1. 线性反向传播

1.1 正向传播的定义

假设有一个函数:

$$z = x \cdot y \qquad (1)$$

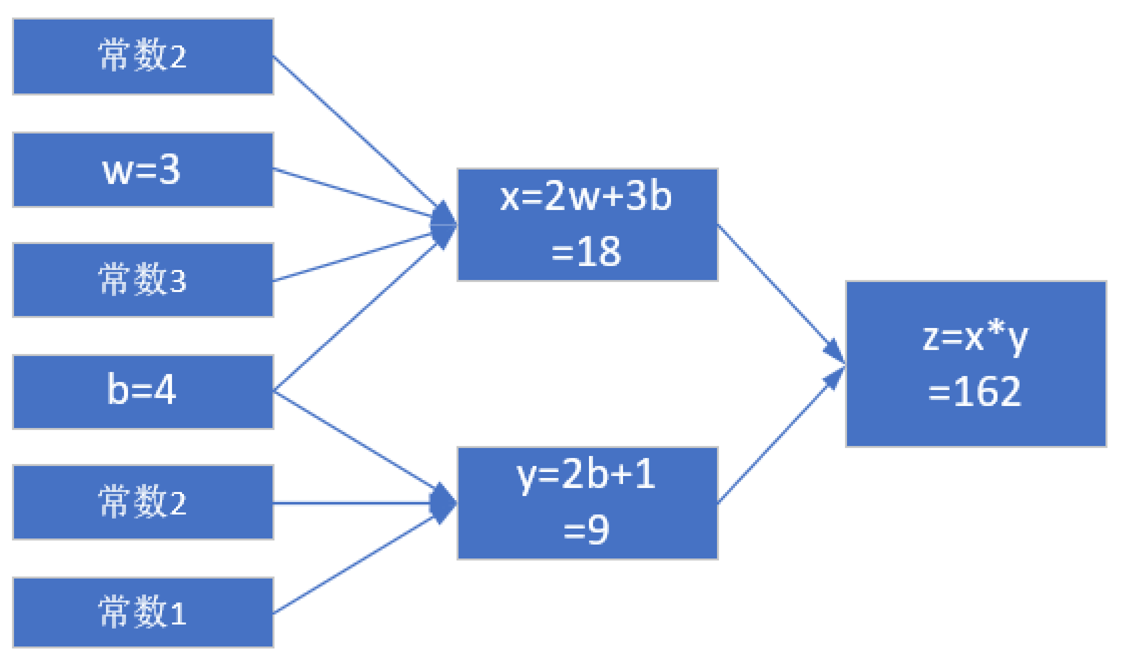
其中:

$$x = 2w + 3b \qquad (2)$$

$$y = 2b + 1 \qquad (3)$$

注意这里 x, y, z 不是变量，只是中间计算结果；w, b 才是变量。因为在后面要学习的神经网络中，要最终求解的目标是 w 和 b 的值，

计算z的值的时候，需要按照公式直接往后计算就好,当 $w = 3, b = 4$ 时，会得到如下的结果：



最终的 z 值，受到了前面很多因素的影响：变量 w，变量 b，计算式 x，计算式 y。

1.2 反向传播求解w

1.2.1 求 w 的偏导

目前 $z = 162$，如果想让 z 变小一些，比如目标是 $z = 150$，w 应该如何变化呢？为了简化问题，先只考虑改变 w 的值，而令 b 值固定为 4。

因为 $z = x \cdot y$，其中 $x = 2w + 3b, y = 2b + 1$

所以：

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial x} \cdot \frac{\partial x}{\partial w} = y \cdot 2 = 18 \qquad (4)$$

其中：

$\frac{\partial z}{\partial x} = \frac{\partial}{\partial x}(x \cdot y) = y = 9$

$\frac{\partial x}{\partial w} = \frac{\partial}{\partial w}(2w+3b) = 2$

当然，我们也可以直接进行求导，我们有：

$$z = x \cdot y = (2w+3b)(2b+1) = 4wb + 2w + 6b^2 + 3b \quad (5)$$

对上式求 w 的偏导：

$$\frac{\partial z}{\partial w} = 4b + 2 = 4 \cdot 4 + 2 = 18 \tag{6}$$

可以看见，两个结果是完全相同的，被称为链式法则

公式4和公式6的含义是：当 w 变化一点点时，z 会产生 w 的变化值18倍的变化。记住我们的目标是让 $z=150$，目前在初始状态时是 $z=162$，所以，问题转化为：当需要 z 从 162 变到 150 时，w 需要变化多少？

既然：

$$\Delta z = 18 \cdot \Delta w$$

则：

$$\Delta w = \frac{\Delta z}{18} = \frac{162-150}{18} = 0.6667$$

所以：

$w = w - 0.6667 = 2.3333$
$x = 2w + 3b = 16.6667$
$z = x \cdot y = 16.6667 \times 9 = 150.0003$

我们一下子就成功地让 z 值变成了 150.0003，与 150 的目标非常地接近，这就是偏导数的威力所在。

当然，我们可以反向传播求b

也可以同时求解 w 和 b 的变化值，如下：

1. 在检查 Δz 时的值时，注意要用绝对值，因为有可能是个负数
2. 在计算 Δb 和 Δw 时，第一次时，它们对 z 的贡献值分别是 $1/63$ 和 $1/18$，但是第二次时，由于 b, w 值的变化，对 z 的贡献值也会有微小变化，所以要重新计算。具体解释如下：

$$\frac{\partial z}{\partial b} = \frac{\partial z}{\partial x} \cdot \frac{\partial x}{\partial b} + \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial b} = y \cdot 3 + x \cdot 2 = 3y + 2x$$

