深度学习
Lab3-linear classification

兰韵诗

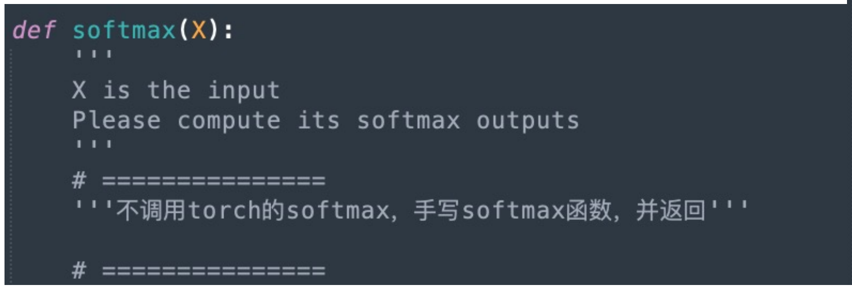
本次Lab有作业，记得3月21号结束之前提交！

Lab3

1.理解机器学习分类算法的代码实现流程
2.熟悉pytorch的自动求导流程
3.实现softmax线性分类模型

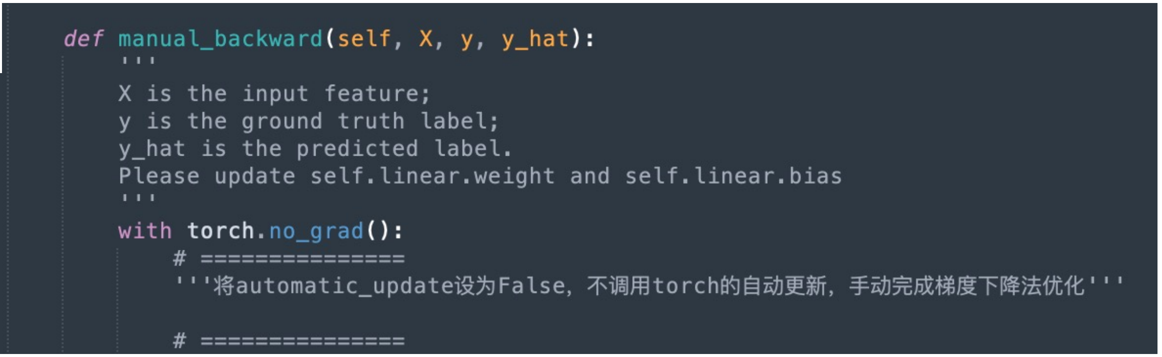
Linear classification

- 完成**softmax函数**并利用**梯度下降法优化softmax线性分类**模型
  - 不能修改项目内部给定的代码，不能import其他工具包，只能在 “to do” 下面书写代码



```
def softmax(X):
    '''
    X is the input
    Please compute its softmax outputs
    '''
    # ===============
    '''不调用torch的softmax，手写softmax函数，并返回'''

    # ===============
```

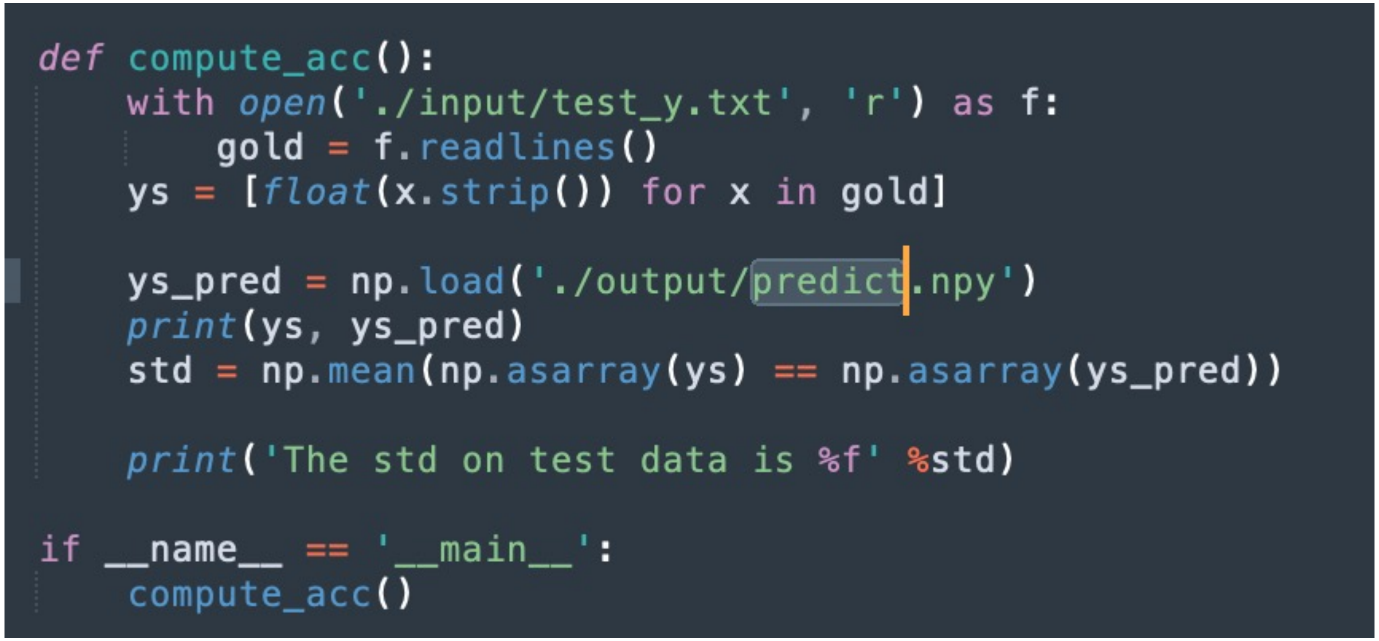


```
def manual_backward(self, X, y, y_hat):
    '''
    X is the input feature;
    y is the ground truth label;
    y_hat is the predicted label.
    Please update self.linear.weight and self.linear.bias
    '''
    with torch.no_grad():
        # ===============
        '''将automatic_update设为False，不调用torch的自动更新，手动完成梯度下降法优化'''

        # ===============
```

  - 提交之后，测试集上的损失值应该降到一个正确的范围内
  - 可多次提交
- **TO DO**： 完成《Linear classification 》项目。

Evaluation脚本



```python
def compute_acc():
    with open('./input/test_y.txt', 'r') as f:
        gold = f.readlines()
    ys = [float(x.strip()) for x in gold]

    ys_pred = np.load('./output/predict.npy')
    print(ys, ys_pred)
    std = np.mean(np.asarray(ys) == np.asarray(ys_pred))

    print('The std on test data is %f' %std)

if __name__ == '__main__':
    compute_acc()
```