neural network train

In [5]:

```
#import libraries and packages
%matplotlib inline
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
import math
#translate from the matlab code
import xlrd
import pandas as pd

from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import InputLayer, Input
from tensorflow.python.keras.layers import Reshape, MaxPooling2D
from tensorflow.python.keras.layers import Conv2D, Dense, Flatten
from keras.models import Sequential
from keras.layers import Dense, Activation

#translate from the matlab code
import xlrd
import pandas as pd
import numpy as np

NumSKU=49
NumVar=13
ID=[1138263,1139362,1139363,1141061,1142731,1143640,1144140,
        1148001,1148010,1148081,1162466,1162467,1162557,1162558,
        1162559,1163152,1163153,1164313,1164961,1164962,1165757,
        1166153,1166984,1166998,1167021,1167087,1167847,1167918,
        1170236,1170372,1170739,1173299,1174241,1174242,1174243,
        1174244,1174275,1174293,1174299,1174313,1174314,1174315,
        1174339,1174340,1175687,1175833,1175835,1175950,1177151]

#train data
new_tv_info = pd.read_excel('C:/Users/wyd15/Downloads/Television.xlsx', sheet_name='new_tv_info') #processed by matla
tv_sim = pd.read_excel('Television.xlsx', sheet_name='Similarity')
#tv_sim.head()
```

```
Using TensorFlow backend.
```

In [44]:

```
print(len(new_tv_info))
new_tv_info.head()
```

```
24697
```

Out[44]:

	ID	Date	SalesQuantity	SalesQuantityLag1	SalesQuantityLag7	SalesQuantityLag14	Price	Discount	InventoryAvailability	WeekOfYear	...	
0	1	2016-01-01	2	1	1	1	2626.27000	1.01	0.93	1	...	17
1	1	2016-01-02	6	2	1	1	2651.69398	1.01	0.93	1	...	18
2	1	2016-01-03	5	6	1	1	2774.57500	1.01	0.93	1	...	18
3	1	2016-01-04	6	5	1	1	2795.76000	1.01	0.93	2	...	18
4	1	2016-01-05	2	6	1	1	2795.76000	1.01	0.93	2	...	18

5 rows × 84 columns

Sequential Model

In [68]:

```
import tensorflow as tf
import numpy as np


def get_placeholder(input1_dim=18, input2_dim=61):
    return tf.placeholder(shape=(None, input1_dim), dtype=tf.float32), \
           tf.placeholder(shape=(None, input2_dim), dtype=tf.float32), \
           tf.placeholder(shape=(None, ), dtype=tf.float32)


def get_output(input1, input2):

    input1_dim = input1.get_shape()[1]
    input2_dim = input2.get_shape()[1]
    W1_1 = tf.get_variable('W1_1', shape=(input1_dim, input1_dim+input2_dim))
    W1_2 = tf.get_variable('W1_2', shape=(input2_dim, input1_dim+input2_dim))
    b1 = tf.get_variable('b1', shape=(1, input1_dim+input2_dim))

    o1_1 = tf.matmul(input1, W1_1)
    o1_2 = tf.matmul(input2, W1_2)
    o1 = o1_1 + o1_2 + b1
    o1 = tf.nn.sigmoid(o1)

    W2_1 = tf.get_variable('W2_1', shape=(input2_dim, 1))
    W2_2 = tf.get_variable('W2_2', shape=(input1_dim+input2_dim, 1))
    b2 = tf.get_variable('b2', shape=(1, 1))
    o2_1 = tf.matmul(input2, W2_1)
    o2_2 = tf.matmul(o1, W2_2)
    o2 = o2_1 + o2_2 + b2
    o2 = tf.reshape(o2, (-1, ))

    saver = tf.train.Saver([W1_1, W1_2, b1, W2_1, W2_2, b2])
    Vars=[W1_1, W1_2, b1, W2_1, W2_2, b2]

    return o2, saver, Vars


def get_loss(labels, predictions):
    return tf.losses.mean_squared_error(labels, predictions)


def main():
    tf.reset_default_graph()

    batch_size = 50
    learning_rate = 0.11
    tf.reset_default_graph()
    x1, x2, y = get_placeholder(18, 61)
    output, saver, Vars = get_output(x1, x2)

    loss = get_loss(y, output)
    optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate)
    train_op = optimizer.minimize(loss, global_step=tf.train.get_global_step())

    #train-test_splie
    msk = np.random.rand(len(new_tv_info)) < 0.85
    tv_train = new_tv_info[msk]
    tv_test = new_tv_info[~msk]

    with tf.Session() as sess:

        sess.run(tf.global_variables_initializer())

        for step in range(493):
            x1_train = tv_train.iloc[step:step+batch_size, 66:84].values
            x2_train = np.concatenate((tv_train.iloc[step:step+batch_size, 3:11].values, tv_train.iloc[step:step+batc
            y_train = tv_train.iloc[step:step+batch_size, 2].values

            x1_test = tv_test.iloc[step:step+batch_size, 66:84].values
            x2_test = np.concatenate((tv_test.iloc[step:step+batch_size, 3:11].values, tv_test.iloc[step:step+batch_s
            y_test = tv_test.iloc[step:step+batch_size, 2].values

            _train_loss, _ = sess.run([loss, train_op],
                                      feed_dict={x1: x1_train,
                                                 x2: x2_train,
                                                 y: y_train})

            _test_loss= sess.run([loss],
                                 feed_dict={x1: x1_test,
                                            x2: x2_test,
                                            y: y_test})
```

```
            #results, _ = sess.run([output]) #print parameters, W1_1 etc..

            if step % 100==0:
                saver.save(sess, 'C:/Users/wyd15/Desktop/tv_model/tv_modelslack.ckpt', global_step=step)
                print("iter: %d, train_loss: %f, test_loss: %f"%(step, _train_loss,  _test_loss[0]))
                #print('output:', )


if __name__ == '__main__':
    main()
```

```
iter: 0, train_loss: 790129.125000, test_loss: 86971.968750
iter: 100, train_loss: 61.847450, test_loss: 2908.143066
iter: 200, train_loss: 28.520618, test_loss: 768.545105
iter: 300, train_loss: 30.584072, test_loss: 317.906342
iter: 400, train_loss: 0.296206, test_loss: 0.626445
```

original tv demand neural network

In [71]:

```
# tensorflow nn
#unfolded version
#reference code: https://github.com/yl3829/Spring2018-Project5-grp5/blob/master/lib/model_tb.py

# D = tf.placeholder(dytpe=np.floate32)
# w1_1 = tf.Variable((18,79),name='W1_1')
# w1_2 = tf.Variable((61,79),name='W1_2')
# b1 = tf.Variable((79,),name='b1')
# param = [w1_1,w1_2,b1]
tf.reset_default_graph()
x1=tf.placeholder(shape=(None, 18), dtype=tf.float32)
x2=tf.placeholder(shape=(None, 61), dtype=tf.float32)
y =tf.placeholder(shape=(None, ), dtype=tf.float32)

#
W1_1 = tf.get_variable('W1_1', shape=(18, 79))
W1_2 = tf.get_variable('W1_2', shape=(61, 79))
b1 = tf.get_variable('b1', shape=(1, 79))

o1_1 = tf.matmul(x1, W1_1)
o1_2 = tf.matmul(x2, W1_2)
o1 = o1_1 + o1_2 + b1
o1 = tf.nn.sigmoid(o1)

W2_1 = tf.get_variable('W2_1', shape=(61, 1))
W2_2 = tf.get_variable('W2_2', shape=(79, 1))
b2 = tf.get_variable('b2', shape=(1, 1))

o2_1 = tf.matmul(x2, W2_1)
o2_2 = tf.matmul(o1, W2_2)
o2 = o2_1 + o2_2 + b2
o2 = tf.reshape(o2, (-1, ))

saver = tf.train.Saver([W1_1, W1_2, b1, W2_1, W2_2, b2])
#

loss = tf.losses.mean_squared_error(y, o2)
optimizer = tf.train.AdamOptimizer(learning_rate=0.11)
train_op = optimizer.minimize(loss, global_step=tf.train.get_global_step())

# layer1_out  = tf.nn.sigmoid(tf.matmul(input1,w1_1) + tf.matmul(input2,w1_2) + b1) # tanh would be better
# # example to minimize layer1 out put
# losses = layer2_out - D
# loss = tf.reduce_mean(losses)
# trainer = tf.train.AdamOptimizer()
# gradients = trainer.compute_gradients(self.loss)
# optimizer = trainer.apply_gradients(gradients)

#train-test_splie
msk = np.random.rand(len(new_tv_info)) < 0.85
tv_train = new_tv_info[msk]
tv_test = new_tv_info[~msk]
batch_size=50

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for step in range(493):
        x1_train = tv_train.iloc[step:step+batch_size, 66:84].values
        x2_train = np.concatenate((tv_train.iloc[step:step+batch_size, 3:11].values, tv_train.iloc[step:step+batch_si
        y_train = tv_train.iloc[step:step+batch_size, 2].values

        x1_test = tv_test.iloc[step:step+batch_size, 66:84].values
        x2_test = np.concatenate((tv_test.iloc[step:step+batch_size, 3:11].values, tv_test.iloc[step:step+batch_size,
        y_test = tv_test.iloc[step:step+batch_size, 2].values

        _train_loss, _ = sess.run([loss, train_op],feed_dict={x1: x1_train,
                                              x2: x2_train,
                                              y: y_train})

        _test_loss= sess.run([loss],feed_dict={x1: x1_test,
                                              x2: x2_test,
                                              y: y_test})
        # predict
        D_predict = sess.run([o2], feed_dict={x1:x1_test,x2:x2_test})
        para_pred = sess.run([W1_1, W1_2, b1, W2_1, W2_2, b2], feed_dict={x1:x1_test,x2:x2_test})
        #results, _ = sess.run([output]) #print parameters, W1_1 etc..