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial x} \cdot \frac{\partial x}{\partial w} + \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial w} = y \cdot 2 + x \cdot 0 = 2y$$

所以，在每次迭代中，要重新计算下面两个值：

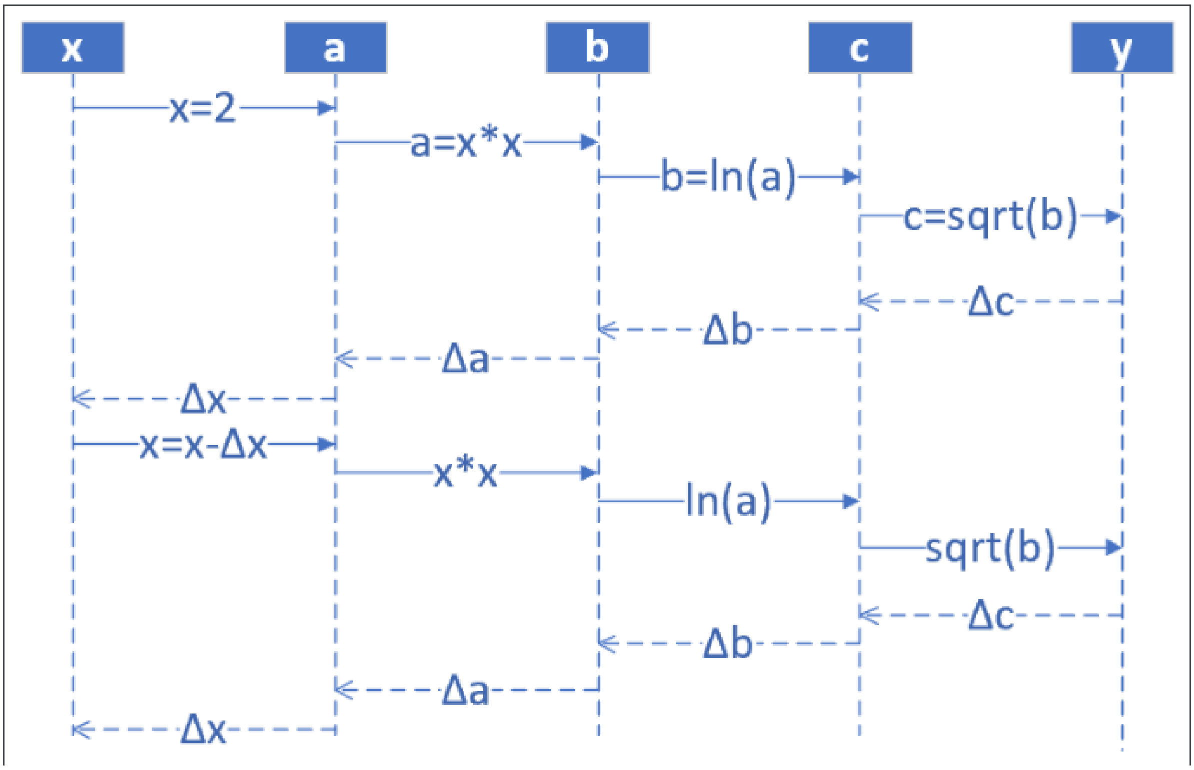
$$\Delta b = \frac{\Delta z}{3y + 2x}$$

$$\Delta w = \frac{\Delta z}{2y}$$

用程序模拟出来的：在每次迭代中都重新计算 $\Delta b, \Delta w$ 的贡献值(`factor_b`和`factor_w`每次都变化)：这个与第一个单变量迭代不同的地方是：这个问题可以有多个解，所以两种方式都可以得到各自的正确解，但是第二种方式效率高，而且满足梯度下降的概念。

2. 非线性反向传播

在上面的线性例子中，我们可以发现，误差一次性地传递给了初始值 w 和 b，即，只经过一步，直接修改 w 和 b 的值，就能做到误差校正。因为从它的计算图看，无论中间计算过程有多么复杂，它都是线性的，所以可以一次传到底。缺点是这种线性的组合最多只能解决线性问题，不能解决更复杂的问题。这个我们在神经网络基本原理中已经阐述过了，需要有激活函数连接两个线性单元。



正向过程

1. 第1个人，输入层，随机输入第一个 x 值，x 的取值范围 $(1, 10]$，假设第一个数是 2；
2. 第2个人，第一层网络计算，接收第1个人传入 x 的值，计算：$a = x^2$；
3. 第3个人，第二层网络计算，接收第2个人传入 a 的值，计算：$b = \ln(a)$；
4. 第4个人，第三层网络计算，接收第3个人传入 b 的值，计算：$c = \sqrt{b}$；
5. 第5个人，输出层，接收第4个人传入 c 的值

反向过程

6. 第5个人，计算 y 与 c 的差值：$\Delta c = c - y$，传回给第4个人
7. 第4个人，接收第5个人传回Δc，计算 $\Delta b = \Delta c \cdot 2\sqrt{b}$
8. 第3个人，接收第4个人传回Δb，计算 $\Delta a = \Delta b \cdot a$
9. 第2个人，接收第3个人传回Δa，计算 $\Delta x = \frac{\Delta}{2x}$
10. 第1个人，接收第2个人传回Δx，更新 $x \leftarrow x - \Delta x$，回到第1步