Q2

February 11, 2018

```
<IPython.core.display.HTML object>
```

0.1 Locally Weighted Linear Regression

In this problem, we will generalize the ideas of linear regression to implement locally weighted linear regression where we want to weigh different training examples differently.

This is the initial loss with parameters equal to zero.

```
Out[21]: 121.390984336008
```

We obtain the parameters learned by unweighted linear regression implemented using normal equation, the solution being

$$\theta = (X^T X)^{-1} X^T Y$$

The first output is the **weight** and second is the **bias**.

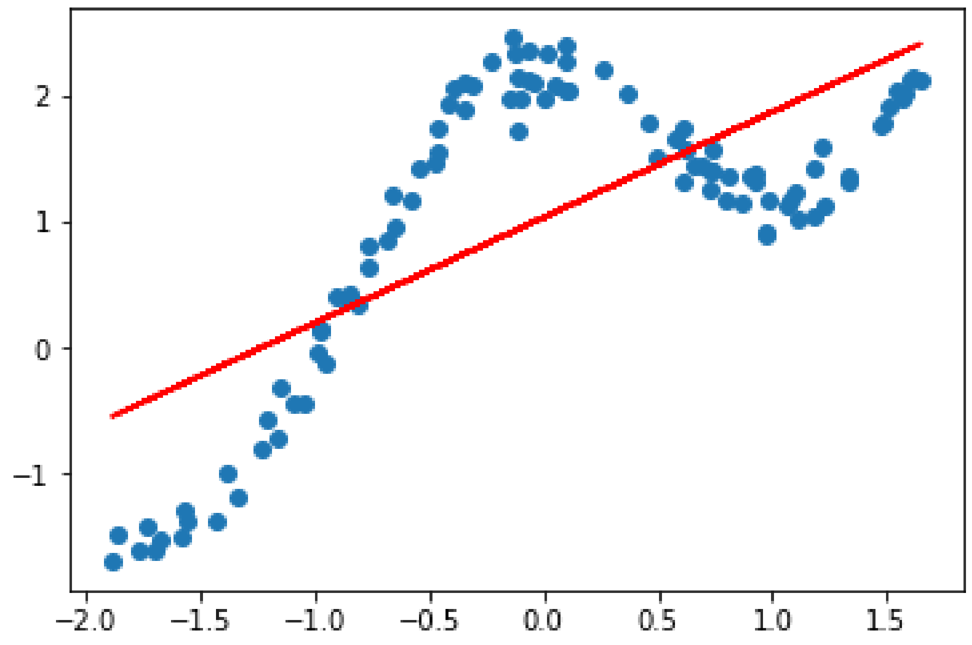
```
Out[40]: array([ 0.83519315,  1.03128116])
```

This is the new loss, it is apparent that the loss has decreased but not by too much.

```
Out[42]: 33.336562701909685
```

This is the plot of unweighted linear regression output and the data, it is clear that our regression has missed important underlying patterns in the data.

```
Out[43]: [<matplotlib.lines.Line2D at 0x7f82fb8d0ef0>]
```

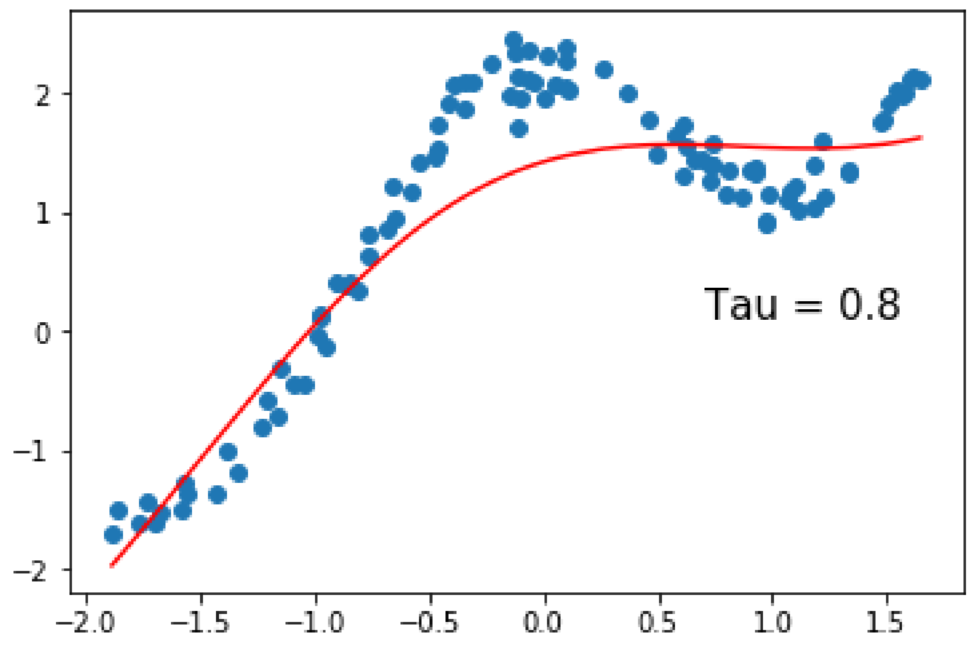


After this we implement locally weighted linear regression. This is not learning per se, as the parameters get learnt at inference time. I did the derivation and found that the normal equation solution for locally weighted LR is