        if step % 100==0:
            saver.save(sess, 'C:/Users/wyd15/Desktop/tv_model/tv_modelslack.ckpt', global_step=step)
            print("iter:%d, train_loss: %f, test_loss: %f"%(step, _train_loss,  _test_loss[0]))
            print('test prediction parameters results:', D_predict)
```

```
            #print('predicted parameters:', para_pred)
    print('last test prediction parameters results:', step, _train_loss,  _test_loss[0], D_predict)
    paras=para_pred
                #print('output:', )
```

```
iter:0, train_loss: 387154.093750, test_loss: 2670.869629
test prediction parameters results: [array([ 75.87928772,  65.21278381,  55.27078629,  58.219944  ,
        65.50083923,  62.80655289,  63.06184769,  51.526371  ,
        49.79753494,  53.8701973 ,  56.84691238,  74.79963684,
        61.30699158,  56.17599487,  45.11227417,  51.51878738,
        43.78727341,  42.5931015 ,  57.85210037,  53.92779922,
        55.53042984,  61.38663101,  55.4371376 ,  59.43503189,
        57.33987808,  56.95289993,  57.04330826,  62.09218216,
        56.30978394,  56.17868042,  63.40144348,  62.89990616,
        37.52988052,  38.19394302,  61.99638748,  37.03100967,
        36.73166275,  49.18375397,  46.22643661,  32.58309555,
        32.4204216 ,  32.52098465,  63.98768234,  64.22125244,
        64.06978607,  64.32680511,  64.23438263,  64.13961029,
        64.39714813,  64.05422974], dtype=float32)]
iter:100, train_loss: 72.489128, test_loss: 5114.185059
test prediction parameters results: [array([ -9.29467850e+01,  -1.13043106e+02,  -9.64898987e+01,
        -8.51466599e+01,  -1.01293991e+02,  -1.01787178e+02,
        -1.01119888e+02,  -1.01140938e+02,  -1.00355583e+02,
        -9.96641998e+01,  -1.09477051e+02,  -1.09302948e+02,
        -1.09816849e+02,  -1.08831421e+02,  -1.09097572e+02,
        -1.08867073e+02,  -1.08428154e+02,  -1.08887955e+02,
        -1.08519981e+02,  -1.08027863e+02,  -1.08598091e+02,
        -1.08367645e+02,  -1.23325386e+01,  -5.26970434e+00,
         3.26803064e+00,  -5.29083776e+00,   9.75400734e+00,
         4.58790874e+00,  -1.00427608e+01,  -7.15459204e+00,
        -2.96243401e+01,   7.83017159e+00,   2.45643091e+00,
        -2.74667883e+00,  -3.54888380e-01,  -4.29374790e+00,
        -5.53133869e+00,  -1.13860903e+01,  -2.56315575e+01,
        -2.65745964e+01,  -2.29370232e+01,  -1.20304012e+01,
        -1.96615810e+01,  -2.85771084e+01,  -2.80239162e+01,
        -1.76924744e+01,   1.55904627e+00,   3.13529444e+00,
        -1.65762454e-02,  -1.08652706e+01], dtype=float32)]
iter:200, train_loss: 46.008579, test_loss: 614.895020
test prediction parameters results: [array([-55.22853088, -55.81244278, -56.85662842, -56.86396027,
       -55.80954742, -55.79956055, -55.93818665,   7.45238161,
         7.52100801,   7.8257947 ,   7.89442873,   7.09101534,
         8.03168106,   8.2272768 ,   8.09126759,  -5.06623411,
        -4.18448973,  -2.57403922,  -3.58089805,  -4.51126432,
       -13.53684521,  -4.2325778 , -17.98591995, -17.54388046,
       -17.78003311, -17.54518509, -17.71140671,   8.88885975,
        16.17218399,  22.60207748,  24.32943344,  14.50481129,
        25.94131088,  31.73221207,  17.55354691,  24.52309036,
        11.09146595,  30.2763176 ,  15.25742817,   2.19598794,
        -0.92703599,   3.26475167,   2.54810739,  13.71952534,
        14.52134418,  14.38533497,   4.2838254 ,   3.48041177,
        17.30356979,  13.25817585], dtype=float32)]
iter:300, train_loss: 24.743704, test_loss: 247.515228
test prediction parameters results: [array([ -1.93935471e+01,  -1.82407265e+01,  -1.88649864e+01,
        -1.91259232e+01,  -1.89816971e+01,   7.83454478e-01,
        -5.41655481e-01,  -2.20991537e-01,  -1.24651897e+00,
         5.93383238e-03,  -4.98225629e-01,   1.37938607e+00,
        -1.56568451e+01,  -1.61518478e+01,  -1.76980324e+01,
        -1.91704082e+01,  -1.83783169e+01,  -1.78364716e+01,
        -1.88541832e+01,  -1.98915787e+01,  -1.90997505e+01,
        -2.60977507e+00,   4.70958464e-02,  -2.06310675e-01,
         6.66795969e+00,   5.86841536e+00,  -2.08129234e+01,
        -2.02704239e+01,  -2.11081810e+01,  -2.21297741e+01,
        -2.16928272e+01,  -2.15525990e+01,  -2.14675789e+01,
        -2.31580429e+01,  -2.23629074e+01,   1.68113194e+01,
         1.44628878e+01,   1.19538469e+01,   1.26868296e+01,
         1.51799440e+01,   1.33082361e+01,   1.36718950e+01,
         2.17172546e+01,   9.28547764e+00,   1.44304399e+01,
         1.22330980e+01,   1.12890711e+01,   1.25240107e+01,
         1.72763062e+01,   9.34364414e+00], dtype=float32)]
iter:400, train_loss: 0.742687, test_loss: 6.349802
test prediction parameters results: [array([ 1.70644295,  1.0930804 ,  1.07649028,  0.44557276,  0.40208402,
        1.27824867,  1.32717168,  1.15918458,  0.8677932 ,  1.28189027,
        1.13654363,  0.28309932,  0.84515226,  1.13732445,  1.05417216,
        1.36258113,  1.56504285,  1.67202032,  1.08326375,  1.64770973,
        1.0887152 ,  0.38845697,  0.23998228,  0.94741905,  0.98755968,
        1.33757937,  0.28707734,  0.21282473,  0.02682509,  1.00412261,
       -0.02501497,  2.77926493,  6.14523363,  1.2398001 ,  1.83809221,
        0.59776199,  3.18276143,  0.41610327,  2.49752522,  1.44049394,
        5.03889704,  7.86020613,  0.72087038, -0.19327268,  3.08367109,
        1.19006073,  2.39342427,  5.89883852,  1.19811642,  1.26000297], dtype=float32)]
```

```
last test prediction parameters results: 492 0.561594 1.17258 [array([ 2.50898361,  3.3695662 ,  2.92975116,  3.52190113,
  1.67425311,
        1.0001899 ,  1.39521515,  0.46583807,  0.93886149,  0.66025746,
        1.21944511,  1.15006864,  0.63560998,  1.14431965,  4.46614456,
        1.30556715,  0.24360265,  0.78973687,  1.09636652,  1.02307022,
        1.25497925,  0.89395893,  0.12779869,  0.83190882,  0.13971843,
        0.78627765,  1.56708205,  1.50119007,  1.39061224,  1.15775216,
        0.64699399,  3.19177532,  0.51240146,  0.83387554,  2.98102975,
        1.10026038,  0.59335816,  0.7036382 ,  0.54637039, -0.03941809,
        0.38444483,  1.18315506, -0.0112444 ,  0.80344379,  0.58730376,
        0.46000564,  0.78147972,  0.06009926,  0.51942933,  0.96379793], dtype=float32)]
```

printing tv nn trained parameters

In [72]:

```
import numpy as np
np.set_printoptions(threshold=1500)
import pickle
para_names=['W1_1', 'W1_2', 'b1', 'W2_1', 'W2_2', 'b2']
for i in range(len(para_names)):
    print(' ===== printing paramter: '+ para_names[i] +' ===== ')
    print(para_pred[i])
```

```
 ===== printing paramter: W1_1 =====
[[  1.65832207e-01  -3.80816460e-02   8.05306971e-01   1.77482605e+00
    1.66702613e-01   7.75866210e-03   9.69515383e-01  -4.41772699e-01
   -1.28720015e-01   8.32836866e-01   1.50457010e-01  -9.32666600e-01
    2.00296116e+00  -1.74888289e+00  -5.21924019e-01  -9.50068086e-02
