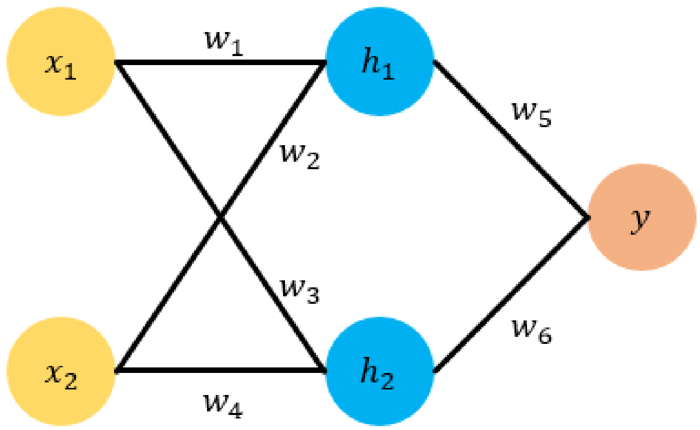


Back Propagation（梯度反向传播）实例讲解

HexUp
创造是最高级的乐趣

来自专栏 · Pure for Fun

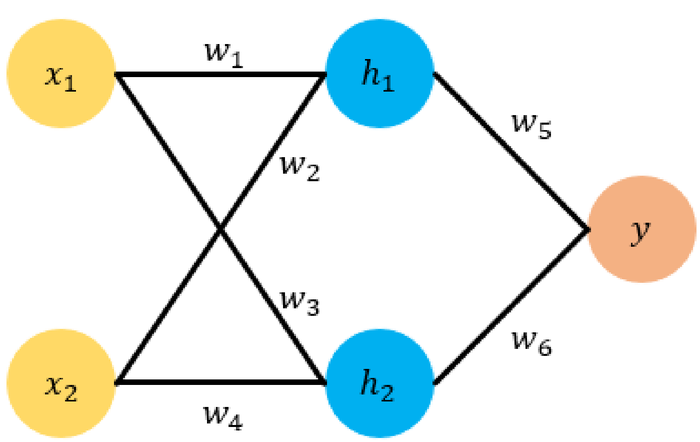
我学习东西的时候喜欢有具体的例子和数字，讨厌冗长而枯燥的公式（我猜大部分人和我一样）。所以当我决定写一篇讲解Back Propagation（梯度反向传播）的文章的时候，我决定用实例来一步步地推导。只要你跟着这篇教程一步步走下来，你就明白什么是Back Propagation了，而且你会发现，其实它的想法很简单。

接下来我们把Back Propagation简称为BP。

BP的目的

首先我们要搞清楚两个问题

1. 为什么要求梯度?
2. 求关于谁的梯度?



上图展示了一个神经网络⁺。神经网络可以看作是一个函数 $y = f_w(x)$ ， x 是输入， y 是输出， w 是 f 的参数。 w 的真实值是我们的目标，但我们有的只是一些 x 和与之对应的真实的 $y = f(x)$ 的值，所以我们要用到这两个值去估计 w 的真实值。这个问题可以看成下面的优化问题（优化问题即求函数最小值）

$$\min_w \sum_x ||f_w(x) - y||^2$$

其中我们令 $E = \sum_x ||f_w(x) - y||^2$ ，并称之为误差项。我们的目标就是求一组 w 使得 E 最小。求解这类问题有个经典的方法叫做梯度下降法⁺（SGD, Stochastic Gradient Descent），这个算法一开始先随机生成一个 w ，然后用下面的公式不断更新 w 的值，最终能够逼近真实结果。

