

深度学习

Lab5-convolutional neural network

兰韵诗

本次Lab有作业，请在4月11日结束前提交！

Lab3参考答案

Python ▾

自动换行

```
1 def softmax(X):
2     '''
3     X is the input
4     Please compute its softmax outputs
5     '''
6     X_exp = X.exp()
7     partition = X_exp.sum(dim=1, keepdim=True)
8     return X_exp / partition
```

```
1 def manual_backward(self, X, y, y_hat):
2     '''
3     X is the input feature;
4     y is the ground truth label;
5     y_hat is the predicted label.
6     Please update self.linear.weight and self.linear.bias
7     '''
8     with torch.no_grad():
9         y_onehot = torch.nn.functional.one_hot(y.long(), self.num_classes)
10        delta_w = - torch.matmul(torch.t(X), (y_hat - y_onehot))/y.size(0)
11        self.linear.weight += self.lr * torch.t(delta_w)
12        delta_b = - torch.sum((y_hat - y_onehot), dim=0)/y.size(0)
13        self.linear.bias += self.lr * delta_b
```

Lab3常见错误

```
1 def manual_backward(self, X, y, y_hat):  
2     with torch.no_grad():  
3         self.linear.weight -= self.lr * torch.matmul(X.T, (y_hat - y))  
4         self.linear.bias -= self.lr * torch.sum(y_hat - y, dim = 0)
```

- 扣分点：偏导数没算对

Lab3常见错误

```
1 def manual_backward(self, X, y, y_hat):
2     with torch.no_grad():
3         alpha = 0.0001
4         N = X.shape[0]
5         # y: 1,2,0 --> [010][001][100]
6         y_cp = torch.zeros(y.shape[0], 3)
7         y_cp[torch.arange(y.shape[0]), y.long()] = 1
8
9         dW = -1/N*(y_cp-y_hat).T@X
10        self.linear.weight -= alpha*dW
```

- 扣分点：没有更新偏置

Lab5

1. 补全cnn_hard_version.py 文件中的CNN模型
2. 在cnn_easy_version.py尝试不同结构的CNN模型, 调整参数, 看看不同CNN模型在图像分类任务上的效果

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降到一个正确的范围内并且cnn_hard_version.py文件是可执行的
 - 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
 - 禁用pytorch自带函数convXd, einsum和as_strided.
- **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train_with_CNN_hard()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