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 2

元敬哲 2022010657 2024 年 5 月 5 日

理论部分

1 单选题（15 分）

1.1 C

1.2 D

1.3 B

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:

依题意，Y 的大小为 (2×2)。

$$Y_{11}=0.5\times 0.1+(-0.2)\times(-0.2)+0.6\times(-0.3)+0.4\times 0.4+b=0.03$$

$$Y_{12}=-0.2\times 0.1+0.3\times(-0.2)+0.4\times(-0.3)+(-0.1)\times 0.4+b=-0.20$$

$$Y_{21} = 0.6 \times 0.1 + 0.4 \times (-0.2) + (-0.4) \times (-0.3) + 0.5 \times 0.4 + b = 0.12$$
$$Y_{22} = 0.4 \times 0.1 + (-0.1) \times (-0.2) + 0.5 \times (-0.3) + 0.2 \times 0.4 + b = -0.01$$

所以

$$Y = \begin{bmatrix} 0.03 & -0.20 \\ 0.12 & -0.01 \end{bmatrix}$$

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 $\frac{\partial L}{\partial X}$。

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。

解：

由题，

$$\frac{\partial L}{\partial X} = \begin{bmatrix} W_4 & W_3 \\ W_2 & W_1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \delta_{Y1} & \delta_{Y2} & 0 \\ 0 & \delta_{Y3} & \delta_{Y4} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

所以

$$\delta_{X1} = 0.3 \times 0.1 = 0.03$$
$$\delta_{X2} = 0.3 \times (-0.2) + 0.1 \times 0.1 = -0.05$$
$$\delta_{X3} = 0.1 \times (-0.2) = -0.02$$
$$\delta_{X4} = 0.3 \times (-0.3) + (-0.4) \times 0.1 = -0.13$$
$$\delta_{X5} = 0.3 \times 0.4 + 0.1 \times (-0.3) + (-0.04) \times (-0.2) + 0.2 \times 0.1 = 0.19$$
$$\delta_{X6} = 0.1 \times 0.4 + 0.2 \times (-0.2) = 0$$
$$\delta_{X7} = -0.4 \times (-0.3) = 0.12$$
$$\delta_{X8} = -0.4 \times (0.4) + 0.2 \times (-0.3) = -0.22$$
$$\delta_{X9} = 0.2 \times 0.4 = 0.08$$

$$\frac{\partial L}{\partial X} = \begin{bmatrix} 0.03 & -0.05 & -0.02 \\ -0.13 & 0.19 & 0 \\ 0.12 & -0.22 & 0.08 \end{bmatrix}$$

编程部分

3 编程作业报告

3.1 探究 batch normalization 和 dropout 的作用

3.1.1 使用默认配置

如图所示，训练准确率最高为 0.906：

```
Epoch 06: loss = 0.446, accuracy on validation set = 0.800
Model saved in checkpoints/default\ckpt_epoch_6.pth

Epoch 07: loss = 0.322, accuracy on validation set = 0.821
Model saved in checkpoints/default\ckpt_epoch_7.pth

Epoch 08: loss = 0.220, accuracy on validation set = 0.823
Model saved in checkpoints/default\ckpt_epoch_8.pth

Epoch 09: loss = 0.148, accuracy on validation set = 0.844
Model saved in checkpoints/default\ckpt_epoch_9.pth

Epoch 10: loss = 0.100, accuracy on validation set = 0.835
Model saved in checkpoints/default\ckpt_epoch_10.pth

Epoch 11: loss = 0.081, accuracy on validation set = 0.871
Model saved in checkpoints/default\ckpt_epoch_11.pth

Epoch 12: loss = 0.080, accuracy on validation set = 0.846
Model saved in checkpoints/default\ckpt_epoch_12.pth

Epoch 13: loss = 0.053, accuracy on validation set = 0.867
Model saved in checkpoints/default\ckpt_epoch_13.pth

Epoch 14: loss = 0.012, accuracy on validation set = 0.896
Model saved in checkpoints/default\ckpt_epoch_14.pth

Epoch 15: loss = 0.002, accuracy on validation set = 0.906
Model saved in checkpoints/default\ckpt_epoch_15.pth
```

