

基于两层神经网络的图像分类器*

本项目的持续时间为一周

November 19, 2020

1 目标

你将练习实习一个基于两层神经网络的简单图像分类器，具体目标：

1. 理解基本图像分类过程（数据驱动的训练/预测阶段）
2. 理解将数据划分为训练/验证/测试集的意义以及使用验证集来对超参数进行微调
3. 熟练使用 numpy 编写高效的矢量化代码。
4. 实现并应用一个两层神经网络

2 开发环境配置

1. 安装 Ubuntu16.04 系统
2. 安装 python 的 IDE “PyCharm” :
<https://www.jetbrains.com/pycharm/download/#section=linux>
3. 安装 python 的环境管理平台 “Anaconda” :
<https://docs.anaconda.com/anaconda/install/linux/>
4. 解压 project1.zip，使用命令行切换进入 project1 的路径
`cd project1`
5. 准备 python 环境，下载所需包及数据
创建一个 python 环境
`conda create --name vclab python=3.7`
激活 vclab 环境（代码将使用 vclab 环境下的包）
`conda activate vclab`
给 vclab 环境安装需要的包
`pip install -r requirements`

*本工程由 IAIR Visual Computing Laboratory 参考斯坦福 CS231n 设计

(如果安装速度很慢, 百度 “ubuntu 换源”, 更改下包源为国内源)

下载 cifar10 数据集

```
cd vclab/datasets
```

```
./get_dataset.sh
```

此时你的文件结构应为

project1

vclab

classifier

 __init__.py

 neural_net.py

 ans_neural_net.py

datasets

 cifar-10-batches-py

 batches.meta

 data_batch_1

 data_batch_2

 data_batch_3

 data_batch_4

 data_batch_5

 readme.html

 test_batch

 get_datasets.sh

 __init__.py

 gradient_check.py

 data_utils.py

 vis_utils.py

 features.py

two_layer_nnet.py

3 实验

1. 在 neural_net.py line 83 实现前向传播 (forward pass), 运行 python two_layer_nnet.py

实现正确后, 输出的 scores 和正确 scores 的误差应该小于 $1e-7$

2. 在 neural_net.py line 101 实现损失计算 (loss function), 运行 python two_layer_nnet.py

实现正确后，输出的 loss 与正确 loss 的误差应该小于 $1e-8$

3. 在 `neural_net.py` line 114 实现梯度计算 (compute gradients)，运行 `python two_layer_nnet.py`

实现正确后，输出的权重 W_1, W_2, B_1, B_2 梯度与正确梯度的误差应该小于 $1e-8$

4. 在 `neural_net.py` line 159 实现产生一个 batch 的数据-标签对，同时在 `neural_net.py` line 175 实现梯度反向传播 (backward pass) 更新权重

实现正确后，输出的训练损失应该小于 0.02；同时，在 cifar10 **验证集** 正确率应该有 $\approx 29\%$

5. 观察可视化出的损失变化以及权重，分析目前在 cifar10 验证集上正确率较低的原因。尝试调整超参数（神经网络隐层维度，学习率，训练时的迭代次数，正则化系数等），将模型在**验证集**上的正确率提升至 $\geq 48\%$

请在报告中解释在调整超参数过程中，模型的正确率，训练速度随之产生的变化

6. 不要局限于超参数调整，请尝试你感兴趣的技术（加深层数，PCA 降维，添加 dropout，使用学习率退火等），将**验证集**上的正确率提升至 $\geq 52\%$

7. 在 cifar10 的测试集上展示你的模型的最终测试结果，请将在**测试集**上的正确率提升至 $\geq 48\%$

8. 通常我们会发现测试集的正确率低于训练集，请总结你利用了那些技术，或者如何调整了哪些超参数，使得正确率的差距在逐渐缩小。