```
!pip install TFANN
import numpy as np
import matplotlib.pyplot as mpl
from sklearn.preprocessing import scale
from TFANN import ANNR
from google.colab import files
from scipy import *
import time

def NNmodeling(Nit,tolv,col,figidx):
  #Number of neurons in the input, output, and hidden layers
  input = 1
  output = 1
  hidden = 50

  #array of layers, 3 hidden and 1 output, along with the tanh activation function
  layers = [('F', hidden), ('AF', 'tanh'), ('F', hidden),
  ('AF', 'tanh'), ('F', hidden), ('AF', 'tanh'), ('F',
  output)]
  #construct the model and dictate params
  mlpr = ANNR([input], layers, batchSize = 256, maxIter =Nit, tol = tolv, reg = 1e-4,
  #number of days for the hold-out period used to access progress
  holdDays = 5
  totalDays = len(dates)
  #fit the model to the data "Learning"
  mlpr.fit(dates[0:(totalDays-holdDays)], prices[0:(totalDays-
  holdDays)])
  #Predict the stock price using the model
  pricePredict = mlpr.predict(dates)
  #Display the predicted reuslts agains the actual data
  mpl.figure(figidx)
  mpl.plot(dates, prices)
  mpl.plot(dates, pricePredict,label='tol='+str(tolv),c=col)
  mpl.legend(loc=0,fontsize=20)
  mpl.xlabel("Time",size='large')
  mpl.ylabel("Stock price",size='large')
if __name__=="__main__":
  Nit=10000
  ts=time.time()
  #files.upload()
  #!ls
  #reads data from the file and ceates a matrix with only the dates and the prices
  stock_data = loadtxt('ZBH_edited.csv', delimiter=",",skiprows=1, usecols=(1, 4))
  #scales the data to smaller values
  stock_data=scale(stock_data)
  #gets the price and dates from the matrix
  prices = stock_data[:, 1].reshape(-1, 1)
  dates = stock_data[:, 0].reshape(-1, 1)
  #creates a plot of the data and then displays it
  NNmodeling(Nit,0.1,'blue',1)
  NNmodeling(Nit,0.2,'red',1)
  te=time.time()
  print("Running time:",te-ts,"s","with Nit=",Nit)


  mpl.show()
```

```
Requirement already satisfied: TFANN in /usr/local/lib/python3.6/dist-packages
Running time: 77.99090147018433 s with Nit= 10000
```