图 1: 默认配置的结果

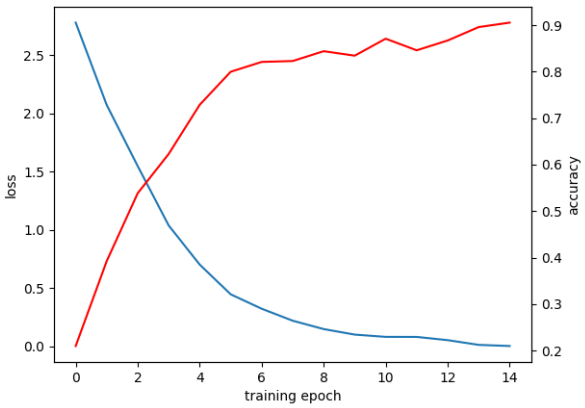

图 2: 默认配置训练可视化

3.1.2 启用 BN

如图所示，训练准确率最高为 0.965：

```
Epoch 07: loss = 0.008, accuracy on validation set = 0.965
Model saved in checkpoints/bn\ckpt_epoch_7.pth

Epoch 08: loss = 0.006, accuracy on validation set = 0.960
Model saved in checkpoints/bn\ckpt_epoch_8.pth

Epoch 09: loss = 0.004, accuracy on validation set = 0.965
Model saved in checkpoints/bn\ckpt_epoch_9.pth

Epoch 10: loss = 0.004, accuracy on validation set = 0.962
Model saved in checkpoints/bn\ckpt_epoch_10.pth

Epoch 11: loss = 0.003, accuracy on validation set = 0.958
Model saved in checkpoints/bn\ckpt_epoch_11.pth

Epoch 12: loss = 0.002, accuracy on validation set = 0.960
Model saved in checkpoints/bn\ckpt_epoch_12.pth

Epoch 13: loss = 0.002, accuracy on validation set = 0.963
Model saved in checkpoints/bn\ckpt_epoch_13.pth

Epoch 14: loss = 0.002, accuracy on validation set = 0.960
Model saved in checkpoints/bn\ckpt_epoch_14.pth

Epoch 15: loss = 0.002, accuracy on validation set = 0.962
Model saved in checkpoints/bn\ckpt_epoch_15.pth

libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
```

