*"""  
 -\*- coding = utf-8 -\*-  
 author chb  
 file 6.py  
 time 2021年03月30日 20:38  
 software PyCharm  
  
 神经网络 效果足够了  
 以四天为训练  
  
 任取四天工作日 预测下一个工作日  
掌握python和tensorflow 简单  
"""*import tensorflow as tf  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif']= 'SimHei'  
data = pd.read\_csv('loadd2.csv')  
data = np.array(data)  
  
print(data.shape)  
  
x\_train =data[:18,1:5]/1000 #  
y\_train=data[:18,5]/1000  
x\_test =data[:18,11:15]/1000 #  
y\_test = data[:18,15]/1000  
print(x\_train)  
# x\_train =data[:4,1:13]/1000 #查看相关度  
# y\_train=data[:4,13]/1000  
# x\_test =data[:4,13:25]/1000 #查看相关度  
# y\_test = data[:4,25]/1000  
  
model= tf.keras.Sequential(  
 [tf.keras.layers.Dense(64,input\_shape=(4,),activation='relu'),  
 tf.keras.layers.Dense(64,activation='relu'),  
 tf.keras.layers.Dense(32,activation='relu'),  
 tf.keras.layers.Dense(1)  
 ]  
 )  
  
model.summary()  
model.compile(  
 optimizer=tf.optimizers.Adam(lr=0.01),  
 loss='mse',)  
  
history=model.fit(x\_train,y\_train,epochs=400)  
y\_pred\_test = model.predict(x\_test)\*1000  
y\_pred\_train = model.predict(x\_train)\*1000  
print(y\_pred\_test)  
  
  
plt.figure()  
  
plt.plot(y\_train\*1000,marker='o',linewidth=4,label='train\_load')  
plt.plot(y\_pred\_train,label='train\_pred')  
plt.plot(y\_train\*400,color='white')  
plt.xlabel('time') # x轴总名称  
plt.ylabel('kwh')  
plt.legend(loc="upper left")  
  
plt.show()  
  
plt.figure()  
plt.plot(y\_test\*1000,marker='o',linewidth=4,label='test\_load')  
plt.plot(y\_pred\_test,label='test\_pred')  
plt.plot(y\_train\*400,color='white')  
plt.xlabel('time') # x轴总名称  
plt.ylabel('kwh')  
plt.legend(loc="upper left")  
# plt.plot(x\_test\*10,y\_pred ,label='train\_load')  
plt.show()  
  
plt.figure()  
plt.plot(history.epoch,history.history.get('loss'))  
plt.xlabel('epoch') # x轴总名称  
plt.ylabel('loss')  
plt.legend(loc="upper left")  
  
plt.show()  
  
#  
#  
#

plt.figure()  
plt.plot(history.epoch,history.history.get('loss'))  
plt.xlabel('epoch') # x轴总名称  
plt.ylabel('loss')  
plt.legend(loc="upper left")  
  
plt.show()