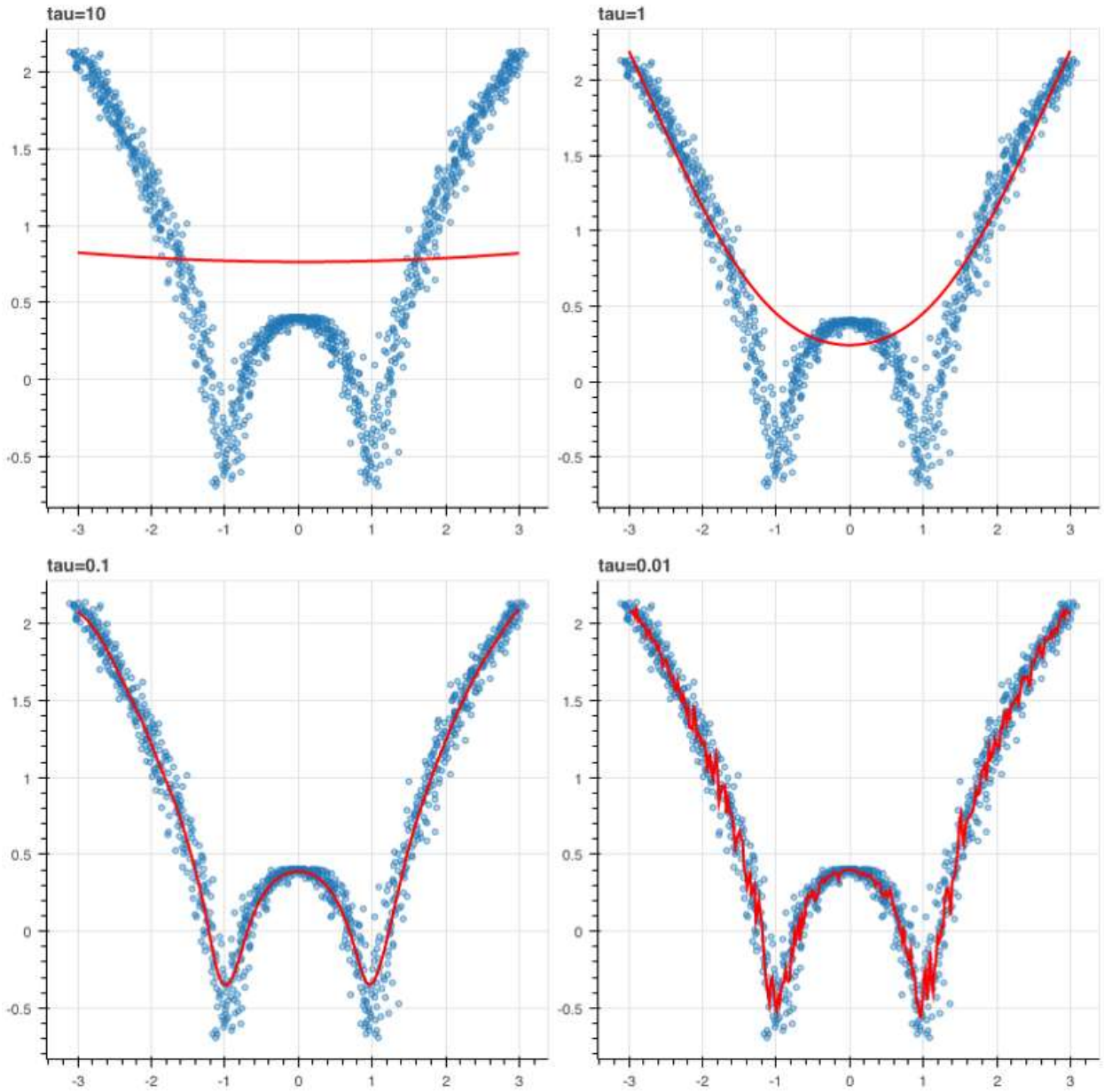
Locally weighted regression

Locally weighted regression is a very powerful non-parametric model used in statistical learning.



To explain how it works, we can begin with a linear regression model and ordinary least squares.

$$h(x) = x^T\hat{\beta}$$
$$\hat{\beta} = \text{argmin}_\beta \sum_{x,y}(y - x^T\beta)^2$$
$$\hat{\beta} = (X^TX)^{-1}X^Ty$$

Given a *dataset* **X**, **y**, we attempt to find a *linear model* **h(x)** that minimizes *residual sum of squared errors*. The solution is given by *Normal equations*.

Linear model can only fit a straight line, however, it can be empowered by polynomial features to get more powerful models. Still, we have to *decide* and *fix* the number and types of features ahead.

Alternate approach is given by locally weighted regression.

$$h(x_o) = x_o{}^T\hat{\beta}(x_o)$$
$$\hat{\beta}(x_o) = \text{argmin}_\beta \sum_{x,y} w(x, x_o)(y - x^T\beta)^2$$
$$w(x, x_o) = e^{-\frac{(x-x_o)^2}{2\tau^2}}$$
$$\hat{\beta}(x_o) = (X^TWX)^{-1}X^TWy$$

Given a *dataset* **X**, **y**, we attempt to find a *model* **h(x)** that minimizes *residual sum of* ***weighted*** *squared errors*. The weights are given by a *kernel function*which can be chosen arbitrarily and in my case I chose a Gaussian kernel. The solution is very similar to *Normal equations*, we only need to insert diagonal weight matrix **W**.

What is interesting about this particular setup? By adjusting meta-parameter **τ**you can get a non-linear model that is as strong as polynomial regression of any degree.

And if you are interested, you can find an excellent explanation of what kernel really does in Andrew Ng's Machine learning course.

This time the notebook contains interactive plot. You can adjust meta-parameter **τ** and watch in realtime its influence on the model. Have fun!