图 3: bn 的结果

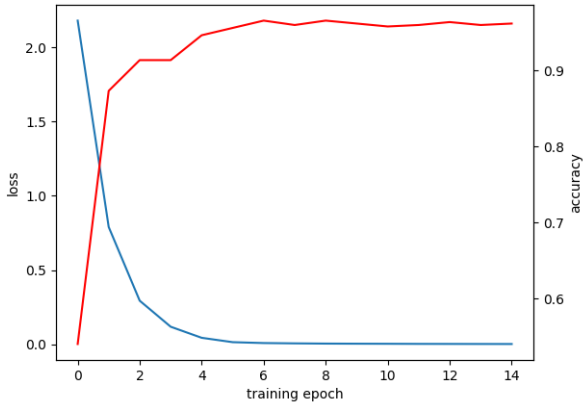

图 4: bn 训练可视化

3.1.3 启用 dropout

如图所示，训练准确率最高为 0.900：

```
Model saved in checkpoints/dropout\ckpt_epoch_5.pth

Epoch 06: loss = 0.767, accuracy on validation set = 0.760
Model saved in checkpoints/dropout\ckpt_epoch_6.pth

Epoch 07: loss = 0.582, accuracy on validation set = 0.775
Model saved in checkpoints/dropout\ckpt_epoch_7.pth

Epoch 08: loss = 0.405, accuracy on validation set = 0.837
Model saved in checkpoints/dropout\ckpt_epoch_8.pth

Epoch 09: loss = 0.323, accuracy on validation set = 0.875
Model saved in checkpoints/dropout\ckpt_epoch_9.pth

Epoch 10: loss = 0.239, accuracy on validation set = 0.873
Model saved in checkpoints/dropout\ckpt_epoch_10.pth

Epoch 11: loss = 0.175, accuracy on validation set = 0.900
Model saved in checkpoints/dropout\ckpt_epoch_11.pth

Epoch 12: loss = 0.175, accuracy on validation set = 0.863
Model saved in checkpoints/dropout\ckpt_epoch_12.pth

Epoch 13: loss = 0.165, accuracy on validation set = 0.856
Model saved in checkpoints/dropout\ckpt_epoch_13.pth

Epoch 14: loss = 0.142, accuracy on validation set = 0.894
Model saved in checkpoints/dropout\ckpt_epoch_14.pth

Epoch 15: loss = 0.088, accuracy on validation set = 0.896
Model saved in checkpoints/dropout\ckpt_epoch_15.pth
```

图 5: dropout 的结果

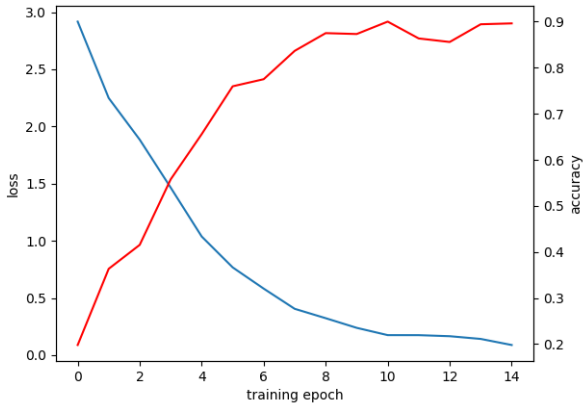

图 6: dropout 训练可视化

3.1.4 测试集上测试

如图所示，默认测试如下：

图 7: 默认测试

准确度为：0.904。

bn 测试如下：

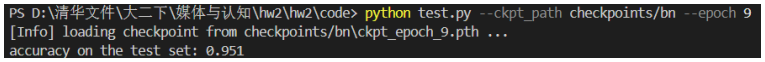

图 8: bn 测试

准确度为：0.951。

dropout 测试如下：

图 9: dropout 测试

准确度为：0.874。

在 bn 之后，Batch Normalization 层还有助于防止梯度爆炸和梯度消失。对每个特征的输入进行归一化，使其均值接近于 0，方差接近于 1，这样可以使输入数据分布更稳定，减少网络层之间的协变量偏移，有助于提高训练效果。此外，它还可以加速训练收敛，改善模型泛化能力，减少对参数初始化的敏感性，并允许使用较大的学习率，进一步加快训练速度，提高准确率。

Dropout 可以通过打破高权重连接的路径来缓解收敛缓慢问题，确保没有一条路径变得过于主导。因此，梯度可以更自由地流动并促进更快的收敛。但是本次实验 dropout 后的效果可能变差，可能是因为丢弃的神经元太多导致模型无法有效运行。

3.2 探究数据增广的作用

在 datasets.py 中定义的数据增广变换所得的图片如下：

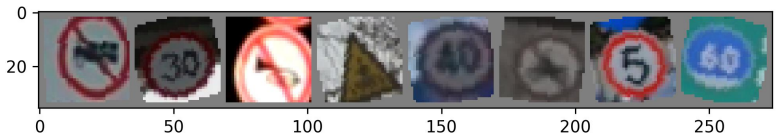

图 10: 数据增广变换后的图片

我所使用的数据增广变换为：改变色彩的亮度、对比度和饱和度。以及在 [-19.5°,19.5°] 间随机旋转。原因是这样的变换比较符合真实的各种情况的数据分布。并且小角度的随机旋转可以模拟不同拍摄角度带来的偏差，且不会改变标志语义。

