题目：MNIST 手写数字识别(用含一层隐含层的 DNN)

上课时间：2019 年 7 月 5 日—11 日

授课老师：梁克维

姓名：杨乾

学号：3170105905

专业：数学与应用数学

班级：数应 1702 班

MNIST手写数字识别(用含一层隐含层的DNN)

成员：3170105905 杨乾

August 28, 2019

题目：用含一层隐含层的DNN，做MNIST手写数字的识别

1.数据处理

MNIST数据集是由http://yann.lecun.com/exdb/mnist/网站提供的一个手写数字数据库, 包括含60000个示例图片的训练集和含10000个示例图片的测试集. 每一张图片都是0到9中的单个数字, 这些数字经过大小规格化处理, 集中在固定大小的图像中, 即均由28×28个像素构成.

网页中给出了如下四个文件：

train-images-idx3-ubyte.gz:	training set images (9912422 bytes)
train-labels-idx1-ubyte.gz:	training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz:	test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz:	test set labels (4542 bytes)

mnist数据集

其中第一个和第三个文件分别储存了60000张训练图片和10000张测试图片在$28 \times 28 = 784$ 个像素点上的灰度值, 第二个和第四个文件则分别储存了60000张训练图片和10000张测试图片所对应的标签.

在以二进制方式读入文件并去掉魔数后, 得到四个文件的格式分别为"47040000x1 double", "60000x1 double", "7840000x1 double"和"10000x1 double". 此处为方便标签与识别结果对比, 将标签形式分别更改成大小为10×60000和10×10000的矩阵, 其中(以0开头)标签为i的图片代表的列向量的第$i+1$ 个分量为1, 其余分量为0. 而将图片形式分别更改成大小为784×60000和784×10000 的矩阵, 则每张图片由一个含784个分量的列向量表示.

在对一些图片的数据reshape还原成大小为28×28的方阵后观察发现: 1)所有灰度值的取值均为区间$[-128, 127]$之间的整数. 2)绝对值较大的灰度值基本出现在数字的轮廓上. 3)数字笔迹未触及到的部分灰度值为0. 4)数字笔迹中间(即最深)部分灰度值为绝对值较小的负数.

结合以上信息, 判断数据读取过程中将无符号数以反码表示方式读入, 故通过将所有负值加上256以还原, 并将所有灰度值除以256从而将数据归一化.

2.网络设计

该神经网络包含一层输入层，一层隐含层以及一层输出层. 输入层节点数为像素点个数784, 隐含层(全连接层)节点数选为28, 输出层节点数设为10 (此处输出表示该网络将某张图片判断为各个数字的强度，最终以强度最大者作为识别的结果).

用正态随机函数得到隐含层和输出层中初始的权重w和偏置项b, 激活函数选取$Sigmoid$函数, 即$\sigma(x) = \frac{1}{1+e^{-x}}$. 误差反馈采用公式$\Delta y = (y^* - y)(1-y)(y-0)$, 并通过$w = w + a(\Delta y * x)$以及$b = b + a\Delta y$更新权重$w$ 和偏置b, 其中a为学习率, 此处设为0.05. 训练步数设为30.

3.结果呈现

利用隐含层和输出层中已训练好的权重w和偏置项b, 对测试集中的10000张图片进行测

试, 将神经网络判断第i张图片为$0\sim 9$十个数字的强度存放在一个大小为10×10000的矩阵的第i列(其中判断为“0”的放在第一行), 然后将该矩阵每列中最大值所在行数j提取出来, 并与标签最大值(即1)所在行数i进行比较.

比较完成之后，则在一个10×10的计数表格中的第i行第j列位置处加1, 于是最后每一行的和表示该行所代表的的数字在样本中的实际数量，每一列的和表示神经网络识别为该行所代表的数字的数量, 对角线上的数则表示相应数字识别正确的数量.最终将每个数字识别的准确率(对角线上数与其所在行的和的比)添加在表格右侧, 并计算出总的准确率.

