In [29]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [30]:

```
X = np.linspace(-3, 3, num=100)
Y = np.log(np.abs(X ** 2 - 1) + .5)
```

In [35]:

```
plt.style.use('seaborn')
```

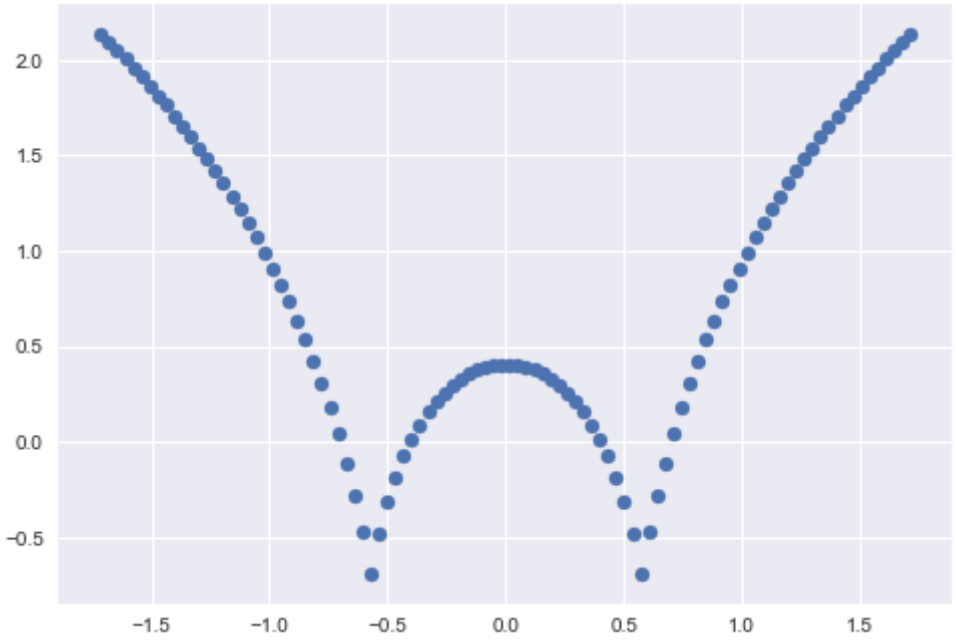
In [32]:

```
X = X.reshape((-1,1))
Y = Y.reshape((-1,1))
print(X.shape)
print(Y.shape)
```

```
(100, 1)
(100, 1)
```

In [36]:

```
# Normailze
u = X.mean()
std = X.std()
X = (X-u)/std

# Visualize
plt.scatter(X,Y)
plt.show()
```



In [41]:

```
def getWeightMatrix(query_point, X, tau):
    M = X.shape[0]
    W = np.mat(np.eye(M))

    for i in range(M):
        xi = X[i]
        x = query_point
        W[i,i] = np.exp(np.dot((xi-x),(xi-x).T)/(-2*tau*tau))

    return W
```

In [45]:

```
X = np.mat(X)
Y = np.mat(Y)
M = X.shape[0]

W = getWeightMatrix(-1,X,0.5)
print(W.shape)
print(W)
```

```
(100, 100)
[[3.59901841e-01 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 3.96423807e-01 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 4.34560818e-01 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 ...
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 8.33473660e-07
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  5.76288304e-07 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 3.96554525e-07]]
```

In [61]:

```
# Make Prediction
def predict(X,Y,query_X,tau):
    ones = np.ones((M,1))
    X_ = np.hstack((X,ones))

    qx = np.mat([query_X,1])

    W = getWeightMatrix(qx,X_,tau)

    theta = np.linalg.pinv(X_.T*(W*X_))*(X_.T*(W*Y))
    pred = np.dot(qx,theta)
    return theta, pred
```

