

学习使用 PyTorch 深度学习框架搭建图像分类器*

本工程的持续时间为一周

November 19, 2020

1 目标

1. 学习并掌握 PyTorch 深度学习框架。PyTorch 是一个在 Tensor 对象上执行动态计算图的系统，其操作方式与 numpy ndarray 类似。然而，它自带强大的自动分化引擎，消除了手动梯度计算和反推的需要。同时可以利用 GPU 的力量来实现自定义神经网络架构的加速训练。

2. 你将练习实现一个基于卷积神经网络的图像分类器，将你在第一周项目上得到的准确精度大幅度提高。

2 开发环境配置

上次的配置指引有些许错误，很高兴看到有同学自己搜索解决了，这次不会再有错误了。

1. 激活之前的 vclab 环境

```
conda activate vclab
```

2. 安装 PyTorch (目前 2020/07/09 的版本是 torch==1.5.1 和 torchvision==0.6.0, 如果你安装下来的不是这两个版本, 请保证 torch>=1.0, torchvision>=0.3.0)

```
pip install torch torchvision
```

*本工程由 IAIR Visual Computing Laboratory 参考斯坦福 CS231n 设计，有问题请联系 lrj466097290@stu.xjtu.edu.cn。

3 任务内容

本作业有 5 个部分。你将在三个不同的抽象层次上学习 PyTorch，这将帮助你更好地理解它，并为你的后续项目做好准备。

1. 第一部分，准备：我们将使用 CIFAR-10 数据集。
2. 第二部分，Barebones PyTorch。抽象层次 1，我们将直接使用最底层的 PyTorch Tensors。
3. 第三部分，PyTorch 模块 API。抽象层次 2，我们将使用 `nn.Module` 来定义任意的神经网络架构。
4. 第四部分，PyTorch Sequential API。抽象层次 3，我们将使用 `nn.Sequential` 来定义一个线性前馈网络，非常方便。
5. 第五部分，CIFAR-10 开放式挑战：请在 CIFAR-10 上实现你自己的网络，以获得尽可能高的精度。你可以用任何层、优化器、超参数或其他高级功能进行实验。

4 实验

1. 第一部分：直接运行 `python part1.py`，下载 `cifar10` 数据集，下载完成后，包含数据的文件夹 `'cifar-10-batches-py'` 将出现在 `vclab/datasets/` 路径下；同时输出的 `'using device:'` 将告诉现在使用的是 CPU 还是 GPU 版本的 `pytorch`（对最终精度无影响，但请大家尽力安装 GPU 版本，飞速训练提升调参效率）。

2. 第二部分：

(1) 在 `part2.py` line 169 根据此行之前的注释文字，模仿已经写好的双层神经网络，通过直接对灵活性最高的底层 `Pytorch Tensors` 进行操作，实现一个三层的卷积神经网络。实现正确后，运行 `python part2.py`，line 195 的代码将输出 `[64, 10]`；line 349 的代码将训练已写好的双层神经网络，无须修改任何超参数，正确率在 40% 以上（**如无特殊说明，均是在验证集上的正确率**）。

(2) 在 `part2.py` line 384 根据此行之前的代码以及注释文字，对三层卷积神经的权重进行随机初始化，对偏置进行全零初始化。实现正确后，运行 `python part2.py`，line 390 的代码将训练你自己实现并按上述方法初始化的三层卷积神经网络，无须修改任何超参数，正确率应该在 42% 以上。

3. 第三部分：

(1) 在 `part3.py` line 102, 114 模仿 line 51-57, 61-63，通过对最常用的 `nn.Module` 形式来定义一个三层的卷积神经网络。实现正确后，运行 `python`

part3.py, line 128 的代码将输出 [64, 10]; line 219 的代码将训练已写好的
双层神经网络网络, 无须修改任何超参数, 正确率在 40% 以上。

(2) 在 part3.py line 241 根据此行之前的代码以及注释文字, 对三层卷
积神经类进行实例化, 设置一个 SGD(stochastic gradient descent) 优化器。
实现正确后, 运行 python part3.py, line 244 的代码将利用 SGD 方法训练
你自己实现并按上述方法实例化的三层卷积神经网络。

4. 第四部分

在 part4.py line 142 模仿 line 45-53, 建立一个三层卷积神经网络, 同
时设置一个 SGD 优化器。实现正确后, 运行 python part4.py, line 147 的
代码将利用 SGD 方法训练你自己实现的三层卷积神经网络, 无须修改任何
超参数, 正确率应在 55% 以上 (细心发现比前面高达到了 55% 的原因)。

5. 第五部分

在 part5.py line 154 使用第三部分的 nn.Module 的形式, 用任何层、优
化器、超参数或其他高级功能进行实验。请仔细阅读其中的注释, 获取一些
指导信息。最终在 **测试集** 上的正确率在 70% 以上。

5 报告要求

第二至四部分请截图你写的代码, 并且将与代码对应的训练过程截图。
每一部分是一个大标题, 部分内部的每一个训练过程是一个小标题。非图结
果 (比如精度数字) 请用表格清晰的展示出来。

第五部分, 首先你应该介绍最终的模型, 你设计或选择的网络结构, 训
练优化器, 超参数, 或者其他你提升最终性能的方法以及你所实现的任何额
外的功能。

然后请提供你在进行尝试的过程中, 训练和评估网络的图表 (画图的
python 代码在第一周任务中有, 请合理使用)。

请参考以下格式:

No.1:

What: 将三层卷积神经网络加深到了十层。

Why: 深网络会抽象更加丰富的特征, 并且堆叠卷积层可以获取更大感
受野内的信息。

How: 在**测试集**上的正确率达到了 XX%; 代码截图如下; 训练过程中,
损失值 (在**训练集**和**验证集**上) 的下降过程如下。

Drawback: 深网络可能会遇到梯度消失问题, 难以训练收敛。