

# 人工智能原理 编程作业1：使用BPNN训练EEG数据

PB16000647 羊达明

## 代码说明

自己用python实现了反向传播神经网络，并使用这个模型训练了EEG数据(其中的一部分)，然后进行了一些初步测试。由于时间并不是特别充分，没有做进一步的模型改进、参数调整以及检验工作。

源代码包括以下几个文件：

- `bpnn.py` - bpnn的实现部分，提供BPNN类以调用，初始化一个实例需要各层节点数目。可以调用的方法包括标准bpnn训练、累积bpnn训练、打印bpnn参数、测试训练好的bpnn、输出测试精确度、保存模型、导入模型。
- `EEG_bpnn.py` - 导入EEG数据，并调用 `bpnn.py` 训练、测试。
- `MyError.py` - 错误信息

## 数据

特征来自[DEAP](#)以及[MAHNOB-HCI](#)，共有160个特征，1749个数据。对应的Label为一维的"唤醒度"，取值为0或1。通过对所有数据进行分层抽样(按0：1比例)，得到训练集、测试集，具体如下：

	0类	1类
训练集	795	604
测试集	199	151

简言之所完成的工作是一个二分类任务。

## 训练

首先，初始化的bpnn实例有160个输入层节点，30个隐层节点，1个输出层节点。

对模型的第一步训练为累积bpnn训练，即根据所有的训练数据计算变化量（batch stochastic gradient descent）。前期这样做可以通过较少的迭代步数得到初步训练结果。实际实验中，对学习率 $\eta = 1, 0.5, 0.1$ 分别进行了1000次训练。

第二步则是标准bpnn训练，根据每一个数据单元计算变化量，这是在累积bpnn下降较缓时有效的改善手段。对学习率 $\eta = 0.01, 0.001, 0.0001$ 分别进行了1000次训练。

## 结果

训练后导出的数据大概有如下形式：

```
0 -> [0.11095977]
0 -> [0.10749874]
0 -> [0.6739511]
1 -> [0.78859744]
0 -> [0.01431799]
0 -> [0.48139089]
0 -> [0.19430477]
0 -> [0.06116301]
1 -> [0.88845738]
```

与0.5比较得到分类结果。

对完全不同于训练集的测试集进行测试，得到的测试结果如下：

	真实的0类	真实的1类
判断为0类	168	24
判断为1类	25	133

$$Recall_0 = \frac{168}{168 + 25} = 87.05\%, Recall_1 = \frac{133}{133 + 24} = 84.71\%$$
$$Precision_0 = \frac{168}{168 + 24} = 87.50\%, Precision_1 = \frac{133}{133 + 25} = 84.18\%$$

另外，训练集最终的总均方误差为（此时并没有将output分成0,1）：

$$E = \frac{1}{2} \sum_i (output_i - real_i)^2 = 74.8$$