第三次上机实验报告

北京大学化学与分子工程学院 李梓烙 2101110396

摘要 本实验运用具有两层卷积层与两层池化层的卷积神经网络，采用MMIST手写数字数据集进行训练，并对其进行测试，计算相应的混淆矩阵、准确率、召回率，从而得出结论：（1）在默认条件下，采用上述卷积神经网络，验证集与测试集的预测准确率均达到了98%以上，且各数字的精确度与召回率也达到了97%以上，表明默认条件下，训练得到的卷积神经网络即有非常良好的表现；（2）各数字中预测错误结果的分布具有一定倾向，其中，误判频率最高的几组数字，在结构上恰好有一定相似之处，如果书写者写字潦草，连人工识别都有可能出错，如4与9、3与5、0与6等；（3）改变学习率与分批大小可在一定程度上改善模型的预测效果，其中，学习率为0.001，分批大小为64，可以在验证集与测试集取得较好的表现；（4）该模型可以识别笔者手写的黑底白字，而无法识别白底黑字，为使该模型具备识别白底黑字的能力，还需输入白底黑字的样本进行训练。

关键词 卷积神经网络 手写数字 识别

1 引言

2 实验部分

2.1 仪器

2.1.1 硬件

Surface Pro（第5代，处理器参数：Intel® Core™ i5-7300U CPU @ 2.60GHz，2.71 GHz，2个内核，4个逻辑处理器；内存容量：8.00 GB）

2.1.2 软件

操作系统：Windows 10家庭版，版本21H1

开发环境：Visual Studio 2019 Community，64位Anaconda 3（版本号2021.05，含64位Python 3.8.8、Conda 4.10.1、NumPy 1.20.1、Pandas 1.2.4、SciPy 1.6.2、Scikit-learn 0.24.1、Matplotlib 3.3.4、Pytorch 1.10.0、Torchvision 0.11.1）

2.1.3 训练和测试数据

训练数据：train-images-idx3-ubyte和train-labels-idx1-ubyte：训练用手写数字和标签（均转存为二进制文件），共计6万个样本。

测试数据：t10k-images-idx3-ubyte和t10k-labels-idx1-ubyte：测试用手写数字和标签（均转存为二进制文件），共计1万个样本。

2.2 实验过程

2.2.1 数据的读入

利用PyTorch包中的torch.utils.data.Dataset，从MNIST官网上下载训练数据与测试数据，保存在工作目录下的data/MNIST中；若该目录下存有数据，则直接读取。

2.2.2 卷积神经网络参数的设定

本次实验采用两隐藏层的卷积神经网络，其中，每一层隐藏层均包含一个卷积层与一个池化层。第一个卷积层采用5×5的卷积核，输出通道数为16，池化层采用2×2的池化核；第二个卷积层采用3×3的卷积核，输出通道数为32。之后的输出层先后连有ReLU与Log-Softmax各一个，其中ReLU将转换后的隐藏层输出数据缩减为128个独立变量，而Log-Softmax则在此基础上进一步缩减为10个独立变量，分别代表输入图片为数字0~9的可能性。

若不特别说明，以下模型默认的超参数如下：一次训练所选取的样本数batch_size为32；初始学习率lr为0.001；权值衰减weight_decay为10^{-5}。

2.2.3 默认参数下，卷积神经网络处理多分类问题的效果

为卷积神经网络设定上述参数，随后采用Adam迭代器，训练卷积神经网络。待训练完毕，对验证集与测试集分别进行预测，并根据预测结果，计算出该模型分类不同数字时的精确度与召回率，由此判断这一模型容易将哪些数字错误指认为其他数字。随后，将卷积神经网络对测试集的预测结果，输出至mnist_test_prediction.csv。

2.2.4 不同超参数下卷积神经网络的表现

为进一步探究不同超参数对卷积神经网络预测表现的影响，笔者决定调节分批大小batch_size与初始学习率lr。本次实验中，可选的batch_size有32、64、128三种，而学习率则有0.1、0.01、0.001、0.0001四种，将上述超参数条件相互组合，并训练三次，取其准确率平均值，作为模型衡量的指标。

