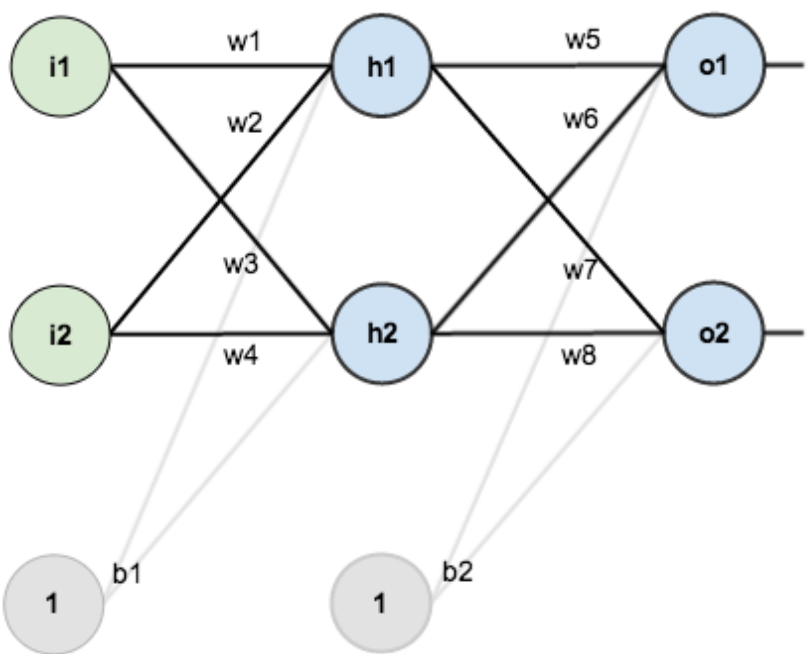
手把手教你神经网络的反向传播

> 原文：A Step by Step Backpropagation Example

概览

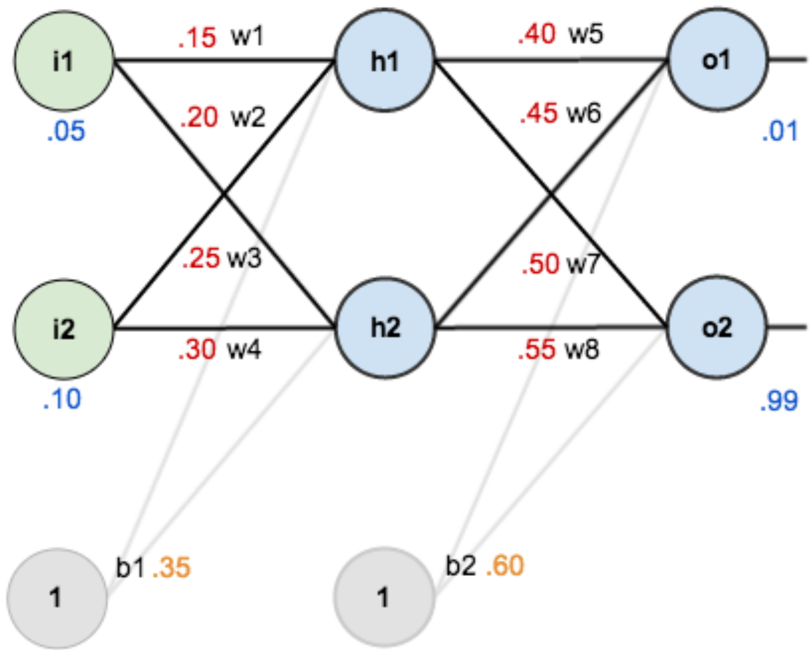
在这篇教程中，我们只会使用两个输出层、两个隐藏层和两个输出。下图展现了这个神经网络的基本结构：

（输入层和隐藏层已经分别补齐了一位数字，即 b_1 和 b_2，其值为 1）



在上图中，i_1, i_2 表示两个输入神经元，w_i 表示权重，h_1, h_2 表示两个隐藏神经元，而 o_1, o_2 则为两个输出神经元。

我们先随便初始化一些权重：



反向传播是优化神经网络的一种方法，其目的是优化各个权重，以便让最终模型的输出更接近我们的预期。在教程的剩下部分里，我们会给予模型 0.05 和 0.10 的输入，并期望神经网络能够输出 0.01 和 0.99。

