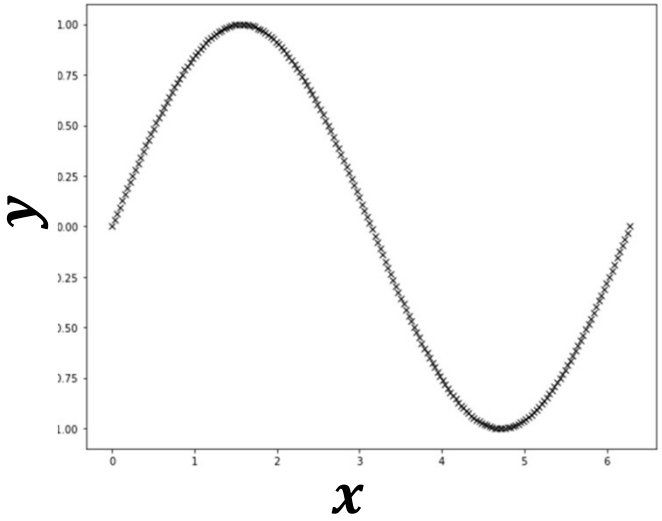
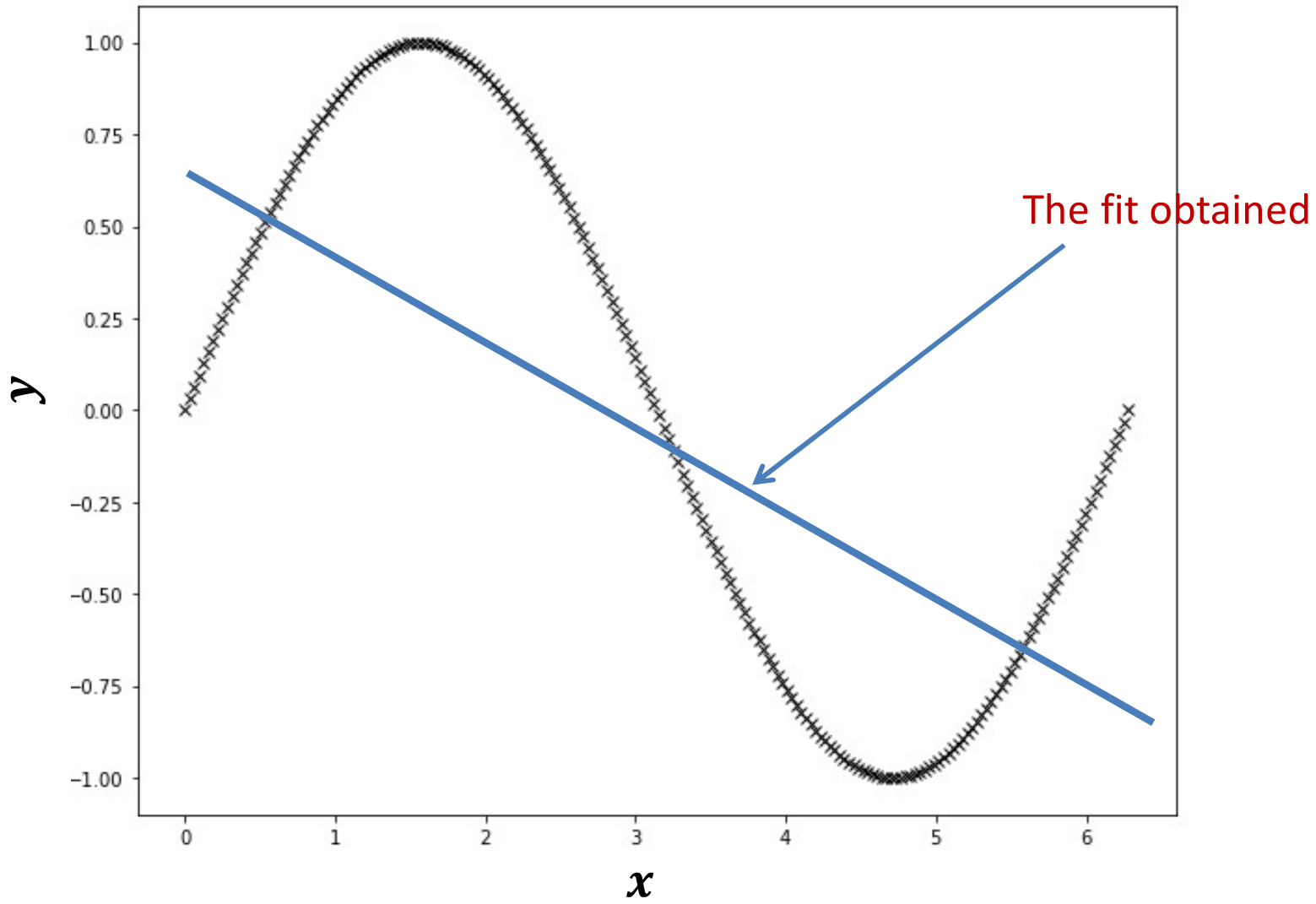
Locally Weighted Linear Regression

