```
# Code from Chapter 4 of Machine Learning: An Algorithmic Perspective (2nd Edition)
# by Stephen Marsland (http://stephenmonika.net)

# You are free to use, change, or redistribute the code in any way you wish for
# non-commercial purposes, but please maintain the name of the original author.
# This code comes with no warranty of any kind.

# Stephen Marsland, 2008, 2014

# The Palmerston North Ozone time series example

import pylab as pl
import numpy as np

PNoz = np.loadtxt('PNOz.dat')
pl.ion()
pl.plot(np.arange(np.shape(PNoz)[0]),PNoz[:,2],'.')
pl.xlabel('Time (Days)')
pl.ylabel('Ozone (Dobson units)')

# Normalise data
PNoz[:,2] = PNoz[:,2]-PNoz[:,2].mean()
PNoz[:,2] = PNoz[:,2]/PNoz[:,2].max()

# Assemble input vectors
t = 2
k = 3

lastPoint = np.shape(PNoz)[0]-t*(k+1)
inputs = np.zeros((lastPoint,k))
targets = np.zeros((lastPoint,1))
for i in range(lastPoint):
    inputs[i,:] = PNoz[i:i+t*k:t,2]
    targets[i] = PNoz[i+t*(k+1),2]

test = inputs[-400:,:]
testtargets = targets[-400:]
train = inputs[:-400:2,:]
traintargets = targets[:-400:2]
valid = inputs[1:-400:2,:]
validtargets = targets[1:-400:2]

# Randomly order the data
change = range(np.shape(inputs)[0])
np.random.shuffle(change)
inputs = inputs[change,:]
targets = targets[change,:]

# Train the network
import mlp
net = mlp.mlp(train,traintargets,3,outtype='linear')
net.earlystopping(train,traintargets,valid,validtargets,0.25)

test = np.concatenate((test,-np.ones((np.shape(test)[0],1))),axis=1)
testout = net.mlpfwd(test)

pl.figure()
pl.plot(np.arange(np.shape(test)[0]),testout,'.')
pl.plot(np.arange(np.shape(test)[0]),testtargets,'x')
pl.legend(('Predictions','Targets'))
print 0.5*np.sum((testtargets-testout)**2)
pl.show()
```