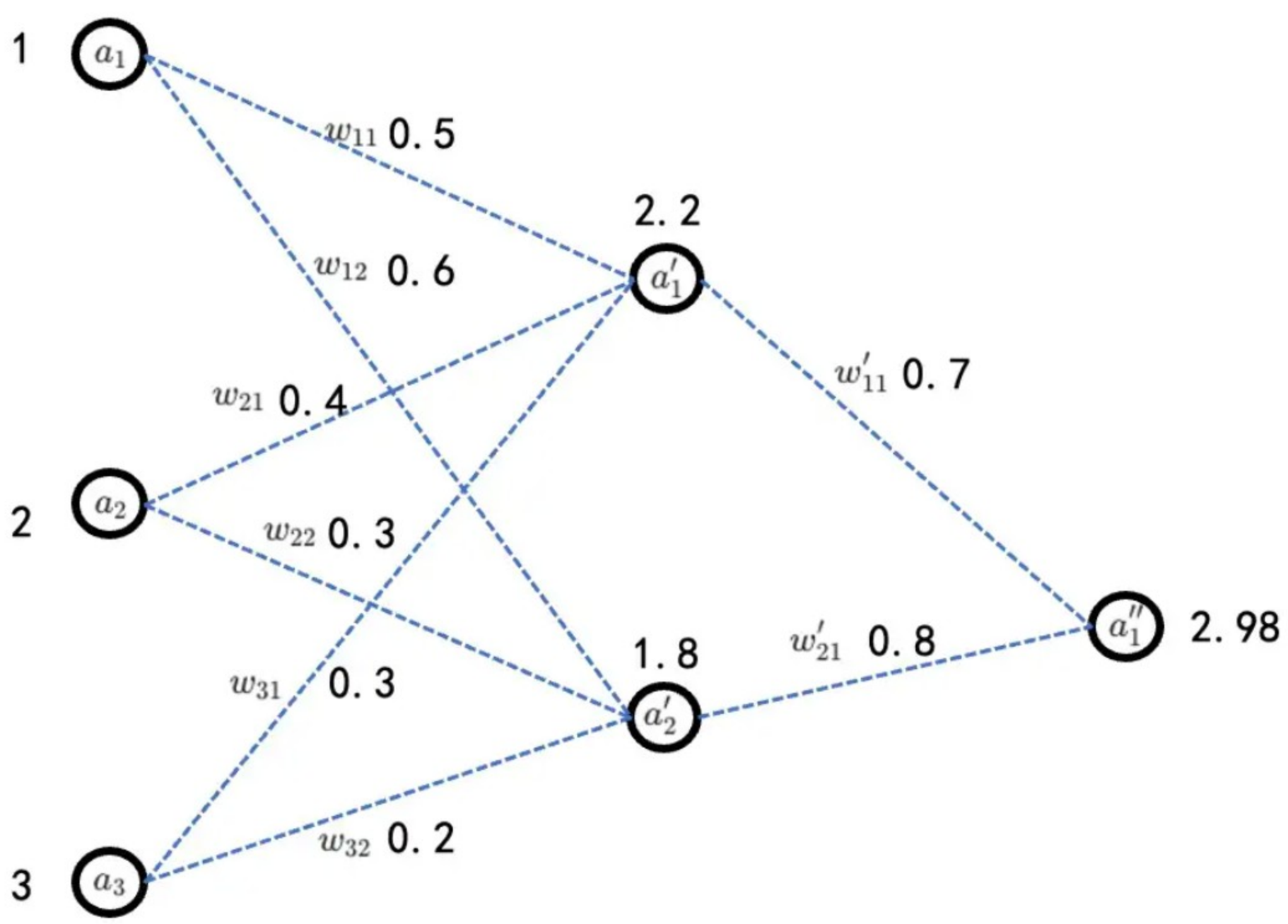
神经网络如何计算结果

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：

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

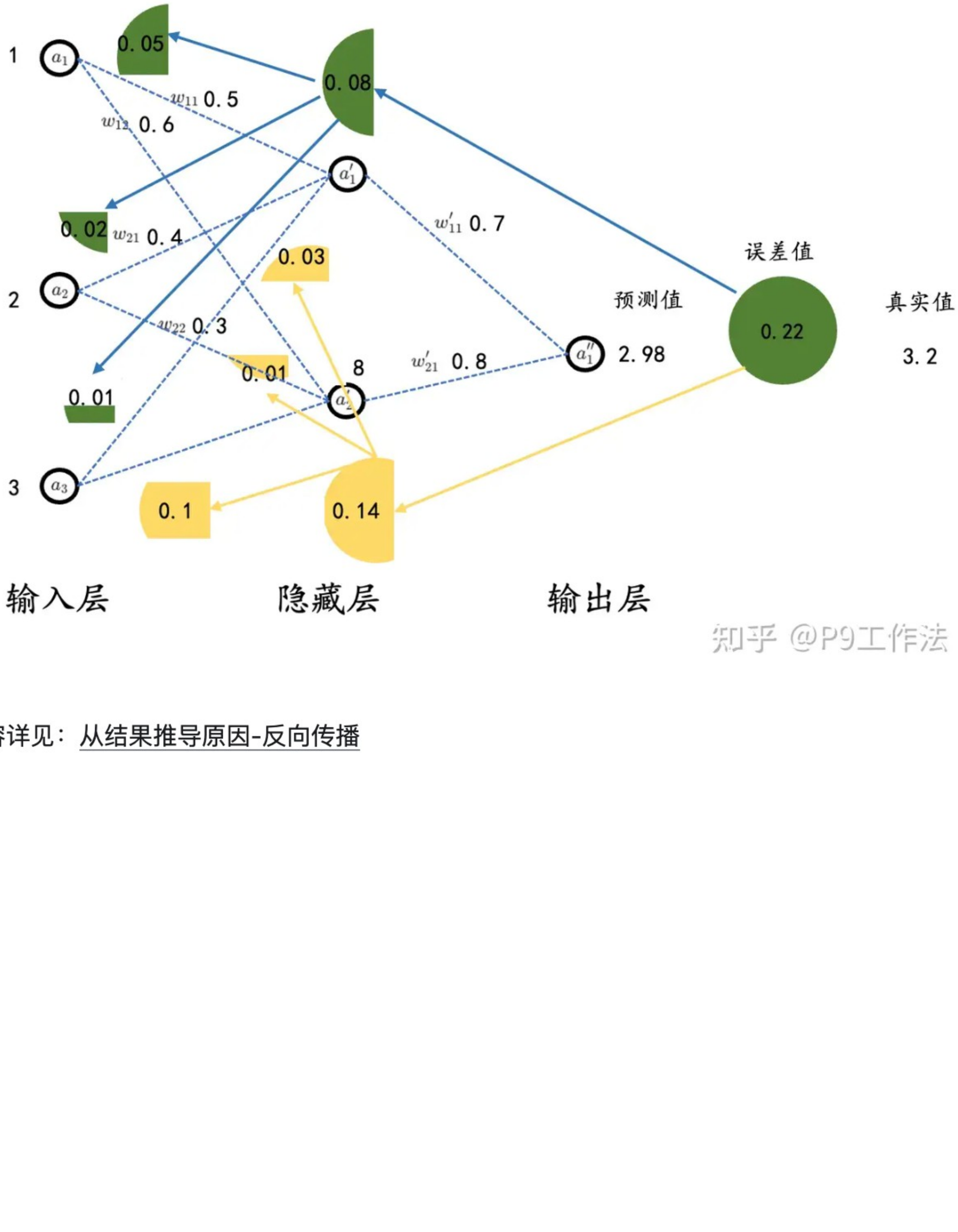
注意，只是修改了 w_{31}, w'_{11}, w'_{21} 的值，最终计算结果 $a''_1 = 3.293$ ，比起来真实值3.2还是有一些误差，但不过已经比较接近了。

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：



更多内容详见：从结果推导原因-反向传播