**机器学习**

**作业：神经网络**

## 一、宗旨：

此作业的目标是熟悉构建具有全连接层的简单神经网络模型以执行分类并在 CIFAR-10 数据集上进行测试。请使用 Google **Colab完成此作业**。随意将Pycharm用于任务 1 和 2 以编辑、测试和调试您的代码。

* 学习如何设计你的神经网络。
* 了解如何为您的训练执行正向传球和反向传球。
* 了解如何训练您的分类模型。
* 了解如何调整超参数和其他技术以提高性能。

二。数据和代码链接：

<https://drive.google.com/drive/folders/1X3lGDvx189n0GWfmnF4KjLklWObQY0xi?usp=sharing>

## 三、描述

任务 1：设计你的神经网络

分类任务创建一个两层全连接神经网络。该网络输入维数为N，隐藏层维数为H，需要实现C类分类输出。您将完成用于更新模型中参数的前向传递、反向传递函数。你所有的功能都会在相应的位置完成 在 网络.py 。您可以完成HW.ipynb任务。 1、完成实施。

* 完成前向传播功能。 （详细评分条款请见评分与提交）
  + 创建前向传球，计算班级分数。
  + 完成前向传播，计算分类器损失。
* 完成backward pass功能（详细评分条件请看评分与提交）
  + 完成backward pass，计算权重的导数。
* 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：网络训练

完成网络设计后，您可以训练您的网络。您将使用玩具数据集和 CIFAR-10 数据集 ( <http://www.cs.toronto.edu/~kriz/cifar.html>) 来训练您的模型。您与培训程序相关的所有功能都将在相应的位置完成 在 网络.py 。你可以通过HW.ipynb完成实现。

* 完成火车功能。 （详细评分条款请见评分与提交）
  + 随机创建训练数据集的小批量。
  + 创建梯度下降来更新网络的参数。
* 完成预测功能。 （详细评分条款请见评分与提交）
  + 完成预测功能。
* 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 3：调整您的网络

获得预测模型后，您可能会发现损失或多或少呈线性下降，这似乎表明学习率可能太低了。此外，训练和验证精度之间没有差距，这表明我们使用的模型容量较低，我们应该增加其尺寸。在此任务中，您可以调整网络以在验证集上实现更好的分类准确性。所有的代码都要在HW.ipynb的相应位置完成

* 实施您自己的技术来提高模型的性能（详细评分条款请参阅评分和提交）
  + 使用训练集和验证集调整超参数或使用任何技术来提高性能。
  + 在测试集上显示你的结果。

## 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：设计您的神经网络（25 分）   * 创建前向传球，计算班级分数。 (5') * 完成forward pass，计算classifier loss（5'） * 完成backward pass，计算权重的导数（5'） * 计算并显示“你的分数和正确分数之间的差异”（5'） * 计算并显示“最大相对误差”(5')   第 2 部分：网络训练（25 分）   * 随机创建一小批训练数据集 (5') * 使用minibatch GD更新网络的参数（5'） * 完成预测功能（5'） * 使用玩具数据并绘制正确的损失历史 (5') * 加载CIFAR-10训练模型，显示验证准确率（5'）   第 3 部分：调整您的网络（20 分）   * 使用一种技术（例如，调整学习率、增加模型容量等）来提高您的表现 (5') * 用 >=50 个单词解释你使用的技术（例如，更改了哪些参数、有哪些新策略和推理）(5') * 绘制损失函数和训练/验证准确度，可视化最佳网络的权重 (5') * 在测试集上显示你的最终结果，显示分类准确率 (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