4.代码实现

```
TrainImage = fopen('train-images.idx3-ubyte', 'rb');    %数据读入
A = fread(TrainImage, inf, 'int8');
A = A(end-60000*784+1:end);
A = reshape(A, 784, 60000);
TrainLabel = fopen('train-labels.idx1-ubyte', 'rb');
B0 = fread(TrainLabel, inf, 'int8');
B0 = B0(end-60000+1:end);
TestImage = fopen('t10k-images.idx3-ubyte', 'rb');
C = fread(TestImage, inf, 'int8');
C = C(end-10000*784+1:end);
C = reshape(C, 784, 10000);
TesstLabel = fopen('t10k-labels.idx1-ubyte', 'rb');
D0 = fread(TesstLabel, inf, 'int8');
D0 = D0(end-10000+1:end);
for i = 1:60000    %处理图片集数据
    for j = 1:784
        if A(j,i) < 0
            A(j,i) = A(j,i) + 256;
        end
        if i <= 10000
            if C(j,i) < 0
                C(j,i) = C(j,i) + 256;
            end
        end
    end
end
A = A/256;
C = C/256;
B = zeros(10, 60000);
D = zeros(10, 10000);
for i = 1:60000    %处理标签集数据
    B(B0(i)+1, i) = 1;
    if i <= 10000
        D(D0(i)+1, i) = 1;
    end
end
step = 30;    %设定参数
a = 0.05;
in = 784;
```

```
hid = 28;
out = 10;
out_w = randn(out, hid);    %初始化权重及偏置
out_b = randn(out, 1);
hid_w = randn(hid, in);
hid_b = randn(hid, 1);
for i = 1 : step    %训练神经网络
    r = randperm(60000);    %打乱图片顺序
    A = A(:, r);
    B = B(:, r);
    for j = 1 : 60000
        x = A(:, j);
        y = B(:, j);
        put0 = F(hid_w, hid_b, x);
        put1 = F(out_w, out_b, put0);
        deltaout = (y − put1). ∗ put1. ∗ (1 − put1);
        deltahid = ((out_w ′) ∗ deltaout). ∗ put0. ∗ (1 − put0);
        out_w = out_w + a ∗ (deltaout ∗ (put0 ′));
        out_b = out_b + a ∗ deltaout;
        hid_w = hid_w + a ∗ (deltahid ∗ (x ′));
        hid_b = hid_b + a ∗ deltahid;
    end
end
E = zeros(10, 10000);
for k = 1 : 10000    %测试神经网络
    x = C(:, k);
    put0 = F(hid_w, hid_b, x);
    E(:, k) = F(out_w, out_b, put0);
    [Res, Result] = max(E);
    [Ans, Answer] = max(D);
end
TABLE = zeros(12, 13);
for i = 1 : 10000    %存放识别结果
    TABLE(Answer(i) + 1, Result(i) + 1) = TABLE(Answer(i) + 1, Result(i) + 1) + 1;
end
for i = 2 : 11    %表格展示以及测试结果分析
    TABLE(i, 1) = i − 2;
    TABLE(1, i) = i − 2;
    TABLE(12, i) = sum(TABLE(2 : 11, i), 1);
    TABLE(i, 12) = sum(TABLE(i, 2 : 11), 2);
    TABLE(i, 13) = TABLE(i, i)/TABLE(i, 12);
end
TABLE(12, 12) = sum(TABLE(2 : 11, 12));
TABLE(12, 13) = sum(diag(TABLE(2 : 11, 2 : 11)))/TABLE(12, 12);
fprintf( ′准确率: %.2f%%\n ′, TABLE(12, 13) ∗ 100);
function [y] = F(w, b, x)    %输出函数
```

$y = w * x + b;$
$n = length(y);$
$for\ i = 1 : n$
$\quad y(i) = 1.0/(1 + exp(-y(i)));$
end
end

5.具体识别结果

由于隐含层和输出层中初始的权重w和偏置项b均是由正态随机函数得到的，故每次运行的结果以及准确率并不完全相同. 以下是10次运行得到的表格和准确率. 可发现整体识别准确率基本保持在94%以上，平均每次运行用时约2分35秒，属于较好地完成了手写数字的识别.

十个数字中识别准确率较高的数字为“1”和“0”，相对而言识别度较低的数字为“5”和“9”，其中“5”最易被识别成“3”或“6”，而“9”易被识别成“4”或“7”. 这一结果符合一般事实.