   -1.90699071e-01   7.74878114e-02   9.15752500e-02  -2.30156630e-01
    7.46385098e-01  -1.33213326e-01   2.05854750e+00  -1.52027547e-01
   -2.35884219e-01   2.46063963e-01   6.28897011e-01   5.51505834e-02
    1.19075581e-01   8.09076369e-01  -1.88010788e+00  -4.44552392e-01
   -1.86735094e+00   7.07626820e-01  -2.39217982e-01  -9.87914085e-01
   -1.64100274e-01  -2.69697607e-02  -1.93583965e-01  -4.55423929e-02
   -9.71991122e-02   8.33238810e-02   1.83138597e+00  -2.32294157e-01
   -1.81040376e-01  -8.32905024e-02   8.84145647e-02  -1.60633922e-01
   -7.66483992e-02   2.33474448e-01   1.25416413e-01   8.60894859e-01
    2.44112805e-01   2.21760288e-01  -1.14922225e-02  -2.08810878e+00
    3.23756486e-02  -2.42462933e-01   1.79701328e+00  -1.61170125e+00
   -1.98492363e-01  -4.33581471e-02  -3.64799201e-02  -3.31570767e-02
   -1.99321985e-01  -8.09774935e-01   6.63374811e-02  -8.08261335e-02
    2.05087408e-01  -1.75246850e-01  -1.04919240e-01   7.50753224e-01
    1.89583942e-01  -1.83912051e+00  -3.20534259e-02   1.63982773e+00
    3.12112272e-03  -4.25938219e-02   2.18203664e+00]
 [  5.98372668e-02   1.57400951e-01   7.95089304e-01   1.83049166e+00
   -2.16778129e-01  -6.40805364e-02  -1.29862085e-01   1.11900821e-01
   -1.19924009e-01   6.57205462e-01  -1.76265329e-01  -5.09391308e-01
    3.51060957e-01   5.92343695e-02  -6.77658737e-01  -7.22189993e-02
   -9.33442414e-02   1.62454739e-01  -6.68073595e-02  -6.61593527e-02
    4.13852870e-01   1.18775740e-01   2.16295481e+00  -1.49354398e-01
    1.86736152e-01  -1.46516114e-01  -6.48510233e-02  -1.07508600e-01
    3.12289745e-02   4.53507900e-01  -1.86035001e+00  -6.08934462e-01
   -1.96573925e+00   8.22081506e-01   1.26442164e-02  -8.40122104e-01
    1.21377602e-01  -1.53384194e-01   7.34471530e-02   1.84213206e-01
   -1.31698415e-01  -1.07175529e-01   1.89922059e+00   1.78414240e-01
   -2.21570194e-01  -2.46045291e-02   3.00017148e-02  -1.75531060e-01
   -7.34468549e-02  -1.71173826e-01  -1.76490396e-02   4.14533198e-01
   -2.20093995e-01  -2.10430205e-01  -1.17861494e-01  -8.87875915e-01
   -1.75529271e-01   8.79710764e-02   1.03729449e-01   1.34301871e-01
    2.44671330e-01   9.44681615e-02   1.35068849e-01   2.44644150e-01
    2.41809338e-02  -6.90649271e-01  -6.47983253e-02  -2.18852803e-01
    1.47742823e-01   1.87219366e-01  -2.03029513e-01   1.04833432e-01
    2.06530616e-01  -1.69237673e+00   1.01653472e-01   1.66740823e+00
    1.34641781e-01   1.40142843e-01   1.94752300e+00]
 [  1.99743822e-01   1.65557429e-01   7.70209312e-01   2.07976961e+00
   -6.24919236e-02   5.07555753e-02   8.28417122e-01   1.15555763e-01
    2.28146121e-01   5.61599910e-01   2.18057528e-01  -8.94000530e-01
    1.76313710e+00  -2.08552241e+00  -4.46497142e-01   1.72880098e-01
    1.36604980e-01   4.82021272e-03   2.23400071e-01   1.79781929e-01
    7.36436069e-01   6.64068609e-02   2.21317363e+00  -9.82072204e-02
   -1.17856219e-01  -2.88255662e-02   8.04717720e-01  -1.63023144e-01
    1.90086409e-01   7.54577577e-01  -1.82053494e+00  -8.81178439e-01
   -2.00304127e+00   8.14051092e-01   2.14570001e-01  -6.44614041e-01
   -1.37088448e-02   1.96717680e-03  -1.69274658e-02  -9.01228562e-02
   -2.07120031e-02  -1.39455065e-01   1.78244293e+00  -2.14553088e-01
   -1.48414016e-01  -1.68216407e-01   1.07075617e-01   3.88474613e-02
   -3.35123688e-02  -2.22043499e-01  -6.26337081e-02   9.45962012e-01
    1.81361586e-02  -1.27804041e-01   8.60949904e-02  -1.70555556e+00
   -7.80617893e-02   7.93669671e-02   2.07720423e+00  -1.75746119e+00
    1.07437864e-01  -1.55895293e-01  -1.31037831e-02  -2.23523974e-01
   -1.66142568e-01  -7.90767670e-01   1.42851338e-01  -1.76596910e-01
    8.38858932e-02   9.28501785e-03  -1.19250700e-01   6.08482778e-01
    1.99098423e-01  -2.13194799e+00   2.08379254e-01   1.59091496e+00
   -9.89494324e-02  -1.88312978e-02   1.78269386e+00]
 [ -5.16465753e-02  -6.51713610e-02   5.42724550e-01   1.87789631e+00
    2.19598874e-01   1.05036244e-01   1.98373515e-02   5.86669110e-02
    3.90737504e-02   7.06057549e-01   1.46035627e-01  -2.51239181e-01
    5.90624250e-02  -2.18414769e-01  -6.19902432e-01   1.46155939e-01
   -2.13271573e-01   2.29835376e-01  -9.40238982e-02   6.93941563e-02
    3.93258482e-01  -5.19827902e-02   2.07452321e+00  -1.19958460e-01
    1.85917929e-01  -3.02343369e-02   1.14110045e-01   6.77776188e-02
   -1.98173821e-01   8.01005125e-01  -2.06552434e+00  -7.02282012e-01
   -1.99095011e+00   7.10021019e-01  -6.73506260e-02  -8.53107035e-01
   -1.23218343e-01   6.25783056e-02   1.26821205e-01   5.53692728e-02
   -1.63885862e-01   1.92172602e-01   1.78216445e+00   9.96810347e-02
```

```
   1.23711035e-01   1.00031063e-01   1.03992209e-01   2.00636819e-01
   6.03325814e-02   1.21654943e-01   1.24123052e-01   6.20940506e-01
   1.81306854e-01   1.55008510e-01   1.26647636e-01  -9.52168107e-01
   1.11514971e-01   2.35824928e-01   1.88153028e-01   1.07956067e-01
  -8.26299340e-02   2.30863497e-01  -8.69663656e-02   5.52691668e-02
   1.32306680e-01  -8.07411432e-01  -2.17233956e-01   2.44153887e-02
   1.02670476e-01  -1.29392847e-01   1.45017654e-02  -2.43476152e-01
   3.99304479e-02  -1.88197744e+00   1.05963930e-01   1.35323930e+00
  -2.12522238e-01   1.01961806e-01   1.82667053e+00]
[ -2.42025807e-01  -1.63938344e-01   7.93513000e-01   2.08969569e+00
  -2.48257518e-01   2.23669812e-01   5.93917608e-01  -3.30811292e-01
   2.08908185e-01   6.98822916e-01  -1.82998747e-01  -9.98336017e-01
   2.00822186e+00  -1.95373952e+00  -7.10775137e-01   7.58536905e-02
   2.40917310e-01   2.42706403e-01   1.54520437e-01   2.72313207e-02
   4.94607896e-01  -1.38878822e-03   2.03347850e+00  -1.14083171e-01
   1.10485777e-01  -1.80121750e-01   6.81238890e-01  -2.41478026e-01
  -2.28096962e-01   5.23255289e-01  -2.24713111e+00  -8.27742934e-01
  -1.93649137e+00   5.51582932e-01  -1.47385031e-01  -6.27506435e-01
   1.11577824e-01  -7.42694139e-02  -1.89377964e-02   1.60311237e-01
  -1.38045937e-01  -1.66257620e-02   1.92199731e+00   1.47366479e-01
  -1.85515136e-01   7.83349127e-02  -1.07544243e-01   2.11844370e-01
   4.51007038e-02   2.30611190e-01  -1.70078442e-01   8.67760420e-01
   2.18473658e-01  -1.22195899e-01   1.76897720e-01  -1.87077975e+00
   3.55958194e-02   6.67091459e-02   2.00552678e+00  -1.47479534e+00
  -8.41296166e-02  -1.87096328e-01  -1.19729042e-01  -1.78201741e-03
   4.72164601e-02  -7.09782362e-01   2.44002566e-01  -1.32184118e-01
   4.32962328e-02   6.95938319e-02   1.95078596e-01   1.00810456e+00
   9.98334587e-03  -1.82308435e+00   2.03397319e-01   1.31568217e+00
  -9.37646031e-02  -1.69433653e-01   1.99136305e+00]
[  1.98734000e-01  -2.07520336e-01   7.62040436e-01   1.77729142e+00
   8.77459794e-02  -6.84843212e-02  -1.81732699e-01  -2.15905141e-02
  -4.52730060e-03   7.93285489e-01   1.24936178e-01  -2.42609113e-01
