

作业 1

高艺轩

2024 年 3 月 19 日

理论部分

1 单选题 (15 分)

1.1 B

1.2 A

1.3 B

1.4 A

1.5 B

2 计算题 (15 分)

2.1 设隐含层为 $\mathbf{z} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$, 其中 $\mathbf{x} \in R^{(m \times 1)}$, $\mathbf{z} \in R^{(n \times 1)}$, $\mathbf{W} \in R^{(m \times n)}$, $\mathbf{b} \in R^{(n \times 1)}$ 均为已知, 其激活函数如下:

$$\mathbf{y} = \delta(\mathbf{z}) = \tanh(\mathbf{z})$$

\tanh 表示双曲正切函数。若训练过程中的目标函数为 L , 且已知 L 对 \mathbf{y} 的导数 $\frac{\partial L}{\partial \mathbf{y}} = [\frac{\partial L}{\partial y_1}, \frac{\partial L}{\partial y_2}, \dots, \frac{\partial L}{\partial y_n}]^T$ 和 $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ 的值。

2.1.1 请使用 \mathbf{y} 表示出 $\frac{\partial \mathbf{y}^T}{\partial \mathbf{z}}$, 这里的 \mathbf{y}^T 为行向量。

解. 首先, 对 $i \neq j$, $\frac{\partial y_i}{\partial z_j} = 0$ 。

同时 $y_i = \tanh(z_i) = \tanh(\arctanh(y_i))$, 因此

$$\frac{\partial y_i}{\partial z_i} = 1 - \tanh^2(z_i) = 1 - y_i^2$$

因此

$$\frac{\partial \mathbf{y}^T}{\partial \mathbf{z}} = \text{diag}\{1 - y_1^2, 1 - y_2^2, \dots, 1 - y_n^2\}$$

□

2.1.2 请使用 \mathbf{y} 和 $\frac{\partial L}{\partial \mathbf{y}}$ 表示 $\frac{\partial L}{\partial \mathbf{x}}$, $\frac{\partial L}{\partial \mathbf{W}}$, $\frac{\partial L}{\partial \mathbf{b}}$ 。

提示: $\frac{\partial L}{\partial \mathbf{x}}$, $\frac{\partial L}{\partial \mathbf{W}}$, $\frac{\partial L}{\partial \mathbf{b}}$ 与 $\mathbf{x}, \mathbf{W}, \mathbf{b}$ 具有相同维度。

解. 由链式法则

$$\frac{\partial L}{\partial \mathbf{x}} = \frac{\partial \mathbf{z}^T}{\partial \mathbf{x}} \frac{\partial \mathbf{y}^T}{\partial \mathbf{z}} \frac{\partial L}{\partial \mathbf{y}} = \mathbf{W} \text{diag}\{1 - y_1^2, 1 - y_2^2, \dots, 1 - y_n^2\} \frac{\partial L}{\partial \mathbf{y}}$$

对于 $\frac{\partial L}{\partial \mathbf{W}}$,

$$\frac{\partial \mathbf{z}^T}{\partial \mathbf{W}} = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \cdots & \mathbf{x} \end{bmatrix}_{m \times n}$$

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{W}} &= \frac{\partial \mathbf{z}^T}{\partial \mathbf{W}} \frac{\partial \mathbf{y}^T}{\partial \mathbf{z}} \frac{\partial L}{\partial \mathbf{y}} \\ &= \begin{bmatrix} \mathbf{x} & \mathbf{x} & \cdots & \mathbf{x} \end{bmatrix}_{m \times n} \text{diag}\{1 - y_1^2, 1 - y_2^2, \dots, 1 - y_n^2\} \frac{\partial L}{\partial \mathbf{y}} \end{aligned}$$

对于 $\frac{\partial L}{\partial \mathbf{b}}$, 由链式法则

$$\frac{\partial L}{\partial \mathbf{b}} = \frac{\partial \mathbf{z}^T}{\partial \mathbf{b}} \frac{\partial \mathbf{y}^T}{\partial \mathbf{z}} \frac{\partial L}{\partial \mathbf{y}} = \mathbf{I}_n \frac{\partial \mathbf{y}^T}{\partial \mathbf{z}} \frac{\partial L}{\partial \mathbf{y}} = \text{diag}\{1 - y_1^2, 1 - y_2^2, \dots, 1 - y_n^2\} \frac{\partial L}{\partial \mathbf{y}} \quad \square$$

