

代码:

```

1 load mnist_uint8;
2
3 train_x = double(reshape(train_x', 28, 28, 60000)) / 255;
4 test_x = double(reshape(test_x', 28, 28, 10000)) / 255;
5 train_y = double(train_y');
6 test_y = double(test_y');
7
8 opts.alpha = 1;
9 opts.batchsize = 50;
10 opts.numepochs = 1;
11
12 % 定义不同的卷积核个数配置, 使用元组 (第一个卷积层卷积核个数, 第二个卷积层卷积核个数)
13 conv_kernels = [4, 8; 6, 12; 8, 16; 10, 20; 12, 24; 14, 28; 16, 32];
14 test_errors = zeros(size(conv_kernels, 1), 1); % 存储每种配置下的测试错误率
15
16 for i = 1:size(conv_kernels, 1)
17     num_kernels_1 = conv_kernels(i, 1);
18     num_kernels_2 = conv_kernels(i, 2);
19
20     rand('state', 0)
21     cnn.layers = {
22         struct('type', 'i') % 输入层
23         struct('type', 'c', 'outputmaps', num_kernels_1, 'kernelsize', 5) %
第一个卷积层
24         struct('type', 's', 'scale', 2) % 第一个子采样层
25         struct('type', 'c', 'outputmaps', num_kernels_2, 'kernelsize', 5) %
第二个卷积层
26         struct('type', 's', 'scale', 2) % 第二个子采样层
27     };
28     cnn = cnnsetup(cnn, train_x, train_y);
29
30     cnn = cnnttrain(cnn, train_x, train_y, opts);
31
32     [er, ~] = cnntest(cnn, test_x, test_y);
33     test_errors(i) = er;
34 end
35
36 % 绘制卷积核个数与测试错误率的关系图
37 figure;
38 plot(conv_kernels(:, 1), test_errors, '-o');
39 xlabel('Number of Kernels in First Conv Layer');
40 ylabel('Test Error Rate');
41 title('Test Error Rate vs Number of Kernels in First Conv Layer');
42 grid on;
43

```

```

44 disp('Test Errors for different number of kernels:');
45 disp(table(conv_kernels(:, 1), conv_kernels(:, 2), test_errors,
    'VariableNames', {'Conv1 Kernels', 'Conv2 Kernels', 'Test Error Rate'}));

```

结果:

命令窗口输出:

```

历时 134.243863 秒。
epoch 1/1
历时 189.792663 秒。
epoch 1/1
历时 247.346603 秒。
epoch 1/1
历时 311.644438 秒。
Test Errors for different number of kernels:
    Conv1 Kernels    Conv2 Kernels    Test Error Rate
    _____    _____    _____
            4            8            0.1172
            6           12            0.1113
            8           16            0.1116
           10           20             0.11
           12           24            0.1261
           14           28            0.1281
           16           32            0.8968

```

工作区变量:

名称	值
cnn	1x1 struct
conv_kernels	7x2 double
er	0.8968
i	7
num_kernels_1	16
num_kernels_2	32
opts	1x1 struct
test_errors	[0.1172;0.1113;0.11...
test_x	28x28x10000 double
test_y	10x10000 double
train_x	28x28x60000 double
train_y	10x60000 double

2

代码:

```

1 load mnist_uint8;
2
3 train_x = double(reshape(train_x', 28, 28, 60000)) / 255;
4 test_x = double(reshape(test_x', 28, 28, 10000)) / 255;
5 train_y = double(train_y');
6 test_y = double(test_y');
7
8 opts.alpha = 1;
9 opts.numepochs = 1;
10
11 % 定义不同的批量大小
12 batch_sizes = [10, 20, 50, 100, 200, 500];
13 test_errors = zeros(length(batch_sizes), 1); % 存储每种配置下的测试错误率
14
15 for i = 1:length(batch_sizes)
16     batch_size = batch_sizes(i);
17     opts.batchsize = batch_size;
18

```

```

19  rand('state', 0)
20  cnn.layers = {
21      struct('type', 'i') % 输入层
22      struct('type', 'c', 'outputmaps', 6, 'kernelsize', 5) % 第一个卷积层
23      struct('type', 's', 'scale', 2) % 第一个子采样层
24      struct('type', 'c', 'outputmaps', 12, 'kernelsize', 5) % 第二个卷积层
25      struct('type', 's', 'scale', 2) % 第二个子采样层
26  };
27  cnn = cnnsetup(cnn, train_x, train_y);
28
29  cnn = cnnttrain(cnn, train_x, train_y, opts);
30
31  [er, ~] = cnntest(cnn, test_x, test_y);
32  test_errors(i) = er;
33  end
34
35  % 绘制批量大小与测试错误率的关系图
36  figure;
37  plot(batch_sizes, test_errors, '-o');
38  xlabel('Batch Size');
39  ylabel('Test Error Rate');
40  title('Test Error Rate vs Batch Size');
41  grid on;
42
43  disp('Test Errors for different batch sizes:');
44  disp(table(batch_sizes, test_errors, 'VariableNames', {'Batch Size', 'Test
Error Rate'}));

```

结果:

The screenshot shows the MATLAB R2023b interface. The Command Window displays the execution of the script, including training and testing times for different batch sizes. The Workspace window shows the variables defined in the script.

