

深度学习

Lab5-convolutional neural network

兰韵诗

本次Lab有作业，请在3月24日结束前提交！

Convolutional Neural Network

- **你需要做的：**根据提示，补全**CNN**代码
 - 利用设定好的输入完成卷积函数(准确的说互相关函数)和汇聚层的前向传播
 - 不能修改给定的对象属性，不能调用其他工具包，只能在“to do”下面书写代码
 - 提交之后，测试集上的准确率应该在一个正确的范围内
 - 可多次提交。即使对自己的代码没有自信也一定要提交，我们会酌情给过程分
- **TO DO：**完成《Convolutional Neural Network》项目。补全cnn_hard_version.py文件使exercise_cnn.py文件中的inference_with_CNN_np()可以顺利执行。

Evaluation脚本

```
def compute_acc(pred_file):  
    with open('./data/test_labels', 'rb') as f:  
        gold = np.asarray(pickle.load(f)).reshape(-1)  
  
    with open(pred_file) as f:  
        pred = f.readlines()  
        pred = [int(sent.strip()) for sent in pred]  
        correct_case = [i for i, _ in enumerate(gold) if gold[i] == pred[i]]  
  
    acc = len(correct_case)*1./len(gold)  
    print('The predicted accuracy is %s' %acc)  
  
if __name__ == '__main__':  
    pred_file = 'data/predict.txt'  
    compute_acc(pred_file)
```

Convolutional Neural Network

- **如果你有额外时间：**尝试不同的CNN模型
 - 读懂exercise_cnn.py文件中的代码，熟悉图像识别任务的代码流程，可通过display_cifar看到数据集的图像
 - 理解通过pytorch版本实现的CNN模型(cnn_easy_version.py中的CNN实现代码)，并尝试修改CNN的参数(如:kernel size, out channel, stride, zero padding size等)或者框架(如:修改卷积块的个数，激活函数等)看看其对模型训练的影响
 - 如果对pytorch的函数不熟悉，可以搜索pytorch的documentation
- **Note：**你们可能会发现自己实现的CNN会比pytorch内置的CNN慢很多，这是因为pytorch包含一些优化的计算策略，因此我们建议大家在cnn_easy_version.py文件中尝试不同的CNN模型，可以更快的训练模型