   2.86321431e-01  -3.56892757e-02  -6.25729680e-01  -1.97855532e-02
   9.79993194e-02  -1.49022400e-01  -2.36173585e-01   1.77389070e-01
   6.03155673e-01  -1.17942259e-01   1.91175842e+00   9.20329541e-02
   1.08462796e-01  -1.55821934e-01  -1.79308832e-01  -1.15753621e-01
   2.32584432e-01   8.75645399e-01  -1.82752454e+00  -4.18309987e-01
  -1.91138458e+00   5.79760730e-01  -5.86882234e-02  -8.65264773e-01
   4.03691381e-02  -1.93257064e-01  -9.31638628e-02   8.42805058e-02
   1.31028488e-01   1.36670306e-01   2.05683708e+00   2.27890149e-01
   8.03390294e-02  -8.35916102e-02  -1.44411191e-01   6.61476105e-02
   5.55987507e-02   1.77604601e-01  -1.29857808e-02   6.51247442e-01
  -1.78329796e-02   2.41126582e-01   1.02339372e-01  -7.48425901e-01
  -2.14384034e-01  -6.27750605e-02  -1.67072549e-01   1.71575665e-01
   2.18938664e-01   1.35195479e-01  -9.42361206e-02   2.90962011e-02
  -3.15423608e-02  -6.92124128e-01  -1.76347882e-01  -1.95192695e-01
   1.94827840e-01   1.58351704e-01   7.46430308e-02   3.29430215e-02
  -2.30989993e-01  -2.14666200e+00  -3.40204239e-02   1.48633027e+00
  -4.43291962e-02  -1.10546365e-01   1.82728708e+00]
[  2.47395486e-02  -1.41146913e-01   5.36127985e-01   1.81656682e+00
  -7.91093260e-02  -1.60540044e-01   9.35127378e-01  -5.19612730e-01
  -8.00452083e-02   7.52150953e-01  -2.21229702e-01  -9.57548022e-01
   2.00803232e+00  -2.05549669e+00  -5.25620341e-01  -2.23008543e-01
   1.25018016e-01   2.44919285e-01   1.40927032e-01  -2.31062457e-01
   4.43330467e-01  -1.53161883e-02   1.84159052e+00   1.22343078e-01
  -1.32488549e-01   1.35600373e-01   1.02974927e+00  -1.28754988e-01
  -1.22335777e-01   8.31803620e-01  -1.80074120e+00  -5.71040273e-01
  -2.11013460e+00   7.77027607e-01  -6.96264356e-02  -7.31634200e-01
   2.18710378e-01   9.77297127e-03   1.71468899e-01  -2.00667515e-01
  -7.90982991e-02  -2.55399942e-02   1.74409020e+00  -1.48056149e-01
   4.01187092e-02  -9.39923525e-02   2.23829105e-01  -1.68237567e-01
   1.14696130e-01  -8.98406506e-02  -2.13513330e-01   8.64475131e-01
   3.47610861e-02  -1.92757547e-01   1.81139752e-01  -1.69236088e+00
   2.36458912e-01  -6.81556314e-02   1.90141964e+00  -1.75638616e+00
  -2.17769340e-01  -1.94591299e-01   1.12991825e-01  -2.29503170e-01
  -2.16394603e-01  -1.10512722e+00   4.73174602e-02  -1.10007003e-01
   4.77579683e-02   1.93480924e-01   7.63742775e-02   7.79358149e-01
  -9.81553346e-02  -1.89219677e+00   2.18546256e-01   1.48123026e+00
  -2.01522604e-01   1.21654227e-01   2.09296918e+00]
[  4.88108844e-02  -2.23479122e-01   5.20159900e-01   2.16985989e+00
   1.94841698e-01   1.67887494e-01   2.42513418e-01  -2.27177650e-01
   1.19283959e-01   7.66496181e-01  -5.59586883e-02  -2.10270897e-01
   7.03153759e-02   1.97556898e-01  -4.58231151e-01   1.81092322e-03
   1.41440019e-01  -2.36133933e-02  -2.57982314e-02  -1.94191173e-01
   7.48350322e-01   1.69733211e-01   2.19343710e+00   9.70095396e-04
   7.62566775e-02   2.20901921e-01  -2.27576405e-01   1.19634286e-01
  -7.28006959e-02   4.19281483e-01  -2.26857638e+00  -9.04892743e-01
  -2.09752536e+00   6.45044208e-01   6.00980669e-02  -6.32604182e-01
  -4.12534773e-02  -1.85605496e-01  -1.28928438e-01   2.03207985e-01
   9.64839309e-02  -2.35942498e-01   1.89761889e+00   1.80690214e-01
   8.82538706e-02   2.18166575e-01  -2.02491030e-01  -1.21439993e-02
   7.45564252e-02  -7.15183467e-02  -1.51225805e-01   8.37416112e-01
```

```
  -1.07817650e-01  -2.57279724e-02  -1.30387902e-01  -8.41491044e-01
  -4.46397364e-02  -6.87389970e-02  -1.02247164e-01   8.44089836e-02
   1.39078423e-01   7.97907561e-02   2.23377839e-01  -1.75504789e-01
   1.42532095e-01  -6.74598217e-01   1.63279966e-01   1.27359852e-01
   8.96989256e-02  -5.35596013e-02   1.87593654e-01  -8.45186636e-02
   1.99531481e-01  -2.08174205e+00   2.18474373e-01   1.41900420e+00
   3.44755799e-02  -5.59442788e-02   1.94011748e+00]
[  2.24580988e-01   7.47760385e-02   5.48781931e-01   2.03244472e+00
  -1.25134587e-01   1.36192456e-01   5.57309628e-01  -4.02059734e-01
  -8.38777125e-02   7.22632945e-01   4.74984944e-03  -1.18159759e+00
   2.07033110e+00  -1.91379023e+00  -5.48415780e-01   8.14877003e-02
  -2.23216504e-01  -1.57302618e-02  -3.59497517e-02   1.98701277e-01
   8.20307076e-01  -1.33419916e-01   2.01402235e+00  -1.34683043e-01
  -1.36675656e-01  -1.78798005e-01   7.61998713e-01  -9.94783640e-03
  -1.15912229e-01   4.59937751e-01  -1.90016937e+00  -4.50633317e-01
  -2.20042872e+00   8.50255370e-01  -1.33885324e-01  -9.78543937e-01
  -9.97935236e-02   4.32982594e-02   8.85916799e-02   2.21955404e-01
  -8.86744559e-02  -2.23163962e-01   1.96143007e+00   6.56888336e-02
  -1.09513536e-01  -1.32362902e-01   9.94142145e-02   2.38077417e-01
   2.39468709e-01  -1.41422465e-01   8.57525617e-02   9.91379261e-01
   2.16499373e-01   8.99843872e-03   7.80384243e-03  -1.83933902e+00
   2.18803898e-01   3.55481952e-02   1.78079200e+00  -1.39571798e+00
   8.10081810e-02  -2.22835928e-01   1.91507772e-01   1.12401284e-01
  -5.73019832e-02  -1.02857423e+00   2.30411246e-01  -2.11062953e-01
  -1.53314516e-01   6.43906742e-02  -1.75720870e-01   7.28752315e-01
   9.00101811e-02  -1.82150137e+00   4.44370061e-02   1.72306406e+00
   1.98393270e-01   2.30422452e-01   1.74171388e+00]
[  3.35315317e-02  -2.00261295e-01   4.41361010e-01   1.93122768e+00
  -1.13254860e-01   5.87746650e-02   8.52433071e-02  -1.41910195e-01
   3.88981551e-02   8.26233149e-01  -1.19525120e-01  -3.18069786e-01
   6.83243796e-02   1.98837712e-01  -5.94215691e-01  -1.10441893e-01
   2.29535863e-01   2.20813498e-01  -5.02138436e-02   5.31293899e-02
   8.50365520e-01   1.89985201e-01   1.82868600e+00  -1.37991160e-01
   5.20351976e-02   2.31817812e-02   5.62204048e-02   1.55323669e-01
   4.28895205e-02   8.67005944e-01  -2.04996872e+00  -7.07427680e-01
  -1.81214166e+00   6.87319160e-01   1.45348683e-01  -1.05767369e+00
  -1.66074723e-01   1.52120098e-01   2.15937719e-01  -2.33345315e-01
   8.29421431e-02  -1.53337359e-01   2.06907773e+00   3.97217423e-02
   1.65175036e-01  -1.67300880e-01  -1.67859972e-01   3.04127485e-02
  -2.14754045e-03   1.15886182e-02   2.52981931e-02   8.43857706e-01
   1.59735218e-01  -1.32371187e-01  -1.62262559e-01  -9.99698162e-01
  -5.51233739e-02  -8.54948610e-02   1.33370116e-01  -2.02457905e-01
   4.26940173e-02  -5.17508239e-02  -2.26460084e-01   1.89688310e-01
   3.34795862e-02  -1.02600133e+00   2.30506912e-01   9.01462585e-02
   9.71504897e-02  -6.68110847e-02  -1.17655694e-02  -1.05216436e-01
   1.81137308e-01  -2.15753436e+00  -2.21965939e-01   1.73645425e+00
  -1.39575899e-02   4.80156690e-02   1.98715150e+00]
[  4.94881123e-02  -1.70308620e-01   4.48226243e-01   1.85296059e+00
   5.38150519e-02  -7.28237033e-02   9.13368881e-01  -2.18515456e-01
   1.11659959e-01   5.19406855e-01  -1.93690702e-01  -1.04022920e+00
