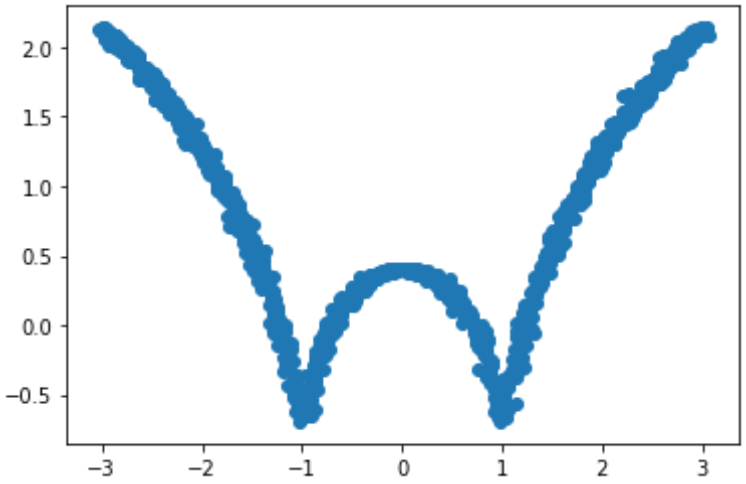
In [50]:
```
import numpy as np
import matplotlib.pyplot as pltt
```

In [53]:
```
x=np.linspace(-3,3,1000)
y=np.log(np.abs((x**2)-1)+0.5)

#adding Jitter
x+=np.random.normal(scale=0.05,size=1000)
```

In [54]:
```
plt.scatter(x,y)
```

Out[54]:
```
<matplotlib.collections.PathCollection at 0x28d92e562b0>
```



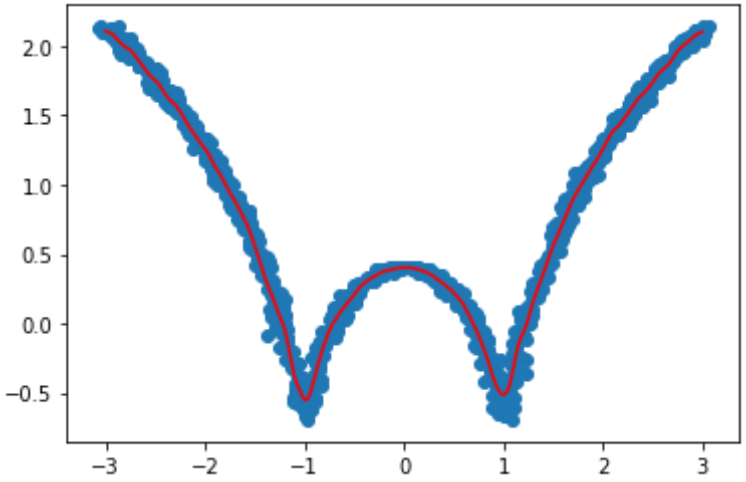
In [44]:
```
def local_reg(x0,x,y,tau):
    # add bias term
    x0 = np.r_[1, x0]
    x = np.c_[np.ones(len(x)), x]

    # fit model: normal equations with kernel
    xw = x.T * radial_kernel(x0, x, tau)
```

```
    beta = np.linalg.pinv(xw @ x) @ xw @ y

    # predict value
    return x0 @ beta


def radial_kernel(x0, x, tau):
    return np.exp(np.sum((x - x0) ** 2, axis=1) / (-2 * tau * tau))
```

In [45]:
```
def plot_lwr(tau):
    # prediction
    domain = np.linspace(-3, 3, num=300)
    prediction = [local_reg(x0, x, y, tau) for x0 in domain]
    plt.scatter(x,y)
    plt.plot(domain, prediction, color='red')
    return plt
```

In [46]:
```
plot_lwr(0.04)
```

Out[46]:
```
<module 'matplotlib.pyplot' from 'C:\\Users\\perum\\Anaconda3\\lib\\site-packages\\matplotlib\\pyplot.py'>
```



In [ ]:
```
https://medium.com/100-days-of-algorithms/day-97-locally-weighted-regre
```

```
ssion-c9cfaff087fb
https://towardsdatascience.com/kernel-function-6f1d2be6091
```