正向传播

先让我们计算一下，在使用上面初始化的权重时，通过 0.05 和 0.1 的输入，模型会返回什么（即计算神经网络的正向传播）。

利用正向传播输出结果

首先，计算从输出层到隐藏层：

需要用权重计算输入层传递给隐藏层的总净输入，然后在利用一个激活函数来计算隐藏层的值。在这儿我们可以使用逻辑回归中常用的 `sigmoid` 函数：

$$sigmoid(z) = \frac{1}{1+e^{-z}}$$

计算过程如下：

1. 使用 net_{hi} 表示隐藏层中第 `i` 个神经元收到的纯输入

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

2. 然后使用 `sigmoid` 函数激活它，使其成为该神经元的值 `h1`，out_{hi} 代表第 `i` 个隐藏神经元的值

$$out_{h1} = sigmoid(net_{h1}) = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

同理，对 `h2` 也这么得到 $out_{h2} = 0.596884378$：

3. 之后，计算从隐藏层到输出层，跟上面一毛一样的逻辑：

$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$
$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$
$out_{o1} = sigmoid(net_{o1}) = \frac{1}{1+e^{-1.105905967}} = 0.75136507$

同理 $out_{o2} = 0.772928465$

计算误差

至此，计算完毕，但明显和我们的期望有差，所以来利用误差函数计算一下其输出的误差。

1. 依次计算每个神经元输出的误差，然后相加求和:

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

举个栗子，我们对 o_1 的期望是 `0.01` ，但实际输出为 `0.75136507` ，因此 o_1 的误差为：

$$E_{o_1} = \frac{1}{2}(target - output)^2 = 0.5 * (0.01 - 0.75136507)^2 = 0.274811083$$

同理，计算得到 $E_{o_2} = 0.023560026$，所以总误差为：

$$E_{total} = E_{o_1} + E_{o_2} = 0.274811083 + 0.023560026 = 0.298371109$$

反向传播

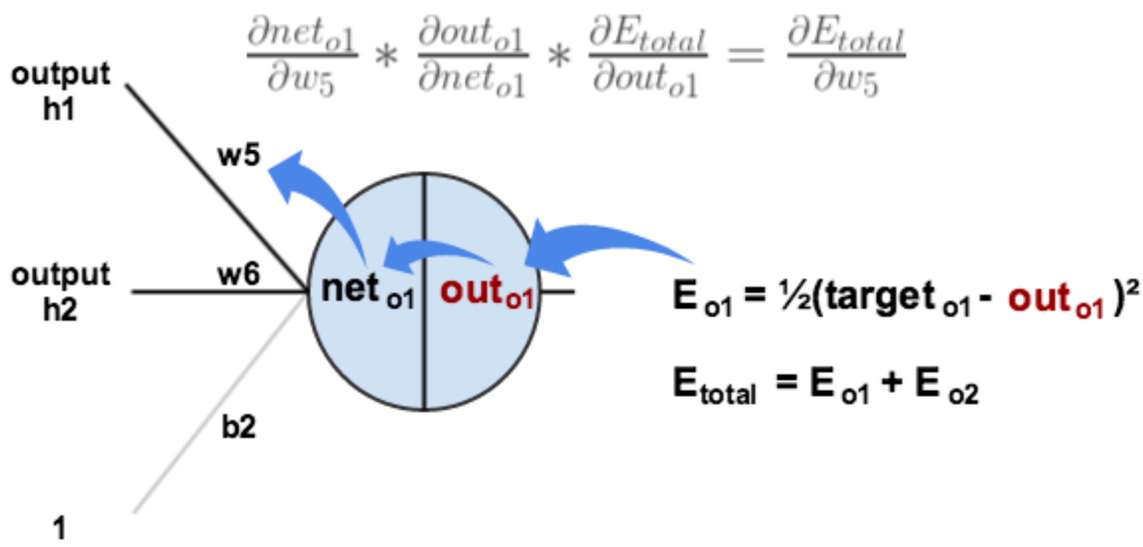
主角来了。我们的目的就是利用反向传播来优化权重，以此让我们的神经网络的输出更加接近期望，也就是最小化误差。既然是反向传播，那就倒序从输出层开始。