- Consider a use case where the relation between the feature vector (independent variables) and the target value (dependent variable) is not linear.
- $y = \sin(x)$
- Suppose we perform linear regression in such case.

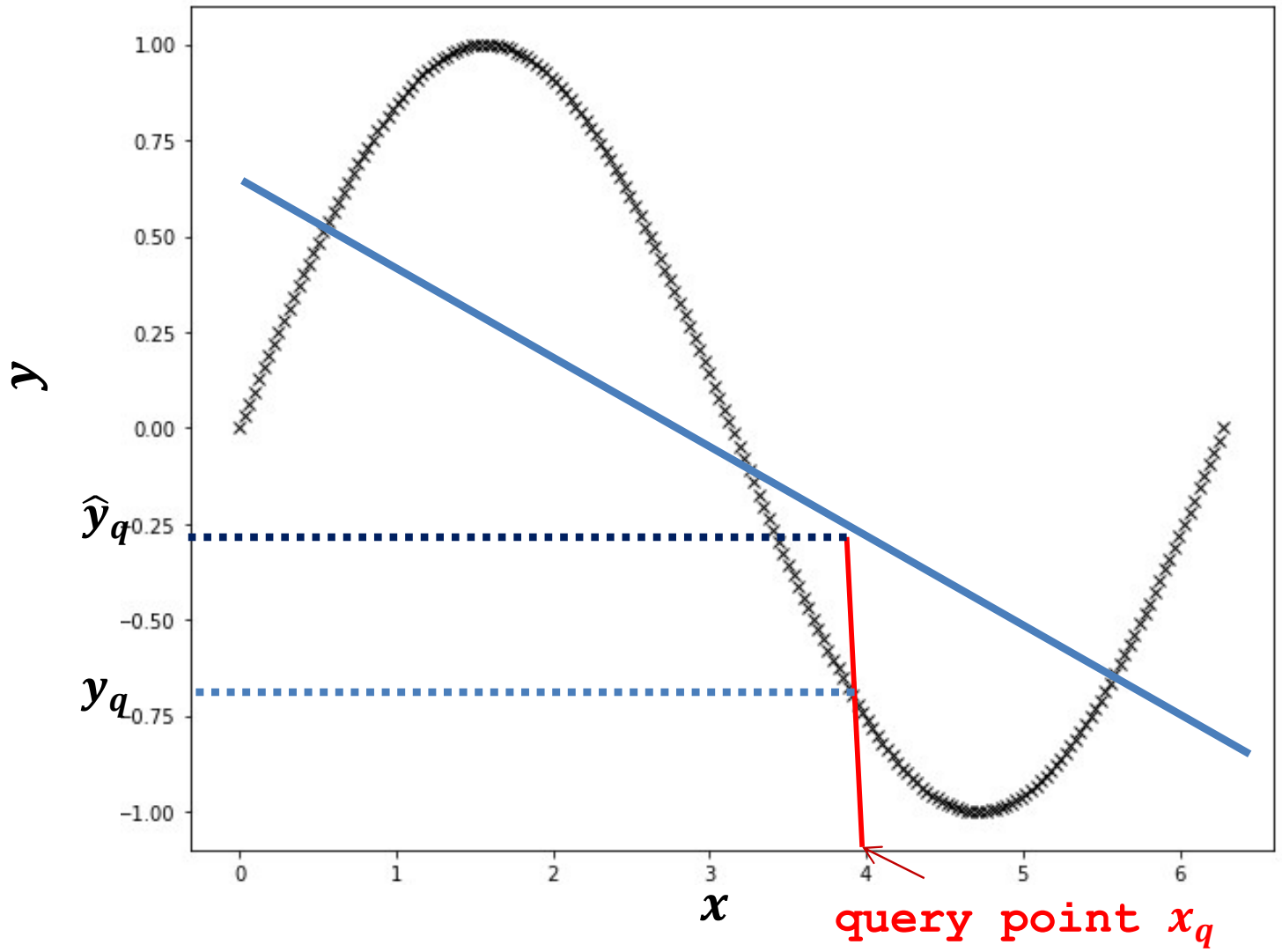


x	y
0	0
π/12	0.25882
π/6	0.5
π/4	0.70711
5π/12	0.86603
π/3	0.86603
7π/12	0.70711
π/2	0.5
2π/3	0.25882
3π/4	0
5π/6	-0.5
11π/12	-0.70711
π	-0.86603
13π/12	-0.86603
7π/6	-0.70711
5π/4	-0.5
17π/12	-0.25882
3π/2	0
19π/12	0.25882

