

目前的问题有：

- 1) 一个图的特征要怎么提取？
- 2) 卷积核怎么选？
- 3) 为什么RELU代替sigmoid可以解决梯度消失问题？
- 4) 全连层的BP算法某些细节不懂。

传统方法提取特征，是用指定的滤波器对图像进行卷积计算，得到的结果就是特征，因为不同类型的图得到的计算结果是不一样的，所以可以用作特征，深度学习就是用的自己学习的卷积核来做卷积计算，这个结果就是特征，深度学习中卷积核是自己学习的，不需要选

可以看出，如果想要训练网络，就需要根据误差更新权重，而如果想要获得误差 E ，不论是MSE,还是K-L divergence, 都需要两种参数：期望输出 r ,和当前层权重 a (回顾公式，即 w, b)。其中期望输出 r 来自标签集，很容易获得，而 a 和误差 E 相互影响。那么，解决方式就很明显，我们可以先固定一方，更新另一方，这是alternating optimazition优化多参数模型的经典思路。CNN的训练方法思路也来自于此，被称作backpropagation。

bp暂时不懂很正常，需要你们先看一部分的吴恩达的课程，或者李航的统计学习，或者周志华的西瓜书，有一点概念的基础就很好懂了，现在不懂没关系

Backpropagation算法，大概可以分为两步：

- 1. 通过训练数据计算网络中的所有 a ，这里可以回想一下 a 的计算方法，最初的 a 只需要输入图像和初始化权重就可以计算，这一步是从输入图像到输出层的计算，即上图中的前向传播。
- 2. 获得所有 a 之后，再我们就可以通过目标函数和期望输出计算出最后一层的 E ,而有了最后一层的 E ，可以计算出倒数第二层的期望输出 r ，以此类推，可以计算误差到第一层，并通过求导更新权重。这是上图的后向误差传播（这里表述不严谨）。

上述1, 2部操作会交替进行。

神经网络中的回传幅度是通过当前位置后一层的梯度确定的，可以看到sigmoid绿色的那个圈圈画出来的地方，曲线接近平滑，也就是说这个函数在这里的梯度（也就是斜率）接近于0，当网络的值落在这个范围，就没啥可以回传的了，因为梯度太小了一次梯度趋于0之后，越往回就越没啥可以传的，这就是梯度消失，relu选择放弃小于0的值，大于0的部分梯度横为1，自然就不会消失了

4. 用(5) 计算

, 并后向传播，用(4)计算出所有其他层的

$l = L - 1, L - 2, \dots, 2$

5. 利用 (6) 更新权重