   1.97578025e+00  -1.91511488e+00  -8.96950722e-01   1.01627395e-01
   8.66725594e-02   1.28383800e-01   1.42601982e-01   1.52730092e-01
   6.24239743e-01   2.11054534e-02   2.31283998e+00  -3.98337990e-02
  -2.32195795e-01  -1.37054086e-01   9.44632053e-01  -9.54696685e-02
   2.16888413e-01   4.96093899e-01  -2.06647992e+00  -4.19933289e-01
  -1.83731091e+00   4.29973781e-01  -9.34620649e-02  -7.40004361e-01
   5.35262078e-02  -1.68005079e-01  -2.30394363e-01  -1.69242755e-01
  -2.16639563e-01  -1.18853286e-01   1.86738110e+00   2.27943853e-01
  -2.01162249e-01  -1.49651289e-01   1.43420175e-01   2.05510423e-01
  -9.60174501e-02   2.26354197e-01  -6.08235002e-02   7.11219788e-01
   2.31979772e-01  -9.66338813e-03  -6.50525093e-03  -1.76969278e+00
   8.97295922e-02  -1.34702444e-01   2.19033170e+00  -1.37022543e+00
   7.70850480e-03  -2.42650732e-01   2.38057449e-01  -1.41180074e-02
   2.46882901e-01  -7.69601882e-01  -2.36595303e-01  -4.27200645e-02
  -5.35189211e-02   7.98579305e-02  -1.82068333e-01   6.46944463e-01
   1.44503519e-01  -1.87257004e+00   5.79869300e-02   1.50982702e+00
   1.22032061e-01  -1.74731016e-02   2.10048509e+00]
[ -7.85470158e-02   7.94442445e-02   7.88903058e-01   2.17649484e+00
   6.27371818e-02   1.25136182e-01  -7.06552491e-02  -1.60672575e-01
   1.14628419e-01   7.47509837e-01  -1.91203266e-01  -4.69024450e-01
   3.08548927e-01  -9.93783474e-02  -6.72062993e-01  -2.45581046e-01
   1.72728881e-01  -1.16383165e-01   3.03941220e-02   3.99150401e-02
   7.02263355e-01   2.26273671e-01   2.08604121e+00  -2.32578069e-01
  -1.12208635e-01   1.04177251e-01   2.33449921e-01  -1.93554208e-01
  -1.13604173e-01   4.64447796e-01  -1.97159469e+00  -6.33025825e-01
  -2.09763741e+00   6.72159612e-01   2.34560475e-01  -8.61063361e-01
  -3.78073454e-02  -2.41285548e-01  -2.43609309e-01  -2.38677025e-01
   1.60184279e-01  -2.14125961e-02   1.99291432e+00  -1.94289356e-01
  -3.93716395e-02   1.94652602e-01  -2.03833684e-01  -1.21182993e-01
  -2.44297221e-01   5.80606312e-02   2.02887431e-01   7.13253736e-01
   2.44507045e-02   1.06684491e-01   1.66290149e-01  -1.14336717e+00
  -1.50917619e-02   1.11792371e-01   2.28482902e-01   2.07319587e-01
```

```
 1.84449106e-02  2.16957554e-01  2.02723727e-01  2.43761092e-02
-1.16723359e-01 -8.89851868e-01 -1.98587775e-04 -2.35370100e-02
-1.38528556e-01  2.12978944e-01  1.30783483e-01 -2.28466943e-01
 1.88694671e-01 -2.03438711e+00 -7.36333430e-03  1.31394935e+00
 1.81317881e-01 -4.93582636e-02  2.19464159e+00]
[ -1.25595957e-01 -3.28064412e-02  7.24058926e-01  2.17513490e+00
-7.61313885e-02 -1.29164368e-01  7.50604510e-01 -1.85211465e-01
-2.70011276e-02  5.46583176e-01 -1.66882887e-01 -1.04000962e+00
 1.90265810e+00 -1.85633206e+00 -7.63159573e-01  1.67130157e-01
 1.85635015e-01 -1.26204938e-01 -2.29597524e-01 -2.45071396e-01
 8.24090779e-01  2.11128756e-01  2.10259843e+00 -1.03807077e-01
-1.25709459e-01 -1.92318469e-01  8.82529438e-01 -2.14885682e-01
-5.82580864e-02  7.60739326e-01 -2.17064691e+00 -7.57084787e-01
-2.16580868e+00  7.04222322e-01 -7.78078735e-02 -9.85396922e-01
 1.45089671e-01 -1.36469364e-01 -1.88631520e-01 -3.70616764e-02
-2.42623746e-01  7.76789039e-02  2.03810263e+00  1.68254122e-01
 1.68113574e-01 -2.30614230e-01  8.56870264e-02  1.81767926e-01
 2.49064714e-02  2.00843468e-01  6.52375370e-02  1.00090361e+00
-3.75402123e-02 -1.07422918e-02 -8.65943879e-02 -2.07197809e+00
 1.97455451e-01 -1.28746510e-01  1.95052040e+00 -1.77003074e+00
-2.13284254e-01 -2.32584476e-01  4.10269648e-02 -2.17429966e-01
-1.37036294e-02 -9.71996486e-01 -7.43056983e-02 -2.25893974e-01
-2.08875626e-01  1.55154184e-01 -2.47632593e-01  8.62650931e-01
-1.65523574e-01 -2.14023972e+00 -1.82194710e-02  1.54189014e+00
 2.41634205e-01  1.64473370e-01  2.12765169e+00]
[ -9.33371335e-02 -1.49825364e-02  8.17287385e-01  1.87633002e+00
-2.35703185e-01 -1.36977404e-01 -9.10024568e-02 -1.50579184e-01
 1.11823231e-02  7.79267311e-01  1.19052634e-01 -3.39347392e-01
 5.18095084e-02 -1.29594151e-02 -6.99954331e-01  4.65482026e-02
 2.89423317e-02 -1.89774990e-01 -4.58712876e-03  8.06770176e-02
 5.10839343e-01 -6.17701113e-02  2.05910754e+00 -7.50026107e-02
 2.07782850e-01  7.58775920e-02 -8.90516862e-02 -2.26146758e-01
 1.19420096e-01  6.38470113e-01 -2.24133062e+00 -9.08089817e-01
-2.05861306e+00  8.21075737e-01  2.40227029e-01 -6.98431015e-01
-7.56345689e-03  2.11184636e-01  2.19037995e-01  1.61449537e-01
 2.32203916e-01  1.80073842e-01  2.14501119e+00  2.32999489e-01
-2.05905870e-01  8.25482160e-02  3.74303311e-02 -8.73602629e-02
-1.68634683e-01 -1.31904781e-02 -5.69906831e-02  5.86412013e-01
-1.09526649e-01  1.95028618e-01 -1.37601346e-02 -8.09059680e-01
-9.91362780e-02 -2.21658006e-01  1.74845964e-01 -2.52845734e-01
-9.44007337e-02  1.11949220e-01 -4.12330776e-02  2.43462190e-01
 2.45893076e-01 -7.89607227e-01  1.95943430e-01 -2.43152022e-01
 1.82082206e-02  3.23175341e-02  5.38748354e-02 -1.00461327e-01
-1.75673306e-01 -1.93618572e+00 -1.12472028e-02  1.71626520e+00
-2.21929997e-01 -1.52485847e-01  2.11319017e+00]
[ -4.74598110e-02 -2.35627592e-03  8.34138215e-01  1.75614333e+00
-8.19655210e-02  1.78536102e-01  7.63970494e-01 -2.66726345e-01
 1.22933194e-01  8.69264960e-01 -2.46188894e-01 -8.12436283e-01
 1.74513936e+00 -2.07593298e+00 -8.47225010e-01 -5.67233711e-02
 3.11402529e-02  1.32371917e-01 -5.71863651e-02 -1.16461918e-01
 7.60099292e-01  4.52374369e-02  2.32573414e+00 -7.31184036e-02
-1.04294434e-01  1.03957504e-02  5.10527849e-01 -4.88496125e-02
-2.32197270e-01  5.92722714e-01 -1.84856963e+00 -7.63263464e-01
-2.05993032e+00  4.98147309e-01  2.23859742e-01 -1.01062763e+00
-4.70458716e-02 -1.04197010e-01 -5.03811240e-02 -5.77300712e-02
 7.97072202e-02 -2.35355228e-01  2.17209530e+00  1.17696896e-01
-9.62229818e-02  1.87315896e-01 -2.19109446e-01  1.16000935e-01
 5.98354489e-02 -1.46217257e-01  2.01256469e-01  1.07253671e+00
-1.69604421e-01  2.60750949e-03  5.27078062e-02 -1.87779009e+00
 1.80258825e-01 -2.80939043e-02  2.04377937e+00 -1.66897297e+00
-1.39493316e-01 -9.00940746e-02 -2.12047935e-01  1.30651847e-01
-2.31646940e-01 -8.32295597e-01  7.14588612e-02 -1.47618487e-01
-1.03108853e-01  6.70653433e-02 -2.16359735e-01  5.43956041e-01
 7.44612515e-03 -1.79568124e+00  7.59415179e-02  1.43983710e+00
 4.68558818e-02 -1.82872862e-01  1.79963827e+00]
[  1.94622979e-01 -2.37642899e-01  8.56741846e-01  1.91063797e+00
 1.83790475e-02 -7.40219057e-02  1.90324277e-01 -4.68650348e-02
-8.25954825e-02  5.01434565e-01 -1.97605699e-01 -2.17185602e-01
 1.58781875e-02  6.83183968e-03 -6.49972260e-01 -1.26143873e-01
-2.43484676e-01  1.10342935e-01  1.14380434e-01  2.33091861e-02
 4.69153404e-01  9.14870203e-03  2.12318659e+00 -2.32636243e-01
-1.00004077e-01 -2.31987834e-01  3.70389386e-03 -1.83246315e-01
 2.12926820e-01  8.40033054e-01 -2.25515056e+00 -6.57755554e-01
