清华大学电子工程系

媒体与认知 课堂 2

2023-2024 学年春季学期

作业 1

毕嘉仪 2024 年 3 月 21 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 y 的导数 $\frac{\partial L}{\partial \mathbf{y}} = [\frac{\partial L}{\partial y_1}, \frac{\partial L}{\partial y_2}, ..., \frac{\partial L}{\partial y_n}]^T$ 和 $\mathbf{y} = [y_1, y_2, ..., y_n]^T$ 的值。

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

解.

$$\frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}}_{n\times n} = \begin{bmatrix} \frac{\partial y_1}{\partial z_1} & \cdots & \frac{\partial y_n}{\partial z_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial z_n} & \cdots & \frac{\partial y_n}{\partial z_n} \end{bmatrix}$$

当 $i \neq j$, 易知 $\frac{\partial y_i}{\partial z_i} = 0$

当 $i = j$,

$$\tanh' z_i = 1 - \tanh^2 z_i$$

$$z_i = \operatorname{arctanh} y_i$$

$$\therefore \tanh' z_i = 1 - y_i^2$$

$$\therefore \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}}_{n\times n} = \operatorname{diag}(1-y_i^2) = \begin{bmatrix} 1-y_1^2 & & & \\ & 1-y_2^2 & & \\ & & \ddots & \\ & & & 1-y_n^2 \end{bmatrix}$$

□

2.1.2 请使用 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 x,W,b 具有相同维度。

解. 对 $\frac{\partial L}{\partial \boldsymbol{x}}$，由链式法则：

$$\frac{\partial L}{\partial \boldsymbol{x}}_{m\times 1} = \frac{\partial \boldsymbol{z}^{\mathrm{T}}}{\partial \boldsymbol{x}}_{m\times n} \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}}_{n\times n} \frac{\partial L}{\partial \boldsymbol{y}}_{n\times 1} = W\dot{\operatorname{diag}}(1-y_i^2)\frac{\partial L}{\partial \boldsymbol{y}}$$

对于 $\frac{\partial L}{\partial W}$，先计算

$$\frac{\partial \boldsymbol{z}_i}{\partial W}_{m\times n} = \begin{bmatrix} \frac{\partial z_i}{\partial W_{11}} & \cdots & \frac{\partial z_i}{\partial W_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_i}{\partial W_{m1}} & \cdots & \frac{\partial z_i}{\partial W_{mn}} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & \frac{\partial z_i}{\partial W_{1i}} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & \frac{\partial z_i}{\partial W_{mi}} & \cdots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} 0 & \cdots & \boldsymbol{x} & \cdots & 0 \end{bmatrix}$$

$$\therefore \frac{\partial L}{\partial W}_{m\times n} = \left(\frac{\partial \boldsymbol{z}_1}{\partial W}_{m\times n} \frac{\partial \boldsymbol{y}^{\mathrm{t}}}{\partial \boldsymbol{z}}_{n\times n} \frac{\partial L}{\partial \boldsymbol{y}}_{n\times 1}, \frac{\partial \boldsymbol{z}_2}{\partial W} \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}} \frac{\partial L}{\partial \boldsymbol{y}}, \ldots, \frac{\partial \boldsymbol{z}_n}{\partial W} \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}} \frac{\partial L}{\partial \boldsymbol{y}} \right)_{m\times n}$$

$$= \left(\boldsymbol{x} \operatorname{diag}(1-y_i^2)\frac{\partial L}{\partial \boldsymbol{y}}, \ldots, \boldsymbol{x} \operatorname{diag}(1-y_i^2)\frac{\partial L}{\partial \boldsymbol{y}} \right)$$