编程部分

3 编程作业报告

完成后的代码也可以在

<https://git.unlockableworld.com/unlockable/MediaNCognition>中找到。

(1) 使用默认配置进行训练和测试。

1) 训练模型。

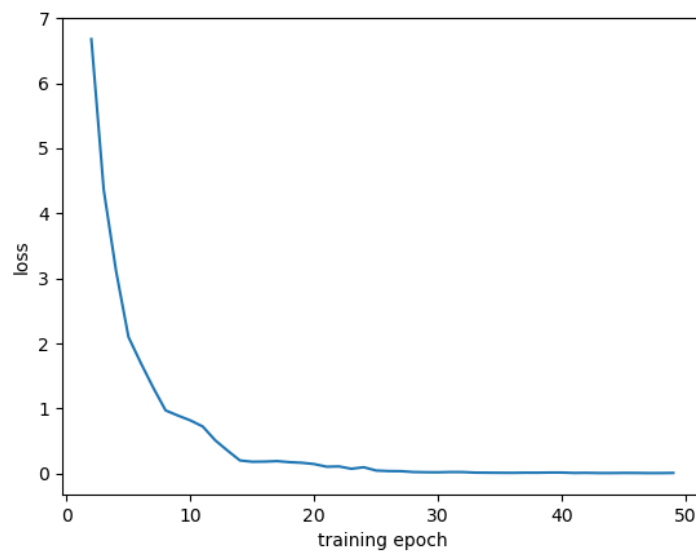
输入:

```
1 | python recognition.py --mode train --model_path ./
   | saved_models/default.pth
```

输出:

```
1 Epoch 01: loss = inf
2 Epoch 02: loss = inf
3 Epoch 03: loss = 6.678
4 Epoch 04: loss = 4.361
5 Epoch 05: loss = 3.110
6 Epoch 06: loss = 2.099
7 Epoch 07: loss = 1.698
8 Epoch 08: loss = 1.320
9 Epoch 09: loss = 0.970
10 Epoch 10: loss = 0.891
11 Epoch 10: validation accuracy = 66.0%
12 Epoch 11: loss = 0.817
13 Epoch 12: loss = 0.723
14 Epoch 13: loss = 0.512
15 Epoch 14: loss = 0.353
16 Epoch 15: loss = 0.202
17 Epoch 16: loss = 0.182
18 Epoch 17: loss = 0.184
19 Epoch 18: loss = 0.191
20 Epoch 19: loss = 0.175
21 Epoch 20: loss = 0.166
22 Epoch 20: validation accuracy = 68.0%
23 Epoch 21: loss = 0.146
24 Epoch 22: loss = 0.105
25 Epoch 23: loss = 0.109
26 Epoch 24: loss = 0.074
27 Epoch 25: loss = 0.097
28 Epoch 26: loss = 0.047
29 Epoch 27: loss = 0.038
30 Epoch 28: loss = 0.037
31 Epoch 29: loss = 0.024
32 Epoch 30: loss = 0.021
33 Epoch 30: validation accuracy = 68.8%
34 Epoch 31: loss = 0.019
35 Epoch 32: loss = 0.024
36 Epoch 33: loss = 0.023
37 Epoch 34: loss = 0.014
38 Epoch 35: loss = 0.013
39 Epoch 36: loss = 0.012
40 Epoch 37: loss = 0.011
41 Epoch 38: loss = 0.013
42 Epoch 39: loss = 0.013
```

```
43 | Epoch 40: loss = 0.016
44 | Epoch 40: validation accuracy = 70.5%
45 | Epoch 41: loss = 0.015
46 | Epoch 42: loss = 0.009
47 | Epoch 43: loss = 0.011
48 | Epoch 44: loss = 0.008
49 | Epoch 45: loss = 0.008
50 | Epoch 46: loss = 0.010
51 | Epoch 47: loss = 0.009
52 | Epoch 48: loss = 0.007
53 | Epoch 49: loss = 0.007
54 | Epoch 50: loss = 0.010
55 | Epoch 50: validation accuracy = 70.5%
56 | Model saved in ./saved_models/default.pth
```



