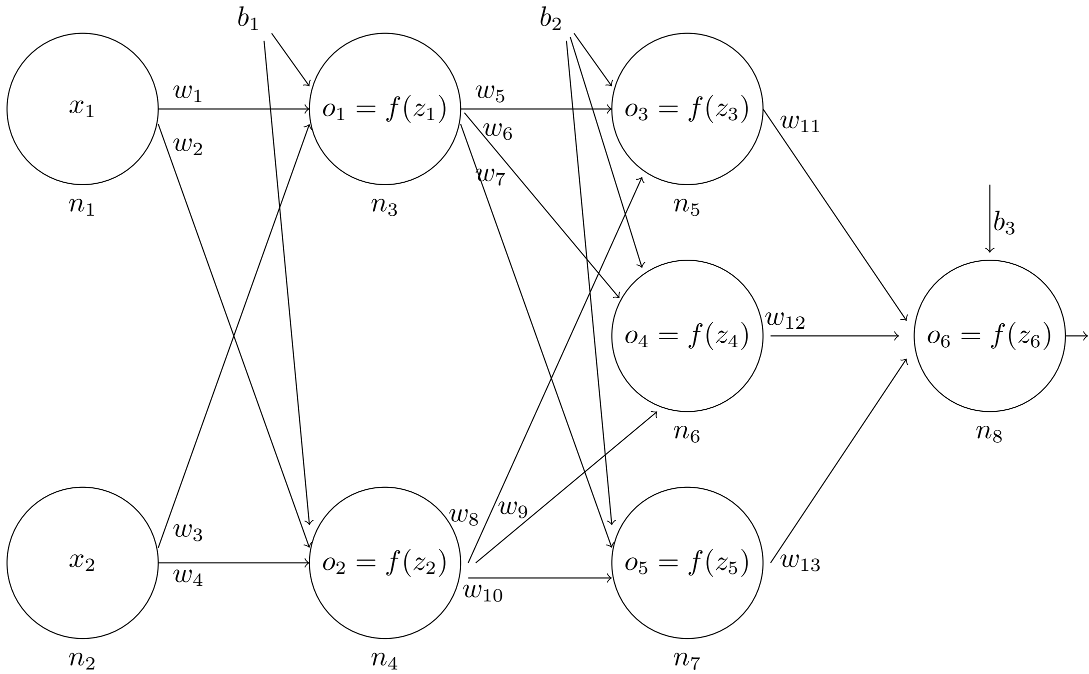
$$
\begin{aligned}
\frac{\partial Err}{\partial b_1} &= \frac{\partial(\frac{1}{2}(o_3-y)^2)}{\partial b_1} \\
&= (o_3-y)\frac{\partial(o_3-y)}{\partial b_1} \\
&= (o_3-y)\frac{\partial o_3}{\partial b_1} \\
&= (o_3-y)\frac{\partial f(z_3)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial z_3}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)\frac{\partial(o_1w_5+o_2w_6+b_2)}{\partial b_1} \\
&= (o_3-y)f^{'}(z_3)(\frac{\partial(o_1w_5)}{\partial b_1}+\frac{\partial(o_2w_6)}{\partial b_1}+\frac{\partial b_2}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial o_1}{\partial b_1}+w_6\frac{\partial o_2}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)(w_5\frac{\partial f(z_1)}{\partial b_1}+w_6\frac{\partial f(z_2)}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)(f^{'}(z_1)w_5\frac{\partial z_1}{\partial b_1}+f^{'}(z_2)w_6\frac{\partial z_2}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)(f^{'}(z_1)w_5\frac{\partial(x_1w_1+x_2w_3+b_1)}{\partial b_1}+f^{'}(z_2)w_6\frac{\partial(x_1w_2+x_2w_4+b_1)}{\partial b_1}) \\
&= (o_3-y)f^{'}(z_3)(f^{'}(z_1)w_5+f^{'}(z_2)w_6)
\end{aligned}
$$

分析这个推导结果，发现一个跟前面不一样的东西：从神经网络的结构图上看，偏差 b_1 是经过两条“路径”传导到 o_3 的，因此，体现在推导结果，就是两条路径微分结果之和。

第四章　两个隐层神经网络 BP 迭代公式的猜测与验证

前面分析的两个神经网络是一个隐层，得到了一些有意思的猜想，也验证了猜想是合理的。

如果神经网络更复杂一点，这些猜想对不对？那就设计一个两个隐层的神经网络验证一下，结构如下图：



我们从前两章得到的“路径”猜想，继续用下去。

从 w_5 到神经网络的输出 o_6，只有一条路径：o_1、$f'(z_3)$、w_{11}、$f'(z_6)$、$o_6 - y$，所以 w_5 的推导结果可以直接写出来：

$$\begin{aligned}\frac{\partial Err}{\partial w_5} &= (o_6-y)f^{'}(z_6)w_{11}f^{'}(z_3)o_1 \\ &= (o_6-y)f^{'}(z_6)f^{'}(z_3)w_{11}o_1\end{aligned}$$

w_6、w_7、w_8、w_9、w_{10}、w_{11}、w_{12}、w_{13}、b_3，跟 w_5 是一样的，也都是一条路径。b_2 到 o_6 是三条路径，跟第二个神经网络的 b_1 推导类似。这些推导结果，就不一一写出来了，体力活。

w_1 跟以前不一样了，从它到 o_6 有三条路径，分别是:

1)x_1，$f(z_1)$，w_5，$f(z_3)$，w_{11}，$f(z_6)$，o_6-y；

2)x_1，$f(z_1)$，w_6，$f(z_4)$，w_{12}，$f(z_6)$，o_6-y；

3)x_1，$f(z_1)$，w_7，$f(z_5)$，w_{13}，$f(z_6)$，o_6-y。

猜测 w_1 的推导结果是三条路径的累加，可以根据路径直接写下来:

$$\begin{aligned}\frac{\partial Err}{\partial w_1} &= x_1f^{'}(z_1)w_5f^{'}(z_3)w_{11}f^{'}(z_6)(o_6-y)+ \\ &\quad x_1f^{'}(z_1)w_6f^{'}(z_4)w_{12}f^{'}(z_6)(o_6-y)+ \\ &\quad x_1f^{'}(z_1)w_7f^{'}(z_5)w_{13}f^{'}(z_6)(o_6-y) \\ &= x_1f^{'}(z_1)(w_5f^{'}(z_3)w_{11}f^{'}(z_6)(o_6-y)+ \\ &\qquad w_6f^{'}(z_4)w_{12}f^{'}(z_6)(o_6-y)+ \\ &\qquad w_7f^{'}(z_5)w_{13}f^{'}(z_6)(o_6-y))\end{aligned}$$

如果对 w_5 推导结果是下面这个结果，表明上面的猜想正确:

$$\frac{\partial Err}{\partial w_1} = x_1f^{'}(z_1)(w_5f^{'}(z_3)w_{11}+w_6f^{'}(z_4)w_{12}+w_7f^{'}(z_5)w_{13})f^{'}(z_6)(o_6-y)$$

...to be continue

w_2、w_3、w_4、b_1 跟 w_1 的逻辑是一样的，推导结果不一一写了，体力活。

现在只缺一个东西了–推导一个通用的抽象的神经网络的 BP 算法，彻底解决理论问题。鉴于我们从最小原型得到的经验和结论，在正式推导之前我们已经对推导结果了然于胸，毫无困难，只需要走个流程。这么看来，BP 算法不难吧?

第五章　to be continue