对于 $\frac{\partial L}{\partial \boldsymbol{b}}$，先计算

$$\frac{\partial \boldsymbol{z}^{\mathrm{T}}}{\partial \boldsymbol{b}}_{n\times n} = \begin{bmatrix} \frac{\partial z_1}{\partial b_1} & \cdots & \frac{\partial z_n}{\partial b_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_1}{\partial b_n} & \cdots & \frac{\partial z_n}{\partial z_n} \end{bmatrix} = I_n$$

$$\therefore \frac{\partial L}{\partial \boldsymbol{b}}_{n\times 1} = \frac{\partial \boldsymbol{z}^{\mathrm{T}}}{\partial \boldsymbol{b}}_{n\times n} \frac{\partial \boldsymbol{y}^{\mathrm{T}}}{\partial \boldsymbol{z}}_{n\times n} \frac{\partial L}{\partial \boldsymbol{y}}_{n\times 1} = \operatorname{diag}(1-y_i^2)\frac{\partial L}{\partial \boldsymbol{y}}$$

□

编程部分

3 编程作业报告

(1) 使用默认配置对模型进行训练和测试。

1) 训练模型：

input:

```
python recognition.py --mode train
```

output:

```
Epoch 01: loss = inf
Epoch 02: loss = inf
Epoch 03: loss = inf
Epoch 04: loss = 4.573
Epoch 05: loss = 3.583
Epoch 06: loss = 2.447
Epoch 07: loss = 1.830
Epoch 08: loss = 1.818
Epoch 09: loss = 1.374
Epoch 10: loss = 1.055
Epoch 10: validation accuracy = 61.3%
Epoch 11: loss = 0.835
Epoch 12: loss = 0.653
Epoch 13: loss = 0.586
Epoch 14: loss = 0.396
Epoch 15: loss = 0.254
Epoch 16: loss = 0.240
Epoch 17: loss = 0.187
Epoch 18: loss = 0.157
Epoch 19: loss = 0.112
Epoch 20: loss = 0.132
Epoch 20: validation accuracy = 62.7%
Epoch 21: loss = 0.096
Epoch 22: loss = 0.061
Epoch 23: loss = 0.060
Epoch 24: loss = 0.056
Epoch 25: loss = 0.048
Epoch 26: loss = 0.054
Epoch 27: loss = 0.038
```

```
Epoch 28: loss = 0.039
Epoch 29: loss = 0.050
Epoch 30: loss = 0.052
Epoch 30: validation accuracy = 64.0%
Epoch 31: loss = 0.036
Epoch 32: loss = 0.032
Epoch 33: loss = 0.028
Epoch 34: loss = 0.026
Epoch 35: loss = 0.025
Epoch 36: loss = 0.026
Epoch 37: loss = 0.024
Epoch 38: loss = 0.024
Epoch 39: loss = 0.021
Epoch 40: loss = 0.027
Epoch 40: validation accuracy = 65.2%
Epoch 41: loss = 0.021
Epoch 42: loss = 0.020
Epoch 43: loss = 0.017
Epoch 44: loss = 0.019
Epoch 45: loss = 0.016
Epoch 46: loss = 0.015
Epoch 47: loss = 0.016
Epoch 48: loss = 0.015
Epoch 49: loss = 0.014
Epoch 50: loss = 0.015
Epoch 50: validation accuracy = 65.5%
Model saved in saved_models/recognition.pth
```

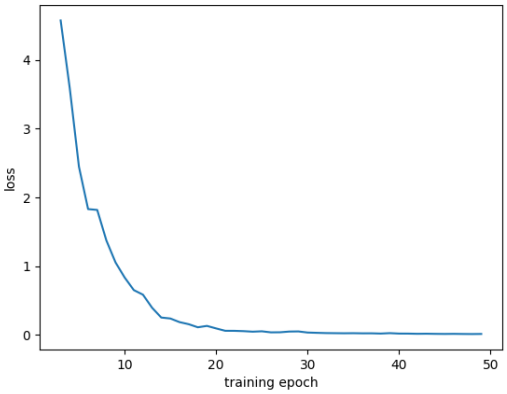


图 1

2) 测试模型：

input:

```
python recognition.py --mode test
```

output:

```
[Info] Load model from saved_models/recognition.pth
[Info] Test accuracy = 67.0%
```

