工科创 IV-J 平时作业 3

叶梓淳 520030910302

2023/3/23

1 被试依存条件

被试依存条件下，需要为 3 个 session 的每名被试单独训练模型，训练数据是本次 session 当前被试的 16 段视频，测试数据是剩余的 8 段视频，计算每个模型的正确率，最终得到 45 个模型的平均正确率。

首先尝试使用 MSE 损失，调用 nn.MSELoss 函数，需要调整数据的输入格式为 [610, 1, 310]，标签的输入格式为 [610, 1, 1]。这里先选择构建单通道的网络模型，代码如下：

- 一维卷积模型

```
class CNN_net(nn.Module):
    def __init__(self):
        super(CNN_net, self).__init__()
        self.conv1 = nn.Sequential(
            nn.Conv1d(in_channels=1, out_channels=8, kernel_size=3),
            #nn.BatchNorm1d(8),
            nn.Softmax(),
            nn.MaxPool1d(kernel_size=2), # out (8, 154)
            #nn.Dropout(0.3),
        )
        self.conv2 = nn.Sequential(
            nn.Conv1d(in_channels=8, out_channels=16, kernel_size=2,
                padding=1),
            #nn.BatchNorm1d(16),
            nn.Softmax(),
            nn.MaxPool1d(kernel_size=5), # out (16, 31)
            #nn.Dropout(0.3),
```

```
        )
        self.conv3 = nn.Sequential(
            nn.Conv1d(in_channels=16, out_channels=1, kernel_size=2,
                padding=1),
            #nn.BatchNorm1d(1),
            nn.Softmax(),
            nn.MaxPool1d(kernel_size=30), # out (1, 1)
            #nn.Dropout(0.3),
        )

        self.softmax = nn.Softmax()
        self.fc_1 = nn.Linear(in_features=1, out_features=4)


    def forward(self, x):
        out = self.conv1(x)
        out = self.conv2(out)
        out = self.conv3(out)
        out = self.fc_1(out)
        out = self.softmax(out)
        return out
```

训练轮数为 500，最终训练集的平均正确率为 0.30，测试集平均正确率为 0.36。增加 BN 层和 Dropout 层正确率并未得到明显提升。

换用交叉熵损失，调用 nn.CrossEntropyLoss 函数，需要调整数据的输入格式为 [610, 1, 310]，标签的输入格式为 [610, 4]，网络结构不改变，得到训练集准确率为 0.39，测试集准确率为 0.40，相比 MSE 损失有了小幅度提升。

调整网络结构，使用五通道一维卷积，得到训练集准确率为 0.35，测试集准确率为 0.42，由此推断可能一维卷积不适合在此情景下的分类。换用二维卷积，需要调整输入格式为 [610, 1, 62, 5]，神经网络结构如下：

- 二维卷积模型

```

class CNN_net(nn.Module):
    def __init__(self):
        super(CNN_net, self).__init__()
```

```
        self.conv1 = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=4, kernel_size=3,
                padding=1),
            nn.AvgPool2d(kernel_size=2, padding=1),
        )
        self.conv2 = nn.Sequential(
            nn.Conv2d(in_channels=4, out_channels=8, kernel_size=3,
                padding=1),
            nn.AvgPool2d(kernel_size=2, padding=1),
        )
        self.fc = nn.Sequential(
            nn.Flatten(),
            nn.Linear(in_features=272, out_features=256),
            nn.LeakyReLU(),
            nn.Linear(256, 128),
            nn.LeakyReLU(),
            nn.Linear(128, 4),
        )


    def forward(self, x):
        out = self.conv1(x)
        out = self.conv2(out)
        out = self.fc(out)
        return out
```

最终得到训练集准确率为 0.87，测试集准确率为 0.52，推测存在过拟合情况。调整训练轮数和 dropout 的比例并没有明显改善，由于训练时长的原因，很难再进行调整。

2 被试独立条件

被试独立条件下，将每个 session 下每名被试的训练集数据与测试集数据合并，由于训练时间的限制，我将三次实验分开运行，取平均准确率。同样，首先使用一维卷积进行测试，训练轮数 500，得到如下表格：

Session	训练集准确率	测试集准确率
1	0.29	0.30
2	0.37	0.32
3	0.35	0.30
平均	0.34	0.31

再使用二维卷积进行测试，训练轮数 500，得到如下表格：

Session	训练集准确率	测试集准确率
1	0.99	0.36
2	0.92	0.37
3	0.94	0.38
平均	0.95	0.37

对比发现，二维卷积对于训练集的拟合效果很好，进行调参甚至能到 0.99 左右，但是普遍存在过拟合情况，因此在测试集的表现上一般。