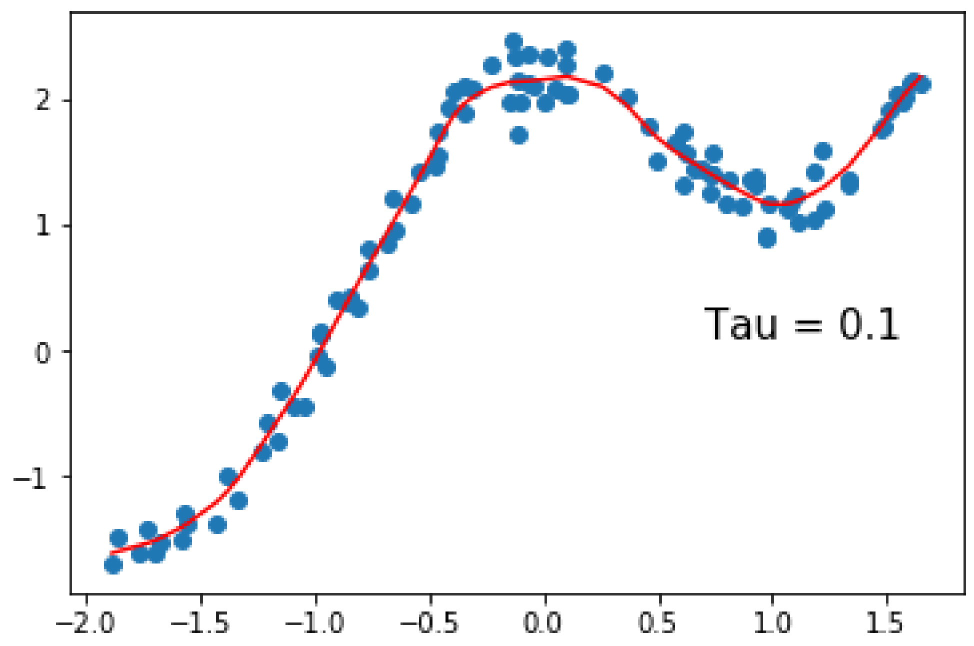
$$\theta = (X^T W X)^{-1} X^T W Y$$

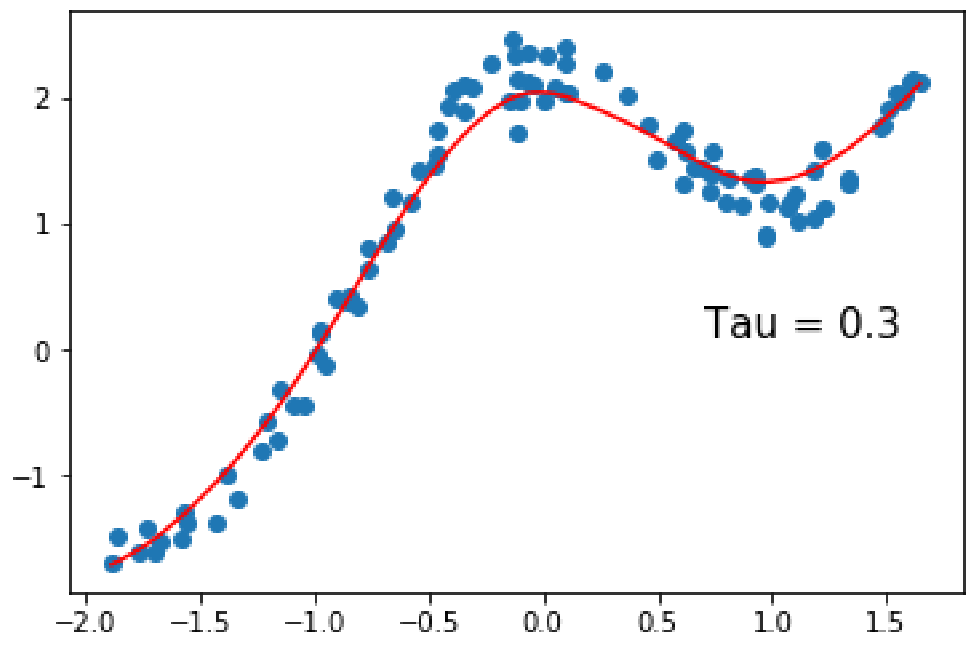
Below is the plot of the output we obtain for locally weighted LR with bandwidth parameter $\tau = 0.8$.

```
Out[49]: Text(0.7,0.1,'Tau = 0.8')
```

