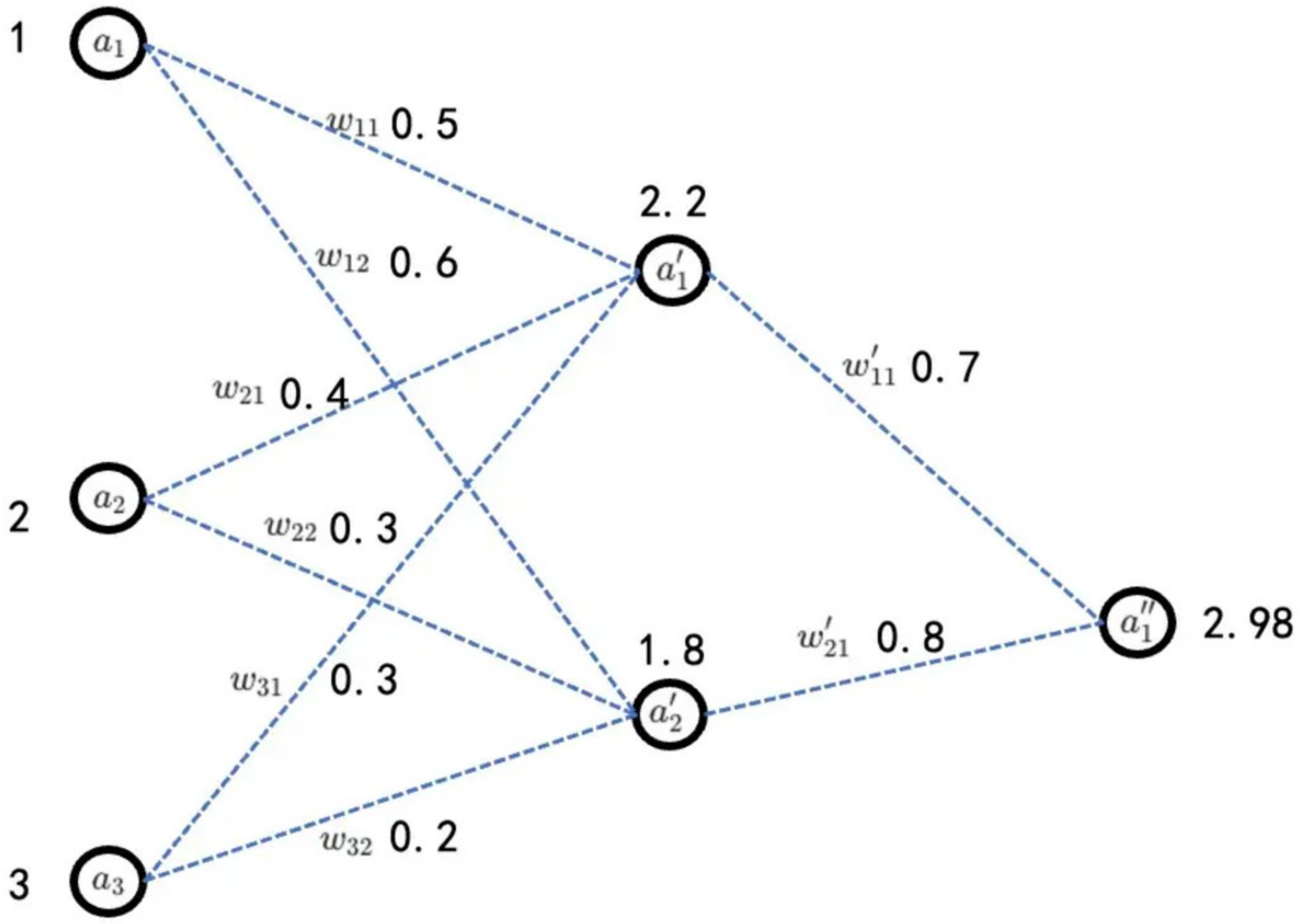
神经网络如何计算结果

上一篇《如何构建神经网络》论述了神经网络的三大步骤，其中最关键的就是参数计算。根据神经网络的结构和计算方法，我们把输入数据设置为 $a_1 = 1, a_2 = 2, a_3 = 3$ ，初始权重也如下图进行设置。



$$a_1' = 1 * 0.5 + 2 * 0.4 + 3 * 0.3 = 2.2$$

$$a_2' = 1 * 0.6 + 2 * 0.3 + 3 * 0.2 = 1.8$$

$$a_1'' = 2.2 * 0.7 + 1.8 * 0.8 = 2.98$$

可以得到 a_1'' 的值为2.98，该值就为预测值。注意这里为了简化计算，省去了偏置参数b，省去了激活函数的使用。

调整参数的两种方法

但实际上我们训练时，输入的数据是这样的： $a_1 = 1, a_2 = 2, a_3 = 3$, $y = 3.2$ ，即真实值为3.2。也就是说预测值2.98与真实值3.2之间有0.22的差异。为了简化理解，先假定我们的目标是使得差异为0。也就是说我们需要调整 $w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, w_{32}, w'_{11}, w'_{21}$ 这些参数的值，使得计算的预测值等于3.2。