$$w^+ = w - \eta \cdot \frac{\partial E}{\partial w}$$

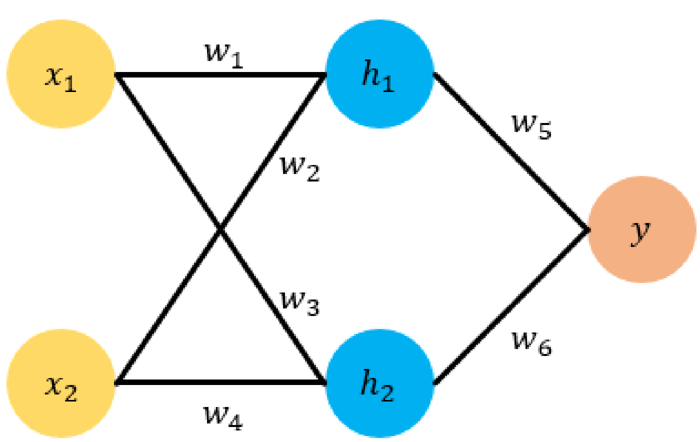
其中 $\frac{\partial E}{\partial w}$ 是当前的误差 E 关于 w 的梯度，它的物理意义是当 w 变化的时候 E 随之变化的幅度， η 叫做学习率，通常在 0.1 以下，用来控制更新的步长（防止步子太大扯到蛋哈哈）。所以，开头的两个问题答案就有了：求梯度的原因是我们需要它来估算真实的 w ，求的是误差项 E 关于参数 w 的梯度。

链式法则⁺--BP的基础

在正式推导BP之前，我们首先需要回忆一下求导数的链式法则，这个是BP的基础和核心。假设 $y = g(x), z = f(y)$ ，那么 $z = h(x), h = f \circ g$ 。我们知道 $\frac{dy}{dx} = g'(x), \frac{dz}{dy} = f'(y)$ ，那么如何求 z 对 x 的导数 $\frac{dz}{dx}$ 呢？这个时候链式法则就出场了，根据微积分的知识 $h'(x) = \frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$ ，即复合函数的求导可以使用乘法法则，也称为链式法则，待会儿我们会用到。上面给出的是单变量的情况，多变量同样适用。

Back Propagation By Example

现在我们用一个例子来讲解BP，如下图所示，我们选取的例子是最简单的feed forward neural network，它有两层，输入层有两个神经元 x_1, x_2 ，隐藏层有两个神经元 h_1, h_2 ，最终输出只有一个神经元 y ，各个神经元之间全连接。为了直观起见，我们给各个参数赋上具体的数值。我们令 $x_1 = 1, x_2 = 0.5$ ，然后我们令 w_1, w_2, w_3, w_4 的**真实值分别是** $1, 2, 3, 4$ ，令 w_5, w_6 的真实值是 $0.5, 0.6$ 。这样我们可以算出 y 的真实目标值是 $t = 4$ 。



网络结构示意图，一个简单的两层Feed Forward Netowrk

那么为了模拟一个Back Propagation的过程，我们假设我们只知道 $x_1 = 1, x_2 = 0.5$ ，以及对应的目标 $t = 4$ 。我们不知道 $w_1, w_2, w_3, w_4, w_5, w_6$ 的真实值，现在我们需要随机为他们初始化值，假设我们的随机化结果是 $w_1 = 0.5, w_2 = 1.5, w_3 = 2.3, w_4 = 3, w_5 = 1, w_6 = 1$ 。下面我们就开始来一步步进行Back Propagation吧。

首先，在计算反向传播之前我们需要计算Feed Forward Pass，也即是预测的 h_1, h_2, y 和误差项 E ，其中 $E = \frac{1}{2}(t-y)^2$ 。根据网络结构示意图，各个变量的计算公式为：

$$\begin{aligned} h_1 &= w_1 \cdot x_1 + w_2 \cdot x_2 = 1.25 \\ h_2 &= w_3 \cdot x_1 + w_4 \cdot x_2 = 3.8 \\ y &= w_5 \cdot h_1 + w_6 \cdot h_2 = 5.05 \\ E &= \frac{1}{2}(y-t)^2 = 0.55125 \end{aligned}$$

现在Feed Forward Pass算完了，我们来计算Backward Pass。 y 是神经网络预测的值，真实的输出是 $t = 4$ 。那么，要更新 w_5 的值我们就要算 $\frac{\partial E}{\partial w_5}$ ，根据链式法则有

$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial w_5}$$

因为 $E = \frac{1}{2}(t-y)^2$ ，所以

$$\begin{aligned} \frac{\partial E}{\partial y} &= 2 \cdot \frac{1}{2} \cdot (t-y) \cdot (-1) \\ &= y - t \\ &= 5.05 - 4 = 1.05 \end{aligned}$$