2.2.5 对手写数字的预测分类

在Windows自带的画图软件中，用Surface Pro的触控屏手写数字0-9并保存，再用Photoshop反色，得到反色后的数字0-9。用2.2.4得到的最佳模型，对两种不同底色的数字进行预测，观察得到的结果。

3 实验数据与结果分析

3.1 代码中卷积神经网络模型的结构

本次实验采用两隐藏层的卷积神经网络如图3-1所示，其中，每一层隐藏层均包含一个卷积层与一个池化层，以提取输入数字的特征。之后的输出层先后连有ReLU与Log-Softmax各一个，目的是将隐藏层输出数据缩减为10个独立变量，分别代表输入图片为数字0~9的可能性。具体的参数设置可参见2.2.2节。

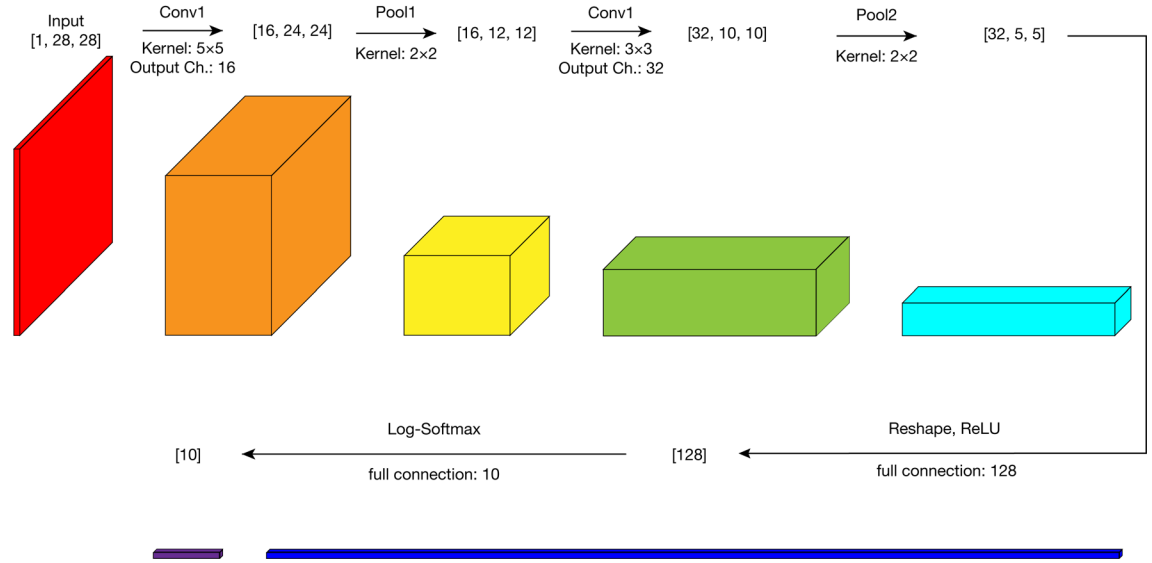

图1 代码中卷积神经网络模型的结构示意图

3.2 给出输入、输出网络每一层数据的维度

根据PyTorch文档的设定，结合程序中设定的参数，可以写出卷积神经网络每一层数据的维度，如表1所示。

表1 卷积神经网络每一层数据的维度

Layer	Input	Output
layer1	[1, 28, 28]	[16, 12, 12]
conv1	[1, 28, 28]	[16, 24, 24]
pool1	[16, 24, 24]	[16, 12, 12]
layer2	[16, 12, 12]	[32, 5, 5]
conv2	[16, 12, 12]	[32, 10, 10]
pool2	[32, 10, 10]	[32, 5, 5]
out_layer	[32, 5, 5]	[10]
fc1	[32, 5, 5]	[128]
fc2	[128]	[10]