那么显而易见可以想到两种方法：

从前到后迭代

这个方法非常简单，就是把 $w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, w_{32}, w'_{11}, w'_{21}$ 的可能取值做一个微调，不断去试探。假设把

$$w_{11} = 0.5, w_{12} = 0.6, w_{21} = 0.4, w_{22} = 0.3, w_{31} = 0.3, w_{32} = 0.2, w'_{11} = 0.7, w'_{21} = 0.8$$

调整为：

$$w_{11} = 0.5, w_{12} = 0.6, w_{21} = 0.4, w_{22} = 0.3, w_{31} = 0.35, w_{32} = 0.2, w'_{11} = 0.78, w'_{21} = 0.83$$

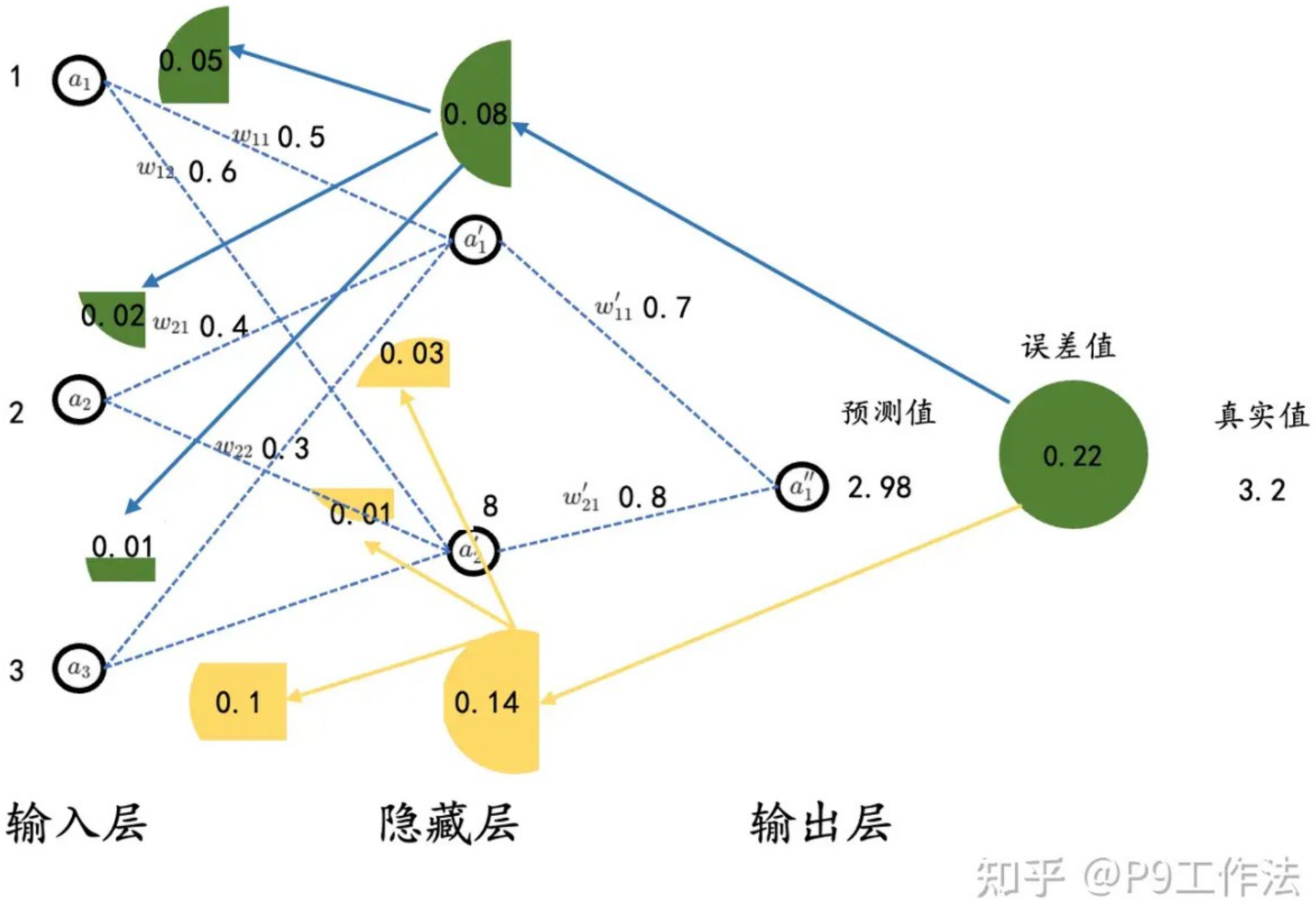
注意，只是修改了 w_{31}, w'_{11}, w'_{21} 的值，最终计算结果 $a''_1 = 3.293$ ，比起来真实值3.2还是有一些误差，但不过已经比较接近了。

