

作业 2

毕嘉仪

2024 年 4 月 11 日

理论部分

1 单选题 (15 分)

1.1 C

1.2 D

1.3 D

1.4 C

1.5 B

2 计算题 (15 分)

2.1 已知某卷积层的输入为 X (该批量中样本数目为 1, 输入样本通道数为 1), 采用一个卷积核 W , 即卷积输出通道数为 1, 卷积核尺寸为 2×2 , 卷积的步长为 1, 无边界延拓, 偏置量为 b :

$$X = \begin{bmatrix} 0.5 & -0.2 & 0.3 \\ 0.6 & 0.4 & -0.1 \\ -0.4 & 0.5 & 0.2 \end{bmatrix}, W = \begin{bmatrix} 0.1 & -0.2 \\ -0.3 & 0.4 \end{bmatrix}, b = 0.04$$

2.1.1 请计算卷积层的输出 Y 。

解.

$$y_{11} = 0.05 + 0.04 - 0.18 + 0.16 + 0.04 = 0.11$$

$$y_{12} = -0.02 - 0.02 - 0.12 - 0.04 + 0.04 = -0.20$$

$$y_{21} = 0.06 - 0.08 + 0.12 + 0.20 + 0.04 = 0.34$$

$$y_{22} = 0.04 + 0.02 - 0.15 + 0.08 + 0.04 = 0.03$$

$$\therefore Y = \begin{bmatrix} 0.11 & -0.20 \\ 0.34 & 0.03 \end{bmatrix}$$

□

2.1.2 若训练过程中的目标函数为 L ，且已知 $\frac{\partial L}{\partial Y} = \begin{bmatrix} 0.3 & 0.1 \\ -0.4 & 0.2 \end{bmatrix}$ ，请计算 $\frac{\partial L}{\partial X}$ 。

解.

$$\begin{aligned} \frac{\partial L}{\partial X} &= \text{zero_padded}\left(\frac{\partial L}{\partial Y}\right) * W^T \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.1 & 0 \\ 0 & -0.4 & 0.2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0.4 & -0.3 \\ -0.2 & 0.1 \end{bmatrix} \\ &= \begin{bmatrix} 0.03 & -0.05 & -0.02 \\ -0.13 & 0.19 & 0 \\ 0.12 & -0.22 & 0.08 \end{bmatrix} \end{aligned}$$

□

注：本题的计算方式不限，但需要提供计算过程以及各步骤的结果。

编程部分

3 编程作业报告

(1) 探究 batch normalization 和 dropout 的作用

1) 使用默认配置（不启用 BN 和 dropout），训练 baseline 模型：

经过多次试验与测试，发现默认参数配置下的模型效果就已经是最好的了。

训练模型：

input:

```
1 python train.py --ckpt_path checkpoints/default
```

output:

```
1 training ...
2 Epoch 01: loss = 2.726, accuracy on validation set = 0.248
3 Model saved in checkpoints/default\ckpt_epoch_1.pth
4
5 Epoch 02: loss = 1.923, accuracy on validation set = 0.431
6 Model saved in checkpoints/default\ckpt_epoch_2.pth
7
8 Epoch 03: loss = 1.284, accuracy on validation set = 0.581
9 Model saved in checkpoints/default\ckpt_epoch_3.pth
10
11 Epoch 04: loss = 0.857, accuracy on validation set = 0.727
12 Model saved in checkpoints/default\ckpt_epoch_4.pth
13
14 Epoch 05: loss = 0.485, accuracy on validation set = 0.802
15 Model saved in checkpoints/default\ckpt_epoch_5.pth
16
17 Epoch 06: loss = 0.305, accuracy on validation set = 0.821
18 Model saved in checkpoints/default\ckpt_epoch_6.pth
19
20 Epoch 07: loss = 0.207, accuracy on validation set = 0.844
21 Model saved in checkpoints/default\ckpt_epoch_7.pth
22
23 Epoch 08: loss = 0.117, accuracy on validation set = 0.858
24 Model saved in checkpoints/default\ckpt_epoch_8.pth
25
26 Epoch 09: loss = 0.050, accuracy on validation set = 0.885
27 Model saved in checkpoints/default\ckpt_epoch_9.pth
28
29 Epoch 10: loss = 0.077, accuracy on validation set = 0.835
30 Model saved in checkpoints/default\ckpt_epoch_10.pth
31
32 Epoch 11: loss = 0.058, accuracy on validation set = 0.867
33 Model saved in checkpoints/default\ckpt_epoch_11.pth
34
35 Epoch 12: loss = 0.034, accuracy on validation set = 0.887
36 Model saved in checkpoints/default\ckpt_epoch_12.pth
37
38 Epoch 13: loss = 0.028, accuracy on validation set = 0.881
39 Model saved in checkpoints/default\ckpt_epoch_13.pth
40
41 Epoch 14: loss = 0.032, accuracy on validation set = 0.879
42 Model saved in checkpoints/default\ckpt_epoch_14.pth
43
44 Epoch 15: loss = 0.042, accuracy on validation set = 0.890
45 Model saved in checkpoints/default\ckpt_epoch_15.pth
```