TABLE
12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	954	2	2	0	1	4	9	3	4	1	980	0.9735	
3	1	0	1119	2	4	0	2	2	2	4	0	1135	0.9859	
4	2	6	4	962	4	8	0	11	20	15	2	1032	0.9322	
5	3	2	1	26	937	1	17	1	9	13	3	1010	0.9277	
6	4	1	1	2	1	925	1	11	6	4	30	982	0.9420	
7	5	8	2	1	23	4	816	18	3	10	7	892	0.9148	
8	6	12	2	5	1	2	10	921	1	4	0	958	0.9614	
9	7	0	10	24	7	3	3	1	960	1	19	1028	0.9339	
10	8	4	2	5	17	9	9	6	9	904	9	974	0.9281	
11	9	5	1	2	16	21	8	0	18	8	930	1009	0.9217	
12	0	992	1144	1031	1010	974	870	980	1031	967	1001	10000	0.9428	
13														

TABLE
12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	962	0	0	2	2	3	7	2	2	0	980	0.9816	
3	1	0	1113	3	4	1	1	4	2	6	1	1135	0.9806	
4	2	9	1	962	13	10	2	5	13	15	2	1032	0.9322	
5	3	1	1	14	930	1	23	3	18	14	5	1010	0.9208	
6	4	2	0	3	0	930	1	10	0	3	33	982	0.9470	
7	5	10	1	3	11	3	829	16	3	9	7	892	0.9294	
8	6	10	3	6	1	9	13	913	0	3	0	958	0.9530	
9	7	1	5	22	5	5	1	2	968	2	17	1028	0.9416	
10	8	6	3	5	13	5	13	8	9	907	5	974	0.9312	
11	9	6	4	0	16	17	6	2	7	6	945	1009	0.9366	
12	0	1007	1131	1018	995	983	892	970	1022	967	1015	10000	0.9459	
13														

TABLE
12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	959	1	4	0	1	3	9	2	1	0	980	0.9786	
3	1	0	1117	2	2	1	1	4	3	4	1	1135	0.9841	
4	2	9	1	960	10	11	2	6	13	16	4	1032	0.9302	
5	3	4	3	19	942	2	11	3	12	10	4	1010	0.9327	
6	4	2	1	9	0	915	2	10	4	3	36	982	0.9318	
7	5	10	1	4	27	5	808	15	3	14	5	892	0.9058	
8	6	12	4	3	3	7	10	914	1	4	0	958	0.9541	
9	7	1	7	12	12	5	1	0	971	5	14	1028	0.9446	
10	8	2	2	7	19	10	11	9	9	904	1	974	0.9281	
11	9	7	3	0	9	17	10	3	22	12	926	1009	0.9177	
12	0	1006	1140	1020	1024	974	859	973	1040	973	991	10000	0.9416	
13														

TABLE
12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	961	0	1	1	3	3	4	1	3	3	980	0.9806	
3	1	0	1113	3	3	1	2	1	2	10	0	1135	0.9806	
4	2	15	0	950	19	10	3	6	9	14	6	1032	0.9205	
5	3	0	0	14	954	2	17	1	10	8	4	1010	0.9446	
6	4	2	1	3	0	926	2	8	2	5	33	982	0.9430	
7	5	13	3	8	24	3	811	4	5	19	2	892	0.9092	
8	6	7	3	7	1	8	11	917	0	4	0	958	0.9572	
9	7	3	6	19	6	8	0	1	970	3	12	1028	0.9436	
10	8	8	2	17	12	6	9	10	5	896	9	974	0.9199	
11	9	6	5	1	11	23	16	3	13	3	928	1009	0.9197	
12	0	1015	1133	1023	1031	990	874	955	1017	965	997	10000	0.9426	
13														

神经网络识别结果分布表(1)~(4)

TABLE

12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	962	1	0	0	0	2	9	3	3	0	980	0.9816	
3	1	0	1111	5	2	0	2	6	2	7	0	1135	0.9789	
4	2	9	1	953	13	11	1	9	10	20	5	1032	0.9234	
5	3	4	1	18	935	0	21	2	11	17	1	1010	0.9257	
6	4	2	1	5	0	921	4	11	4	6	28	982	0.9379	
7	5	8	0	4	21	3	821	11	4	15	5	892	0.9204	
8	6	11	3	2	0	9	20	903	3	7	0	958	0.9426	
9	7	3	9	13	10	4	1	0	968	3	17	1028	0.9416	
10	8	4	3	4	10	4	10	9	7	921	2	974	0.9456	
11	9	6	7	2	6	30	8	0	12	20	918	1009	0.9098	
12	0	1009	1137	1006	997	982	890	960	1024	1019	976	10000	0.9413	
13														