-1.89280283e+00  4.46189880e-01  1.87600479e-01 -6.97686911e-01
-2.78450996e-02  1.55556455e-01 -1.87395483e-01 -7.03498255e-04
 3.21845859e-02  2.27517113e-01  2.17682338e+00  2.02514276e-01
 1.21617392e-01 -1.16354048e-01  3.86312753e-02  2.10375890e-01
 1.59791902e-01 -2.47291878e-01 -4.88090515e-02  5.74291289e-01
 1.56643018e-01  1.79387942e-01  2.10428759e-01 -9.09958124e-01
 5.70677072e-02  6.81984574e-02  2.22743332e-01 -9.32071209e-02
-1.87218428e-01  2.38599330e-02  7.20964819e-02 -2.57723182e-02
 1.38653681e-01 -8.16857576e-01 -1.06801912e-01 -2.18281120e-01
```

```
   2.05058947e-01   9.24275666e-02  -8.04151595e-03   1.90629154e-01
  -8.45310539e-02  -2.14299870e+00  -2.20113933e-01   1.50934815e+00
   8.98083299e-02   7.46912509e-02   2.16667438e+00]
 [  1.48039207e-01  -2.47560009e-01   5.78125417e-01   2.08300614e+00
  -1.01672694e-01   6.85147494e-02   8.86714339e-01  -2.01018274e-01
   1.82084098e-01   5.62950194e-01   2.28548631e-01  -9.09286022e-01
   2.23585558e+00  -1.89008880e+00  -6.34286582e-01  -2.79946923e-02
   1.86028525e-01  -1.34239390e-01  -1.22295335e-01  -4.82009053e-02
   7.65700519e-01  -8.12218785e-02   2.19931626e+00   1.05684474e-01
  -1.44777358e-01  -1.39325321e-01   6.21287525e-01   6.25997335e-02
   1.19671062e-01   4.79961991e-01  -2.17383242e+00  -4.50469375e-01
  -1.92072439e+00   5.13846457e-01   2.65732259e-02  -7.45604634e-01
   2.16273963e-03   2.99748629e-02  -2.21228048e-01  -8.57152194e-02
  -9.36636776e-02  -1.02602288e-01   2.09868050e+00  -2.34774306e-01
   2.05746070e-01  -1.91474497e-01  -1.00217134e-01   1.99014649e-01
  -5.50734997e-02  -2.16720611e-01  -5.23145497e-02   7.09800541e-01
   8.20207745e-02  -1.76575780e-02   1.20686457e-01  -2.09934020e+00
   1.32370248e-01   1.49859771e-01   2.11364198e+00  -1.42545676e+00
  -1.28256366e-01   1.90185860e-01   9.19551700e-02   1.91688135e-01
  -7.24083185e-03  -9.48736787e-01  -1.31615460e-01  -1.26987010e-01
  -2.36941233e-01  -5.88907152e-02  -1.93257719e-01   8.85663152e-01
   1.28409818e-01  -1.86803734e+00  -9.97711122e-02   1.53746247e+00
  -2.82140374e-02   2.40057155e-01   1.70841599e+00]
 [ -2.41547808e-01   7.96557516e-02   8.27170074e-01   2.08742619e+00
  -6.53181821e-02   1.76709697e-01   1.79825693e-01  -1.28409177e-01
   6.02146834e-02   6.25599265e-01  -1.67369246e-01  -3.41628581e-01
   3.42362732e-01  -1.62365511e-01  -4.53504741e-01   3.47519070e-02
   2.32924417e-01  -2.06670374e-01  -1.95176557e-01  -7.30662942e-02
   4.04561579e-01   8.70470554e-02   1.84728861e+00   1.68247774e-01
   1.06279507e-01   1.07523665e-01  -1.05542786e-01  -3.08556408e-02
   2.48300686e-01   4.12614107e-01  -1.87030566e+00  -6.24496520e-01
  -2.28524184e+00   6.57202780e-01   1.15944132e-01  -5.96607208e-01
  -1.64974183e-01   1.65468320e-01   5.64086884e-02  -7.23413825e-02
   7.81549960e-02  -2.41039276e-01   1.71054924e+00   3.00304741e-02
  -1.95375800e-01   1.33419141e-01   1.52862385e-01   4.09836322e-02
  -5.94470948e-02   1.43764630e-01   6.32874519e-02   6.43073261e-01
  -1.78028643e-02  -7.00569302e-02  -5.28255701e-02  -1.08873582e+00
   8.34128410e-02   2.10774228e-01  -1.97789147e-01   3.09987236e-02
  -1.35915533e-01  -5.10884076e-02  -1.34903401e-01   2.07945898e-01
   1.85363933e-01  -1.11016691e+00   1.91562667e-01  -2.27307081e-02
  -4.26879227e-02   1.25987843e-01   2.33195946e-01  -1.66292656e-02
  -1.43369347e-01  -1.90149057e+00  -1.31410241e-01   1.41556001e+00
   2.28107110e-01  -2.08259165e-01   1.99290478e+00]]
 ===== printing paramter: W1_2 =====
[[ 0.02530299  0.10664599  0.75140953 ..., -0.03674847 -0.18071637
   1.88281071]
 [-0.06630699  0.02294751  0.55038959 ..., -0.0719011  -0.02760842
   1.77603829]
 [ 0.13194145  0.08225454  0.51377738 ...,  0.03874175 -0.17059845
   1.93120897]
 ...,
 [ 0.02974263 -0.09989332  0.13426502 ...,  0.06340556 -0.1886238
   0.01122489]
 [ 0.06979986 -0.08174054 -0.0568376  ...,  0.04328896 -0.155077
   1.75352585]
 [-0.02668688 -0.06067537  0.53500485 ..., -0.09523957  0.01341355
   1.80642748]]
 ===== printing paramter: b1 =====
[[ 0.13687319  0.17148703  0.69647878  1.75060809  0.22554955 -0.0064162
  -0.0184671  -0.24005878  0.00568217  0.8563655   0.23870921 -0.29065374
   0.17701834 -0.28736883 -0.7496897  -0.14680375 -0.13777561 -0.22236285
   0.05340073 -0.21950448  0.58757567  0.20932561  1.98077786  0.0311892
  -0.03346266 -0.11549847  0.03857461 -0.21707445 -0.01456001  0.90722561
  -2.13461113 -0.75945801 -1.78550851  0.82460248  0.09916598 -0.90579152
   0.21888503 -0.15578601  0.09826031 -0.14759701  0.11455441 -0.01027551
   2.09978795 -0.23110932  0.2585597   0.21885037  0.17419869  0.14655173
  -0.20838152  0.09218395 -0.10508348  0.83441991  0.20921668 -0.00639179
  -0.08305596 -1.00351059  0.1095348  -0.03247124  0.05970433  0.0780407
  -0.24636836  0.13756457 -0.08918247 -0.19076183 -0.02166271 -0.8275488
   0.00632507  0.07047081  0.1113776   0.23781842 -0.15508422  0.01934866
   0.16854849 -2.03269362 -0.03828563  1.27750325 -0.23667207 -0.18786973
   1.98043132]]
 ===== printing paramter: W2_1 =====
[[  3.33816916e-01]
 [  1.36356831e-01]
 [  2.43870378e-01]
 [  9.27413988e-04]
 [  1.09346345e-01]
 [ -1.43763453e-01]
 [ -2.19639614e-02]
 [  3.04999924e+00]
 [  6.65651858e-02]
```

```
 [ -3.91194224e-03]
 [  2.87218988e-01]
 [ -1.11604556e-01]
 [ -3.03706259e-01]
 [  9.48012173e-02]
 [  2.07673490e-01]
 [ -2.63090163e-01]
 [  2.06208348e-01]
 [ -1.80283248e-01]
 [ -2.93109447e-01]
 [  1.54309630e-01]
 [  1.46588326e-01]
 [ -1.40121922e-01]
 [ -1.83752477e-01]
 [ -5.04435897e-02]
 [ -2.42155924e-01]
 [ -4.82445061e-02]
 [  7.89219141e-03]
 [ -2.73755074e-01]
 [  4.82209027e-02]
 [ -9.20683295e-02]
 [  1.81924522e-01]
 [  2.47912645e-01]
 [  2.85911500e-01]
 [ -1.64748117e-01]
 [  1.43802464e-01]
 [  2.34588385e-01]
 [  2.77402997e-02]
 [  2.82362342e-01]
 [ -1.89830899e-01]
 [  2.64611304e-01]
 [  2.20568478e-01]
 [  4.72953022e-02]
 [  1.78688258e-01]
 [  9.58612561e-03]
 [  1.02572680e-01]
 [  3.31576467e-02]
 [  2.03575134e-01]
 [ -1.98973119e-01]
 [  9.50732827e-02]
 [ -9.72243845e-02]
 [ -2.73980260e-01]
 [ -1.10217527e-01]
 [  1.16249740e-01]
 [  2.47378051e-01]
 [  3.21474075e-01]
 [  2.32860163e-01]
 [  4.58871685e-02]
 [ -4.59330231e-01]
 [ -4.93214041e-01]
 [ -3.54674459e-02]
 [ -9.60324833e-04]]
 ===== printing paramter: W2_2 =====
[[ -1.81038320e-01]
 [ -1.85704321e-01]
 [  3.04802150e-01]
 [ -1.66571605e+00]
 [ -1.59739077e-01]
 [  5.56683987e-02]
 [ -2.49775127e-01]
 [ -1.18146546e-01]
 [  1.66695386e-01]
 [  9.30079520e-02]
 [ -2.00534135e-01]
 [  2.60881279e-02]
 [ -1.52657270e-01]
 [ -1.00202060e+00]
 [ -4.97798353e-01]
 [ -7.68049732e-02]
 [ -4.53725755e-02]
 [  4.16269228e-02]
 [ -3.66294414e-01]
 [  2.40183651e-01]
 [  1.73492819e-01]
 [  1.99483633e-01]
 [  1.33802056e-01]