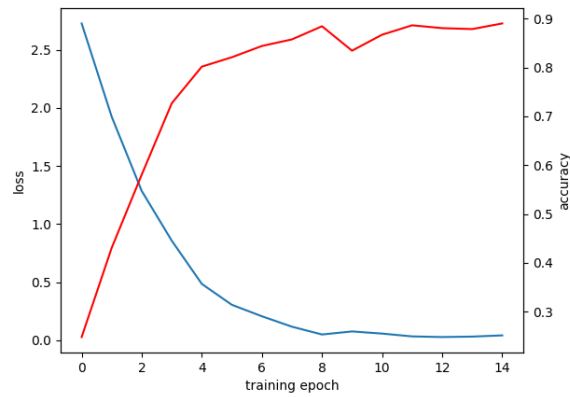


图 1

测试模型:

input:

```
1 python test.py --ckpt_path checkpoints/default --epoch 10
```

output:

```
1 [Info] loading checkpoint from checkpoints/default\ckpt_epoch_10.
   pth ...
2 accuracy on the test set: 0.806
```

2) 启用 batch normalization:

经过多次试验与测试，发现默认参数配置下的模型效果就已经是最好的了。

训练模型:

input:

```
1 python train.py --ckpt_path checkpoints/bn --bn
```

output:

```
1 training ...
2 Epoch 01: loss = 2.257, accuracy on validation set = 0.588
3 Model saved in checkpoints/bn\ckpt_epoch_1.pth
4
5 Epoch 02: loss = 0.854, accuracy on validation set = 0.835
6 Model saved in checkpoints/bn\ckpt_epoch_2.pth
7
8 Epoch 03: loss = 0.325, accuracy on validation set = 0.896
9 Model saved in checkpoints/bn\ckpt_epoch_3.pth
```

```
10
11 Epoch 04: loss = 0.129, accuracy on validation set = 0.931
12 Model saved in checkpoints/bn\ckpt_epoch_4.pth
13
14 Epoch 05: loss = 0.041, accuracy on validation set = 0.938
15 Model saved in checkpoints/bn\ckpt_epoch_5.pth
16
17 Epoch 06: loss = 0.014, accuracy on validation set = 0.958
18 Model saved in checkpoints/bn\ckpt_epoch_6.pth
19
20 Epoch 07: loss = 0.007, accuracy on validation set = 0.963
21 Model saved in checkpoints/bn\ckpt_epoch_7.pth
22
23 Epoch 08: loss = 0.005, accuracy on validation set = 0.960
24 Model saved in checkpoints/bn\ckpt_epoch_8.pth
25
26 Epoch 09: loss = 0.004, accuracy on validation set = 0.963
27 Model saved in checkpoints/bn\ckpt_epoch_9.pth
28
29 Epoch 10: loss = 0.003, accuracy on validation set = 0.960
30 Model saved in checkpoints/bn\ckpt_epoch_10.pth
31
32 Epoch 11: loss = 0.003, accuracy on validation set = 0.960
33 Model saved in checkpoints/bn\ckpt_epoch_11.pth
34
35 Epoch 12: loss = 0.002, accuracy on validation set = 0.960
36 Model saved in checkpoints/bn\ckpt_epoch_12.pth
37
38 Epoch 13: loss = 0.002, accuracy on validation set = 0.963
39 Model saved in checkpoints/bn\ckpt_epoch_13.pth
40
41 Epoch 14: loss = 0.002, accuracy on validation set = 0.960
42 Model saved in checkpoints/bn\ckpt_epoch_14.pth
43
44 Epoch 15: loss = 0.001, accuracy on validation set = 0.962
45 Model saved in checkpoints/bn\ckpt_epoch_15.pth
```