TABLE

12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	959	0	2	2	2	6	5	1	2	1	980	0.9786	
3	1	0	1120	3	2	0	1	2	3	3	1	1135	0.9868	
4	2	10	5	955	8	10	2	11	13	18	0	1032	0.9254	
5	3	3	1	12	946	2	15	1	12	16	2	1010	0.9366	
6	4	2	0	2	0	938	0	13	3	2	22	982	0.9552	
7	5	8	2	4	21	3	811	15	7	13	8	892	0.9092	
8	6	14	2	7	1	4	11	913	1	5	0	958	0.9530	
9	7	2	10	22	2	6	1	0	970	3	12	1028	0.9436	
10	8	5	5	10	11	6	9	11	9	906	2	974	0.9302	
11	9	12	7	0	7	27	9	1	16	16	914	1009	0.9058	
12	0	1015	1152	1017	1000	998	865	972	1035	984	962	10000	0.9432	
13														

TABLE

12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	951	0	2	0	2	6	9	2	5	3	980	0.9704	
3	1	0	1113	6	2	1	2	4	1	6	0	1135	0.9806	
4	2	5	2	947	19	12	5	8	13	20	1	1032	0.9176	
5	3	2	0	15	940	3	20	2	9	16	3	1010	0.9307	
6	4	1	1	2	1	937	0	12	2	3	23	982	0.9542	
7	5	11	0	2	27	5	799	16	4	23	5	892	0.8957	
8	6	9	3	6	1	14	7	907	1	9	1	958	0.9468	
9	7	1	9	20	2	7	1	1	970	4	13	1028	0.9436	
10	8	5	1	7	14	10	14	12	5	902	4	974	0.9261	
11	9	9	6	2	10	42	9	4	10	12	905	1009	0.8969	
12	0	994	1135	1009	1016	1033	863	975	1017	1000	958	10000	0.9371	
13														

TABLE

12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	962	0	1	3	1	3	7	2	1	0	980	0.9816	
3	1	0	1110	5	3	1	2	5	4	5	0	1135	0.9780	
4	2	16	1	965	14	7	4	8	10	6	1	1032	0.9351	
5	3	2	2	18	938	1	19	1	6	16	7	1010	0.9287	
6	4	1	1	4	3	923	1	10	3	2	34	982	0.9399	
7	5	11	1	5	29	4	809	7	2	16	8	892	0.9070	
8	6	11	3	4	1	9	10	911	1	8	0	958	0.9509	
9	7	1	8	13	3	5	1	0	974	3	20	1028	0.9475	
10	8	10	4	7	12	9	9	7	10	901	5	974	0.9251	
11	9	11	4	1	8	24	12	1	8	4	936	1009	0.9277	
12	0	1025	1134	1023	1014	984	870	957	1020	962	1011	10000	0.9429	
13														