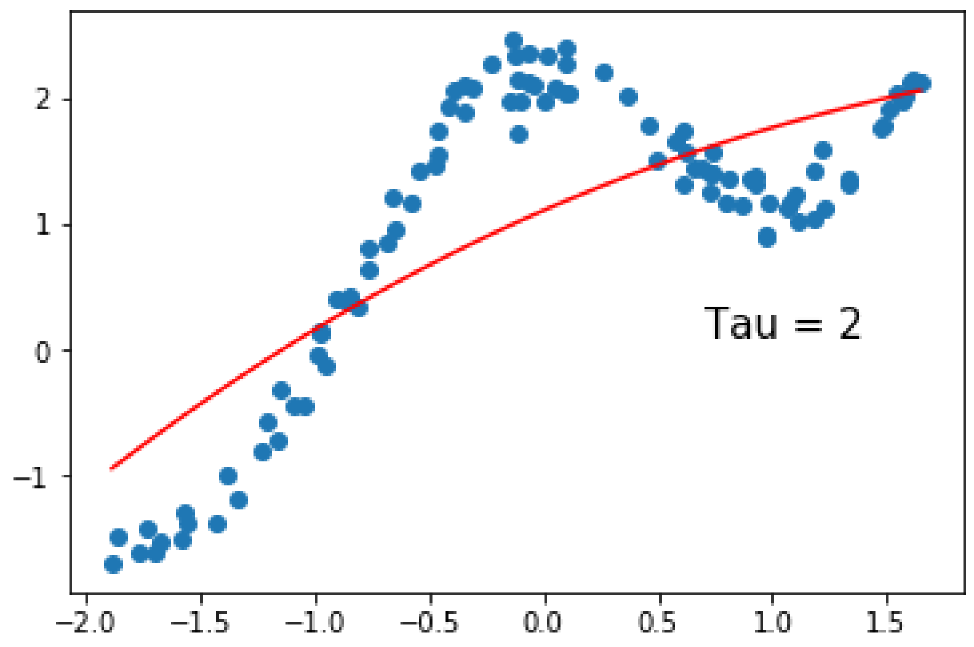


Finally we vary the bandwidth τ and obtain the following plots. It seems that a value of 0.3 for the bandwidth results in the best regression output.

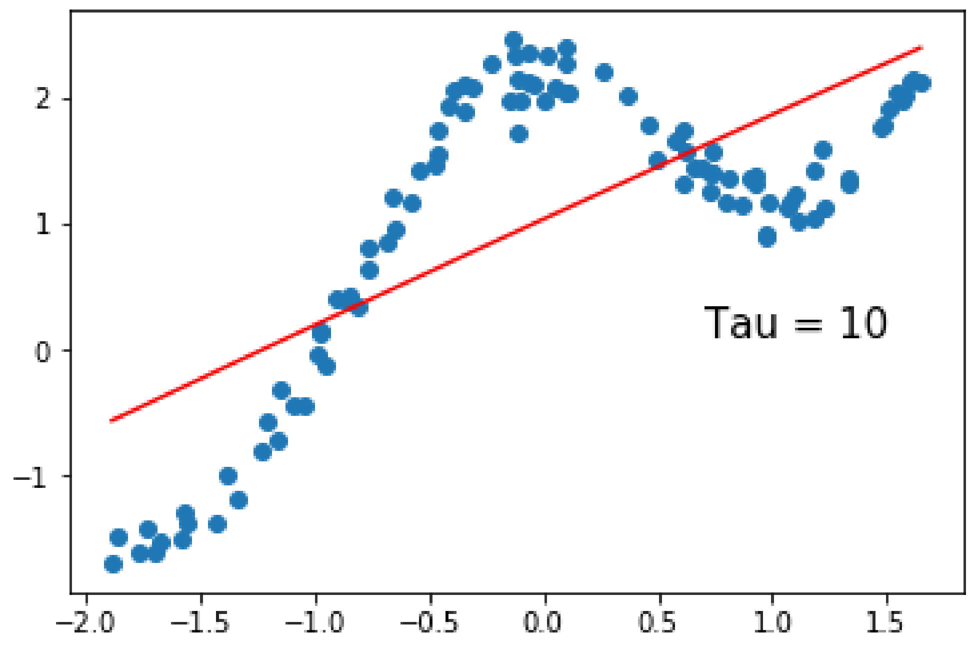




Tau = 0.3
2
1
0
−1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



Tau = 2
2
1
0
−1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



When tau is too small the model starts fitting the small errors in the data and not the underlying pattern, on the other hand when tau is too large the "local" part matters less and we approach unweighted regression, that is the regression starts giving near equal weight to all points no matter how close they are to the target.