 [  2.06851304e-01]
 [ -1.14606947e-01]
 [  1.59115911e-01]
 [ -6.73852682e-01]
 [  2.09876597e-02]
 [  3.93256592e-03]
```

```
 [  2.38760397e-01]
 [ -1.89198300e-01]
 [ -6.30752265e-01]
 [ -1.16403967e-01]
 [  8.53191614e-02]
 [ -1.93640709e-01]
 [  4.14185464e-01]
 [ -3.56203169e-02]
 [ -1.73160836e-01]
 [ -2.65002638e-01]
 [  1.47656405e+00]
 [ -1.00735724e-02]
 [ -2.00575590e-03]
 [ -1.26206255e+00]
 [  4.31252837e-01]
 [ -2.29623690e-01]
 [  4.00821865e-02]
 [ -3.26730996e-01]
 [ -1.19020484e-01]
 [  5.21979183e-02]
 [  2.32781768e-01]
 [  1.40278965e-01]
 [  4.32450436e-02]
 [ -2.73148447e-01]
 [ -3.36600989e-02]
 [  8.17259401e-02]
 [  8.68587121e-02]
 [ -1.64684355e-01]
 [  1.11448050e-01]
 [  1.25419766e-01]
 [  1.22124457e+00]
 [  5.31361401e-02]
 [ -9.97697562e-02]
 [  1.42760485e-01]
 [  7.84369409e-02]
 [ -6.28681630e-02]
 [ -2.14106157e-01]
 [  9.37047228e-02]
 [ -2.08912358e-01]
 [ -6.94034547e-02]
 [ -3.12658161e-01]
 [ -8.83209854e-02]
 [  1.00811529e+00]
 [ -2.84144789e-01]
 [ -1.78060782e+00]
 [  4.52728420e-02]
 [  5.71304131e+00]
 [ -7.24753886e-02]
 [ -3.68157685e-01]
 [ -1.08535910e+00]]
 ===== printing paramter: b2 =====
[[ 0.06405534]]
```

plug in parameters from tv nn and train optimization nn

```
simulation note:
- ignored constant vector for the optimization, i.e. set as zero vector
- price_focal trained here has length 61 as in tv_nn
```

In [76]:

```python
import time
#optimize the loss funciton by adding negative sign
start_time= time.time()
tf.reset_default_graph()
p_f = tf.get_variable('p_f', shape=(1, 61))
W1_1 = tf.constant(paras[0], dtype=tf.float32, name='W1_1' )
W1_2 = tf.constant(paras[1], dtype=tf.float32, name='W1_2' )
b1 = tf.constant(paras[2], dtype=tf.float32, name='b1' )
W2_1 = tf.constant(paras[3], dtype=tf.float32, name='W2_1' )
W2_2 = tf.constant(paras[4], dtype=tf.float32, name='W2_2' )
b2 = tf.constant(paras[5], dtype=tf.float32, name='b2' )
p_sub =  tf.placeholder(shape=(None, 18), dtype=tf.float32)
D = tf.matmul(tf.nn.sigmoid(tf.matmul(p_f, W1_2) + tf.matmul(p_sub, W1_1)+b1), W2_2)+tf.matmul(p_f, W2_1)+b2

p_loss = tf.reduce_mean(-p_f*D) #self defined loss

###parameters
batch_size = 50
learning_rate = 0.5
trainer = tf.train.AdamOptimizer(learning_rate=learning_rate)
gradients = trainer.compute_gradients(p_loss)
optimizer = trainer.apply_gradients(gradients)

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for step in range(493):
        #parameters
        x1_train = tv_train.iloc[step:step+batch_size, 66:84].values
        x1_test = tv_test.iloc[step:step+batch_size, 66:84].values


        loss_tr, _ = sess.run([p_loss, optimizer], feed_dict={p_sub: x1_train})
        _test_loss = sess.run([p_loss], feed_dict={p_sub: x1_test})

        # predict
        D_pred_tr = sess.run([D], feed_dict={p_sub:x1_train})
        D_pred_te = sess.run([D], feed_dict={p_sub:x1_test})
        pf_pred_tr = sess.run([p_f], feed_dict={p_sub:x1_train})
        pf_pred_te = sess.run([p_f], feed_dict={p_sub:x1_test})

        if step % 100 == 0:
            print("iter:%d, train_loss: %f, test_loss: %f"%(step, loss_tr,  _test_loss[0]))
            print("simulated train_revenue: %f, simulated test_revenue: %f"%(-loss_tr,  -_test_loss[0]))
            print('train D prediction parameters results:', D_pred_tr[0][0])
            print('test D prediction parameters results:', D_pred_te[0][0])
            print('train p_f prediction parameters results:', pf_pred_tr[0][0])
            print('test p_f prediction parameters results:', pf_pred_te[0][0])

end_time=time.time()

print('trained total elasped time:', end_time-start_time)
```

```
iter:0, train_loss: 0.015254, test_loss: -1.931153
simulated train_revenue: -0.015254, simulated test_revenue: 1.931153
train D prediction parameters results: [ 4.08376312]
test D prediction parameters results: [ 4.08376312]
train p_f prediction parameters results: [ 0.27519163  0.71603501  0.40172035  0.2465378   0.65532756  0.72448307
  0.26340601  0.68892837  0.73887032  0.30339321  0.74439216  0.25574148
  0.58710772  0.22948661  0.37563109  0.48797232  0.25176153  0.32835129
  0.78147292  0.3213383   0.33693913  0.5167436   0.58838892  0.80850506
  0.44930342  0.28909355  0.59339476  0.48936957  0.23207715  0.70006233
  0.23459548  0.38393834  0.7051819   0.5100767   0.45062947  0.58456802
  0.54376048  0.52921164  0.31723985  0.60836881  0.74959129  0.61383998
  0.48322216  0.39377356  0.70933682  0.28857201  0.48687404  0.63943958
  0.38513982  0.4271408   0.6301102   0.28774959  0.75927979  0.2149156
  0.71745455  0.64149898  0.55777717  0.26488853  0.53295612  0.6465013
  0.34597835]
test p_f prediction parameters results: [ 0.27519163  0.71603501  0.40172035  0.2465378   0.65532756  0.72448307
  0.26340601  0.68892837  0.73887032  0.30339321  0.74439216  0.25574148
  0.58710772  0.22948661  0.37563109  0.48797232  0.25176153  0.32835129
  0.78147292  0.3213383   0.33693913  0.5167436   0.58838892  0.80850506
  0.44930342  0.28909355  0.59339476  0.48936957  0.23207715  0.70006233
  0.23459548  0.38393834  0.7051819   0.5100767   0.45062947  0.58456802
  0.54376048  0.52921164  0.31723985  0.60836881  0.74959129  0.61383998
  0.48322216  0.39377356  0.70933682  0.28857201  0.48687404  0.63943958
  0.38513982  0.4271408   0.6301102   0.28774959  0.75927979  0.2149156
  0.71745455  0.64149898  0.55777717  0.26488853  0.53295612  0.6465013
  0.34597835]
iter:100, train_loss: -32877.410156, test_loss: -33660.800781
simulated train_revenue: 32877.410156, simulated test_revenue: 33660.800781
```

```
train D prediction parameters results: [ 659.20294189]
test D prediction parameters results: [ 659.63519287]
train p_f prediction parameters results: [ 69.28001404  69.73469543  69.41191101  69.27007294  69.67552185
  68.87541199  69.28619385  69.67295837  69.76075745  69.32836914
  69.75111389  69.03865051 -68.30260468  69.24838257  69.38613892
 -68.34574127  69.2615509   66.04453278 -68.09355927  69.33564758
  69.35146332  68.78456879  65.84838104  69.82216644 -68.25084686
  69.30109406  69.61952209 -68.36791229  69.25392914  69.62873077
  69.24595642  69.39058685  69.71393585  67.5896759   69.46438599
  69.59360504  69.56583405  69.53381348  64.72014618  69.61633301
  69.76107025  69.63965607  69.49585724  69.41732025  69.73049927
  69.31558228  69.49734497  62.87686157  69.40433502  69.32061005
 -68.23214722  69.09980011  69.7755127   69.22385406  69.72133636
  69.65061188  69.57968903 -68.67276764 -68.40973663  69.66360474
  69.36969757]
test p_f prediction parameters results: [ 69.28001404  69.73469543  69.41191101  69.27007294  69.67552185
  68.87541199  69.28619385  69.67295837  69.76075745  69.32836914
  69.75111389  69.03865051 -68.30260468  69.24838257  69.38613892
 -68.34574127  69.2615509   66.04453278 -68.09355927  69.33564758