训练及测试结果如下：

```
Epoch 04: loss = 0.305, accuracy on validation set = 0.912
Model saved in checkpoints/bn_aug\ckpt_epoch_4.pth

Epoch 05: loss = 0.215, accuracy on validation set = 0.938
Model saved in checkpoints/bn_aug\ckpt_epoch_5.pth

Epoch 06: loss = 0.131, accuracy on validation set = 0.954
Model saved in checkpoints/bn_aug\ckpt_epoch_6.pth

Epoch 07: loss = 0.112, accuracy on validation set = 0.946
Model saved in checkpoints/bn_aug\ckpt_epoch_7.pth

Epoch 08: loss = 0.092, accuracy on validation set = 0.956
Model saved in checkpoints/bn_aug\ckpt_epoch_8.pth
Model saved in checkpoints/bn_aug\ckpt_epoch_10.pth

Epoch 11: loss = 0.054, accuracy on validation set = 0.938
Model saved in checkpoints/bn_aug\ckpt_epoch_11.pth

Epoch 12: loss = 0.060, accuracy on validation set = 0.960
Model saved in checkpoints/bn_aug\ckpt_epoch_12.pth

Epoch 13: loss = 0.081, accuracy on validation set = 0.954
Model saved in checkpoints/bn_aug\ckpt_epoch_13.pth

Epoch 14: loss = 0.043, accuracy on validation set = 0.967
Model saved in checkpoints/bn_aug\ckpt_epoch_14.pth

Epoch 15: loss = 0.020, accuracy on validation set = 0.973
Model saved in checkpoints/bn_aug\ckpt_epoch_15.pth
```

图 11: 增广后的结果

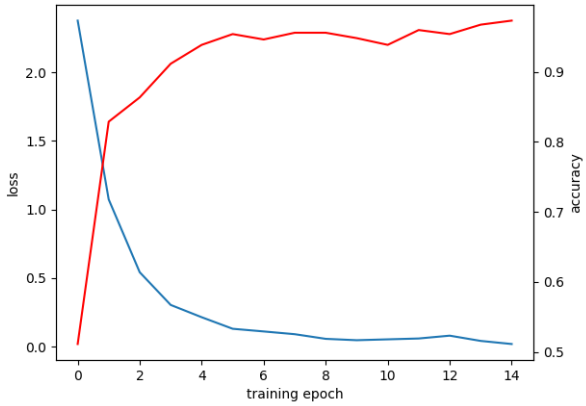

图 12: dropout 增广可视化

训练的准确率为 0.973。

```
PS D:\清华文件\大二下\媒体与认知\hw2\hw2\code> python test.py --ckpt_path checkpoints/bn_aug --epoch 15
[Info] loading checkpoint from checkpoints/bn_aug\ckpt_epoch_15.pth ...
accuracy on the test set: 0.971
```

图 13: 增广后测试结果

测试准确率为 0.971。

3.3 探究空间变换网络 (STN) 的作用

使用 stn 后，训练及测试结果如下：

```
Epoch 02: loss = 1.449, accuracy on validation set = 0.740
Model saved in checkpoints/stn\ckpt_epoch_2.pth

Epoch 03: loss = 0.664, accuracy on validation set = 0.840
Model saved in checkpoints/stn\ckpt_epoch_3.pth

Epoch 04: loss = 0.391, accuracy on validation set = 0.833
Model saved in checkpoints/stn\ckpt_epoch_4.pth

Epoch 05: loss = 0.371, accuracy on validation set = 0.923
Model saved in checkpoints/stn\ckpt_epoch_5.pth

Epoch 06: loss = 0.277, accuracy on validation set = 0.923
Model saved in checkpoints/stn\ckpt_epoch_6.pth

Epoch 07: loss = 0.148, accuracy on validation set = 0.927
Model saved in checkpoints/stn\ckpt_epoch_7.pth

Epoch 08: loss = 0.134, accuracy on validation set = 0.935
Model saved in checkpoints/stn\ckpt_epoch_8.pth

Epoch 09: loss = 0.100, accuracy on validation set = 0.946
Model saved in checkpoints/stn\ckpt_epoch_9.pth

Epoch 10: loss = 0.046, accuracy on validation set = 0.962
Model saved in checkpoints/stn\ckpt_epoch_10.pth

Epoch 11: loss = 0.026, accuracy on validation set = 0.973
Model saved in checkpoints/stn\ckpt_epoch_11.pth

Epoch 12: loss = 0.041, accuracy on validation set = 0.948
Model saved in checkpoints/stn\ckpt_epoch_12.pth

Epoch 13: loss = 0.074, accuracy on validation set = 0.917
Model saved in checkpoints/stn\ckpt_epoch_13.pth

Epoch 14: loss = 0.076, accuracy on validation set = 0.948
Model saved in checkpoints/stn\ckpt_epoch_14.pth

Epoch 15: loss = 0.042, accuracy on validation set = 0.958
Model saved in checkpoints/stn\ckpt_epoch_15.pth
```

图 14: 用 stn 的结果

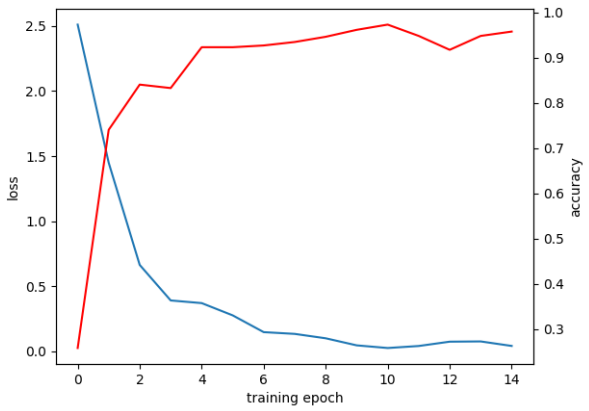

图 15: stn 可视化

训练准确率最高为 0.973，高于不用 stn 的情况。

```
PS D:\清华文件\大二下\媒体与认知\hw2\hw2\code> python test.py --ckpt_path checkpoints/stn --epoch 11
[Info] loading checkpoint from checkpoints/stn\ckpt_epoch_11.pth ...
accuracy on the test set: 0.963
```

图 16: stn 测试结果

使用 stn 后测试准确率为 0.963。

3.4 可视化

3.4.1 卷积核可视化

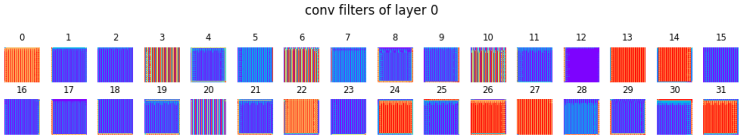

图 17: 第 0 层卷积层的卷积核

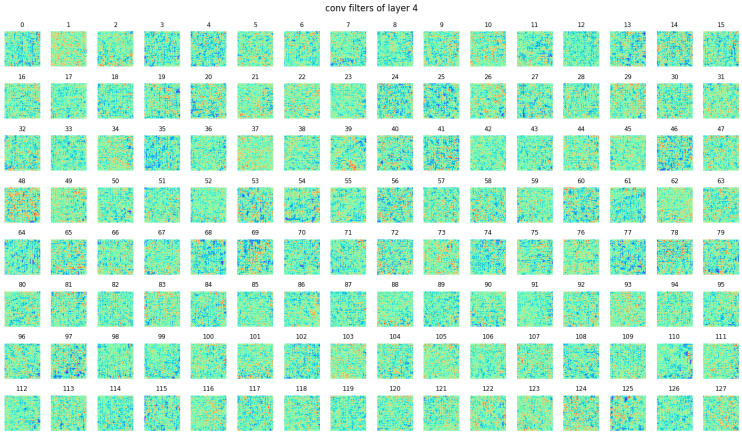

图 18: 第 4 层卷积层的卷积核

由图可见，浅层卷积核颜色、方向简单，深层的卷积核有更加复杂多样的模式。

3.4.2 卷积层的输出特征图

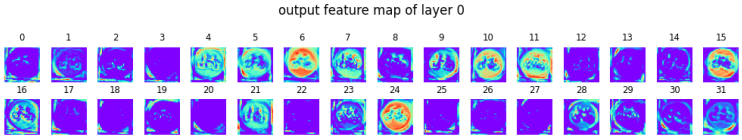

