机器学习实验---神经网络

一、实验目的

1. 理解神经网络算法原理，能实现神经网络分类算法；
2. 针对特定应用场景及数据，实现神经网络分类。

二、实验内容

1. 从 UCI 数据库中下载一个分类数据集，进行数据说明；
2. 用 80%的数据训练，余下的做测试，计算分类准确度。
3. 换一个别的数据集，加深神经网络，调试结果。
4. 调试不同超参，lr、优化器类别、迭代次数等等。

三、实验报告要求

1. 按实验内容撰写实验过程；
2. 报告中涉及到的代码，每一行需要有详细的注释；
3. 按自己的理解重新组织，禁止粘贴复制实验内容。

四、实验记录

ADULT 数据集：

Adult 数据集（即“人口普查收入”数据集），由美国人口普查数据集库 抽取而来，其中共包含 48842 条记录，年收入大于 50k 美元的占比 23.93%，年收入小于 50k 美元的占比 76.07%，并且已经划分为训练数据 32561 条和测试数据 16281 条。 该数据集类变量为年收入是否超过 50k 美元，属性变量包括年龄、工种、学历、职业等 14 类重要信息，其中有 8 类属于类别离散型变量，另外 6 类属于数值连续型变量。该数据集是一个分类数据集，用来预测年收入是否超过 50k 美元。

属性	类型	含义
Age	Continuous	年龄
Workclass	Discrete	工作类别
Fnlwgt	Continuous	人口普查员序号
Education	Discrete	受教育程度
Education-num	Continuous	受教育时间
Marital-status	Discrete	婚姻状况
occupation	Discrete	职业
Relationship	Discrete	社会角色
Race	Discrete	种族
Sex	Discrete	性别
Capital-gain	Continuous	资本收益
Capital-loss	Continuous	资本支出
Hours-per-week	Continuous	每周工作时间
Native-country	Discrete	国际

```
# 通过 pandas 包中 read_csv 方法，给每一列加上属性名
columns = ['Age', 'Workclass', 'fnlgwt', 'Education', 'EdNum', 'MaritalStatus',
           'Occupation', 'Relationship', 'Race', 'Sex', 'CapitalGain',
           'CapitalLoss', 'HoursPerWeek', 'Country', 'Income']
dataset = pd.read_csv('adult.csv', names=columns)
```

下载数据集并导入。

	Age	Workclass	fnlgwt	Education	EdNum	MaritalStatus	Occupation	Relationship	Race	Sex	CapitalGain	CapitalLoss	HoursPerWeek	Country	Income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K

```
# 因为 fnlgwt 属性记录的是人口普查员的 ID，对预测结果无影响，故删除该列
dataset.drop('fnlgwt', axis=1, inplace=True)

# 进行数据清洗，将数据集中‘?’字符替换为‘Unknown’
for i in dataset.columns:
    dataset[i].replace('?', 'Unknown', inplace=True)

    # 去掉非 int64 类型数据中的点和空格，以提高算法精度
    for col in dataset.columns:
        if dataset[col].dtype != 'int64':
            dataset[col] = dataset[col].apply(lambda val: val.replace(" ", ""))
            dataset[col] = dataset[col].apply(lambda val: val.replace(".", ""))

# Education（受教育程度）和 Ednum（受教育时间）特征相似，为减少干扰因素， 删除 Education 属性；除此之外，Country 对年收入的影响也不大，故同样删除
dataset.drop(['Education', 'Country'], axis=1, inplace=True)
```

