CV Report I

18300290007 加兴华

0. 实验地址

GitHub Ropo：

huakyouin/CV-PJ1: 计算机视觉的PJ仓库 (github.com)

模型地址：

Google网盘

1. 模型实现与设置

激活函数

我实现了ReLu 和 Sigmoid 激活函数，在我的模型框架中，激活函数作为独立的一层被模块化。经过前期的实验探索，发现使用ReLu 作为激活函数效果远好于Sigmoid，因此后续在参数查找时我使用sigmoid来突出参数组合的性能差异；而在之后计算最优组合的模型精度时使用ReLu。

反向传播

在我的模型框架中，各个 'layer' 类里都定义好了自己的反向传播函数，它们可以被损失函数类中的gradient函数调用，从而实现整个网络的反向传播。

优化器

我首先实现了基本的SGD优化器，然后在此基础上实现了带动量的Momentum优化器:

$$w^{t+1}=w^t-\alpha_t\nabla f\left(w^t\right)+\beta_t\left(w^t-w^{t-1}\right),\quad \beta_t=0.9$$

由于前期尝试效果不理想，我又实现了效果更强的RMSprop优化器:

$$S_{grad}=\beta S_{grad}+(1-\beta)grad^2,\quad \beta=0.98$$

$$W_{next}=W-lr\cdot\frac{grad}{\sqrt{S_{grad}}+\epsilon}$$

在我的模型框架中，优化器作为独立的模块可以在计算好模型梯度后进行调用。

学习率衰减策略

在我的模型框架中，学习率衰减策略采用指数衰减的方式，形如下式：

$$lr = lr_0 \cdot \gamma^{\text{epoch\_num}}$$

注：学习率衰减的目的是解决mini_batch训练时产生的过拟合现象，而像RMSprop等优化器的设计旨在处理优化时遇到鞍点，因此是可以共用的。论文《DECOUPLED WEIGHT DECAY REGULARIZATION》4.1节对此有类似的阐述。当然，由于使用了性能更好的优化器，γ 的取值可以设的很小，在本项目中取为了0.999。

它被嵌入在优化器模块中。

正则化

二范数正则化公式如下：

$$\frac{\partial Loss}{\partial W} = g + 2\lambda W$$

它同样被嵌入在优化器模块中。

模型存储与调用

在我的模型框架中，每次运行都需要重新定义网络结构，涉及到存储和读取的只有网络中的参数。模型的训练、测试、保存、读取函数都在框架最外层的路由模块中实现了。

2. 参数查找

我首先对以下范围进行了查找：

```
nhiddens=[100,128,150,200,300]
lrs=[1e-5,1e-4,1e-3,1e-2]
penaltys=[1e-7,1e-6,1e-5,1e-4,1e-3,1e-2]
```

设定运行2轮次，无学习率衰减，sigmoid激活，各组合及其测试集精确度详见MNIST文件夹中的result.txt。

观察模型发现，不论隐含层单元个数多少，学习率总是在0.001时模型最佳，因此我取定学习率初始值为0.001。

接着，我发现组合结果在隐含层单元和正则项强度维度明显不对称，因此对范围进行了扩展继续查找：

```
nhiddens=[200,300,350,400]
lrs=[1e-3]
penaltys=[5e-8,1e-7,5e-7,1e-6,5e-6]
```