Locally Weighted Linear Regression



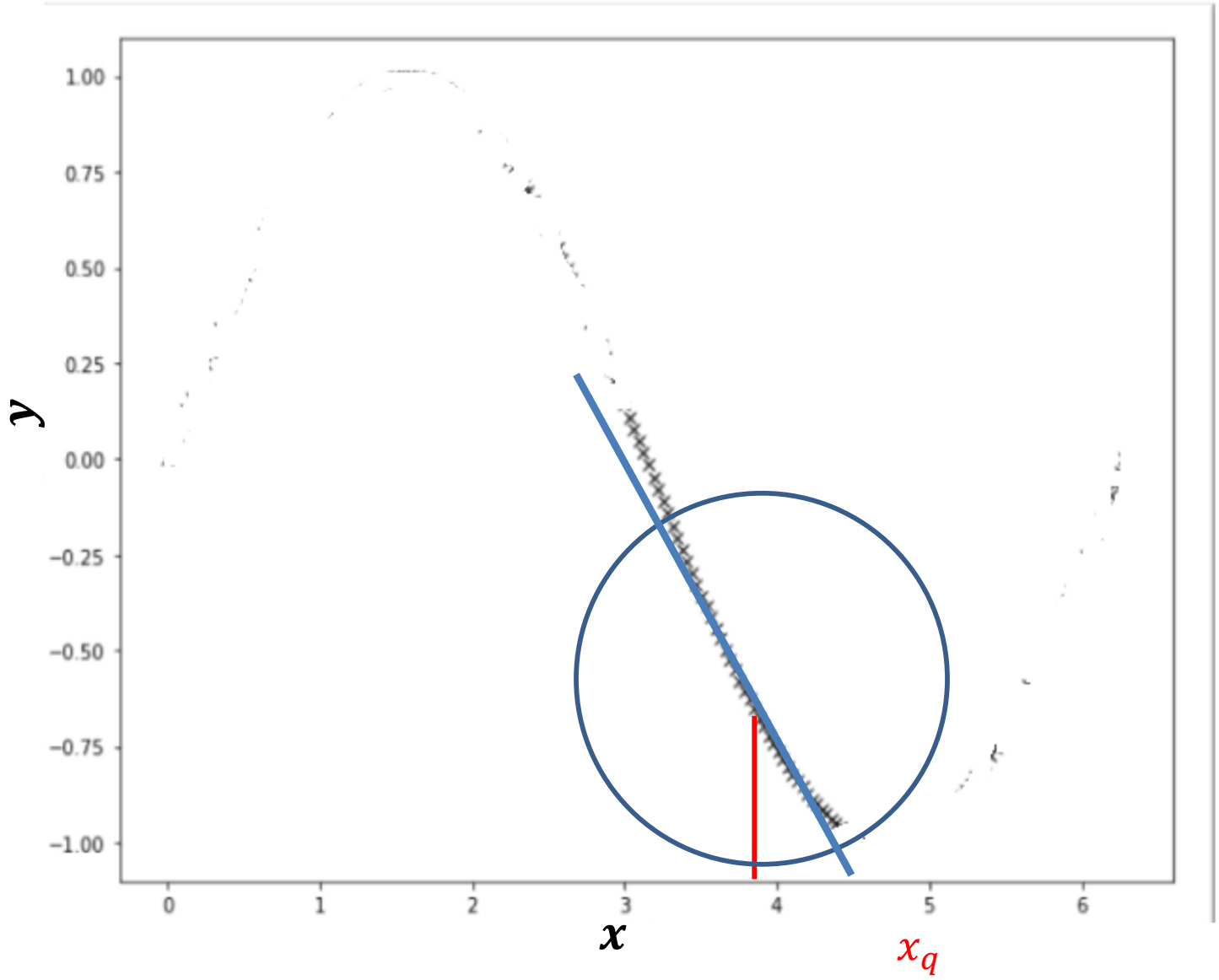
Locally Weighted Linear Regression



Locally Weighted Linear Regression

- Obviously, linear regression does not work.
- The principle of locally weighted regression is as follows.
- Suppose, we know the query point x_q where the value of $\boldsymbol{y_q}$ is to be calculated.
- The samples in the vicinity of the query point contribute more to linear regression, than the samples farther from the query point.