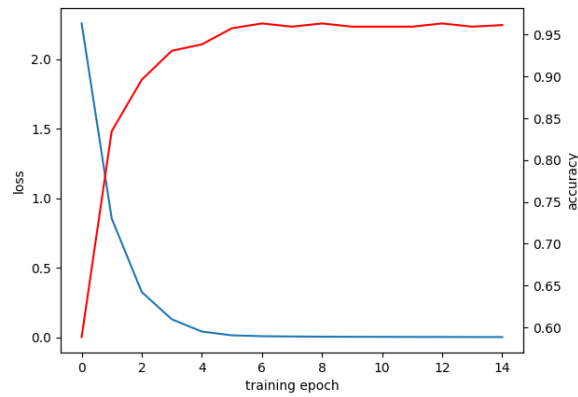


图 2

测试模型:

input:

```
1 python test.py --ckpt_path checkpoints/bn --epoch 10
```

output:

```
1 [Info] loading checkpoint from checkpoints/bn\ckpt_epoch_10.pth ...
2 accuracy on the test set: 0.962
```

分析: batchnorm 对各通道输出的数据进行归一化以后, 有效缓解了梯度消失, 测试准确率明显提升, 收敛速度也大大加快。

3) 启用 dropout 并设置概率为 0.3: 训练模型:

input:

```
1 python train.py --ckpt_path checkpoints/dropout_epoch25 --dropout
  0.3 --epoch 25
```

output:

```
1 training ...
2 Epoch 01: loss = 2.881, accuracy on validation set = 0.206
3 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_1.pth
4
5 Epoch 02: loss = 2.147, accuracy on validation set = 0.413
6 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_2.pth
7
8 Epoch 03: loss = 1.667, accuracy on validation set = 0.498
9 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_3.pth
10
```

```
11 Epoch 04: loss = 1.283, accuracy on validation set = 0.612
12 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_4.pth
13
14 Epoch 05: loss = 0.928, accuracy on validation set = 0.715
15 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_5.pth
16
17 Epoch 06: loss = 0.656, accuracy on validation set = 0.817
18 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_6.pth
19
20 Epoch 07: loss = 0.474, accuracy on validation set = 0.800
21 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_7.pth
22
23 Epoch 08: loss = 0.380, accuracy on validation set = 0.833
24 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_8.pth
25
26 Epoch 09: loss = 0.268, accuracy on validation set = 0.863
27 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_9.pth
28
29 Epoch 10: loss = 0.177, accuracy on validation set = 0.869
30 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_10.pth
31
32 Epoch 11: loss = 0.153, accuracy on validation set = 0.879
33 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_11.pth
34
35 Epoch 12: loss = 0.134, accuracy on validation set = 0.888
36 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_12.pth
37
38 Epoch 13: loss = 0.103, accuracy on validation set = 0.908
39 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_13.pth
40
41 Epoch 14: loss = 0.062, accuracy on validation set = 0.896
42 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_14.pth
43
44 Epoch 15: loss = 0.083, accuracy on validation set = 0.929
45 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_15.pth
46
47 Epoch 16: loss = 0.070, accuracy on validation set = 0.919
48 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_16.pth
49
50 Epoch 17: loss = 0.047, accuracy on validation set = 0.931
51 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_17.pth
52
53 Epoch 18: loss = 0.050, accuracy on validation set = 0.931
54 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_18.pth
55
56 Epoch 19: loss = 0.105, accuracy on validation set = 0.921
57 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_19.pth
58
59 Epoch 20: loss = 0.107, accuracy on validation set = 0.906
60 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_20.pth
61
```

```

62 Epoch 21: loss = 0.067, accuracy on validation set = 0.912
63 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_21.pth
64
65 Epoch 22: loss = 0.042, accuracy on validation set = 0.938
66 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_22.pth
67
68 Epoch 23: loss = 0.014, accuracy on validation set = 0.933
69 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_23.pth
70
71 Epoch 24: loss = 0.045, accuracy on validation set = 0.917
72 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_24.pth
73
74 Epoch 25: loss = 0.043, accuracy on validation set = 0.925
75 Model saved in checkpoints/dropout_epoch25\ckpt_epoch_25.pth