(2) 调整优化器为 Adam 优化器，修改其他参数，观察网络训练、验证和测试的性能。

1) 训练模型

input:

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

output:

```
Epoch 01: loss = inf
Epoch 02: loss = inf
Epoch 03: loss = inf
Epoch 04: loss = inf
Epoch 05: loss = inf
Epoch 06: loss = inf
Epoch 07: loss = inf
Epoch 08: loss = inf
Epoch 09: loss = inf
Epoch 10: loss = 3.594
Epoch 10: validation accuracy = 59.5%
Epoch 11: loss = 2.529
Epoch 12: loss = 2.012
Epoch 13: loss = 1.733
Epoch 14: loss = 1.475
Epoch 15: loss = 1.215
Epoch 16: loss = 1.047
Epoch 17: loss = 1.050
Epoch 18: loss = 1.053
Epoch 19: loss = 0.776
Epoch 20: loss = 0.761
Epoch 20: validation accuracy = 65.0%
Epoch 21: loss = 0.768
```

```
Epoch 22: loss = 0.737
Epoch 23: loss = 0.683
Epoch 24: loss = 1.086
Epoch 25: loss = 0.613
Epoch 26: loss = 0.630
Epoch 27: loss = 0.670
Epoch 28: loss = 0.562
Epoch 29: loss = 0.683
Epoch 30: loss = 0.488
Epoch 30: validation accuracy = 72.0%
Epoch 31: loss = 0.597
Epoch 32: loss = 0.858
Epoch 33: loss = 0.559
Epoch 34: loss = 0.635
Epoch 35: loss = 0.462
Epoch 36: loss = 0.551
Epoch 37: loss = 0.470
Epoch 38: loss = 0.403
Epoch 39: loss = 0.412
Epoch 40: loss = 0.524
Epoch 40: validation accuracy = 75.5%
Epoch 41: loss = 0.544
Epoch 42: loss = 0.359
Epoch 43: loss = 0.611
Epoch 44: loss = 0.331
Epoch 45: loss = 0.334
Epoch 46: loss = 0.487
Epoch 47: loss = 0.448
Epoch 48: loss = 0.334
Epoch 49: loss = 0.313
Epoch 50: loss = 0.464
Epoch 50: validation accuracy = 78.2%
Model saved in .\saved_models\ADAM_opt.pth
```

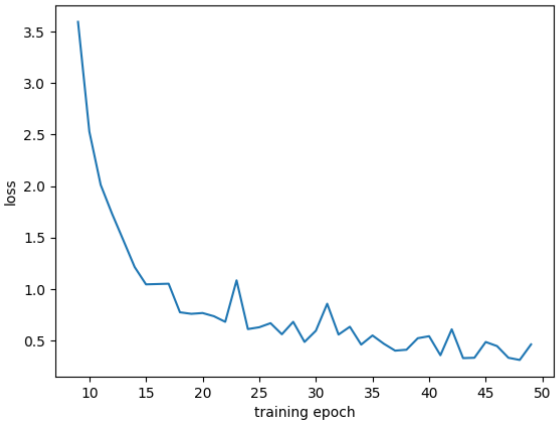


图 2

2) 测试模型

input:

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

output:

```
[Info] Load model from .\saved_models\ADAM_opt.pth
[Info] Test accuracy = 80.0%
```