Locally Weighted Linear Regression



Locally Weighted Linear Regression

$$Find\ \theta\ to\ minimize\ \frac{1}{2n}\sum_{i=1}^{n} w^i \left(\widehat{y^i} - y^i\right)^2$$

$$where\ w^i = exp\left(-\frac{\left(x^i - x_q\right)^2}{2\tau^2}\right)$$

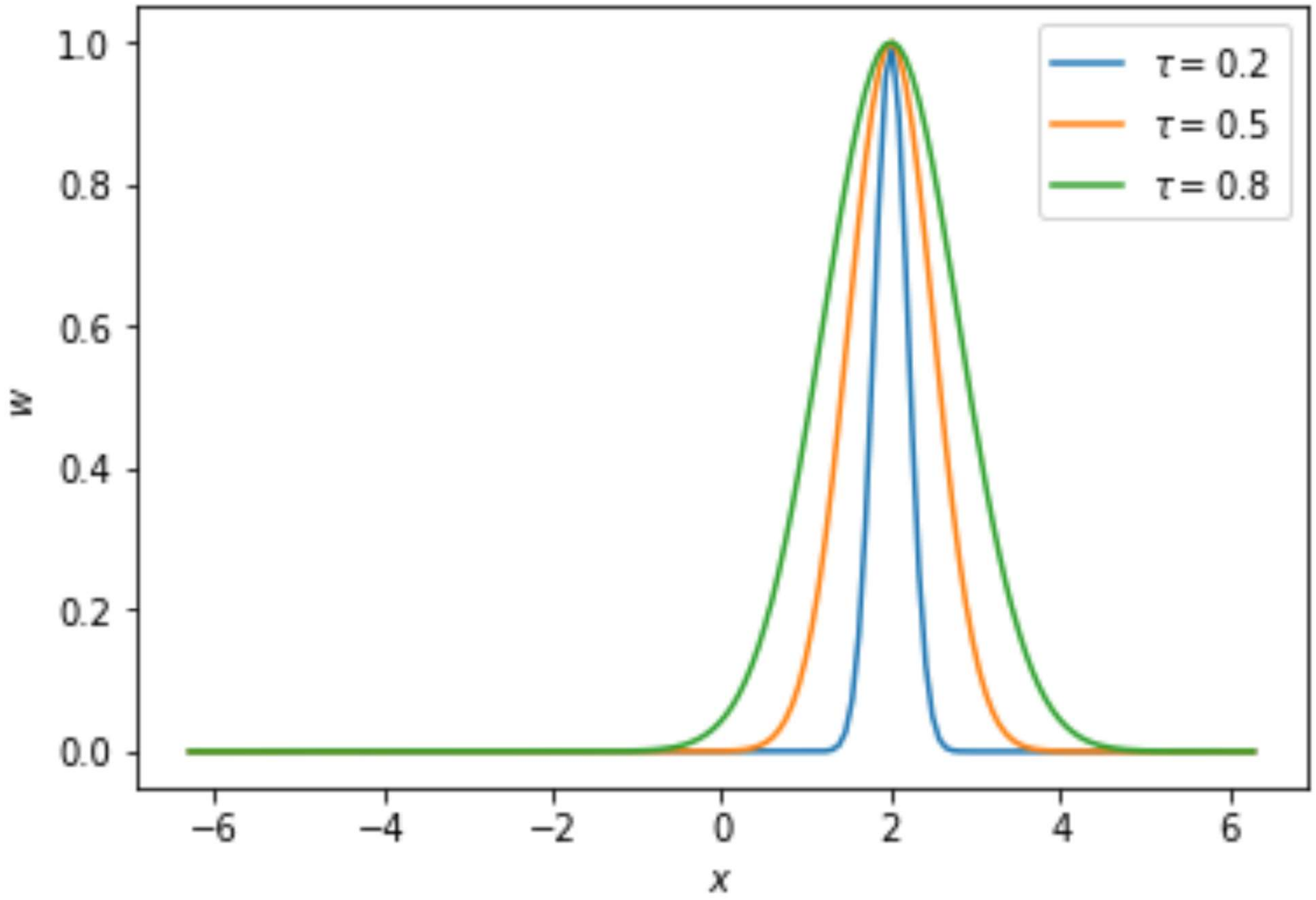
$\boldsymbol{w^i}$ is the weight given to the error of regression contributed by sample $\boldsymbol{i}$ for updating the model parameters.

The farther is $\boldsymbol{x^i}$ from the query point $\boldsymbol{x_q}$, smaller is the weight.

Locally Weighted Linear Regression

- τ controls how quickly the weight of a training example falls off with distance of $x(i)$ from the query point x_q;
- τ is called the bandwidth parameter,

Locally Weighted Linear Regression



Locally Weighted Linear Regression

$$Find\ \theta\ to\ minimize\ J(\theta) = \frac{1}{2n}\sum_{i=1}^{n} w^i \left(h_\theta\left(x^i\right) - y^i\right)^2$$

$$where\ w^i = exp\left(-\frac{\left(x^i - x_q\right)^2}{2\tau^2}\right)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{n}\sum_{i=1}^{n} w^i \left(h_\theta\left(x^i\right) - y^i\right)x_j^i = \frac{1}{n}\sum_{i=1}^{n} w^i\, E^i x_j^i$$

LWR-Batch Gradient Descent

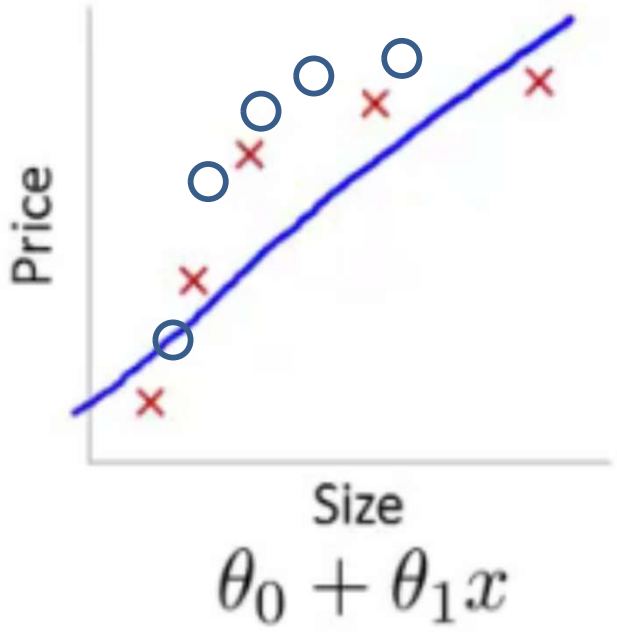
- $g = W * E * X = \begin{bmatrix} W^1 \\ W^2 \\ \\ \\ W^n \end{bmatrix} \begin{bmatrix} E^1 \\ E^2 \\ \\ \\ E^n \end{bmatrix} * \begin{bmatrix} x_0^1 & x_1^1 & x_2^1 \\ x_0^2 & x_1^2 & x_2^2 \\ & & \\ & & \\ x_0^n & x_1^n & x_2^n \end{bmatrix}$
- $grad = \frac{1}{n} np.sum(g, axis = 0)$
- $[\theta_0, \theta_1, \theta_2] = [\theta_0, \theta_1, \theta_2] - \alpha * gra$