```

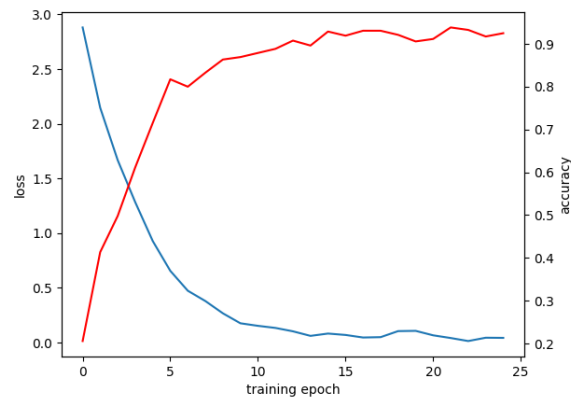


图 3

测试模型:

input:

```
1 python test.py --ckpt_path checkpoints/dropout_epoch25 --epoch 25
```

output:

```

1 [Info] loading checkpoint from checkpoints/dropout_epoch25\
  ckpt_epoch_25.pth ...
2 accuracy on the test set: 0.908

```

分析: dropout 通过丢弃某些节点的激活值, 缓解过拟合问题; 模型泛化能力提高, 测试准确率明显提升, 收敛速度有所降低。

(2) 探究数据增广的作用

1) 数据增广变换 input:

```
1 python unit_test.py data_loader
```

output:

```
1 I H M H B K J X
2 Clipping input data to the valid range for imshow with RGB data
  ([0..1] for floats or [0..255] for integers).
```

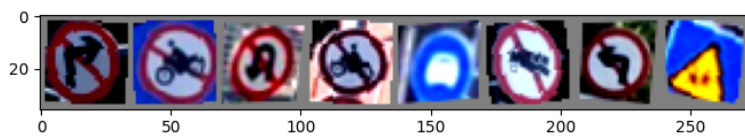


图 4

其中，所使用的增广变换为

```
1 if mode == "train" and augment:
2     data_transforms.append(transforms.RandomRotation(10))
3     data_transforms.append(transforms.RandomPerspective(0.2))
```

选择原因：首先，数据图像已将路牌主体呈现在画面中央，因此 random crop 效果并不会好；其次，由于拍摄角度问题，路牌图像的旋转角度和透视效果不一，因此选择这两种 transform 可以实现比较合理的数据增广。

2) 模型训练与测试训练模型:

input:

```
1 python train.py --ckpt_path checkpoints/bn_aug --bn --augment --
  epoch 13
```

output:

```
1 training ...
2 Epoch 01: loss = 2.478, accuracy on validation set = 0.333
3 Model saved in checkpoints/bn_aug\ckpt_epoch_1.pth
4
5 Epoch 02: loss = 1.306, accuracy on validation set = 0.756
6 Model saved in checkpoints/bn_aug\ckpt_epoch_2.pth
7
8 Epoch 03: loss = 0.661, accuracy on validation set = 0.875
```

```

9 Model saved in checkpoints/bn_aug\ckpt_epoch_3.pth
10
11 Epoch 04: loss = 0.398, accuracy on validation set = 0.887
12 Model saved in checkpoints/bn_aug\ckpt_epoch_4.pth
13
14 Epoch 05: loss = 0.246, accuracy on validation set = 0.935
15 Model saved in checkpoints/bn_aug\ckpt_epoch_5.pth
16
17 Epoch 06: loss = 0.145, accuracy on validation set = 0.904
18 Model saved in checkpoints/bn_aug\ckpt_epoch_6.pth
19
20 Epoch 07: loss = 0.142, accuracy on validation set = 0.960
21 Model saved in checkpoints/bn_aug\ckpt_epoch_7.pth
22
23 Epoch 08: loss = 0.097, accuracy on validation set = 0.954
24 Model saved in checkpoints/bn_aug\ckpt_epoch_8.pth
25
26 Epoch 09: loss = 0.065, accuracy on validation set = 0.967
27 Model saved in checkpoints/bn_aug\ckpt_epoch_9.pth
28
29 Epoch 10: loss = 0.078, accuracy on validation set = 0.963
30 Model saved in checkpoints/bn_aug\ckpt_epoch_10.pth
31
32 Epoch 11: loss = 0.071, accuracy on validation set = 0.967
33 Model saved in checkpoints/bn_aug\ckpt_epoch_11.pth
34
35 Epoch 12: loss = 0.040, accuracy on validation set = 0.967
36 Model saved in checkpoints/bn_aug\ckpt_epoch_12.pth
37
38 Epoch 13: loss = 0.027, accuracy on validation set = 0.962
39 Model saved in checkpoints/bn_aug\ckpt_epoch_13.pth