Command Window Output:

```

epoch 1/1
历时 55.715522 秒。
epoch 1/1
历时 54.517788 秒。
epoch 1/1
历时 55.877546 秒。
epoch 1/1
历时 45.676214 秒。
Test Errors for different batch sizes:
    Batch Size    Test Error Rate
    _____    _____
         10         0.0486
         20         0.0693
         50         0.1113
        100         0.1761
        200         0.6316
        500         0.8863
fx >>

```

Workspace Variables:

名称	值
batch_size	500
batch_sizes	[10,20,50,100,200,500]
cnn	1x1 struct
conv_kernels	7x2 double
er	0.8863
i	6
num_kernels_1	16
num_kernels_2	32
opts	1x1 struct
test_errors	[0.0486;0.0693;0.1113;0.1761;0.6316;0.8863]
test_x	28x28x10000 double
test_y	10x10000 double
train_x	28x28x60000 double
train_y	10x60000 double

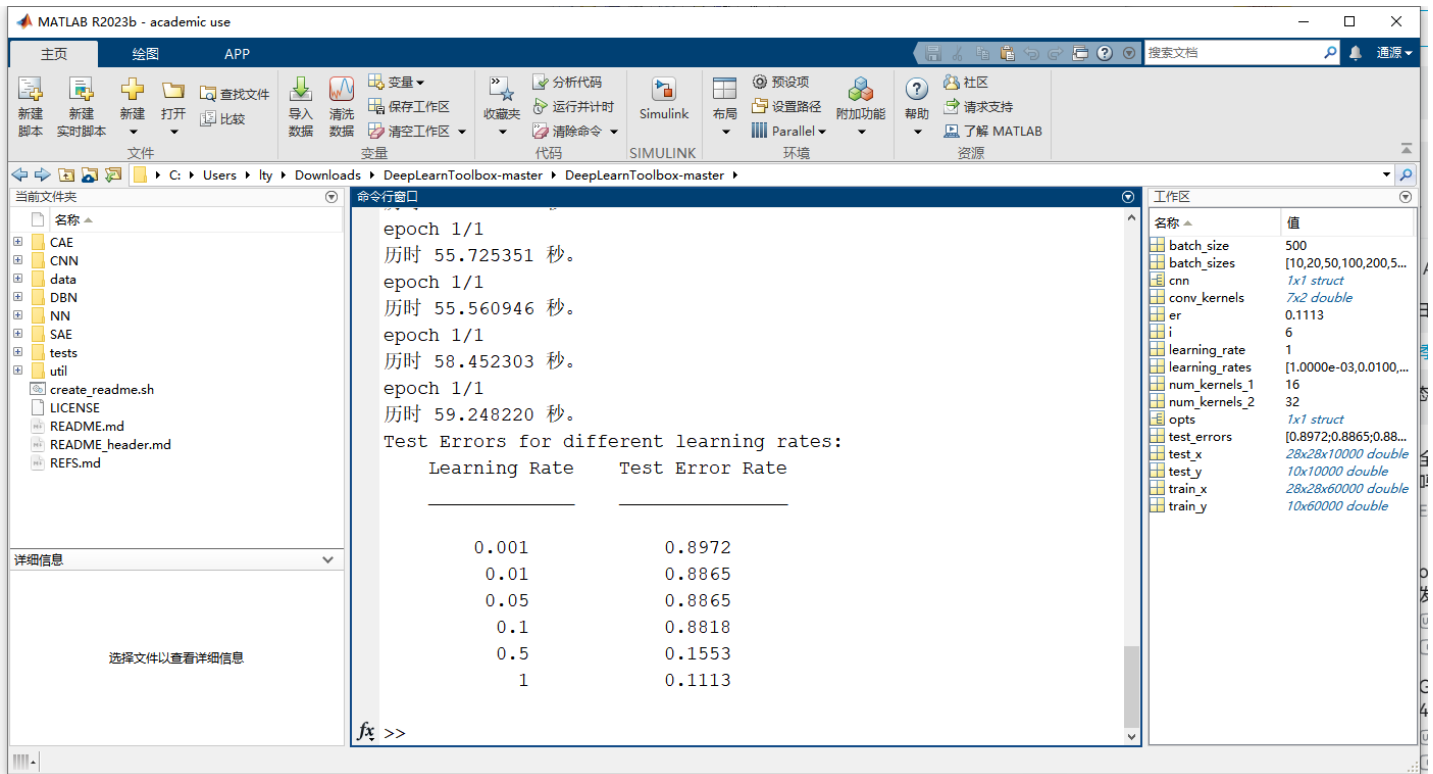
代码:

```

1 load mnist_uint8;
2
3 train_x = double(reshape(train_x', 28, 28, 60000)) / 255;
4 test_x = double(reshape(test_x', 28, 28, 10000)) / 255;
5 train_y = double(train_y');
6 test_y = double(test_y');
7
8 opts.batchsize = 50;
9 opts.numepochs = 1;
10
11 % 定义不同的学习率
12 learning_rates = [0.001, 0.01, 0.05, 0.1, 0.5, 1];
13 test_errors = zeros(length(learning_rates), 1); % 存储每种配置下的测试错误率
14
15 for i = 1:length(learning_rates)
16     learning_rate = learning_rates(i);
17     opts.alpha = learning_rate;
18
19     rand('state', 0)
20     cnn.layers = {
21         struct('type', 'i') % 输入层
22         struct('type', 'c', 'outputmaps', 6, 'kernelsize', 5) % 第一个卷积层
23         struct('type', 's', 'scale', 2) % 第一个子采样层
24         struct('type', 'c', 'outputmaps', 12, 'kernelsize', 5) % 第二个卷积层
25         struct('type', 's', 'scale', 2) % 第二个子采样层
26     };
27     cnn = cnnsetup(cnn, train_x, train_y);
28
29     cnn = cnnttrain(cnn, train_x, train_y, opts);
30
31     [er, ~] = cnntest(cnn, test_x, test_y);
32     test_errors(i) = er;
33 end
34
35 % 绘制学习率与测试错误率的关系图
36 figure;
37 plot(learning_rates, test_errors, '-o');
38 xlabel('Learning Rate');
39 ylabel('Test Error Rate');
40 title('Test Error Rate vs Learning Rate');
41 grid on;
42
43 disp('Test Errors for different learning rates:');
44 disp(table(learning_rates, test_errors, 'VariableNames', {'Learning Rate',
45     'Test Error Rate'}));