Bias and Variance of a Model

- Consider the problem of predicting y from $x \in R$.
- In regression we have several options (linear and non-linear) of fitting $y = h_\theta(x)$ to a dataset.
- The following questions are important to judge the choice.
- How well does the fit/model perform?
- How is its training error often referred to as bias?
- How is its testing error over different datasets often referred to as variance?

Bias and Variance of a Model

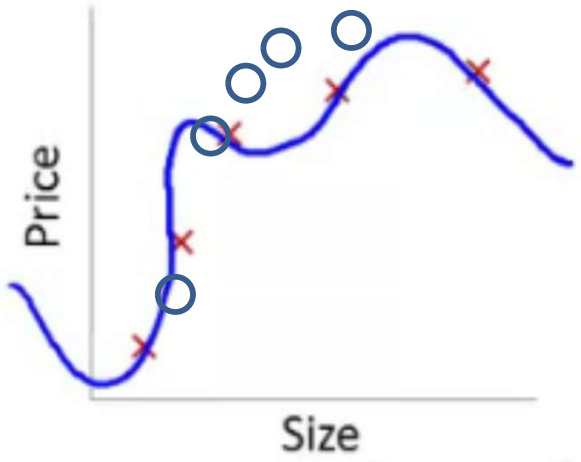
- Consider the problem of predicting y from $x \in R$.
- The figure shows the result of fitting $y = h_\theta(x) = \theta_0 + \theta_1 x_1$ to a dataset.
- The fit is not very good.
- The model has high bias and suffers from underfit.



High bias (underfit)

Bias and Variance of a Model

- The figure is the result of fitting a 4-th order polynomial.
- The Fitted curve passes through the data perfectly.
- The model has less bias.
- But, the variance i.e the test error may be high.
- This is often referred to as overfitting.

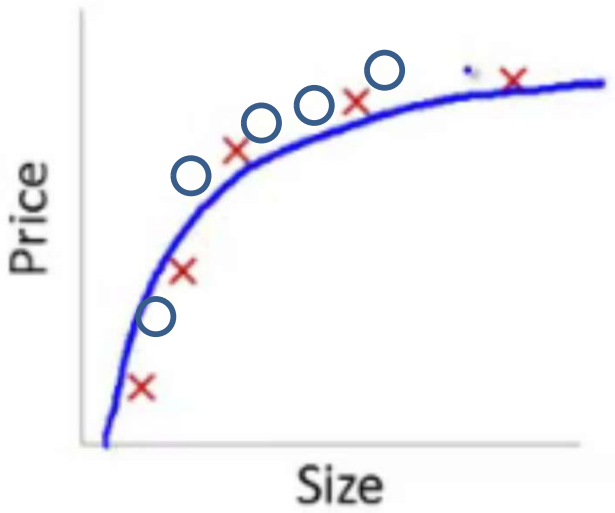


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

Bias and Variance of a Model

- Instead, if we perform non-linear regression, and fit $y = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$, then we obtain a slightly better fit to the data.
- This model has low bias and low variance and proves to be a right fit.



Over fitting

- Over fitting refers to a model that fits the training data too well.
- Let $error_t(h)$ and $error_D(h)$ be the errors of hypothesis h over training samples and the entire distribution of samples respectively.
- Hypothesis $h \in H$ is said to *overfit* training data *iff* there is an alternate hypothesis $h' \in H$ such that
  - $error_t(h) < error_t(h')$ *(h performs well on training data but h' doesn't)*
  - $error_D(h) > error_D(h')$ *(h does not perform well on test data but h' does)*