```

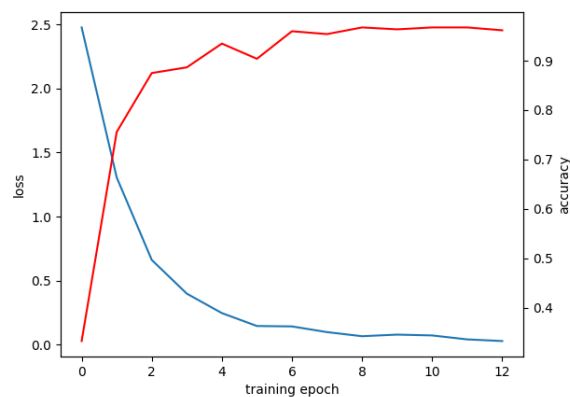


图 5

测试模型:

input:

```
1 python test.py --ckpt_path checkpoints/bn_aug --epoch 12
```

output:

```
1 [Info] loading checkpoint from checkpoints/bn_aug\ckpt_epoch_12.pth
   ...
2 accuracy on the test set: 0.968
```

(3) 探究空间变换网络 (STN) 的作用训练模型:

input:

```
1 python train.py --ckpt_path checkpoints/stn_trial --bn --stn --epoch 40
```

output:

```
1 training ...
2 Epoch 01: loss = 2.673, accuracy on validation set = 0.167
3 Model saved in checkpoints/stn_trial\ckpt_epoch_1.pth
4
5 Epoch 02: loss = 2.178, accuracy on validation set = 0.408
6 Model saved in checkpoints/stn_trial\ckpt_epoch_2.pth
7
8 Epoch 03: loss = 1.720, accuracy on validation set = 0.463
9 Model saved in checkpoints/stn_trial\ckpt_epoch_3.pth
10
11 Epoch 04: loss = 1.190, accuracy on validation set = 0.575
12 Model saved in checkpoints/stn_trial\ckpt_epoch_4.pth
13
14 Epoch 05: loss = 0.780, accuracy on validation set = 0.804
15 Model saved in checkpoints/stn_trial\ckpt_epoch_5.pth
16
17 Epoch 06: loss = 0.577, accuracy on validation set = 0.767
18 Model saved in checkpoints/stn_trial\ckpt_epoch_6.pth
19
20 Epoch 07: loss = 0.468, accuracy on validation set = 0.860
21 Model saved in checkpoints/stn_trial\ckpt_epoch_7.pth
22
23 Epoch 08: loss = 0.346, accuracy on validation set = 0.865
24 Model saved in checkpoints/stn_trial\ckpt_epoch_8.pth
25
26 Epoch 09: loss = 0.338, accuracy on validation set = 0.837
27 Model saved in checkpoints/stn_trial\ckpt_epoch_9.pth
28
29 Epoch 10: loss = 0.281, accuracy on validation set = 0.850
30 Model saved in checkpoints/stn_trial\ckpt_epoch_10.pth
31
```

```
32 Epoch 11: loss = 0.279, accuracy on validation set = 0.873
33 Model saved in checkpoints/stn_trial\ckpt_epoch_11.pth
34
35 Epoch 12: loss = 0.219, accuracy on validation set = 0.896
36 Model saved in checkpoints/stn_trial\ckpt_epoch_12.pth
37
38 Epoch 13: loss = 0.235, accuracy on validation set = 0.888
39 Model saved in checkpoints/stn_trial\ckpt_epoch_13.pth
40
41 Epoch 14: loss = 0.209, accuracy on validation set = 0.892
42 Model saved in checkpoints/stn_trial\ckpt_epoch_14.pth
43
44 Epoch 15: loss = 0.285, accuracy on validation set = 0.865
45 Model saved in checkpoints/stn_trial\ckpt_epoch_15.pth
46
47 Epoch 16: loss = 0.273, accuracy on validation set = 0.894
48 Model saved in checkpoints/stn_trial\ckpt_epoch_16.pth
49
50 Epoch 17: loss = 0.261, accuracy on validation set = 0.887
51 Model saved in checkpoints/stn_trial\ckpt_epoch_17.pth
52
53 Epoch 18: loss = 0.222, accuracy on validation set = 0.881
54 Model saved in checkpoints/stn_trial\ckpt_epoch_18.pth
55
56 Epoch 19: loss = 0.254, accuracy on validation set = 0.892