(3) 选择效果最好的模型，对新的图像进行识别。

1) 选择模型：

经过一些尝试，以下模型预测准确值为 85.5%，遂选取该模型进行预测：

input:

```
python recognition.py --mode train --hsize 64 --lr 1e
    -3 --optim_type adam --momentum 15 --weight_decay
    0.1 --epoch 100 --model_path .\saved_models\
    ADAM_opt_trial_lr1e-3_momentum15_epoch100.pth
```

output:

```
Epoch 01: loss = inf
Epoch 02: loss = inf
Epoch 03: loss = inf
```

```
Epoch 04: loss = inf
Epoch 05: loss = inf
Epoch 06: loss = inf
Epoch 07: loss = inf
Epoch 08: loss = inf
Epoch 09: loss = inf
Epoch 10: loss = inf
Epoch 10: validation accuracy = 41.0%
Epoch 11: loss = inf
Epoch 12: loss = inf
Epoch 13: loss = inf
Epoch 14: loss = inf
Epoch 15: loss = inf
Epoch 16: loss = inf
Epoch 17: loss = 3.093
Epoch 18: loss = 2.537
Epoch 19: loss = 2.197
Epoch 20: loss = 1.908
Epoch 20: validation accuracy = 56.8%
Epoch 21: loss = 1.571
Epoch 22: loss = 1.438
Epoch 23: loss = 1.177
Epoch 24: loss = 1.031
Epoch 25: loss = 0.942
Epoch 26: loss = 0.913
Epoch 27: loss = 0.707
Epoch 28: loss = 0.613
Epoch 29: loss = 0.622
Epoch 30: loss = 0.481
Epoch 30: validation accuracy = 61.3%
Epoch 31: loss = 0.441
Epoch 32: loss = 0.407
Epoch 33: loss = 0.492
Epoch 34: loss = 0.408
Epoch 35: loss = 0.369
Epoch 36: loss = 0.385
Epoch 37: loss = 0.316
Epoch 38: loss = 0.472
Epoch 39: loss = 0.345
Epoch 40: loss = 0.351
Epoch 40: validation accuracy = 65.5%
Epoch 41: loss = 0.318
```

```
Epoch 42: loss = 0.313
Epoch 43: loss = 0.302
Epoch 44: loss = 0.284
Epoch 45: loss = 0.196
Epoch 46: loss = 0.380
Epoch 47: loss = 0.268
Epoch 48: loss = 0.158
Epoch 49: loss = 0.239
Epoch 50: loss = 0.315
Epoch 50: validation accuracy = 70.2%
Epoch 51: loss = 0.338
Epoch 52: loss = 0.300
Epoch 53: loss = 0.257
Epoch 54: loss = 0.227
Epoch 55: loss = 0.216
Epoch 56: loss = 0.219
Epoch 57: loss = 0.282
Epoch 58: loss = 0.276
Epoch 59: loss = 0.260
Epoch 60: loss = 0.270
Epoch 60: validation accuracy = 77.8%
Epoch 61: loss = 0.301
Epoch 62: loss = 0.218
Epoch 63: loss = 0.241
Epoch 64: loss = 0.280
Epoch 65: loss = 0.300
Epoch 66: loss = 0.299
Epoch 67: loss = 0.283
Epoch 68: loss = 0.304
Epoch 69: loss = 0.313
Epoch 70: loss = 0.276
Epoch 70: validation accuracy = 81.0%
Epoch 71: loss = 0.334
Epoch 72: loss = 0.300
Epoch 73: loss = 0.302
Epoch 74: loss = 0.287
Epoch 75: loss = 0.303
Epoch 76: loss = 0.301
Epoch 77: loss = 0.314
Epoch 78: loss = 0.320
Epoch 79: loss = 0.319
Epoch 80: loss = 0.327
```

```
Epoch 80: validation accuracy = 76.8%
Epoch 81: loss = 0.327
Epoch 82: loss = 0.341
Epoch 83: loss = 0.336
Epoch 84: loss = 0.324
Epoch 85: loss = 0.324
Epoch 86: loss = 0.357
Epoch 87: loss = 0.370
Epoch 88: loss = 0.352
Epoch 89: loss = 0.371
Epoch 90: loss = 0.343
Epoch 90: validation accuracy = 83.5%
Epoch 91: loss = 0.327
Epoch 92: loss = 0.343
Epoch 93: loss = 0.353
Epoch 94: loss = 0.347
Epoch 95: loss = 0.361
Epoch 96: loss = 0.355
Epoch 97: loss = 0.343
Epoch 98: loss = 0.357
Epoch 99: loss = 0.404
Epoch 100: loss = 0.352
Epoch 100: validation accuracy = 82.8%
Model saved in .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth
```

