**机器学习**

**作业： PyTorch中的神经网络**

**最后期限： 周一 11/07 晚上 11:59 (CDT)**

## 一、宗旨：

Pytorch实现重新审视双层神经网络作业。将使用相同的玩具示例和 CIFAR-10 数据集。**所有代码都将在笔记本中实现。** 请使用 Google **Colab完成此作业**。

* 练习PyTorch
* 了解如何在PyTorch中设计您的神经网络。
* PyTorch中准备数据。
* 了解如何在PyTorch中执行训练管道。
* 了解如何在PyTorch中训练您的模型。

二。数据和代码链接：

[h ttps://drive.google.com/drive/folders/12jJrwPLaddPPKoSskIqpdmt0qhxm5sem?usp=sharing](https://drive.google.com/drive/folders/12jJrwPLaddPPKoSskIqpdmt0qhxm5sem?usp=sharing)

## 三、描述

任务 1：设计神经网络和数据准备

分类任务创建一个两层全连接神经网络。该网络的输入维度为 N，隐藏层维度为 H，输出类别为 C。您将设计您的网络结构，并在CoLab笔记本中完全设置您的学习参数。你可以通过HW.ipynb 并完成实施。

* 完成模型设计（ “class bmodel ()”）。 （详细评分条款请见评分与提交）
  + 定义您需要的所有层。
  + 设计您的网络结构。
* 完成数据准备。 （详细评分条款请见评分与提交）
  + Numpy数组传输到 Tensor。
  + 为迭代创建数据加载器。
  + 设置您的标准（损失）和优化器。
* 遍历ipynb并在每个部分显示结果。 （详细评分条款请见评分与提交）

任务 2 和 3：构建和训练Training Pipeline

完成网络设计后，您可以使用网络和数据构建训练管道。所有的训练过程都会在笔记本中的“Training”功能中完成。请按照说明完成所有代码。您可以使用 toy 数据集和 cifar-10 数据集来训练您的模型。您的所有功能都将在笔记本中完成。

* 完成Training Pipeline。 （详细评分条款请见评分与提交）
  + 完成“train”函数中的所有代码。
* 使用toy数据集来训练您的toy model。 （详细评分条款请见评分与提交）
  + 完成训练过程。
  + 通过ipynb并绘制损失和准确性
* 加载 cifar-10 数据集并训练您的 cifar-10 模型
  + 完成数据准备、pipeline setting 和training过程。
  + 通过ipynb并绘制损失和准确性。

任务 4：提高绩效

在此任务中，您可以重新设计网络以在验证集上实现更好的分类准确性。所有的代码都应该在notebook中完成。

* PyTorch提供的现有网络，来取代简单的“ bmodel ( )”（详细评分条款请参见评分和提交）
  + 使用训练集和验证集重新设计网络或使用任何技术来提高性能。
  + 在验证集上显示您的结果。

## 四、评分和提交

* 作业总分为 70 分。一般根据下表给出基本分数。

|  |
| --- |
| 第 1 部分：设计神经网络和数据准备（15 分）   * 完成模型设计（“class bmodel ( )”）。 (5') * 完成数据加载器以进行验证。 (5') * 设置您的标准（损失）和优化器。 (5')   第 2 部分：构建 Training Pipeline（25 分）   * 将数据和模型推送到 GPU。 (5') * 定义前向路径。 (5') * 计算损失。 (5') * 定义后向路径。 (5') * 在训练部分的基础上完成验证部分。 (5')   第 3 部分：训练您的网络（15 分）   * 训练玩具模型，展示“损失与时代”的情节和准确性。 (5') * 训练 cifar-10 模型，显示“ loss vs. epoch”图和准确性。 (5') * 将深度模型保存到本地。 pt文件并可以加载文件 (5')   第 4 部分：调整您的网络（15 分）   * PyTorch提供的现有网络。 (5') * 绘制新的“loss vs. epoch”图和 train/ val accuracies (5') * 用 >=50 字描述您的修改。 (5') |

对于每个 5' 刻度。

5' = 完全正确

4' = 小瑕疵

3' = 大部分不正确

2' = 完全不正确

1' = 做某事

0'=什么都不做