In [65]:

```
theta, pred = predict(X,Y,1,1)
print(theta)
print(pred)
```

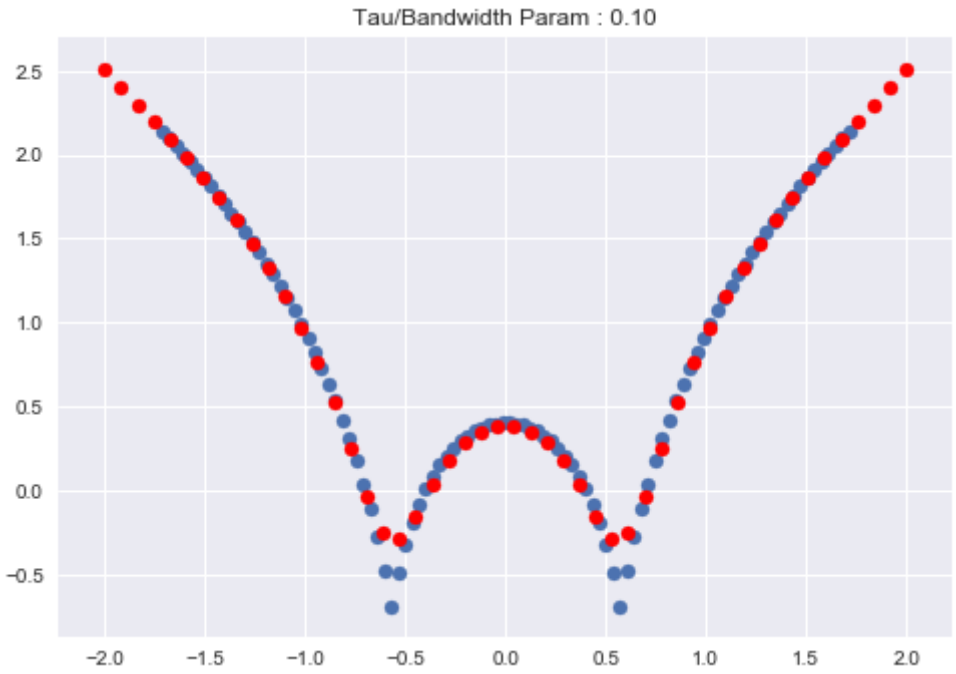
```
 [[0.65529605]
  [0.3017585 ]]
 [[0.95705455]]
```

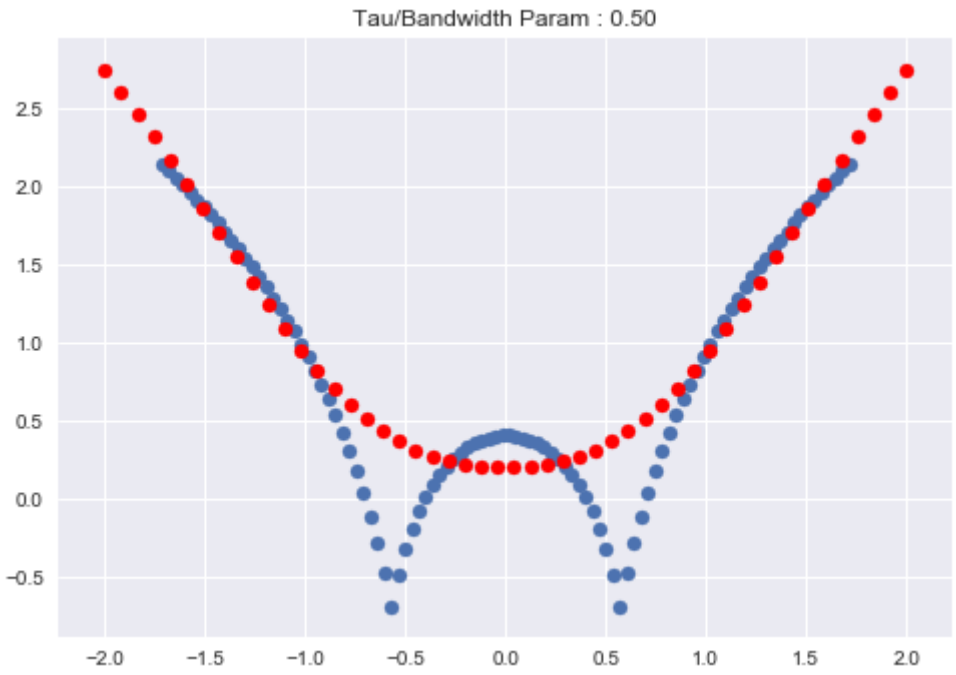
In [75]:

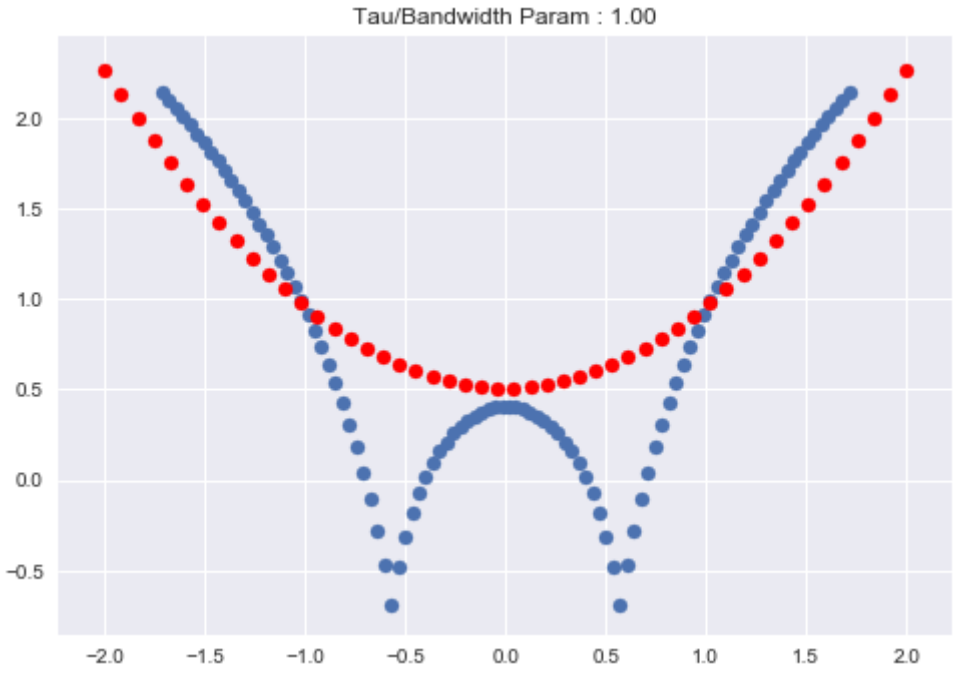
```
def plotPrediction(tau):
    X_text = np.linspace(-2,2,50)
    Y_test = []

    for xq in X_text:
        theta, pred = predict(X,Y,xq,tau)
        Y_test.append(pred[0][0])

    Y_test = np.array(Y_test)

    XO = np.array(X)
    YO = np.array(Y)

    plt.title("Tau/Bandwidth Param : %.2f"%tau)
    plt.scatter(XO,YO)
    plt.scatter(X_text,Y_test, color='red')
    plt.show()
```

