

cnn网络结构说明

**1. 卷积层**

**Conv1: nn.Conv2d(3, 64, kernel\_size=3, padding=1)**

输入: 3 通道（RGB 图像）

输出: 64 个特征图

核大小: 3x3

填充: 1（保持特征图大小不变）

激活函数: torch.relu

说明: 这一层从输入图像中提取低级特征，如边缘和纹理。

**Conv2: nn.Conv2d(64, 128, kernel\_size=3, padding=1)**

输入: 64 个特征图

输出: 128 个特征图

核大小: 3x3

填充: 1（保持特征图大小不变）

激活函数: torch.relu

说明: 这一层提取更高层次的特征，如简单形状和模式。

**Conv3: nn.Conv2d(128, 256, kernel\_size=3, padding=1)**

输入: 128 个特征图

输出: 256 个特征图

核大小: 3x3

填充: 1（保持特征图大小不变）

激活函数: torch.relu

说明: 这一层进一步提取更复杂的特征，如部分对象。

**Conv4: nn.Conv2d(256, 512, kernel\_size=3, padding=1)**

输入: 256 个特征图

输出: 512 个特征图

核大小: 3x3

填充: 1（保持特征图大小不变）

激活函数: torch.relu

说明: 这一层提取更高级别的特征，如完整的对象和背景信息。

**2. 池化层（Pooling Layer）**

Max Pooling: nn.MaxPool2d(kernel\_size=2, stride=2, padding=0)

输入: 各卷积层的特征图

输出: 下采样后的特征图

池大小: 2x2

步长: 2

填充: 0

说明: 这一层进行空间下采样，减小特征图尺寸，减少计算量，同时保留最重要的特征。

**3. Dropout 层**

Dropout: nn.Dropout(0.5)

说明: Dropout 是一种正则化技术，用于防止过拟合。它以 0.5 的概率随机丢弃一些神经元。

4. 全连接层（Fully Connected Layers）

FC1: nn.Linear(512 \* 2 \* 2, 1024)

输入: 展平后的特征图（512 \* 2 \* 2）

输出: 1024 个神经元

激活函数: torch.relu

说明: 这一层将卷积层的特征映射到一个较高维度的特征空间，以进行进一步的特征组合。

FC2: nn.Linear(1024, 512)

输入: 1024 个神经元

输出: 512 个神经元

激活函数: torch.relu

说明: 这一层进一步压缩特征维度，提取更高层次的抽象特征。

FC3: nn.Linear(512, 10)

输入: 512 个神经元

输出: 10 个神经元（类别数）

说明: 最后一层输出最终的类别概率分布，用于分类任务。

前向传播过程

**卷积层与激活函数**：输入图像通过 conv1 进行卷积运算，再经过 ReLU 激活函数，然后通过 pool 进行最大池化。相同的过程在 conv2、conv3 和 conv4 上重复。

**展平特征图**：x = x.view(-1, 512 \* 2 \* 2)：将池化后的特征图展平，准备进入全连接层。

**全连接层与激活函数**：展平后的特征图通过 fc1 进行全连接，再经过 ReLU 激活函数，然后通过 dropout 层进行随机丢弃。相同的过程在 fc2 上重复。

**输出层**：最后一层 fc3 输出最终的类别概率分布。

**总结**

**卷积层**：用于从输入图像中提取不同层次的特征，最初的卷积层提取简单的边缘和纹理等低级特征，后续的卷积层逐步提取更加复杂的形状和物体等高级特征。

**池化层**：通过对特征图进行下采样，减少其空间尺寸，同时保留关键信息，常见的池化方式包括最大池化和平均池化。

**全连接层**：将前面卷积层和池化层提取的特征展平成一维向量，并进行进一步的处理，用于分类或回归任务。

**Dropout**：在训练过程中随机丢弃部分神经元，以防止模型过拟合，从而提高模型在新数据上的泛化能力。

**激活函数**（如ReLU）：引入非线性变换，使得神经网络能够学习和表示更加复杂的模式和特征，相比于线性激活函数，ReLU能够更有效地缓解梯度消失问题。