The application of SoftMaxlayer

基科 31 白可 2013012245

这次作业，在 multilayer perception 的最后的 evaluation function 中加入了 softlayer. 上次作业我采用了控制变量的方法，调整各层的权重，分别讨论了两层、三层的 Relu 和 sigmoid。先看实验结果再讨论内容。但是效果不是很好，最终的实验报告显得有些混乱，意义不大。因此这次作业中，我将以得到的结论为主，顺带分析实验数据，也解释了我实验一没有弄明白的几个点。

1. 多层的 hidden layer 网络不一定优于单层的网络。

类型：sigmoid + softmaxlayer

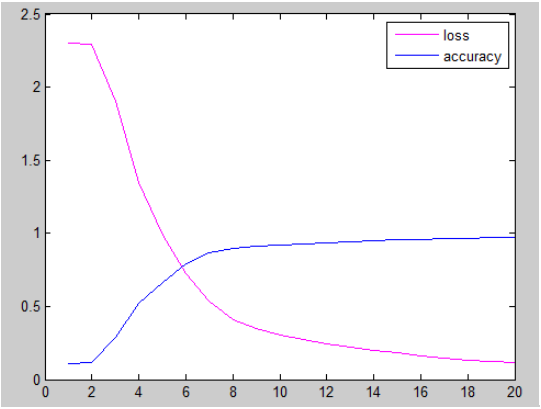
步长：0.08 – 0.03 -0.01

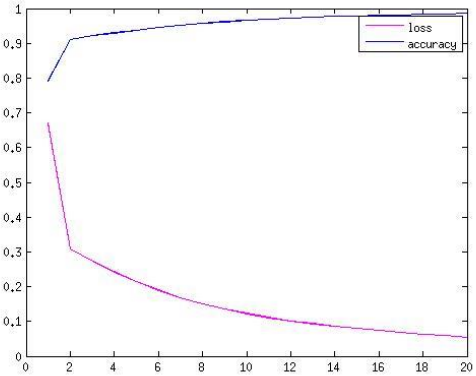
	训练准确度	测试准确度	训练损失	测试损失
784 x 256 x 10 x 10	97.28%	96.37%	0.14	0.137
784 x 256 x 10	98.62%	97.44%	0.05	0.089

步长：0.08-0.03-0.01

	训练准确度	测试准确度	训练损失	测试损失
784 x 256 x 64 x 10	98.14%	97.00%	0.069	0.102
784 x 256 x 10	98.62%	97.44%	0.05	0.089

收敛速度：左侧为三层模型，右侧为两层模型。差别不大。





原因 1：

backpropagation 中，我们能够发现，过程是个“乘法”，尤其在 sigmoid 中，（1-y）y, 是一个小于零的值，两层过后，这个值可能会非常小，因此对第一层的影响就不大了。无法达到调整权重的目的。

但是这里我们只有两层，有时不会影响很大。

原因 2：

两层的 hidden layer 的参数自由度更大，容易“过拟合”

2. Softmaxlayer 的效果要优于 euclidenlayer.

Sigmoid

步长：0.08-0.03 两层

	训练准确度	测试准确度	训练损失	测试损失	速度
euclid	95.09 %	94.31%	0.089	0.105	106s
Softmax	98.62%	97.44%	0.05	0.089	100s

Relu

步长：0.08-0.03 两层

	训练准确度	测试准确度	训练损失	测试损失	速度
euclid	99.14 %	97.98%	0.0028	0.039	87s
Softmax	100.00%	98.15%	0.0027	0.0687	84s

准确率提升，训练损失减小，速度相近。

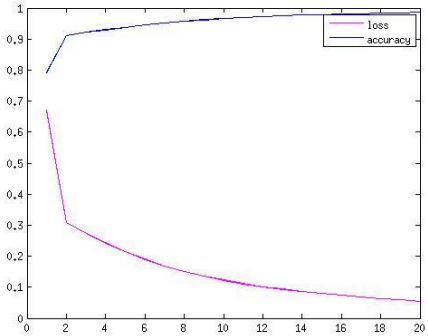
原因：

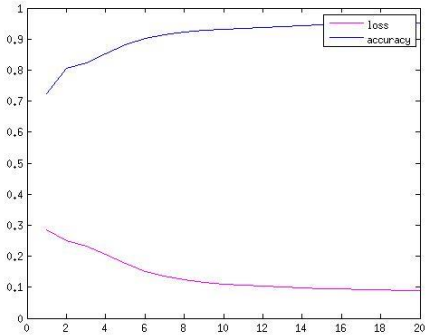
我看来，结构上，这两者是相似的：fc 层 + 转化为 $0-1$ 之间的“概率”。有一点不太一样。在最后一层，sigmoid 结构始终是在自己的输入上变化。但是 softmax 加入了归一化因子，这意味着每个神经元的输出与其他的神经元相关。这也更加符合生理模型。这可以解释更高的正确率。

