

# 使用教程

- 1、替换训练集（把需要预测的数据集放入程序的文件夹中）  
然后把红框位置的地方换成数据集的文件名字

```
weight_decay = 0.001 # 权重衰减系数
tcn_OutputChannelList = [32, 64] # TCN各层输出维度
tcn_KernelSize = 3 # tcn卷积核尺寸
tcn_Dropout = 0.1 # TCN dropout概率
num_blocks = 1 # gru堆叠次数
dim = 64 # 隐层维度
interval_length = 2977 # 预测数据长度，最长不可以超过总数据条数
scalar = True # 是否使用归一化
scalar_contain_labels = True # 归一化过程是否包含目标值的历史数据
target_value = '预测量' # 需要预测的列名，可以在excel中查看
# 多步，单步标签
if output_length > 1:
    forecasting_model = 'multi_steps'
else:
    forecasting_model = 'one_steps'

# 读取数据
df = pd.read_excel("data.xlsx")
df = df[:interval_length]
features_num = 6 # 请手动输入特征维度数量
if features_num > 1:
    features_ = df.values
else:
    features_ = df[target_value].values
labels_ = df[target_value].values
# 初步划分训练集、验证集、测试集
split_train_val, split_val_test = int(len(features_)*train_ratio), \
    int(len(features_)*train_ratio)+int(len(features_)*val_ratio)

# 数据标准化
```

- 2、修改预测的特征数量

红框位置处的数字是特征的数量，例如图上，总共 5 个变量特征，一个预测量。因此此处应该填  $5+1=6$

	A	B	C	D	E	F	G	H
1	特征1	特征2	特征3	特征4	特征5	预测量		
2	10.7	6.4	8.7	75	40.3	10153.32		
3	11.15	5.9	8.7	70.5	38.23	9865.78		

```
dim = 64 # 隐层维度
interval_length = 2977 # 预测数据长度，最长不可以超过总数据条数
scalar = True # 是否使用归一化
scalar_contain_labels = True # 归一化过程是否包含目标值的历史数据
target_value = '预测量' # 需要预测的列名，可以在excel中查看
# 多步，单步标签
if output_length > 1:
    forecasting_model = 'multi_steps'
else:
    forecasting_model = 'one_steps'

# 读取数据
df = pd.read_excel("data.xlsx")
df = df[:interval_length]
features_num = 6 # 请手动输入特征维度数量
if features_num > 1:
    features_ = df.values
else:
    features_ = df[target_value].values
labels_ = df[target_value].values
# 初步划分训练集、验证集、测试集
split_train_val, split_val_test = int(len(features_)*train_ratio), \
    int(len(features_)*train_ratio)+int(len(features_)*val_ratio)

# 数据标准化
```

### 3、修改预测的特征名称（数据集中，要预测的数据的最上面的文字）

	A	B	C	D	E	F	G	H
	特征1	特征2	特征3	特征4	特征5	预测量		
2	10.7	6.4	8.7	75	40.3	10153.32		
3	11.15	5.9	8.7	70.5	38.23	9865.78		

```
interval_length = 2977 # 预测数据长度，最长不可以超过总数据条数
scalar = True # 是否使用归一化
# 归一化过程是否包含目标值的历史数据
target_value = '预测量' # 需要预测的列名，可以在excel中查看
# 多步，单步标签
if output_length > 1:
    forecasting_model = 'multi_steps'
else:
    forecasting_model = 'one_steps'

# 读取数据
df = pd.read_excel("data.xlsx")
df = df[:interval_length]
features_num = 6 # 请手动输入特征维度数量
if features_num > 1:
    features_ = df.values
else:
    features_ = df[target_value].values
labels_ = df[target_value].values
# 初始化训练集，验证集，测试集
```

### 4、修改总数据集的长度（例如你的数据集有 6000 个，但是你只想使用前 3000 个，那此处就填 3000）

```
weight_decay = 0.001 # 权重衰减系数
tcn_OutputChannelList = [32, 64] # TCN各层输出维度
tcn_KernelSize = 3 # tcn卷积核尺寸
tcn_Dropout = 0.1 # TCN dropout概率
num_blocks = 1 # gru堆叠次数
dim = 64 # 隐层维度
interval_length = 2977 # 预测数据长度，最长不可以超过总数据条数
scalar = True # 是否使用归一化
scalar_contain_labels = True # 归一化过程是否包含目标值的历史数据
target_value = '预测量' # 需要预测的列名，可以在excel中查看
# 多步，单步标签
if output_length > 1:
    forecasting_model = 'multi_steps'
else:
    forecasting_model = 'one_steps'

# 读取数据
df = pd.read_excel("data.xlsx")
df = df[:interval_length]
```