3.3 简要解释下列名词的含义及其在网络训练中发挥的功能

Batch size：（样本）分批大小，指一次训练中采用的样本数量，它的作用是在内存消耗、模型精度、模型收敛速率之间达成平衡；

Adam Optimizer：全称自适应矩估计优化算法，其在计算每一步优化步长时均会考虑梯度均值及梯度平方的影响，它的作用是拟合卷积神经网络的各项参数。

Learning Rate：学习率，指每次梯度下降中允许的损失函数变化的（相对）大小，它决定了单次梯度下降的快慢，其作用为在训练时间与模型精度间寻找平衡。

Weight Decay：权值衰减，是放在正则项（regularization）前面的一个系数，相当于岭回归中的正则项系数C，它的作用是防止模型过拟合。

NLL Loss与Cross Entropy Loss的联系与区别：NLL Loss与Cross Entropy Loss均为损失函数，其作用是衡量预测值与实际值的偏差程度。两者不同之处在于，Cross Entropy Loss需要将输入值用Log_Softmax处理后，再用NLL Loss计算，即Cross Entropy Loss = Log_Softmax + NLL Loss。

Epoch：（训练的）代数，当采用全部的样本数据，对神经网络模型进行训练，此时这一过程称为“一代（训练）”，增加训练代数，有利于提高模型的准确程度。

3.4 默认参数下，卷积神经网络处理多分类问题的效果

笔者采用默认参数，对现有的卷积神经网络进行训练。由于训练方法采用了随机梯度下降，为保证实验的可信度，总共进行三次实验，得到的对验证集与测试集的预测结果如表2至表6所示。

表2 卷积神经网络对验证集与测试集预测的准确率

	验证集	测试集
第一次	0.9883	0.9908
第二次	0.9864	0.9867
第三次	0.9895	0.9886
平均值	0.9881	0.9887

表3 卷积神经网络对验证集的混淆矩阵

	0	1	2	3	4	5	6	7	8	9
0	1000/1010/1004	1/0/0	0/0/2	1/0/0	0/0/0	0/0/0	4/0/3	1/0/0	2/1/0	2/0/2
1	0/0/0	1103/1103/1112	4/2/1	2/1/2	1/0/0	1/0/0	2/0/0	0/9/0	1/0/0	1/0/0
2	1/1/0	0/0/0	954/957/961	3/1/1	1/0/0	0/0/0	0/1/0	4/3/3	4/4/2	0/0/0
3	0/0/0	0/0/1	0/0/0	1035/1034/1033	0/0/0	3/3/2	0/0/1	0/0/0	4/3/4	1/3/2
4	0/2/1	3/3/2	2/1/1	0/0/0	985/980/992	0/0/0	2/1/1	0/1/0	2/1/0	5/10/2
5	1/3/0	0/0/0	0/0/0	4/6/3	0/0/0	874/871/873	1/1/3	0/0/0	1/0/1	0/0/1
6	1/10/2	0/2/0	0/0/0	0/0/0	0/0/0	2/12/4	970/948/966	0/0/0	0/1/1	0/0/0
7	0/0/0	1/0/5	2/3/4	2/1/1	3/2/4	0/0/0	0/0/0	977/979/971	2/2/1	1/1/2
8	2/4/2	0/3/1	2/3/3	0/0/0	1/0/3	1/5/2	3/1/6	0/0/0	1021/1014/1012	2/2/3
9	1/3/1	0/0/0	0/0/0	0/2/0	6/4/8	5/6/4	0/0/0	9/5/5	6/3/2	964/968/971