2) 测试模型。

输入：

```
1 | python recognition.py --mode test --model_path .\
   | saved_models\default.pth
```

输出：

```
1 | [Info] Load model from .\saved_models\default.pth
2 | [Info] Test accuracy = 72.0%
```

(2) 调整参数、使用 Adam 优化器训练并测试。

1) 训练模型。

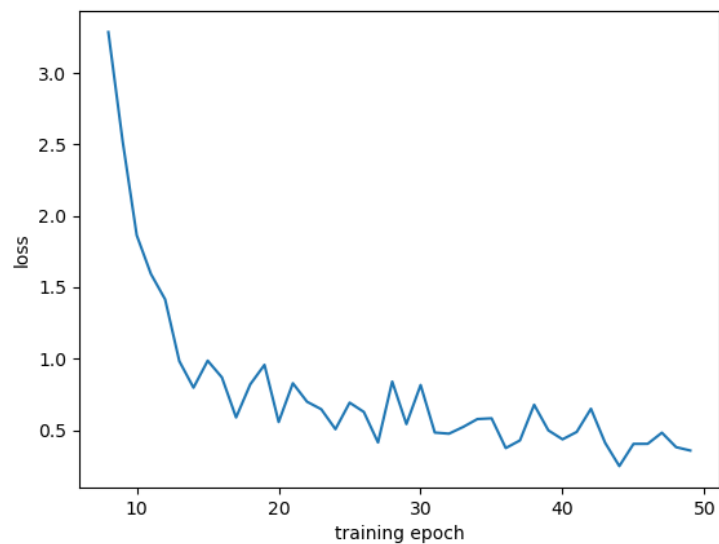
输入：

```
1 python recognition.py --mode train --hsize 64 --lr 2e
  -3 --optim_type adam --momentum 0 --weight_decay
  0.1 --model_path .\saved_models\adam_optim.pth
```

输出：

```
1 Epoch 01: loss = inf
2 Epoch 02: loss = inf
3 Epoch 03: loss = inf
4 Epoch 04: loss = inf
5 Epoch 05: loss = inf
6 Epoch 06: loss = inf
7 Epoch 07: loss = inf
8 Epoch 08: loss = inf
9 Epoch 09: loss = 3.286
10 Epoch 10: loss = 2.528
11 Epoch 10: validation accuracy = 57.5%
12 Epoch 11: loss = 1.864
13 Epoch 12: loss = 1.593
14 Epoch 13: loss = 1.416
15 Epoch 14: loss = 0.983
16 Epoch 15: loss = 0.798
17 Epoch 16: loss = 0.986
18 Epoch 17: loss = 0.870
19 Epoch 18: loss = 0.589
20 Epoch 19: loss = 0.821
21 Epoch 20: loss = 0.957
22 Epoch 20: validation accuracy = 66.2%
23 Epoch 21: loss = 0.557
24 Epoch 22: loss = 0.829
25 Epoch 23: loss = 0.699
26 Epoch 24: loss = 0.647
27 Epoch 25: loss = 0.507
28 Epoch 26: loss = 0.693
29 Epoch 27: loss = 0.628
30 Epoch 28: loss = 0.414
31 Epoch 29: loss = 0.840
32 Epoch 30: loss = 0.543
33 Epoch 30: validation accuracy = 66.8%
```

```
34 | Epoch 31: loss = 0.816
35 | Epoch 32: loss = 0.483
36 | Epoch 33: loss = 0.476
37 | Epoch 34: loss = 0.523
38 | Epoch 35: loss = 0.578
39 | Epoch 36: loss = 0.583
40 | Epoch 37: loss = 0.375
41 | Epoch 38: loss = 0.428
42 | Epoch 39: loss = 0.678
43 | Epoch 40: loss = 0.499
44 | Epoch 40: validation accuracy = 72.2%
45 | Epoch 41: loss = 0.436
46 | Epoch 42: loss = 0.488
47 | Epoch 43: loss = 0.650
48 | Epoch 44: loss = 0.412
49 | Epoch 45: loss = 0.249
50 | Epoch 46: loss = 0.404
51 | Epoch 47: loss = 0.405
52 | Epoch 48: loss = 0.482
53 | Epoch 49: loss = 0.381
54 | Epoch 50: loss = 0.358
55 | Epoch 50: validation accuracy = 79.5%
56 | Model saved in .\saved_models\adam_optim.pth
```