57 Model saved in checkpoints/stn_trial\ckpt_epoch_19.pth
58
59 Epoch 20: loss = 0.155, accuracy on validation set = 0.915
60 Model saved in checkpoints/stn_trial\ckpt_epoch_20.pth
61
62 Epoch 21: loss = 0.164, accuracy on validation set = 0.904
63 Model saved in checkpoints/stn_trial\ckpt_epoch_21.pth
64
65 Epoch 22: loss = 0.217, accuracy on validation set = 0.894
66 Model saved in checkpoints/stn_trial\ckpt_epoch_22.pth
67
68 Epoch 23: loss = 0.154, accuracy on validation set = 0.912
69 Model saved in checkpoints/stn_trial\ckpt_epoch_23.pth
70
71 Epoch 24: loss = 0.214, accuracy on validation set = 0.875
72 Model saved in checkpoints/stn_trial\ckpt_epoch_24.pth
73
74 Epoch 25: loss = 0.233, accuracy on validation set = 0.900
75 Model saved in checkpoints/stn_trial\ckpt_epoch_25.pth
76
77 Epoch 26: loss = 0.165, accuracy on validation set = 0.890
78 Model saved in checkpoints/stn_trial\ckpt_epoch_26.pth
79
80 Epoch 27: loss = 0.173, accuracy on validation set = 0.919
81 Model saved in checkpoints/stn_trial\ckpt_epoch_27.pth
82
```

```
83 Epoch 28: loss = 0.134, accuracy on validation set = 0.919
84 Model saved in checkpoints/stn_trial\ckpt_epoch_28.pth
85
86 Epoch 29: loss = 0.109, accuracy on validation set = 0.927
87 Model saved in checkpoints/stn_trial\ckpt_epoch_29.pth
88
89 Epoch 30: loss = 0.110, accuracy on validation set = 0.931
90 Model saved in checkpoints/stn_trial\ckpt_epoch_30.pth
91
92 Epoch 31: loss = 0.148, accuracy on validation set = 0.904
93 Model saved in checkpoints/stn_trial\ckpt_epoch_31.pth
94
95 Epoch 32: loss = 0.117, accuracy on validation set = 0.929
96 Model saved in checkpoints/stn_trial\ckpt_epoch_32.pth
97
98 Epoch 33: loss = 0.079, accuracy on validation set = 0.927
99 Model saved in checkpoints/stn_trial\ckpt_epoch_33.pth
100
101 Epoch 34: loss = 0.127, accuracy on validation set = 0.940
102 Model saved in checkpoints/stn_trial\ckpt_epoch_34.pth
103
104 Epoch 35: loss = 0.143, accuracy on validation set = 0.879
105 Model saved in checkpoints/stn_trial\ckpt_epoch_35.pth
106
107 Epoch 36: loss = 0.172, accuracy on validation set = 0.925
108 Model saved in checkpoints/stn_trial\ckpt_epoch_36.pth
109
110 Epoch 37: loss = 0.092, accuracy on validation set = 0.942
111 Model saved in checkpoints/stn_trial\ckpt_epoch_37.pth
112
113 Epoch 38: loss = 0.083, accuracy on validation set = 0.929
114 Model saved in checkpoints/stn_trial\ckpt_epoch_38.pth
115
116 Epoch 39: loss = 0.097, accuracy on validation set = 0.946
117 Model saved in checkpoints/stn_trial\ckpt_epoch_39.pth
118
119 Epoch 40: loss = 0.118, accuracy on validation set = 0.921
120 Model saved in checkpoints/stn_trial\ckpt_epoch_40.pth
```