表4 卷积神经网络对测试集的混淆矩阵

	0	1	2	3	4	5	6	7	8	9
0	974/978/974	0/0/1	0/0/2	0/0/0	0/0/0	0/0/0	3/0/2	2/1/1	1/1/0	0/0/0
1	0/0/1	1124/1128/1130	1/0/0	3/2/3	0/0/0	1/2/0	5/0/1	0/3/0	1/0/0	0/0/0
2	1/3/1	1/0/2	1023/1023/1024	0/0/2	1/1/0	0/0/0	0/1/1	4/3/1	2/1/1	0/0/0
3	0/0/0	0/0/1	1/2/1	1006/1004/1002	0/0/0	1/3/3	0/0/0	1/0/0	1/1/1	0/0/2
4	0/0/0	0/0/0	0/0/1	0/0/0	978/964/978	0/0/0	0/3/0	0/1/0	1/2/0	3/11/3
5	1/3/1	0/0/0	0/0/0	6/4/7	0/0/0	884/884/881	1/1/1	0/0/0	0/0/2	0/0/0
6	4/20/3	1/3/2	0/0/0	0/0/0	3/1/2	3/10/1	945/924/950	0/0/0	2/0/0	0/0/0
7	0/0/0	1/0/11	5/9/15	0/0/0	0/0/1	0/0/0	0/0/0	1019/1016/997	1/1/2	2/2/2
8	1/4/2	0/0/0	3/1/2	0/0/0	0/0/4	0/3/1	0/0/2	0/0/0	968/964/960	2/2/3
9	0/2/0	0/0/3	0/1/0	0/1/0	8/3/8	2/12/5	0/0/0	5/4/1	7/4/2	987/982/990

表5 卷积神经网络对验证集与测试集各数字的预测精确度

	验证集	验证集平均值	测试集	测试集平均值
0	0.9940/0.9777/0.9940	0.9886	0.9929/0.9683/0.9919	0.9844
1	0.9955/0.9928/0.9920	0.9934	0.9973/0.9973/0.9826	0.9924
2	0.9896/0.9907/0.9887	0.9897	0.9903/0.9875/0.9799	0.9859
3	0.9885/0.9895/0.9933	0.9904	0.9911/0.9921/0.9882	0.9905
4	0.9880/0.9939/0.9851	0.9890	0.9879/0.9948/0.9849	0.9892
5	0.9865/0.9710/0.9864	0.9813	0.9921/0.9672/0.9888	0.9827
6	0.9878/0.9958/0.9857	0.9898	0.9906/0.9946/0.9927	0.9926
7	0.9859/0.9819/0.9918	0.9865	0.9884/0.9883/0.9970	0.9912
8	0.9789/0.9854/0.9892	0.9845	0.9837/0.9897/0.9917	0.9884
9	0.9877/0.9837/0.9878	0.9864	0.9930/0.9850/0.9900	0.9893

表6 卷积神经网络对验证集与测试集各数字的预测召回率

	验证集	验证集平均值	测试集	测试集平均值
0	0.9891/0.9990/0.9931	0.9937	0.9939/0.9980/0.9939	0.9953
1	0.9892/0.9892/0.9973	0.9919	0.9903/0.9938/0.9956	0.9932
2	0.9866/0.9897/0.9938	0.9900	0.9913/0.9913/0.9922	0.9916
3	0.9923/0.9914/0.9904	0.9914	0.9960/0.9941/0.9921	0.9941
4	0.9860/0.9810/0.9930	0.9867	0.9959/0.9817/0.9959	0.9912
5	0.9921/0.9886/0.9909	0.9905	0.9910/0.9910/0.9877	0.9899
6	0.9969/0.9743/0.9928	0.9880	0.9864/0.9645/0.9916	0.9808
7	0.9889/0.9909/0.9828	0.9875	0.9912/0.9883/0.9698	0.9831
8	0.9893/0.9826/0.9806	0.9842	0.9938/0.9897/0.9856	0.9897
9	0.9727/0.9768/0.9798	0.9764	0.9782/0.9732/0.9812	0.9775

根据表2至表6的数据，我们有如下发现：首先，在默认条件下，验证集与测试集的预测准确率均达到了98%以上，且各数字的精确度与召回率也达到了97%以上，表明默认条件下，训练得到的卷积神经网络即有非常良好的表现。

其次，对于各数字中预测错误的结果，其分布也具有一定倾向，这些误判倾向整理后如表7所示。在这些错误分类的数字中，误判频率最高的几组数字，恰好是结构上有一定相似之处的数字，如果书写者写字潦草，连人工识别都有可能出错，如4与9、3与5、0与6等。