同样设定运行2轮次，无学习率衰减，sigmoid激活，各组合及对应测试机精确度继续写入在reusult.txt中。

程序运行结束输出如下：

```
测试集准确率:91.56%
最优参数组合为[350, 0.001, 5e-06],精度为91.69%
```

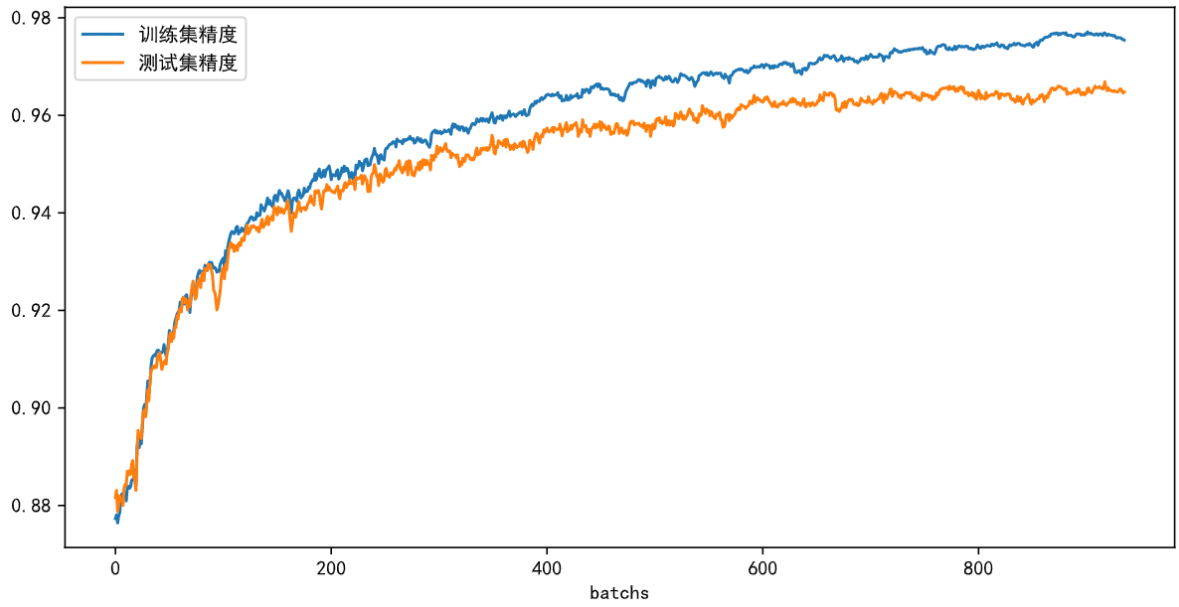
于是接下来在这组参数组合的基础上加上学习率衰减，使用relu激活函数，并迭代至收敛，程序运行结束输出如下：

```
测试集准确率:96.79%
```