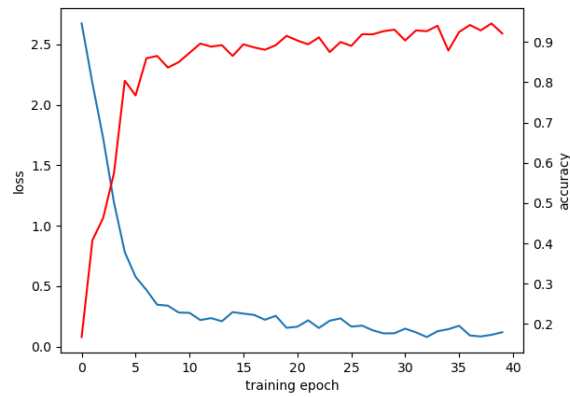


图 6

测试模型:

input:

```
1 python test.py --ckpt_path checkpoints/stn_trial --epoch 39
```

output:

```
1 [Info] loading checkpoint from checkpoints/stn_trial\ckpt_epoch_39.pth
   ...
2 accuracy on the test set: 0.947
```

(4) 可视化

分别输入以下命令:

```
1 python visualize.py --type filter --ckpt_path checkpoints/bn --
   layer_idx 0
2 python visualize.py --type filter --ckpt_path checkpoints/bn --
   layer_idx 1
3 python visualize.py --type filter --ckpt_path checkpoints/bn --
   layer_idx 2
4 python visualize.py --type filter --ckpt_path checkpoints/bn --
   layer_idx 3
5 python visualize.py --type filter --ckpt_path checkpoints/bn --
   layer_idx 4
6
7 python visualize.py --type feature --ckpt_path checkpoints/bn --
   layer_idx 0 --image_idx 50
8 python visualize.py --type feature --ckpt_path checkpoints/bn --
   layer_idx 1 --image_idx 50
```

```

9 python visualize.py --type feature --ckpt_path checkpoints/bn --
  layer_idx 2 --image_idx 50
10 python visualize.py --type feature --ckpt_path checkpoints/bn --
  layer_idx 3 --image_idx 50
11 python visualize.py --type feature --ckpt_path checkpoints/bn --
  layer_idx 4 --image_idx 50
12
13 python visualize.py --type tsne --ckpt_path checkpoints/bn
14
15 python visualize.py --type stn --ckpt_path checkpoints/stn_trial --
  epoch 39

```

1) 可视化各卷积层的卷积核:

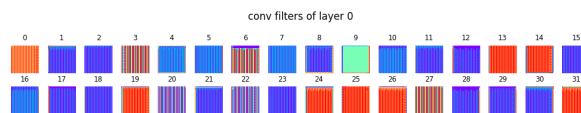


图 7

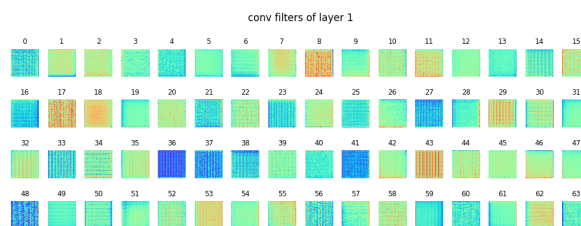


图 8

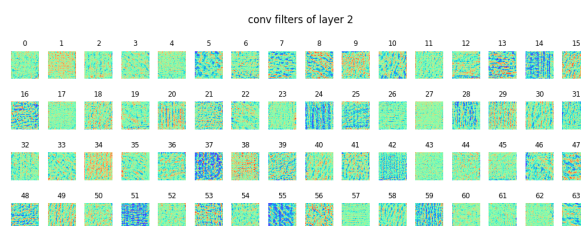


图 9

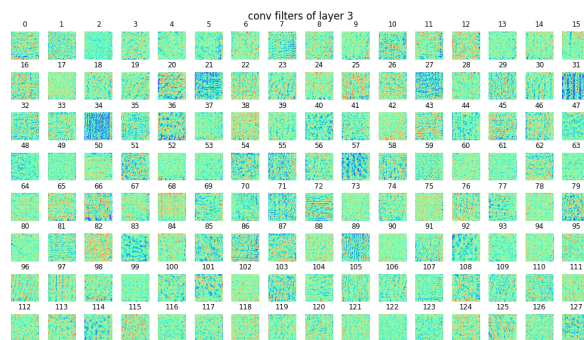


图 10

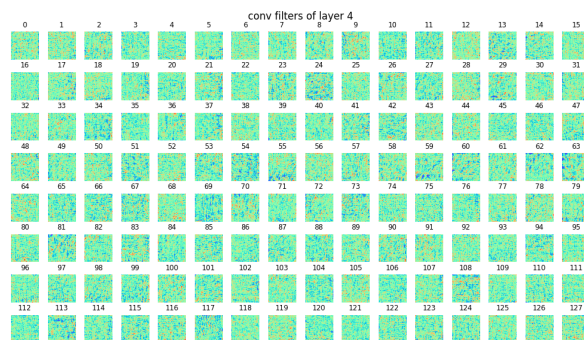


图 11

2) 可视化第 50 张图像个卷积层的输出特征图:

