

搭建CNN模型技术文档

1. 初始化方法

在 `init` 方法中，定义了网络的所有层。

卷积层:

```
1 # 卷积层，用于提取图像特征
2 self.conv1 = nn.Conv2d(in_channels=3, out_channels=64, kernel_size=(3, 3),
   padding=(1, 1), stride=(1, 1))
3 self.conv2 = nn.Conv2d(in_channels=64, out_channels=128, kernel_size=(3, 3),
   padding=(1, 1), stride=(1, 1))
4 self.conv3 = nn.Conv2d(in_channels=128, out_channels=64, kernel_size=(3, 3),
   padding=(1, 1), stride=(1, 1))
5 self.conv4 = nn.Conv2d(in_channels=64, out_channels=32, kernel_size=(3, 3),
   padding=(1, 1), stride=(1, 1))
```

- `self.conv1`: 第一个卷积层，接收3个输入通道（对应于RGB图像的三个颜色通道），输出64个通道，使用3x3的卷积核，步长为1，填充为1以保持输入和输出尺寸相同。
- `self.conv2`: 第二个卷积层，接收64个输入通道，输出128个通道，同样使用3x3的卷积核，步长为1，填充为1。
- `self.conv3`: 第三个卷积层，将通道数从128减少到64，使用3x3的卷积核，步长为1，填充为1。
- `self.conv4`: 第四个卷积层，将通道数从64减少到32，使用3x3的卷积核，步长为1，填充为1。

池化层:

```
1 # 池化层，用于减少特征图的尺寸
2 self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
   ceil_mode=False)
3 self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
   ceil_mode=False)
4 self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
   ceil_mode=False)
```

```
5 self.pool4 = nn.MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
    ceil_mode=False)
```

- `self.pool1`, `self.pool2`, `self.pool3`, `self.pool4`: 每个卷积层后面都跟着一个最大池化层，使用2x2的池化核，步长为2，没有填充。这些层用于减少特征图的尺寸，增加特征的抽象程度。

展平层:

```
1 # 展平层，将多维特征图展平为一维
2 self.flat = nn.Flatten()
```

- `self.flat`: 将多维的特征图展平成一维，以便输入到全连接层。

全连接层:

```
1 # 全连接层，将特征映射到指定维度
2 self.fc1 = nn.Linear(6272, 784)
3 self.fc2 = nn.Linear(784, 98)
4 self.fc3 = nn.Linear(98, 2)
```

- `self.fc1`, `self.fc2`, `self.fc3`: 三个全连接层，分别将特征从6272维映射到784维，784维映射到98维，最后98维映射到2维（假设是二分类任务）。

激活函数和Dropout:

```
1 # 激活函数和Dropout层
2 self.softmax = nn.LogSoftmax(dim=1)
3 self.relu = nn.ReLU()
4 self.dropout = nn.Dropout(0.25)
```

- `self.softmax`, `self.relu`: 激活函数，分别用于输出层的对数softmax激活和隐藏层的ReLU激活。
- `self.dropout`: Dropout层，用于防止过拟合，这里设置了0.25的dropout率。

2. 前向传播方法

在 `forward` 方法中，定义了数据通过网络的前向传播路径。

```
1 def forward(self, x):
2     x = self.pool1(self.relu(self.conv1(x)))
3     x = self.pool2(self.relu(self.conv2(x)))
4     x = self.pool3(self.relu(self.conv3(x)))
5     x = self.pool4(self.relu(self.conv4(x)))
6     x = self.flat(x)
7     x = self.relu(self.fc1(x))
8     x = self.dropout(x)
9     x = self.fc2(x)
10    x = self.dropout(x)
11    x = self.fc3(x)
12    x = self.softmax(x)
13    return x
```

- 数据通过每个卷积层后，先进行ReLU激活，再进行最大池化。
- 展平后，数据通过全连接层，中间使用ReLU激活和Dropout。
- 最后，通过softmax层得到输出。