表7 不同数字在验证集与测试集的误判倾向

数字	验证集误判倾向	测试集误判倾向
0	6、8、9（主要）；1、2、3、7（次要）	6、7（主要）；1、2、8（次要）
1	2、3、7（主要）；4、5、6、8、9（次要）	3、5、6、7（主要）；0、2、8（次要）
2	3、7、8（主要）；0、4、6（次要）	0、7、8（主要）；1、3、4、6（次要）
3	5、8、9（主要）；1、6（次要）	2、5、8（主要）；1、7、9（次要）
4	1、9（主要）；0、2、6、8（次要）	9（主要）；2、6、7、8（次要）
5	3（主要）；0、6、8（次要）	3（主要）；0、6、8（次要）
6	0、5（主要）；1、8（次要）	0、5（主要）；1、4、8（次要）
7	1、2、4、8（主要）；3、9（次要）	1、2（主要）；4、8、9（次要）
8	0、2、5、6、9（主要）；1、4（次要）	0、2、9（主要）；4、5、6（次要）
9	4、5、7、8（主要）；0、3（次要）	4、5、7、8（主要）；0、1、2、3（次要）

3.5 不同超参数下卷积神经网络的表现

在卷积神经网络的框架大致不变时，可供调节的超参数主要有学习率、分批大小、最大训练代数等，其中学习率控制梯度下降的幅度，过大会导致梯度可能会在最小值附近来回振荡，甚至可能无法收敛，过小则有可能下降缓慢，无法在较少步数内抵达损失函数极小值；分批大小控制一次学习的样本数，过大可能导致占用内存过高，而过小则同样导致梯度振荡；最大训练代数控制模型优化的次数，过大会使模型收敛后仍在训练，无意义地消耗时间，过小则使模型无法彻底达到收敛。

限于紧张的测试时间，笔者按2.2.4所述，仅调节学习率与分批大小，以期获得更好的预测准确率，而未测试最大训练代数，结果见表8与表9。

表8 不同超参数下卷积神经网络预测验证集的准确率（括号为平均值）

学习率\分批大小	32	64	128
0.1	0.0991/0.1011/0.0999 (0.1000)	0.1115/0.1115/0.0973 (0.1068)	0.1032/0.1115/0.1350 (0.1166)
0.01	0.9825/0.9669/0.9763 (0.9752)	0.9845/0.9855/0.9817 (0.9839)	0.9885/0.9833/0.9854 (0.9857)
0.001	0.9883/0.9864/0.9895 (0.9881)	0.9869/0.9877/0.9857 (0.9868)	0.9886/0.9874/0.9873 (0.9878)
0.0001	0.9848/0.9849/0.9869 (0.9855)	0.9833/0.9845/0.9839 (0.9839)	0.9806/0.9816/0.9810 (0.9811)

表9 不同超参数下卷积神经网络预测测试集的准确率（括号为平均值）

学习率\分批大小	32	64	128
0.1	0.1009/0.0980/0.0982 (0.0990)	0.1135/0.1135/0.0958 (0.1076)	0.0974/0.1135/0.1424 (0.1178)
0.01	0.9858/0.9645/0.9784 (0.9762)	0.9868/0.9876/0.9836 (0.9860)	0.9901/0.9870/0.9874 (0.9882)
0.001	0.9908/0.9867/0.9886 (0.9887)	0.9898/0.9908/0.9900 (0.9902)	0.9903/0.9899/0.9890 (0.9897)
0.0001	0.9871/0.9876/0.9873 (0.9873)	0.9865/0.9856/0.9839 (0.9853)	0.9819/0.9821/0.9843 (0.9828)

观察表8与表9，笔者发现，当学习率为0.1时，即使将学习率调节至128，整个模型的预测能力依然极差，准确率仅有10-12%左右，与随机猜测一个数字的准确率相当，查看输出文件，发现此时的模型倾向于将所有数字预测为同一个数字，表明学习率为0.1时，该模型基本没有提取出数字的形状信息。而当学习率降低至不大于0.01时，所有模型均能达到97.5%以上的预测准确率，即在不大于0.01的学习率下，该卷积神经网络方能提取出有效的形状信息。