图 19: 第 50 张图像第 0 层卷积层的输出特征图

由图可知，层数越大，特征提取越抽象和细节。

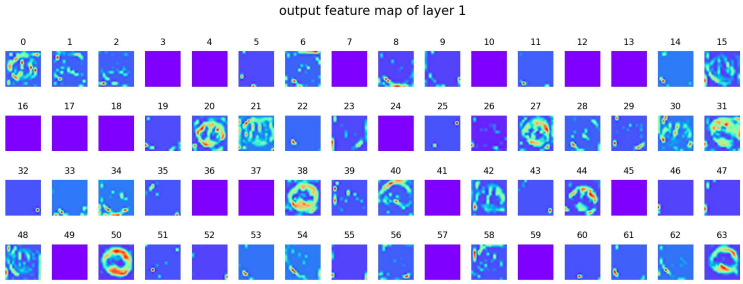

图 20: 第 50 张图像第 1 层卷积层的输出特征图

3.4.3 t-SNE 可视化最后一层隐藏层的输出特征

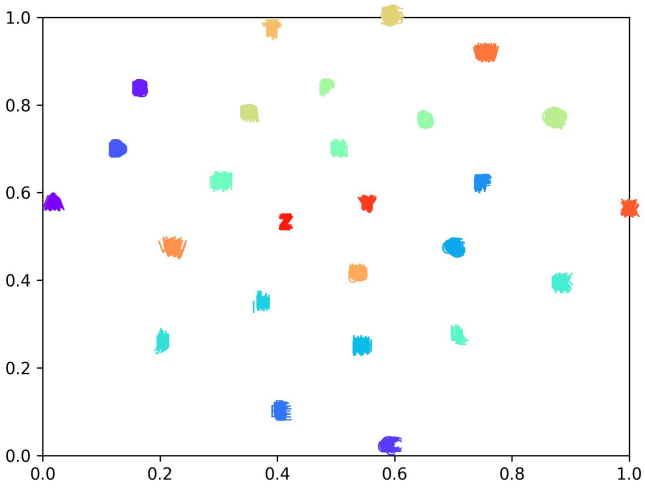

图 21: t-SNE 可视化最后一层隐藏层的输出特征图

将最后一层隐藏层的输出特征降维到 2 维并绘制，可见各个分类点比较分散，分类情况良好。

3.4.4 可视化 STN 学习到的变换

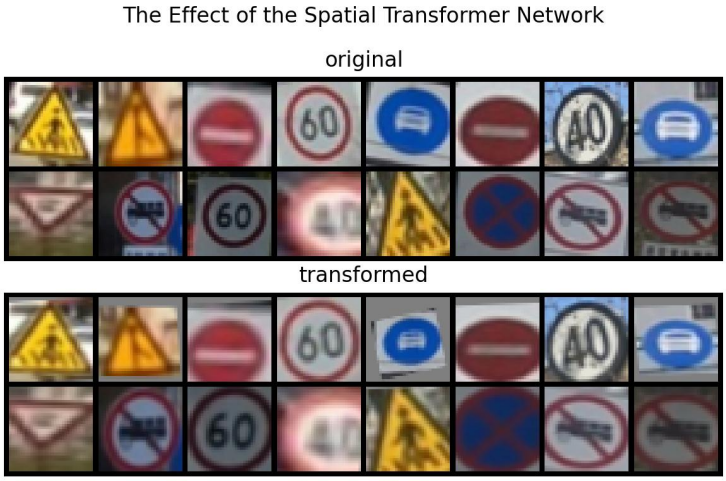

图 22: 可视化 STN 学习到的变换图

由图，可见 stn 对图片做了裁切放大和变换视角的操作。

3.5 总结与反思

本次实验遇在调试的时候遇到了代码中使用 inplace 操作导致的计算梯度消失的问题。经过不断 debug，发现是使用了 += 操作符, 后来改为两个式子相加，out=out+x, 才解决了这一问题。本次实验略微使用了大语言模型，在初始化全连接网络最后一个线性层的权重/偏置时使用 LLM，解决了自己写导致的维度不匹配的问题。