2) 测试性能。

输入：

```
1 python recognition.py --mode test --hsize 64 --  
  model_path .\saved_models\adam_optim.pth
```

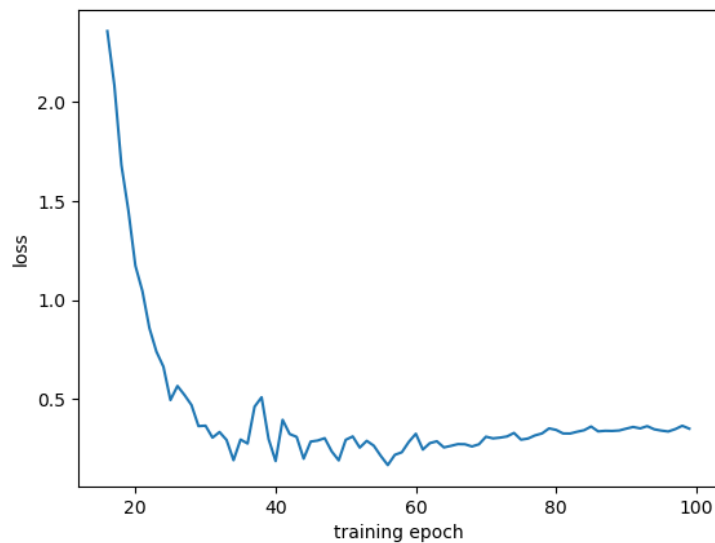
输出：

```
1 [Info] Load model from .\saved_models\adam_optim.pth  
2 [Info] Test accuracy = 85.0%
```

(3) 使用效果最佳的模型测试。经过简单的尝试，发现使用

```
1 python recognition.py --mode train --hsize 64 --lr 1e-3 --  
  optim_type adam --momentum 10 --weight_decay 0.1 --  
  epoch 100 --model_path .\saved_models\adam_optim_lr1e-3  
  _epoch100_momentum10.pth
```

可以使测试集准确率达到 88.8%，有略微的提升。训练的 loss 曲线：



使用它进行预测：



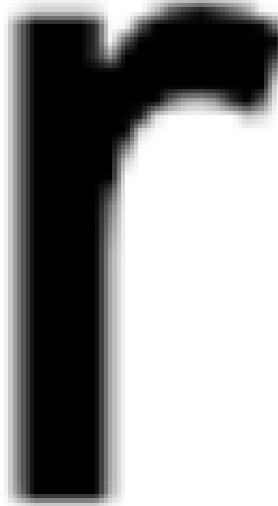
(a) 预测: A



(b) 预测: B



(c) 预测: M



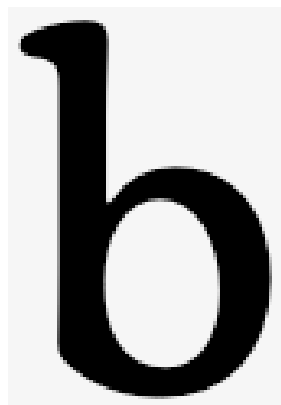
(d) 预测: R



(e) 预测: M



(f) 预测: O



(g) 预测: B



(h) 预测: W

(4) 遇到的问题及解决方法

- 1) 代码中对灰度图像的矩阵进行标准化时，`numpy`显示不能对`NumpyGenericArray`进行对`float`的`/`操作。改用`np.div()`解决了这个问题。
 - 2) 在利用训练好的模型进行预测时，发现自己找到的大部分模型都预测错误；最后与训练集的图片进行了对比，发现主要问题是裁切字母时留下了过大的边距，导致模型不能正确理解输入。重新裁剪边框后，得到正确的结果。
- (5) 建议：希望下次发布作业代码可以利用清华的 `git`。