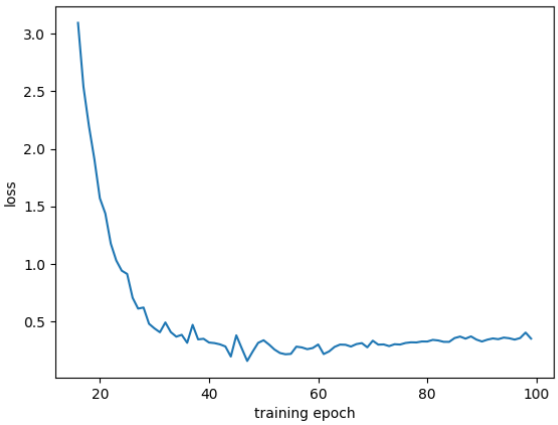


图 3

input:

```
python recognition.py --mode test --hsize 64 --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth
```

output:

```
[Info] Load model from .\saved_models\
    ADAM_opt_trial_lr1e-3_momentum15_epoch100.pth
[Info] Test accuracy = 85.5%
```

2) 进行预测：

input respectively:

```
python recognition.py --mode predict --im_path data/
    character_classification/new_images/1_W.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/2_I.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/3_S.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/4_H.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/5_Y.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/6_O.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/7_U.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/8_A.jpg --
```

```
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/9_N.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/10_I.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/11_C.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/12_E.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/13_D.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/14_A.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
python recognition.py --mode predict --im_path data/
    character_classification/new_images/15_Y.jpg --
    model_path .\saved_models\ADAM_opt_trial_lr1e-3
    _momentum15_epoch100.pth --hsize 64
```

prediction results:

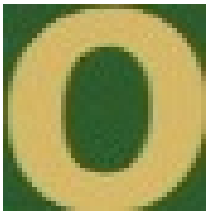
(a) pred: W

(b) pred: I

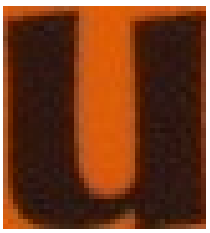
(c) pred: S

(d) pred: H

(e) pred: Y



(f) pred: O

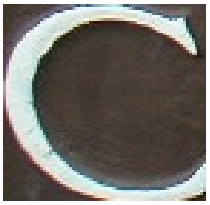


(g) pred: U

(h) pred: A

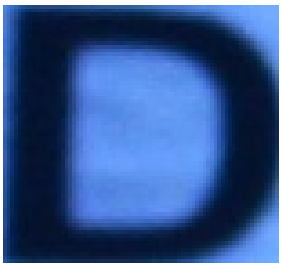
(i) pred: N

(j) pred: M



(k) pred: E

(l) pred: E



(m) pred: D

(n) pred: A

(o) pred: Y

(4) 遇到的问题及解决方法。

1) 在将样本图片转化为一维向量而使用 tensor.flatten(input, start_dim=0) 函数时，没有注意到 tensor 变量 imgs 的第 0 维是 batch_size，导致后续计算 tensor 维数错误。因为一开始就对 tensor.flatten(input, start_dim=0) 用法不确定，多留意了一下，侥幸发现了这个问题。

2) 使用 tensor.sum() 时总是搞不清楚叠加的维度究竟是哪一维。借助浏览器的力量解决了这一问题。

3) 没有注意到 opt 的 hsize 参数 default= 32，而模型训练的时候用的是 64；进行预测时又没有指定 hsize，导致 tensor 维度不对应。在重新查看了 run command 之后发现了这个小错。

(5) 建议：也许可以多介绍一下目录里各个类型的文件都是用来做什么的？第一次搭模型，很多文件类型都没见过，不知道它们有什么用。