这样的做法看起来可行，但要真的迭代出来这样的参数组合代价是实在是太大了。首先是这样的取值组合有无数个，就这8个参数就非常计算了。如果参数多大上千亿，这样的迭代法计算开销实在太大，在有限的时间肯定是不可能完成的。

从后到前计算

既然从前到后迭代不行，可以尝试把误差值传到模型中去，反向去找哪个参数对误差的贡献最大，然后去调整这个参数即可。这显然是有极大的难度，因为相当于知道了结果但是要反向去推导哪个原因的贡献最大。举个下围棋为例，每当下完一盘棋，最后的结果要么赢要么输。我们会思考哪几步棋导致了最后的胜利，或者又是哪几步棋导致了最后的败局。如何判断每一步棋的贡献就是贡献度分配问题，这是一个非常困难的问题。

可以形象化理解为把误差这个饼不断往前去分，直到分干净为止，如下图所示：



如何做反向传播✦

形象化的表达只能简化理解，而真正要运用还需要用数学公式表达出来。把上面的过程稍作归纳和整理，得到完整的反向传播应该有如下三个过程：

1、前向传播

输入层到隐藏层： $z_1 = W_1 x + b_1$, $a_1 = f(z_1)$ ，其中，f是激活函数（如ReLU、Sigmoid等）。

隐藏层到输出层： $z_2 = W_2 * a_1 + b_2$, $\hat{y} = f(z_2)$ 。如果有多个隐藏层则依次类推。

这里 W_1 , W_2 表示为矩阵，即 $W_1 = w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, w_{32}$ ， $W_2 = w'_{11}, w'_{21}$

2、计算损失

假定损失函数如下： $L = \frac{1}{2}\|\hat{y} - y\|^2$ （为什这里需要一个 $\frac{1}{2}$ ？），只是上面的例子为 $L = \hat{y} - y$ ，更多的损失函数接下来将单独篇章讲解。得到损失函数L后，我们的目的就是求出来一组参数，使得损失L最小，即数学表达式为 $minL$ 。

如果把这个函数 L 展开，肯定可以看到这是一个关于 $w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, w_{32}, w'_{11}, w'_{21}$ 的函数。

$L = \frac{1}{2}\|W_2 * A_2 + b_2 - y\|^2$ ，再把 a_1 展开，
$L = \frac{1}{2}\|f_2(W_2 * f_1(W_1 A_1 + b_1) + b_2) - y\|^2$

此时 $A2 = a'_1, a'_2$ ， $A1 = a_1, a_2, a_3$ 。

3、反向传播

上面的损失函数 $L = \frac{1}{2}\|W_2 * A_2 + b_2 - y\|^2$ 中，在反向传播过程中其实只有W和b是参数，因为真实值y和输入数据 A_1 都是明确的。

所以求解 $minL$ 问题，其实就是求解什么W和b的取值下，L的取值最小。而函数的极值求解其实是一个函数求导数（如果是高阶函数就是求偏导），若导数为0的地方则表示该这里的增速为0，应该是极致点（是否是最大/最小值不一定，要看函数的性质）。

所以在这里可以简单将最小化损失函数的求解问题理解为求导问题（肯定不严谨，但不妨碍理解），令导数为0，再去解方程组就能够得到具体的W和b。

至此，终于找到了一个数学工具去寻找最优的参数值，而不是完全靠迭代试错的暴力计算。但这里还是没有具体求解出来参数W和b，下面一篇会继续剖解梯度下降法✦，来具体求出来参数W和b。