输出层

考虑一下 w_5 这个权重。我们想要知道它对于结果的输出有多大的影响，因此，需要计算 E_{total} 和 w_5 之间的变化关系，即斜率：

$$\frac{\Delta E_{total}}{\Delta w_5} = \frac{\Delta E_{total}}{\Delta out_{o_1}} * \frac{\Delta out_{o_1}}{\Delta net_{o_1}} * \frac{\Delta net_{o_1}}{\Delta w_5}$$

用下图表示：



接下来计算各个变化率：

$$E_{total} = \frac{1}{2}(target_{o1} - out_{o_1})^2 + \frac{1}{2}(target_{o2} - out_{o_2})^2$$

1. 针对 out_{o1} 求导：

$$\frac{\Delta E_{total}}{\Delta out_{o_1}} = 2 * \frac{1}{2}(target_{o1} - out_{o_1})^{2-1} + 0$$

得到：

$$\frac{\Delta E_{total}}{\Delta out_{o_1}} = -(target_{o1} - out_{o_1}) = -(0.01 - 0.75136507) = 0.74136507$$

2. 针对 net_{o1} 求导：

已知：

$$out_{h1} = sigmoid(net_{h1}) = \frac{1}{1 + e^{-net_{h1}}}$$

则：

$$\frac{\Delta out_{o_1}}{\Delta net_{o_1}} = out_{o1}(1 - out_{o1}) = 0.75136507 * (1 - 0.75136507) = 0.186815602$$

3. 针对 w_5 求导

已知：

$$neto_1 = w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1$$

则:

$$\frac{\Delta net_{o_1}}{\Delta w_5} = out_{h_1} + 0 + 0 = 0.593269992$$

于是得到:

$$\frac{\Delta E_{total}}{\Delta w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

为了降低误差，我们需要用这个值来更新我们的权重（跟梯度下降一样，使用一个学习速率。在这儿设为 $\eta = 0.5$）：

$$w_5 = w_5 - \eta(\frac{\Delta E_{total}}{\Delta w_5}) = 0.4 - 0.5 * 0.082167041 = 0.35891648$$

同理，使用这种方法依次计算出 w_6, w_7, w_8:

$w_6 = 0.408666186$
$w_7 = 0.511301270$
$w_8 = 0.561370121$

隐藏层

下一步，继续使用反向传播计算 w_1, w_2, w_3, w_4。

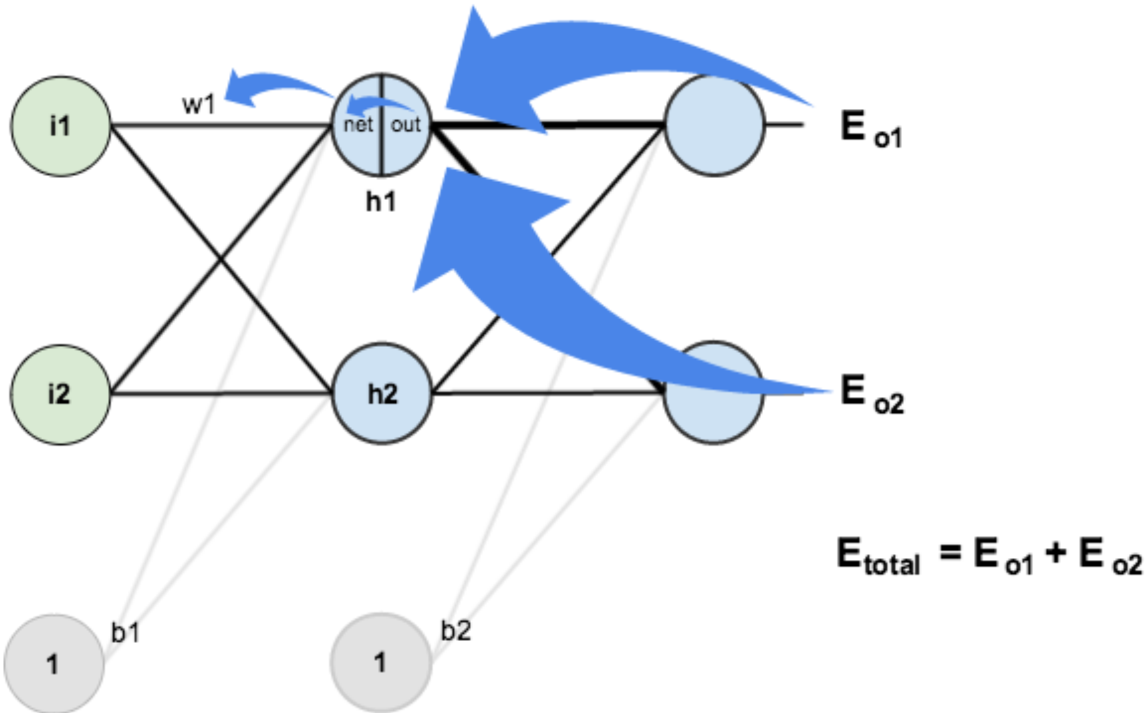
以计算 w_1 的影响为例:

$$\frac{\Delta E_{total}}{\Delta w_1} = \frac{\Delta E_{total}}{\Delta out_{h_1}} * \frac{\Delta out_{h_1}}{\Delta net_{h_1}} * \frac{\Delta net_{h_1}}{\Delta w_1}$$

如下图:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



1. 计算 $\frac{\Delta E_{total}}{\Delta out_{h1}}$

需要注意的是，鉴于我们的 out_{h1} 对于两个输出神经元 o_1 和 o_2 都有权重的影响，因此，需要像上图中表示的那样，把 E_{total} 拆分成 E_{o_1} 和 E_{o_2} 进行计算。

$$\frac{\Delta E_{total}}{\Delta out_{h_1}} = \frac{\Delta E_{o_1}}{\Delta out_{h_1}} + \frac{\Delta E_{o_2}}{\Delta out_{h_1}}$$

首先来看 $\frac{\Delta E_{o_1}}{\Delta out_{h_1}}$：

$$\frac{\Delta E_{o_1}}{\Delta out_{h_1}} = \frac{\Delta E_{o_1}}{\Delta net_{o_1}} * \frac{\Delta net_{o_1}}{\Delta out_{h_1}}$$

计算 $\frac{\Delta E_{o_1}}{\Delta net_{o_1}}$：

$$\frac{\Delta E_{o_1}}{\Delta net_{o_1}} = \frac{\Delta E_{o_1}}{\Delta out_{o_1}} * \frac{\Delta out_{o_1}}{\Delta net_{o_1}} = 0.74136507 * 0.186815602 = 0.138498562$$

计算 $\frac{\Delta net_{o_1}}{\Delta out_{h_1}}$：

由：

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

得：

$$\frac{\Delta net_{o1}}{\Delta out_{h1}} = w_5 = 0.40$$

因此，

$$\frac{\Delta E_{o1}}{\Delta out_{h1}} = 0.138498562 * 0.4 = 0.055399425$$

同理，计算得到 $\frac{\Delta E_{o2}}{\Delta out_{h1}} = -0.019049119$

所以 $\frac{\Delta E_{total}}{\Delta out_{h1}} = 0.055399425 - 0.019049119 = 0.036350306$

别忘了我们还有 $\frac{\Delta out_{h1}}{\Delta net_{h1}}$ 和 $\frac{\Delta net_{h1}}{\Delta w1}$ 没有计算：

还记得吧，$out_{h1} = sigmoid(net_{h1}) = \frac{1}{1+e^{-net_{h1}}}$，

$$\frac{\Delta out_{h1}}{\Delta net_{h1}} = out_{h1}(1 - out_{h1}) = 0.59326999 * (1 - 0.59326999) = 0.241300709$$

而 $net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$，

$$\frac{\Delta net_{h1}}{\Delta w_1} = i_1 = 0.05$$

合并在一起得出结果：

$$\frac{\Delta E_{total}}{\Delta w_1} = \frac{\Delta E_{total}}{\Delta out_{h1}} * \frac{\Delta out_{h1}}{\Delta net_{h1}} * \frac{\Delta net_{h1}}{\Delta w_1} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$$

现在可以更新 w_1 了：

$$w_1 = w_1 - \eta(\frac{\Delta E_{total}}{\Delta w_1}) = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

同理，更新 w_2, w_3, w_4。

$w_2 = 0.19956143$
$w_3 = 0.24975114$
$w_4 = 0.29950299$

至此，我们已经全部更新了权重！当一开始使用 `0.05` 和 `0.1` 作为输入时，得到的误差为 `0.298371109`；在第一次反向传播后，误差降为 `0.291027924`。看起来改变不大，但是如果我们重复

10,000 次反向传播，则误差可减低至 `0.0000351085` 。

这就是基本的反向传播过程了，希望通过这个例子能够使你对它有更好的认识。