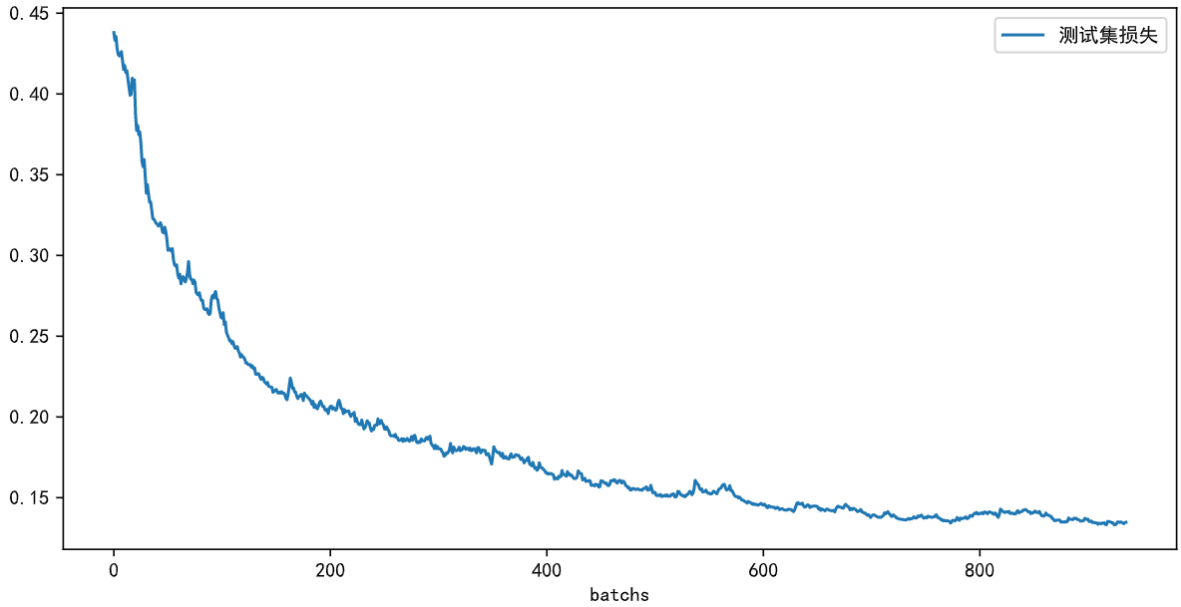
3. 结果可视化

网络性能

模型的accuary曲线



模型的loss曲线（测试集上）

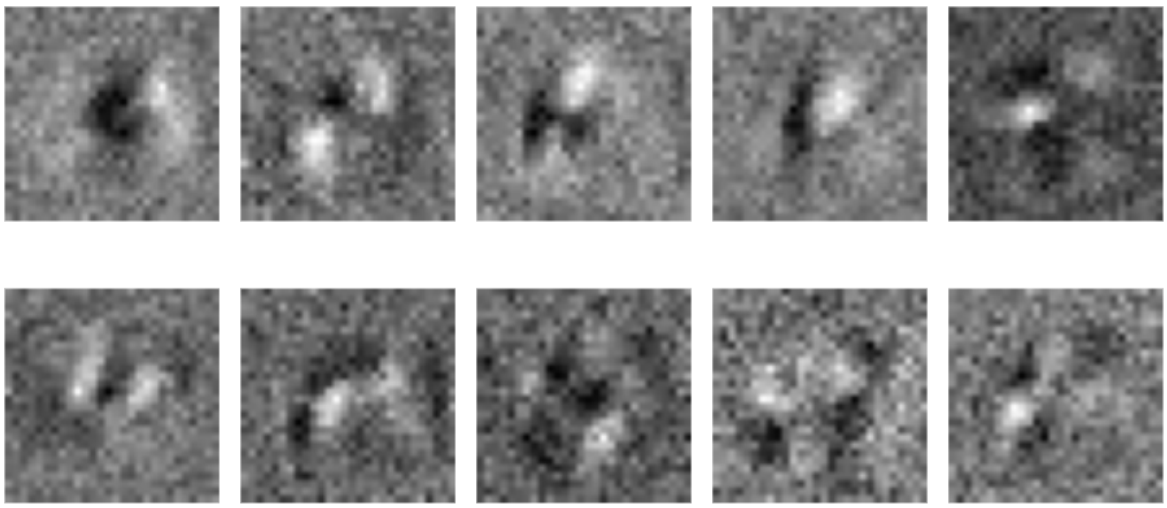


可以看出随着训练，模型在训练集和测试集上的表现是比较一致的，网络泛化性能较好。

网络参数可视化

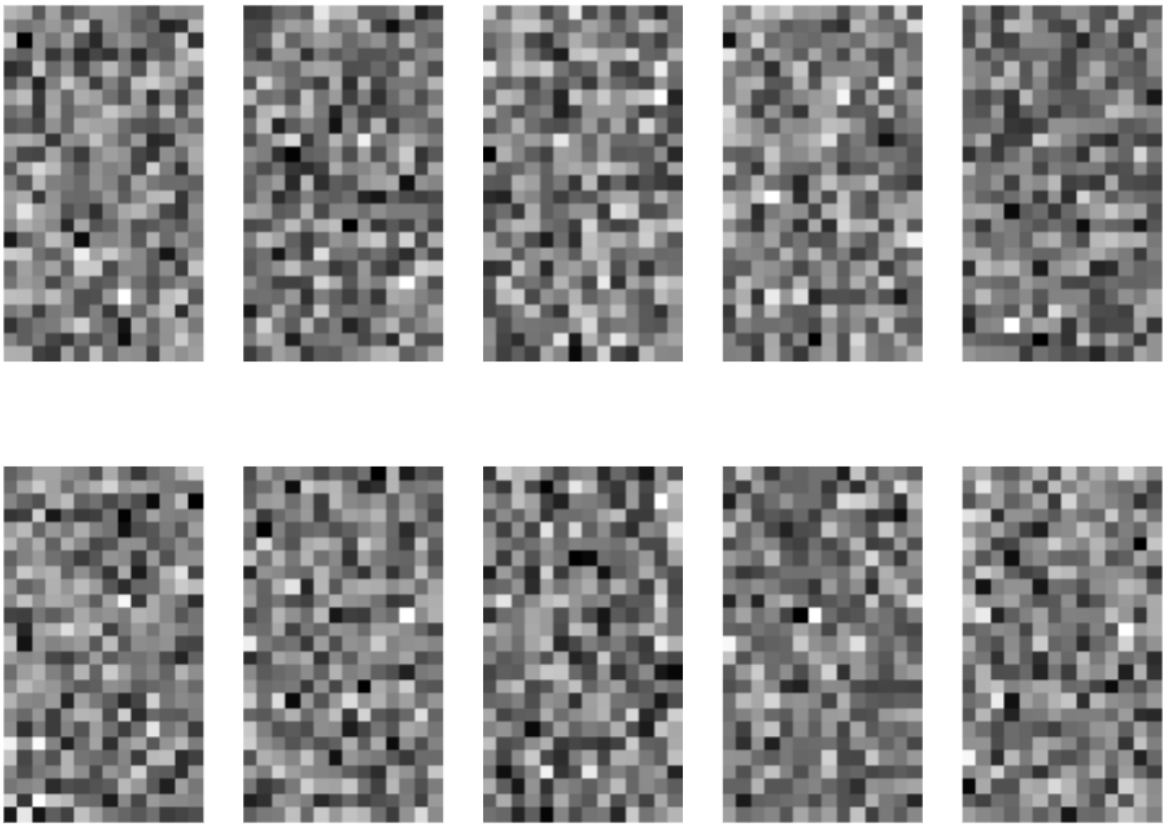
输入到隐含层

这部分网络的形状为784x350，应用主成分分析对第二个维度提取10个主成分，然后把每个主成分矩阵化后可视化结果如下：



隐含层到输出层

这部分网络的形状为350x10，把指向每个分类器的向量矩阵化后可视化结果如下：