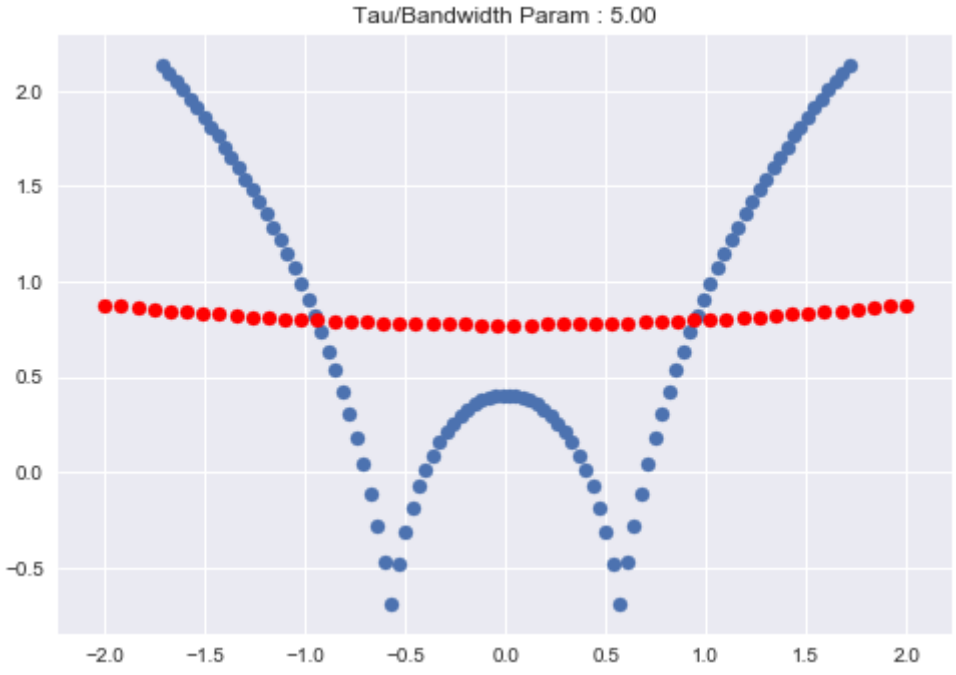
In [76]:

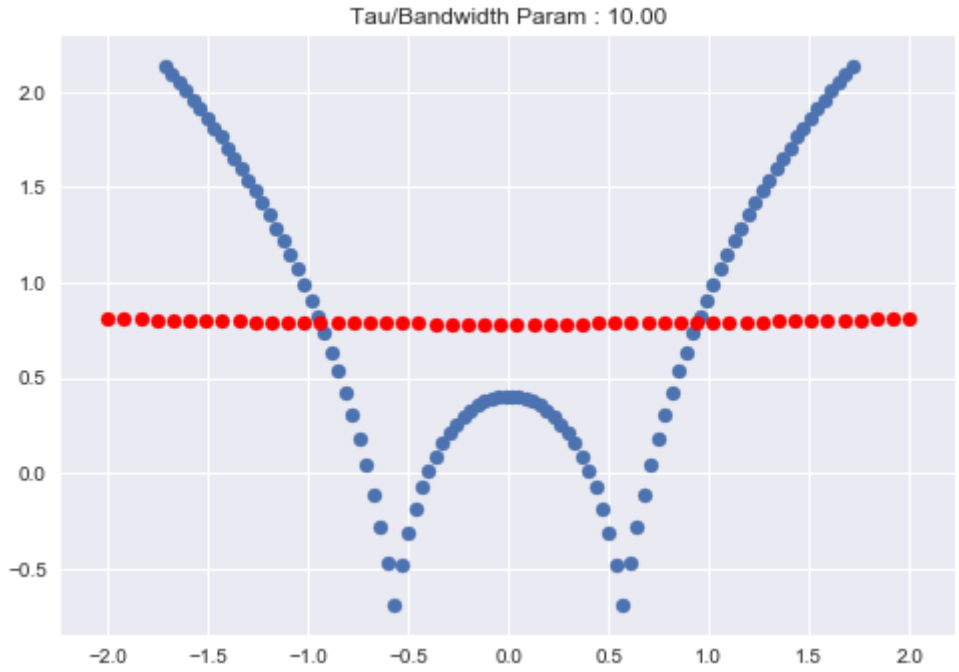
```
taus = [0.1, 0.5, 1, 5, 10]
for t in taus:
    plotPrediction(t)
```






Tau/Bandwidth Param : 1.00
2.0
1.5
1.0
0.5
0.0
−0.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Tau/Bandwidth Param : 5.00
2.0
1.5
1.0
0.5
0.0
−0.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Tau/Bandwidth Param : 10.00
2.0
1.5
1.0
0.5
0.0
−0.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

In [ ]: