Pytorch中搭建分类网络实例

2022.07.21

按照Pytorch官方教程如何去构建一个神经网络模型

神经网络由对数据执行操作的层/模块组成。torch.nn 提供了构建自己的神经网络所需的所有构建块。PyTorch 中的每个模块都是nn.Module的子模块。一个神经网络本身是由其他模块（层）组成的模块。这种嵌套结构允许轻松构建和管理复杂的架构。

一、搭建步骤

1.导入模块

```
import os
import torch
from torch import nn
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
```

2.Get Device for training

```
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"Using {device} device")
```

3.定义网络架构

```
class NeuralNetwork(nn.Module): # 所有定义的网络都继承自nn.Module
    def __init__(self):
        super(NeuralNetwork, self).__init__() # 实例化调用父类__init__方法
        self.flatten = nn.Flatten() # 定义flatten
        self.linear_relu_stack = nn.Sequential( # 按照顺序去传入一些层
            # 5层
            nn.Linear(28*28, 512), # 线性层（MLP、前馈网络全连接层）,输入维度--->输出维
度
            nn.ReLU(), # ReLu激活函数
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10), # 10分类
        )

    def forward(self, x): # forward代表对定义的网络的前向运算
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits # 得到10个类别的概率
```

```
# 实例化定义的网络
model = NeuralNetwork().to(device)
print(model)
```

4.使用

```
X = torch.rand(1, 28, 28, device=device) # 数据
logits = model(X)
pred_probab = nn.Softmax(dim=1)(logits) # 调用softmax，在dim=1维度上求和是等于1
y_pred = pred_probab.argmax(1) # 取最大值，即可以判断分类
print(f"Predicted class: {y_pred}")
```

二、详细说明各网络模块

```
input_image = torch.rand(3,28,28)
print(input_image.size())
```

1.nn.Flatten

```
flatten = nn.Flatten() # 从第一维到最后一维都变成一个维度 ---> (batch_size，其余维度相乘)
flat_image = flatten(input_image)
print(flat_image.size())
```

2.nn.Linear

```
layer1 = nn.Linear(in_features=28*28, out_features=20) # 输入维度 ---> 输出维度
hidden1 = layer1(flat_image)
print(hidden1.size())
```

3.nn.ReLu

```
print(f"Before ReLU: {hidden1}\n\n")
hidden1 = nn.ReLU()(hidden1)
print(f"After ReLU: {hidden1}")
```

4.nn.Sequential

```
# 按顺序把网络层串起来
seq_modules = nn.Sequential(
    flatten,
    layer1,
    nn.ReLU(),
    nn.Linear(20, 10)
)
input_image = torch.rand(3,28,28)
logits = seq_modules(input_image)
```

5.nn.Softmax

不同维度图解：https://zhuanlan.zhihu.com/p/525276061

```
softmax = nn.Softmax(dim=1) # 因为logits是二维的（batch_size,xxx），所以在dim=1上做softmax
pred_probab = softmax(logits)
```

6.Model Parameters

打印模型参数

```
print(f"Model structure: {model}\n\n")

for name, param in model.named_parameters():
    print(f"Layer: {name} | Size: {param.size()} | Values : {param[:2]} \n")
```