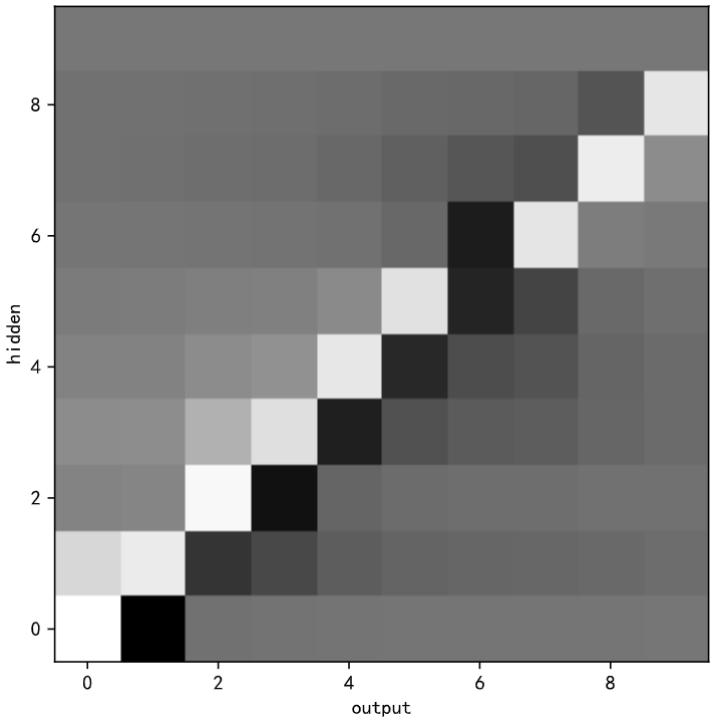


对比发现：

- 第一层权重的可读性较好，权值大的区域隐约可以与字迹的轮廓对上
- 第一层权重更加平滑，噪声较少

隐含层到输出层--反向分析

我继续对第二层网络进行反向主成分分析，将350x10降维到10x10，可视化结果如下：



可以看到，这时隐含层与输出层有很明显的线性关系，说明第二层网络的分类效果与主成分分析类似，将350维的数据降至10维。

总结

通过对参数网络的可视化，我得出以下结论：

- 第一层网络作用为特征提取，从图像数据中提取字迹的轮廓特征，可读性高
- 第二层网络的作用为降维，将轮廓特征降维到分类量级，可读性低