然而，降低学习率并不总能带来预测准确率的提升，当学习率从0.01逐步降至0.0001时，在给定的三组分批大小下，准确率均先小幅上升，随后小幅降低。查看训练的输出文件发现，当学习率为0.0001时，其每次更新时误差函数的减小值，以及准确率的提升程度均不及学习率为0.001时的情形，即过低的学习率不利于模型参数的迅速收敛，进而导致准确率的小幅下降。

另一方面，观察准确率随分批大小的变化，可以发现，当学习率不小于0.01时，适当增大分批大小（如选为64或128），可以提高模型预测准确度，改善预测效果；然而，当学习率不大于0.001时，增加分批大小未必能改善模型的预测效果，反而有可能使预测准确度有所下降。

综合表8与表9的数据，笔者认为，学习率为0.001，分批大小为64，可以在验证集与测试集取得较好的表现，适用于下一阶段中手写数字的识别。

3.6 对手写数字的预测分类

笔者用学习率为0.001，分批大小为64的卷积神经网络模型，对手写的黑底白字与白底黑字进行预测，结果如图2所示。显然，左边除9以外，对数字0-9的预测基本准确，而前述的训练过程中常把9错认为4，因此这一错误情有可原；而右边仅正确预测1、2、5、7，其余数字均猜错，这是由于训练时没有采用白底黑字的样本，导致卷积神经网络错将黑字当作背景，而将白底视为数字，进而预测失误。

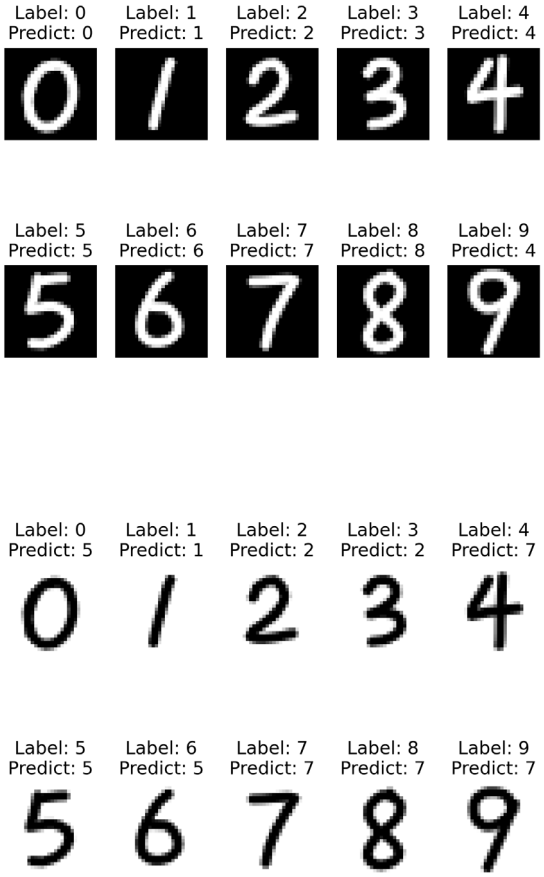

图2 两种背景的数字的预测结果。左：黑底白字；右：白底黑字

3.7 实验结论

（1）在默认条件下，采用上述卷积神经网络，验证集与测试集的预测准确率均达到了98%以上，且各数字的精确度与召回率也达到了97%以上，表明默认条件下，训练得到的卷积神经网络即有非常良好的表现；

（2）各数字中预测错误结果的分布具有一定倾向，其中，误判频率最高的几组数字，在结构上恰好有一定相似之处，如果书写者写字潦草，连人工识别都有可能出错，如4与9、3与5、0与6等；

（3）改变学习率与分批大小可在一定程度上改善模型的预测效果，其中，0.1的学习率会使训练后的模型难以提取出数字的形状特征，故不予采纳，而0.001的学习率则在验证集与测试集中均有良好表现，故学习率可定为0.001；分批大小可以改善训练模型，但改善程度与其大小未必成正比，需谨慎选取；

（4）学习率为0.001，分批大小为64，可以在验证集与测试集取得较好的表现；

（5）该模型可以识别笔者手写的黑底白字，而无法识别白底黑字，为使该模型具备识别白底黑字的能力，还需输入白底黑字的样本进行训练。