对数据集中缺失值和分类 Label 进行预处理，便于将其转化为数值属性，删除无效指标。

	Age	Workclass	EdNum	MaritalStatus	Occupation	Relationship	Race	Sex	CapitalGain	CapitalLoss	HoursPerWeek	Income
0	39	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	<=50K
1	50	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	<=50K
2	38	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	<=50K
3	53	Private	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	<=50K
4	28	Private	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...
32556	27	Private	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	<=50K
32557	40	Private	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	>50K
32558	58	Private	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	<=50K
32559	22	Private	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	<=50K

```
# 调用 sklearn-pandas 包中的 DataFrameMapper 类对 AgeGroup、AgeGroup、Workclass、Occupation 等列进行标签编码，转化为连续的数值型变量，大大提高了代码的简洁性，一步到位。同时调用 sklearn.preprocessing 中的 LabelEncoder()进行编码
mapper = DataFrameMapper([('Workclass', LabelEncoder()),('MaritalStatus', LabelEncoder()),
                          ('Occupation', LabelEncoder()),('Relationship', LabelEncoder()),
                          ('Race', LabelEncoder()),('Sex', LabelEncoder()),
                          ('Income', LabelEncoder())], df_out=True, default=None)
dataset = mapper.fit_transform(dataset.copy())
```

	Workclass	MaritalStatus	Occupation	Relationship	Race	Sex	Income	Age	EdNum	CapitalGain	CapitalLoss	HoursPerWeek
0	6	4	0	1	4	1	0	39	13	2174	0	40
1	5	2	3	0	4	1	0	50	13	0	0	13
2	3	0	5	1	4	1	0	38	9	0	0	40
3	3	2	5	0	2	1	0	53	7	0	0	40
4	3	2	9	5	2	0	0	28	13	0	0	40
...	...	...	...	...	...	...	...	...	...	...	...	...
32556	3	2	12	5	4	0	0	27	12	0	0	38
32557	3	2	6	0	4	1	1	40	9	0	0	40
32558	3	6	0	4	4	0	0	58	9	0	0	40
32559	3	4	0	3	4	1	0	22	9	0	0	20
32560	4	2	3	5	4	0	1	52	9	15024	0	40

```
import torch.nn as nn
import torch.nn.functional as F


class Adult_model(nn.Module):
    def __init__(self):
        super(Adult_model, self).__init__()
        self.input = nn.Linear(11, 8)
        self.layer = nn.Linear(8, 5)
        self.output = nn.Linear(5, 2)

    def forward(self, x):
        x = F.relu(self.input(x))
        x = F.relu(self.layer(x))
        x = self.output(x)
        return x
```

构建全连接神经网络由于有 11 个属性标签，所以输入维度为 11，而 Income 一共有两个类比，所以输出维度为 2。

```
from torch.utils.data import DataLoader, random_split

dataset_size = len(dataset)
test_size = int(dataset_size * 0.2)       # 测试集大小为数据集大小的 20%
train_size = dataset_size - test_size    # 训练集大小为数据集大小减去测试集大小
```

```
train_dataset, test_dataset = random_split(dataset, lengths=[train_size, test_size])

train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True, drop_last=False)
test_loader = DataLoader(test_dataset, batch_size=64, shuffle=False, drop_last=False)
model = Adult_model()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001)
criterion = torch.nn.CrossEntropyLoss()
```

选择优化器和损失函数，并分割数据集。
最后在训练集上进行训练，并进行测试。

五、运行结果

学习率（优化器 SGD，迭代次数 1000）：
学习率 0.01

```
Income预测准确率 0.7756794104099494
```

学习率 0.001

```
Income预测准确率 0.780439121756487
```

优化器（学习率 0.001，迭代次数 1000）：
优化器 SGD

```
Income预测准确率 0.780439121756487
```

优化器 Adam：

```
Income预测准确率 0.8203592814371258
```

迭代次数（学习率 0.001，优化器 Adam）
迭代次数 1000：

```
Income预测准确率 0.8203592814371258
```

迭代次数 5000：

```
Income预测准确率 0.8387839705204975
```

注：训练过程代码以及最终准确率计算部分未给出，需要自己在实验报告中补全

六、实验小结

本次实验是理解并实现神经网络算法的原理，输入是已标签的特征向量，输出为实例的分类类别。使用了 torch 深度学习框架并使用 pycharm 作为编译器。