而 $y = w_5 * h1 + w_6 * h2$ ，所以

$$\frac{\partial y}{\partial w_5} = h_1 + 0 = h_1 = 1.25$$

把上面两项相乘我们得到

$$\begin{aligned} \frac{\partial E}{\partial w_5} &= \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial w_5} = (y-t) \cdot h_1 \\ &= 1.05 \cdot 1.25 = 1.3125 \end{aligned}$$

运用之前梯度下降法的公式更新 w_5 ，得到新的 w_5^+ 。其中我们假设 $\eta = 0.1$ （并且后面所有的 η 都等于 0.1 ）

$$\begin{aligned} w_5^+ &= w_5 - \eta \cdot \frac{\partial E}{\partial w_5} \\ &= 1 - 0.1 \cdot 1.3125 \\ &= 0.86875 \end{aligned}$$

同理，我们可以按照相同的步骤计算 w_6^+ 的更新公式

$$\begin{aligned} w_6^+ &= w_6 - \eta \cdot \frac{\partial E}{\partial w_6} \\ &= 1 - 0.1 \cdot 3.99 \\ &= 0.601 \end{aligned}$$

下面我们再来看 w_1, w_2, w_3, w_4 ，由于这四个参数在同一层，所以求梯度的方法是相同的，因此我们这里仅展示对 w_1 的推导。根据链式法则

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1}$$

其中 $\frac{\partial E}{\partial y} = y - t$ 在求 $\frac{\partial E}{\partial w_5}$ 的时候已经求过了。而根据 $y = w_5 \cdot h_1 + w_6 \cdot h_2$ 我们可以得到

$$\frac{\partial y}{\partial h_1} = w_5 + 0 = w_5$$

又根据 $h_1 = w_1 \cdot x_1 + w_2 \cdot x_2$ 我们可以得到

$$\frac{\partial h_1}{\partial w_1} = x_1 + 0 = x_1$$

因此我们有下面的公式

$$\frac{\partial E}{\partial w_1} = (y - t) \cdot w_5 \cdot x_1$$

现在我们代入数字并使用梯度下降法更新 w_1

$$\begin{aligned} w_1^+ &= w_1 - \eta \cdot \frac{\partial E}{\partial w_1} \\ &= w_1 - \eta \cdot (y - t) \cdot w_5 \cdot x_1 \\ &= 0.5 - 0.1 \cdot 1.05 \cdot 1 \cdot 1 \\ &= 0.395 \end{aligned}$$

然后重复这个步骤更新 w_2, w_3, w_4

$$\begin{aligned} w_2^+ &= w_2 - \eta \cdot \frac{\partial E}{\partial w_2} \\ &= w_2 - \eta \cdot (y - t) \cdot w_5 \cdot x_2 \\ &= 1.5 - 0.1 \cdot 1.05 \cdot 1 \cdot 0.5 \\ &= 1.4475 \end{aligned}$$

$$\begin{aligned} w_3^+ &= w_3 - \eta \cdot \frac{\partial E}{\partial w_3} \\ &= w_3 - \eta \cdot (y - t) \cdot w_6 \cdot x_1 \\ &= 2.3 - 0.1 \cdot 1.05 \cdot 1 \cdot 1 \\ &= 2.195 \end{aligned}$$

$$
\begin{aligned}
w_4^+ &= w_4 - \eta \cdot \frac{\partial E}{\partial w_4} \\
&= w_2 - \eta \cdot (y - t) \cdot w_5 \cdot x_2 \\
&= 3 - 0.1 \cdot 1.05 \cdot 1 \cdot 0.5 \\
&= 2.9475
\end{aligned}
$$

Great！现在我们已经更新了所有的梯度，完成了一次梯度下降法。我们用得到的新的 w^+ 再来预测一次网络输出值，根据Feed Forward Pass得到 $y^+ = 3.1768$ ，那么新的误差是 $E^+ = 0.3388$ ，相比于之前的 $E = 0.55125$ 确实是下降了呢，说明我们的模型预测稍微准了一点。只要重复这个步骤，不断更新网络参数我们就能学习到更准确的模型啦。

如果你看到了这里，那么恭喜你，你已经学会Back Propagation了。这么看下来，Back Propagation是不是很简单呢?

最后，给自己的公众号打个广告：kffuniverse

编辑于 2020-12-16 17:36