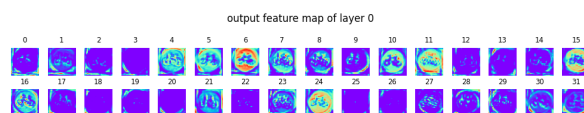


图 12

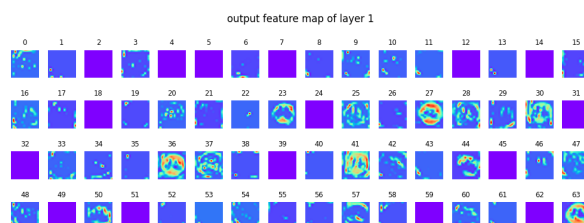


图 13

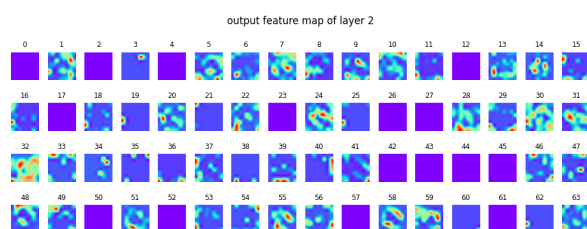


图 14

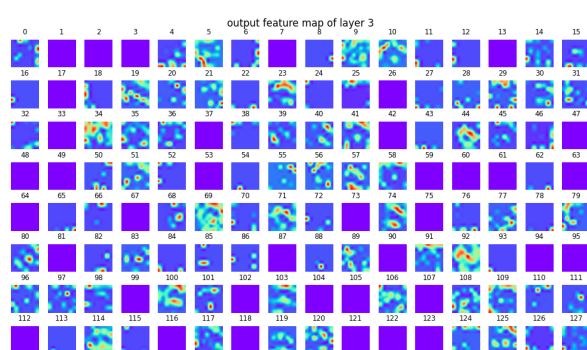


图 15

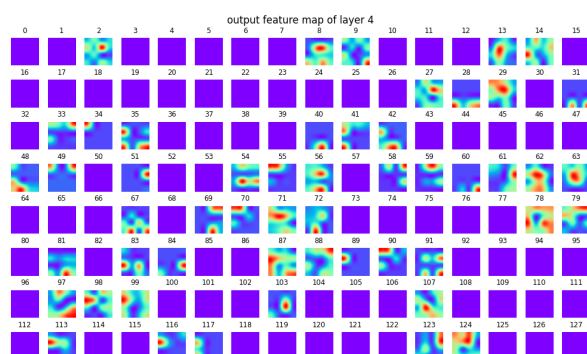


图 16

可见，随着卷积层数的增加，图像的不同特征不断被提取、分离、放大，在最后一张特征图中最为鲜明。

3) t-SNE 可视化最后一层隐藏层的输出特征：

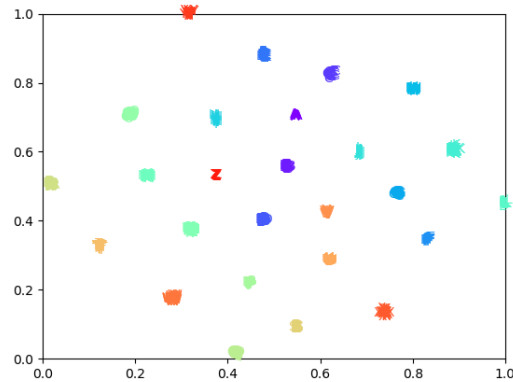


图 17

可见，各个类型的标牌被明确地分成了不同类别。

4) 可视化 STN 学习到的变换：

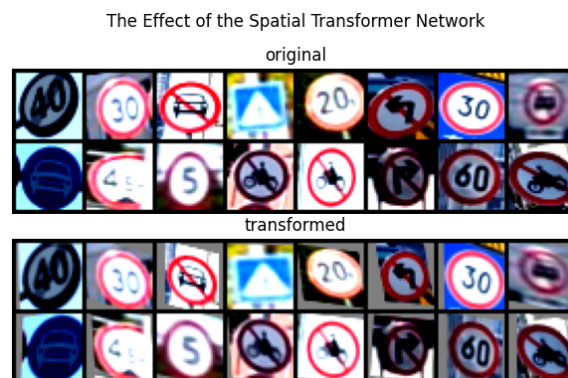


图 18

可见，经过 STN 后，图像数据产生了一些细微的变换，在提高数据多样性的同时保证其特征没有受到过大影响，从而增加了整体网络的泛化能力。

4 遇到的问题及解决方法

- (1) 没有明白 stn 网络是如何构建的，完全面向官方文档编程，但也没有太搞明白一些参数是怎么设置的。听了课上讲解之后理解了不少。

- (2) 本次的模型训练中，默认设置的网络识别准确率极高，导致后面进行数据增广和 stn 网络训练后准确率不增反掉，寻找合适的增广变换和对 stn 的调整花费了很多时间。

5 建议

希望可以大概讲一下各个参数和模型结构设置的业界习惯之类的，不然感觉自己敲出来的代码看上去非常外行……谢谢！