```

结果:



4

代码:

```
1 load mnist_uint8;
2
3 train_x = double(reshape(train_x', 28, 28, 60000)) / 255;
4 test_x = double(reshape(test_x', 28, 28, 10000)) / 255;
5 train_y = double(train_y');
6 test_y = double(test_y');
7
8 opts.alpha = 1;
9 opts.batchsize = 50;
10
11 % 定义不同的训练轮数
12 num_epochs = [1, 3, 5, 7];
13 test_errors = zeros(length(num_epochs), 1); % 存储每种配置下的测试错误率
14
15 for i = 1:length(num_epochs)
16     opts.numepochs = num_epochs(i);
17
18     rand('state', 0)
19     cnn.layers = {
20         struct('type', 'i') % 输入层
21         struct('type', 'c', 'outputmaps', 6, 'kernelsize', 5) % 第一个卷积层
22         struct('type', 's', 'scale', 2) % 第一个子采样层
23         struct('type', 'c', 'outputmaps', 12, 'kernelsize', 5) % 第二个卷积层
```

```

24     struct('type', 's', 'scale', 2) % 第二个子采样层
25 };
26 cnn = cnnsetup(cnn, train_x, train_y);
27
28 cnn = cnnttrain(cnn, train_x, train_y, opts);
29
30 [er, ~] = cntest(cnn, test_x, test_y);
31 test_errors(i) = er;
32 end
33
34 % 绘制训练轮数与测试错误率的关系图
35 figure;
36 plot(num_epochs, test_errors, '-o');
37 xlabel('Number of Epochs');
38 ylabel('Test Error Rate');
39 title('Test Error Rate vs Number of Epochs');
40 grid on;
41
42 disp('Test Errors for different number of epochs:');
43 disp(table(num_epochs', test_errors, 'VariableNames', {'Number of Epochs',
    'Test Error Rate'}));
44

```

结果:

The screenshot shows the MATLAB R2023b interface. The Command Window displays the execution of a script, including epoch durations and a table of test error rates. The Workspace window shows the values of variables defined in the script.

Command Window Output:

```

epoch 3/7
历时 62.258748 秒。
epoch 4/7
历时 63.976844 秒。
epoch 5/7
历时 63.373779 秒。
epoch 6/7
历时 62.848447 秒。
epoch 7/7
历时 59.953390 秒。
Test Errors for different number of epochs:
Number of Epochs    Test Error Rate
-----
1                    0.1113
3                    0.061
5                    0.0456
7                    0.0341
fx >>

```

Workspace Variables:

名称	值
batch_size	500
batch_sizes	[10,20,50,100,200,5...
cnn	1x1 struct
conv_kernels	7x2 double
er	0.0341
i	4
learning_rate	1
learning_rates	[1.0000e-03,0.0100,...
num_epochs	[1,3,5,7]
num_kernels_1	16
num_kernels_2	32
opts	1x1 struct
test_errors	[0.1113;0.0610;0.04...
test_x	28x28x10000 double
test_y	10x10000 double
train_x	28x28x60000 double
train_y	10x60000 double