TABLE

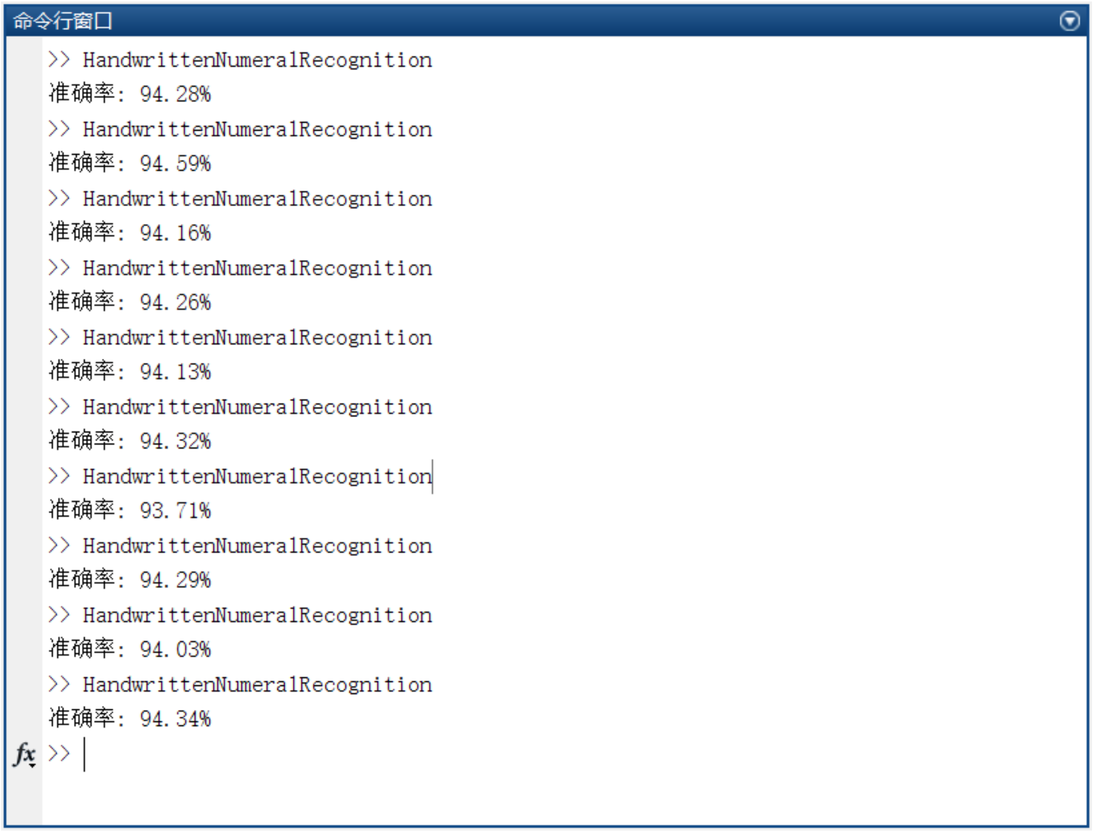
12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	956	1	2	1	1	5	9	2	3	0	980	0.9755	
3	1	0	1110	7	4	0	1	3	1	9	0	1135	0.9780	
4	2	12	3	969	7	10	1	3	9	15	3	1032	0.9390	
5	3	4	1	19	942	0	12	2	10	16	4	1010	0.9327	
6	4	1	1	4	0	918	1	11	6	3	37	982	0.9348	
7	5	10	1	4	31	5	792	12	3	25	9	892	0.8879	
8	6	11	3	5	1	6	8	921	0	3	0	958	0.9614	
9	7	0	13	22	9	4	0	1	966	1	12	1028	0.9397	
10	8	8	5	7	22	5	18	8	7	891	3	974	0.9148	
11	9	12	9	1	13	14	2	0	12	8	938	1009	0.9296	
12	0	1014	1147	1040	1030	963	840	970	1016	974	1006	10000	0.9403	
13														

12x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	2	3	4	5	6	7	8	9	0	0	
2	0	962	0	2	2	0	4	6	2	2	0	980	0.9816	
3	1	0	1116	2	3	0	1	5	3	5	0	1135	0.9833	
4	2	11	5	954	14	13	1	6	11	17	0	1032	0.9244	
5	3	1	1	21	935	1	24	0	12	15	0	1010	0.9257	
6	4	1	1	7	1	930	1	8	5	5	23	982	0.9470	
7	5	6	3	5	21	3	833	8	1	6	6	892	0.9339	
8	6	9	3	6	1	3	13	915	2	6	0	958	0.9551	
9	7	1	9	23	3	4	1	0	965	2	20	1028	0.9387	
10	8	5	4	6	17	8	13	9	7	901	4	974	0.9251	
11	9	8	5	2	15	24	4	3	20	5	923	1009	0.9148	
12	0	1004	1147	1028	1012	986	895	960	1028	964	976	10000	0.9434	
13														

神经网络识别结果分布表(5)~(10)



```
命令行窗口
>> HandwrittenNumeralRecognition
准确率: 94.28%
>> HandwrittenNumeralRecognition
准确率: 94.59%
>> HandwrittenNumeralRecognition
准确率: 94.16%
>> HandwrittenNumeralRecognition
准确率: 94.26%
>> HandwrittenNumeralRecognition
准确率: 94.13%
>> HandwrittenNumeralRecognition
准确率: 94.32%
>> HandwrittenNumeralRecognition
准确率: 93.71%
>> HandwrittenNumeralRecognition
准确率: 94.29%
>> HandwrittenNumeralRecognition
准确率: 94.03%
>> HandwrittenNumeralRecognition
准确率: 94.34%
>>
```

神经网络整体识别准确率