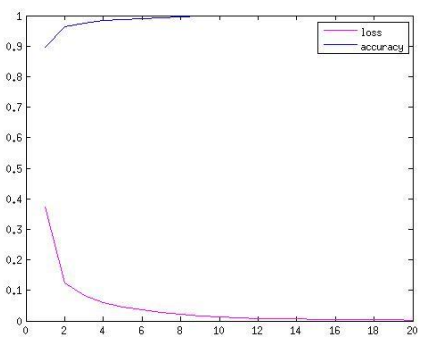
速度的优化可能是因为在计算第二层的 fc layer 时候，eucliden distance 的类型，需要计算乘法即$\delta = (\text{output} - \text{right answer}) \times \big(1 - \text{f}(\text{x})\big)\big(\text{f}(x)\big)$, 且多了一层，会增加参数传递，函数调用的时间。

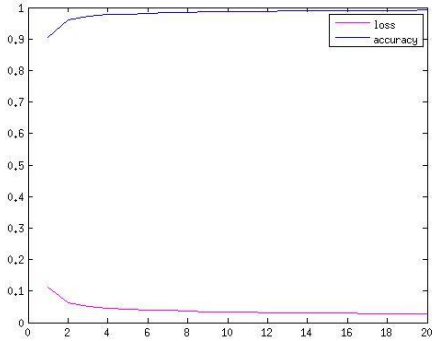
收敛速度：

左侧为 sigmoid + softmaxlayer 右侧为 sigmoid + euclidenlayer，可以看到，softmax 的收敛速度更快，最后的准确度也更高。









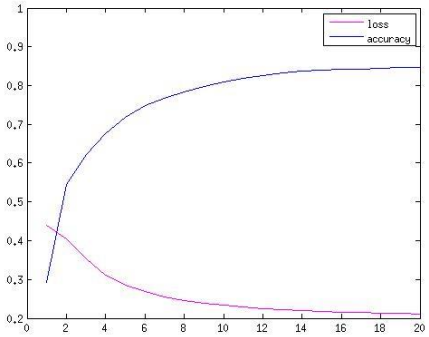
左侧为 Relu + softmaxlayer　右侧为 Relu + euclidenlayer，可以看到，softmax 最后的准确度也更高，收敛速度相对较快，不过由于 Relu 本身收敛就较快，所以两者差异不明显。

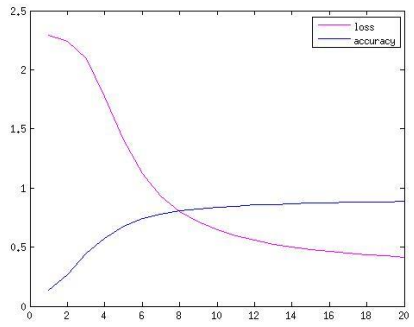
3. softmaxlayer 和 euclidenlayer 在步长较小时都会进入局部极小值

步长：0.001-0.001 两层

	训练准确度	测试准确度	训练损失	测试损失	速度
euclid	84.76%	85.11%	0.211	0.20	116s
Softmax	88.63%	89.32%	0.4145	0.39	112s

收敛速度：





左侧是 euclid ，右侧是 softmax，可以看到，他们最后的准确度不是很高，但是 softmax 稍微优于 EUCLID，但不是很明显。

分析：

两者结构类似，步长较小，陷入局部极小值是正常。因此需要寻找合适的步长。

4. relu 的效果好于 sigmoid

这个上一回实验报告我们曾经讨论过，根据 section 2 的实验结果也可以看到 relu 的训练正确率已经到了 100%。但是上回没有说清楚。

原因一：

Relu 函数是“稀疏”的一种表现，与人脑的神经元类似。稀疏特征处在高维特征空间中。而 relu 函数可以从数据集中抽象出鲁棒性更强的特征。可能某一个特征就决定了某个关

键步骤，优化分类。

但是这样的稀疏性也是有一定限制的。我在 Relu 中，将原函数变成了

$$\begin{cases} x & x > 2 \\ 0 & x < 2 \end{cases}$$

收敛速度和正确率有所下降。

原因二：

减轻了 vanish gradient problem. 使用 sigmoid 时两端饱和较快，想到这里我把激活函数变成了线性（完全去掉了激活函数），正确率大概 40%，因此，可以看到非线性带来的良好效果。

且反向传播时，error 成倍衰减。这解释了我在实验一中，为什么当第一层的步长取相对第二层较大时，效果较好。但是在 Relu 中就不存在这样的问题。单端饱和，梯度变化较快，收敛速度很快。

总结：

Softmaxlayer 的思想在于他定义了一种可以被当做误差函数计算的“概率”。考虑到一个 cell 在整个群体中所起到的作用，因为最后的概率是一个输出值除以所有的加和。让我体会到了生理模型在 ANN 的神奇之处。

但是 Relu 的值域趋近于无穷。显然是不符合生理模型的。但我想也可以这么理解。可能被激活的函数所对应的不是一个 Neuron 而是一群，他们的和就可以表达一个任意大的输出。