  69.35146332  68.78456879  65.84838104  69.82216644 -68.25084686
  69.30109406  69.61952209 -68.36791229  69.25392914  69.62873077
  69.24595642  69.39058685  69.71393585  67.5896759   69.46438599
  69.59360504  69.56583405  69.53381348  64.72014618  69.61633301
  69.76107025  69.63965607  69.49585724  69.41732025  69.73049927
  69.31558228  69.49734497  62.87686157  69.40433502  69.32061005
 -68.23214722  69.09980011  69.7755127   69.22385406  69.72133636
  69.65061188  69.57968903 -68.67276764 -68.40973663  69.66360474
  69.36969757]
iter:200, train_loss: -159975.890625, test_loss: -161748.203125
simulated train_revenue: 159975.890625, simulated test_revenue: 161748.203125
train D prediction parameters results: [ 1443.44458008]
test D prediction parameters results: [ 1444.49743652]
train p_f prediction parameters results: [ 152.14485168  152.58067322  152.26750183  152.09127808  152.51397705
  151.51593018  152.10157776  152.5652771   152.59588623  152.14717102
  152.61070251  151.76763916 -151.40153503  152.09075928  152.24156189
 -151.58267212  152.11987305  148.34356689 -151.22523499  152.18882751
  152.20075989  151.44154358  148.09901428  152.60592651 -151.7318573
  152.09544373  152.43670654 -151.55334473  152.08532715  152.36798096
  152.10215759  152.25216675  152.5723877   150.11952209  152.31576538
  152.4513855   152.3928833   152.39660645  146.74775696  152.47673035
  152.61579895  152.47125244  152.34628296  152.23829651  152.56600952
  152.14430237  152.35534668  144.37252808  152.24461365  152.07125854
 -151.39933777  151.81254578  152.62142944  152.08140564  152.58750916
  152.50968933  152.4115448  -151.64305115 -151.36729431  152.46876526
  152.18606567]
test p_f prediction parameters results: [ 152.14485168  152.58067322  152.26750183  152.09127808  152.51397705
  151.51593018  152.10157776  152.5652771   152.59588623  152.14717102
  152.61070251  151.76763916 -151.40153503  152.09075928  152.24156189
 -151.58267212  152.11987305  148.34356689 -151.22523499  152.18882751
  152.20075989  151.44154358  148.09901428  152.60592651 -151.7318573
  152.09544373  152.43670654 -151.55334473  152.08532715  152.36798096
  152.10215759  152.25216675  152.5723877   150.11952209  152.31576538
  152.4513855   152.3928833   152.39660645  146.74775696  152.47673035
  152.61579895  152.47125244  152.34628296  152.23829651  152.56600952
  152.14430237  152.35534668  144.37252808  152.24461365  152.07125854
 -151.39933777  151.81254578  152.62142944  152.08140564  152.58750916
  152.50968933  152.4115448  -151.64305115 -151.36729431  152.46876526
  152.18606567]
iter:300, train_loss: -384427.062500, test_loss: -387221.312500
simulated train_revenue: 384427.062500, simulated test_revenue: 387221.312500
train D prediction parameters results: [ 2232.00585938]
test D prediction parameters results: [ 2232.37768555]
train p_f prediction parameters results: [ 235.51501465  235.94140625  235.62705994  235.42498779  235.85948181
  234.65940857  235.445755    235.95098877  235.92758179  235.48995972
  235.96792603  235.00457764 -234.95266724  235.43190002  235.60430908
 -235.18969727  235.48774719  231.34373474 -234.76849365  235.54992676
  235.54328918  234.60757446  230.91267395  235.89376831 -235.65081787
  235.39021301  235.76940918 -235.19819641  235.41856384  235.63938904
  235.46220398  235.61714172  235.93157959  233.19287109  235.66438293
  235.80973816  235.71214294  235.75349426  229.64103699  235.83573914
  235.97291565  235.82229614  235.69363403  235.55479431  235.89616394
  235.49435425  235.71932983  226.82846069  235.58207703  235.33187866
 -234.93113708  235.08920288  235.96531677  235.4354248   235.95716858
  235.87171936  235.73579407 -235.0927124  -234.79588318  235.79547119
  235.4984436 ]
test p_f prediction parameters results: [ 235.51501465  235.94140625  235.62705994  235.42498779  235.85948181
  234.65940857  235.445755    235.95098877  235.92758179  235.48995972
  235.96792603  235.00457764 -234.95266724  235.43190002  235.60430908
 -235.18969727  235.48774719  231.34373474 -234.76849365  235.54992676
  235.54328918  234.60757446  230.91267395  235.89376831 -235.65081787
  235.39021301  235.76940918 -235.19819641  235.41856384  235.63938904
  235.46220398  235.61714172  235.93157959  233.19287109  235.66438293
```

```
  235.80973816  235.71214294  235.75349426  229.64103699  235.83573914
  235.97291565  235.82229614  235.69363403  235.55479431  235.89616394
  235.49435425  235.71932983  226.82846069  235.58207703  235.33187866
 -234.93113708  235.08920288  235.96531677  235.4354248   235.95716858
  235.87171936  235.73579407 -235.0927124  -234.79588318  235.79547119
  235.4984436 ]
iter:400, train_loss: -702253.687500, test_loss: -705878.625000
simulated train_revenue: 702253.687500, simulated test_revenue: 705878.625000
train D prediction parameters results: [ 3012.96826172]
test D prediction parameters results: [ 3013.19726562]
train p_f prediction parameters results: [ 318.07928467  318.48495483  318.19143677  317.96038818  318.41647339
  317.1362915   317.94943237  318.53979492  318.48617554  318.01037598
  318.54168701  317.51049805 -317.65673828  317.98968506  318.16567993
 -318.05651855  318.04452515  313.49377441 -317.52142334  318.10037231
  318.1105957   317.08352661  313.18582153  318.42410278 -318.50909424
  317.91738892  318.30752563 -317.95700073  317.97576904  318.12695312
  318.02053833  318.18280029  318.49780273  315.58746338  318.22741699
  318.37643433  318.27539062  318.33078003  311.53640747  318.40219116
  318.53518677  318.35192871  318.26428223  318.11065674  318.46353149
  318.01873779  318.27584839  308.31829834  318.14419556  317.83032227
 -317.77496338  317.53726196  318.52856445  318.00549316  318.52346802
  318.43377686  318.29751587 -317.72537231 -317.43667603  318.31292725
  318.0524292 ]
test p_f prediction parameters results: [ 318.07928467  318.48495483  318.19143677  317.96038818  318.41647339
  317.1362915   317.94943237  318.53979492  318.48617554  318.01037598
  318.54168701  317.51049805 -317.65673828  317.98968506  318.16567993
 -318.05651855  318.04452515  313.49377441 -317.52142334  318.10037231
  318.1105957   317.08352661  313.18582153  318.42410278 -318.50909424
  317.91738892  318.30752563 -317.95700073  317.97576904  318.12695312
  318.02053833  318.18280029  318.49780273  315.58746338  318.22741699
  318.37643433  318.27539062  318.33078003  311.53640747  318.40219116
  318.53518677  318.35192871  318.26428223  318.11065674  318.46353149
  318.01873779  318.27584839  308.31829834  318.14419556  317.83032227
 -317.77496338  317.53726196  318.52856445  318.00549316  318.52346802
  318.43377686  318.29751587 -317.72537231 -317.43667603  318.31292725
  318.0524292 ]

trained total elasped time: 2.4205403327941895
```

In [ ]: