LINEAR REGRESSION & REGULARIZATION

Narges Norouzi

EXTENDING LINEAR REGRESSION TO MORE COMPLEX MODELS

- The inputs $\boldsymbol{X}$ for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs (log, exp, square root, square, etc.)
  - Polynomial transformation (example: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$)
  - Dummy encoding of categorical inputs
  - Interactions between variables (example: $x_3 = x_1 \times x_2$)
- This allows use of linear regression techniques to fit non-linear datasets.

LINEAR BASIS FUNCTION MODEL

- Generally,

$$h_\theta(x) = \sum_{j=0}^{d} \theta_j \phi_j(x)$$

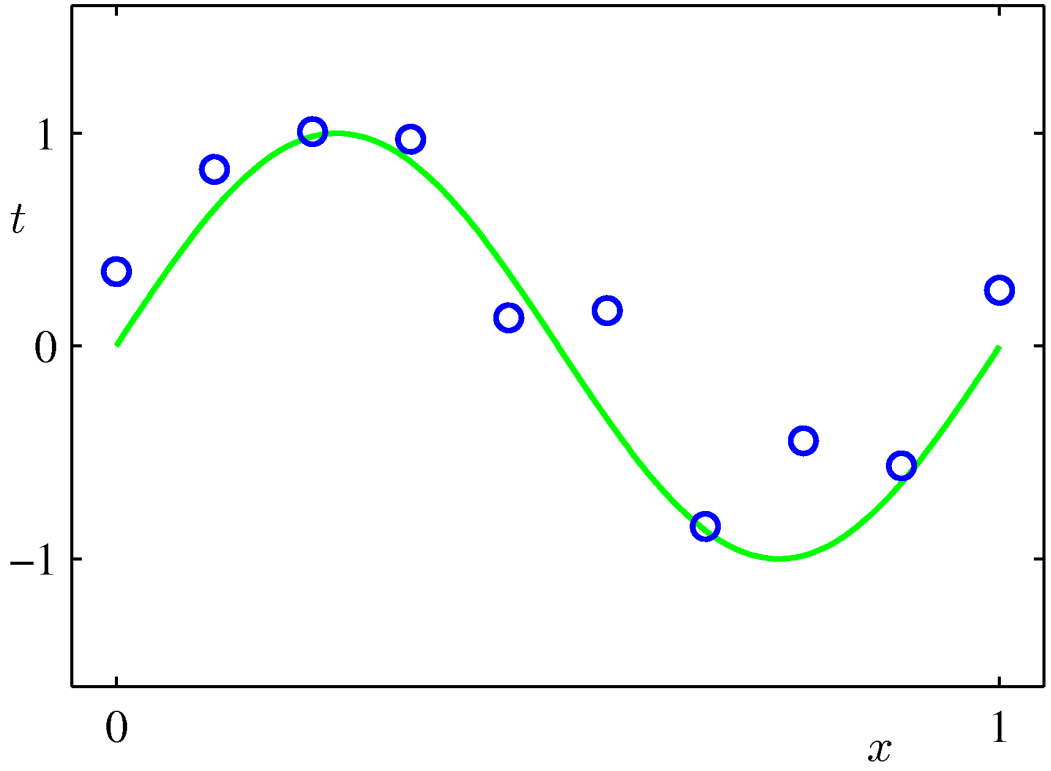
Basis Function

- Typically, $\phi_0(x) = 1$ so that θ_0 acts as a bias.
- In the simplest case, we can use linear basis function:

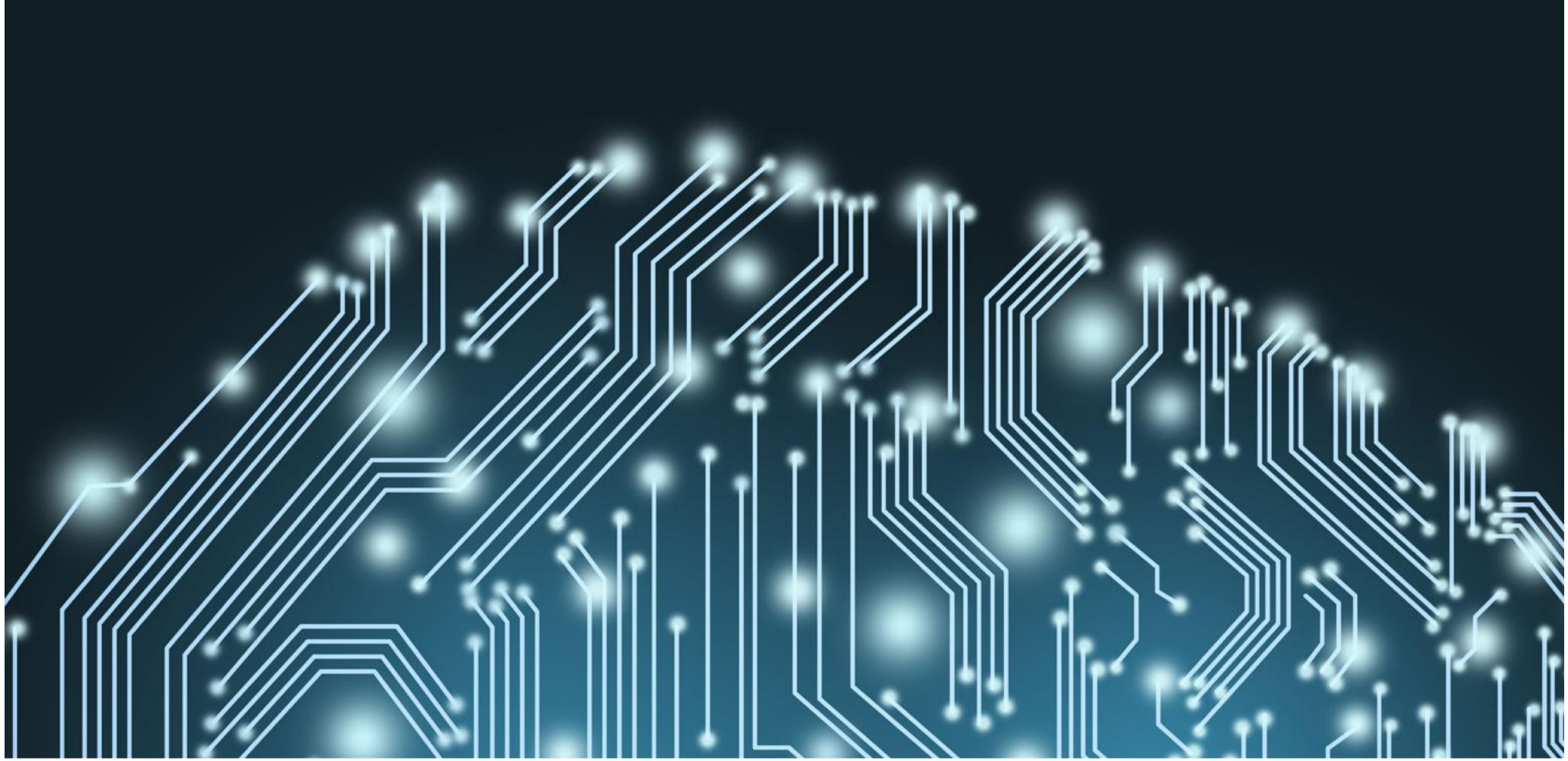
$$\phi_j(x) = x_j$$

- Polynomial basis function: $\phi_j(x) = x^j$
- Gaussian basis function: $\phi_j(x) = e^{\frac{\left(x-\mu_j\right)^2}{2s^2}}$

EXAMPLE OF FITTING A POLYNOMIAL CURVE WITH A LINEAR MODEL



$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_d x^d = \sum_{j=0}^{d} \theta_j x^d$$



VECTOR CALCULATION

LINEAR ALGEBRA CONCEPTS

- Vector in $\mathbb{R}^d$ is an ordered set of d real numbers.
- $v = [1, 2, 3, 4]$ is in $\mathbb{R}^4$ and is a column vector.
- An m-by-n matrix is an object with m rows and n columns, where each entry is a real number

$$\begin{bmatrix} 1 & 0 & 2 \\ 4 & 0.5 & 7.8 \end{bmatrix}$$

- Transposing the matrix

$$\begin{bmatrix} a \\ b \end{bmatrix}^T = [a \quad b], \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix}^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Note: $(Ax)^T = x^T A^T$

LINEAR ALGEBRA CONCEPTS

- Vector norms:
  - L_p norm of $v = (v_1, v_2, \ldots, v_k) = (\sum_i |v_i|^p)^{\frac{1}{p}}$
  - Common norms L_1 and L_2
  - Length of the vector v is $L_2(v)$

LINEAR ALGEBRA CONCEPTS

- Vector dot product: $u \,.\, v = (u_1 \;\; u_2) \,.\, (v_1 \;\; v_2) = u_1 \times v_1 + u_2 \times v_2$
  - The dot product of u with itself $= length(u)^2 = \|u\|_2^2$
- Matrix product:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$A \times B = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

LINEAR ALGEBRA CONCEPTS

- Vector products:
  - Dot product:

$$u \,.\, v = u^T v = [u_1 \;\; u_2].\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = u_1 \times v_1 + u_2 \times v_2$$

  - Outer product:

$$uv^T = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}[v_1 \;\; v_2] = \begin{bmatrix} u_1 v_1 & u_1 v_2 \\ u_2 v_1 & u_2 v_2 \end{bmatrix}$$

VECTORIZATION

- Benefits of vectorization
  - More compact equations
  - Faster code (using optimized matrix libraries)
- Consider our model: $h(x) = \sum_{j=0}^{d} \theta_j x_j$
- Let $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$, $x = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$

We can write our model in vectorized form as $h(x) = x^T \theta$

VECTORIZATION

- Consider our model for n instances: $h\left(x^{(i)}\right) = \sum_{j=0}^{d} \theta_j x_j^{(i)}$
- Let

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \in \mathbb{R}^{d+1 \times 1}, \qquad X = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_d^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_d^{(2)} \\ \vdots & & \ddots & \vdots \\ 1 & x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix} \in \mathbb{R}^{n \times d+1}$$

- We can write the vectorized form as $h_\theta(x) = X\theta$

VECTORIZATION

- For the linear regression cost function:

$$Cost(\theta) = \frac{1}{2\times n}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2\times n}\sum_{i=1}^{n}\left(x^{(i)^T}\theta - y^{(i)}\right)^2$$

$$= \frac{1}{2\times n}(X\theta - Y)^T(X\theta - Y)$$

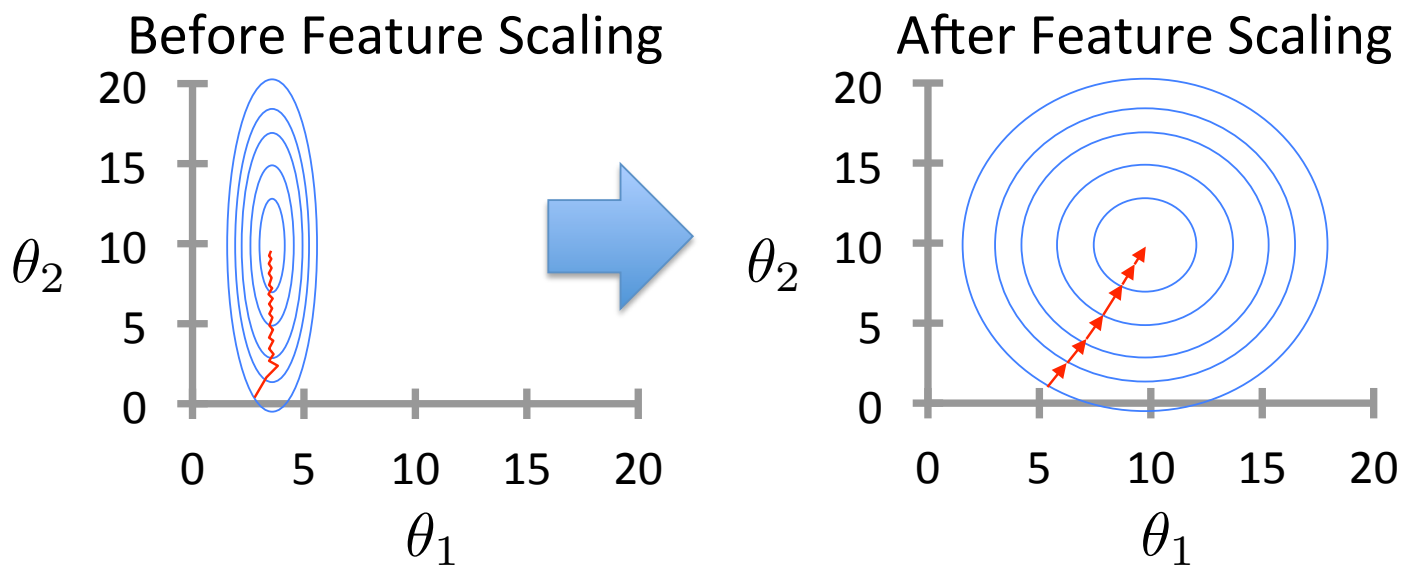
- Note that:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n\times 1}, \qquad X \in \mathbb{R}^{n\times d+1}, \qquad \theta \in \mathbb{R}^{d+1\times 1} \rightarrow \quad X\theta \in \mathbb{R}^{n\times 1}$$

$$X\theta - Y \in \mathbb{R}^{n\times 1}, \quad (X\theta - Y)^T \quad \in \mathbb{R}^{1\times n} \rightarrow Cost(\theta) \in \mathbb{R}$$

IMPROVING LEARNING: FEATURE SCALING

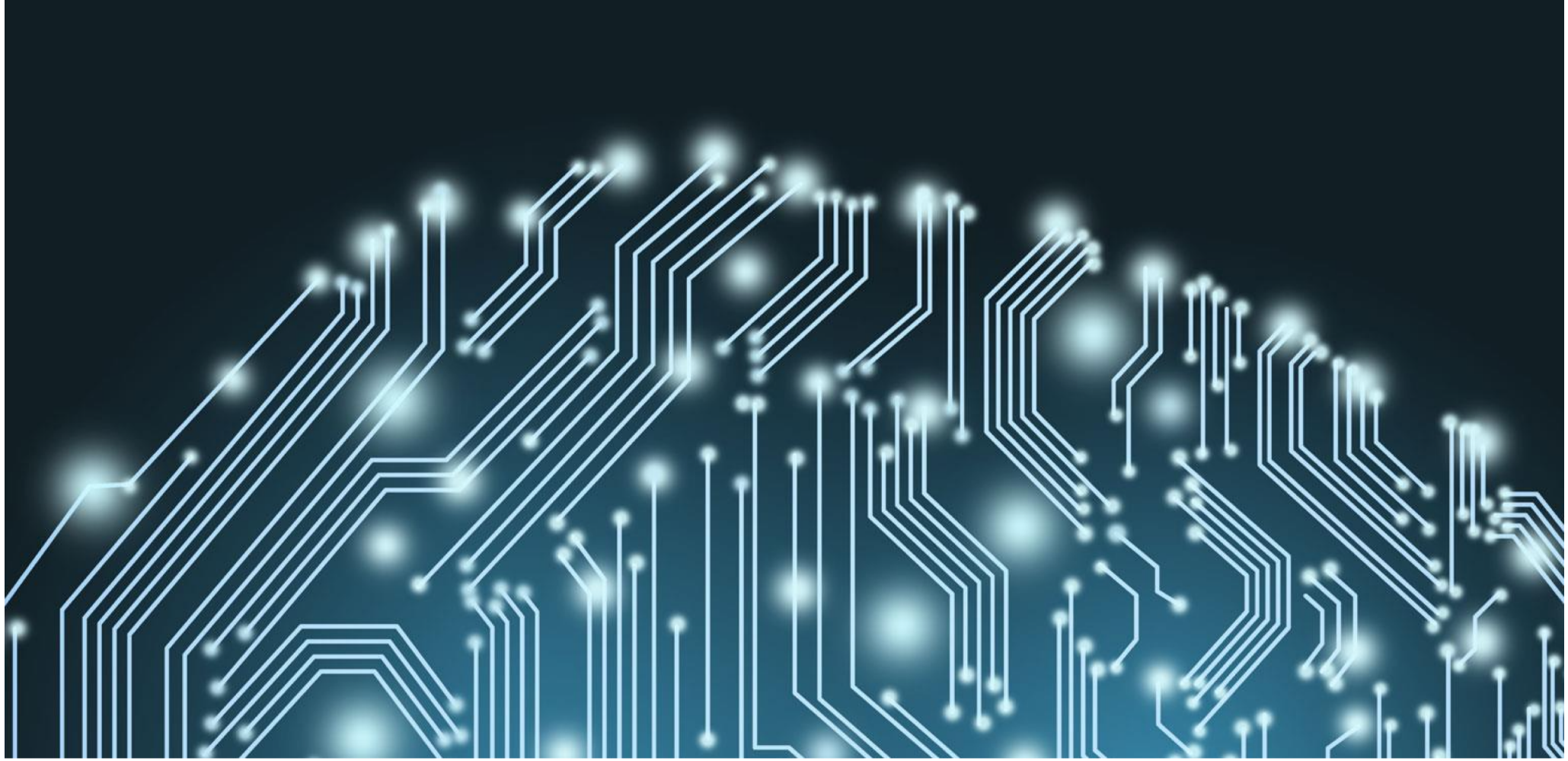
- Idea: ensure that features have similar scales



- Makes gradient descent converge much faster

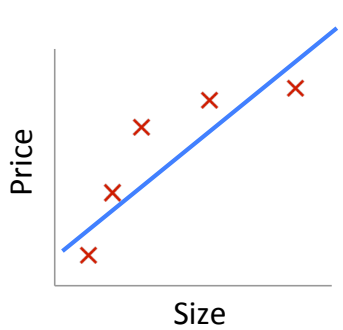
CLASS EXERCISE

bit.ly/ce-6



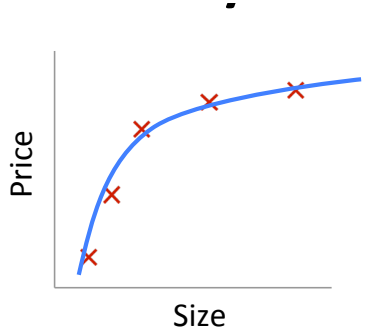
REGULARIZATION AND OVERFITTING

QUALITY OF FIT



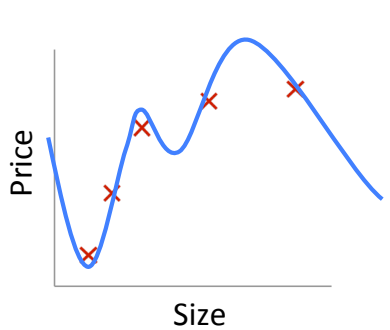
$\theta_0 + \theta_1 x$

Underfitting
(high bias)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Correct fit



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

Overfitting
(high variance)

- Overfitting: The learned hypothesis may fit the training set very well
  - but fails to generalize to new examples

REGULARIZATION

- A method for automatically controlling the complexity of the learned hypothesis
- Idea: penalize large values of θ_j
- There are other techniques we will learn later such as dropout technique in neural networks or dimensionality reduction.

REGULARIZATION

- Linear regression objective function

$$Cost(\theta) = \left\{\frac{1}{2\times n}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2\right\} + \left\{\lambda\sum_{j=1}^{d}\theta_j^2\right\}$$

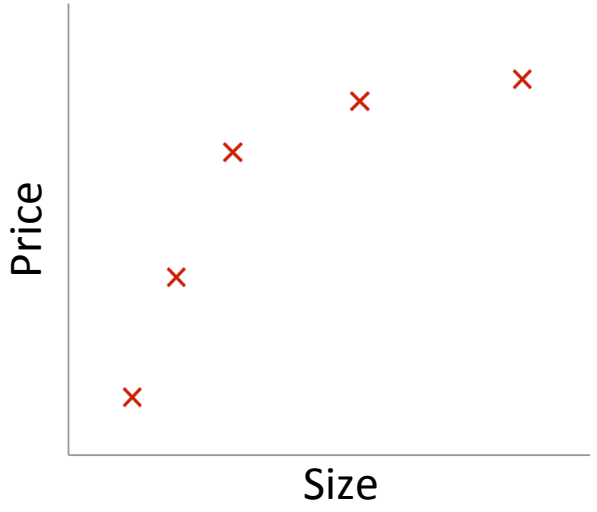
Model fit to data Regularization

- λ is the regularization parameter ($\lambda \geq 0$)
- No regularization on θ_0
- This exact regularizer pulls coefficients/parameters to 0

UNDERSTANDING REGULARIZATION

- What happens if we set λ too huge?

$$Cost(\theta) = \frac{1}{2\times n}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{d}\theta_j^2$$



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

UNDERSTANDING REGULARIZATION

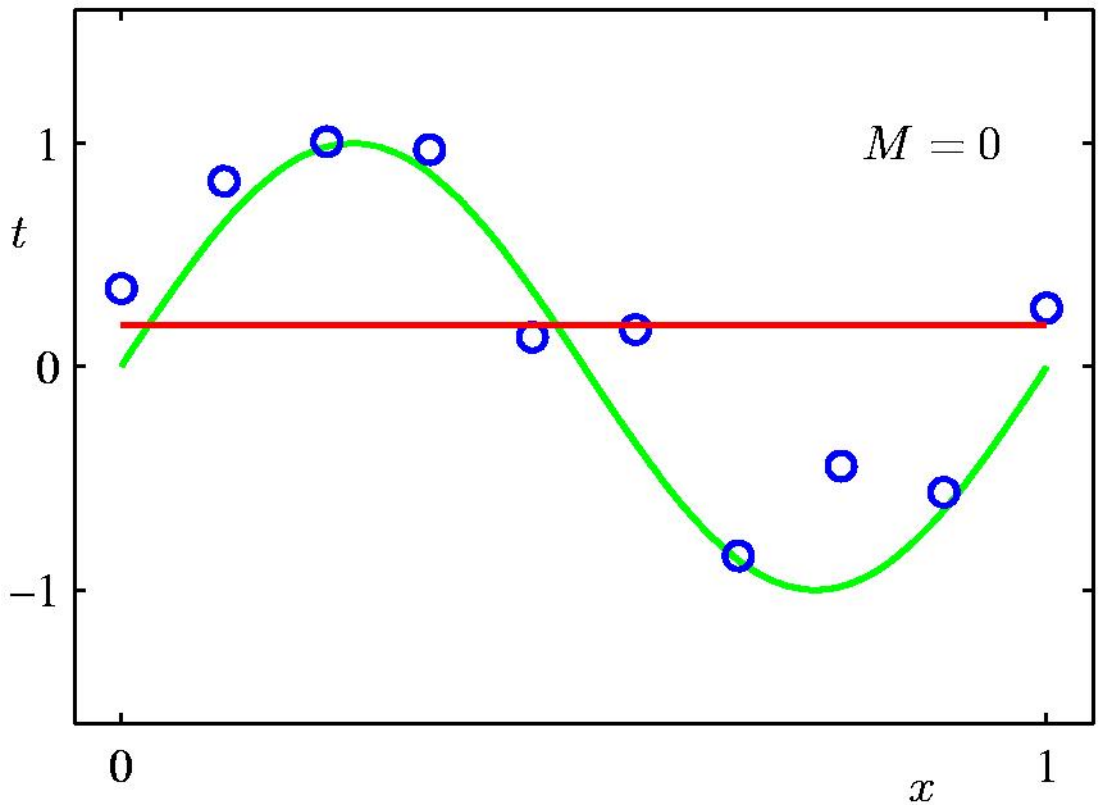
- What happens if we set λ too huge?

$$Cost(\theta) = \frac{1}{2\times n}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{d}\theta_j^2$$

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

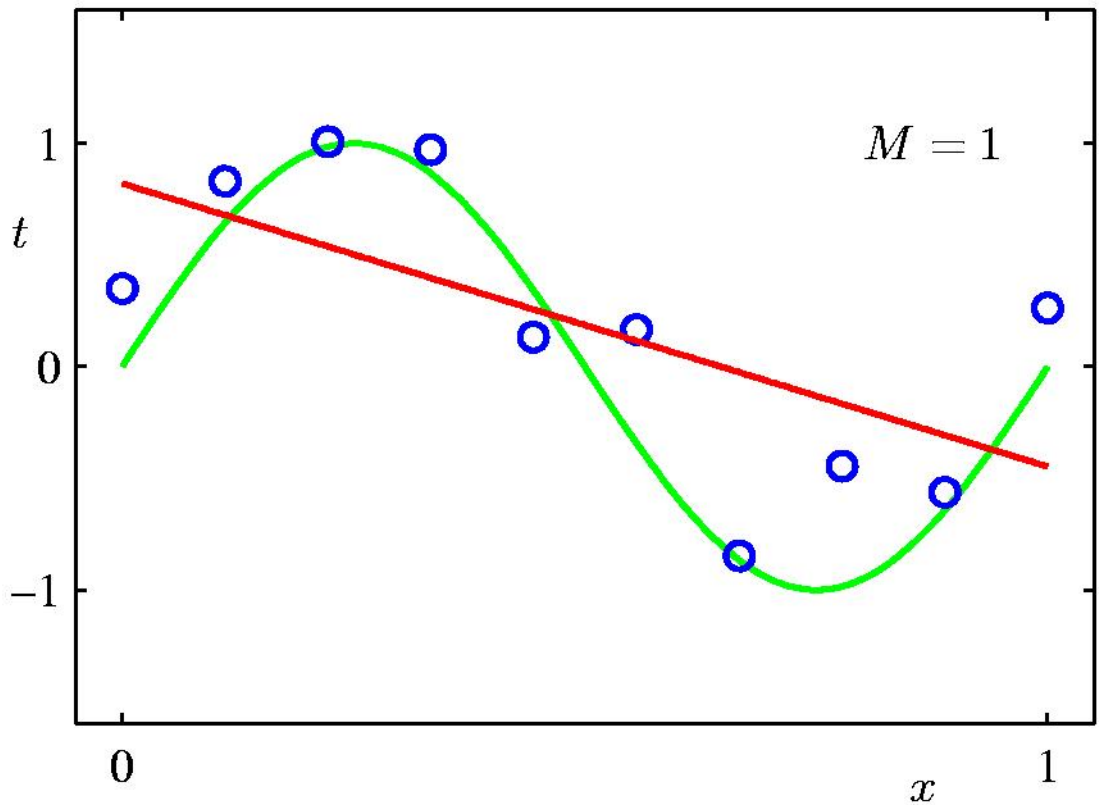
0 0 0 0

0TH ORDER POLYNOMIAL

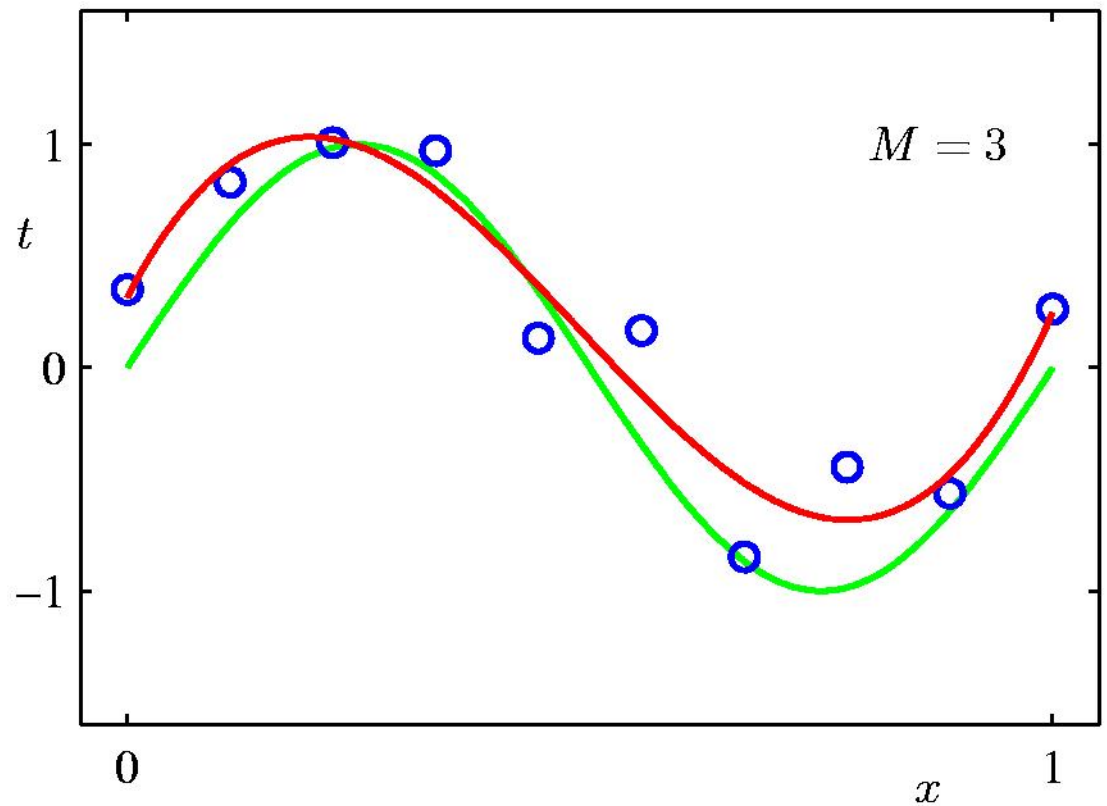


N = 10

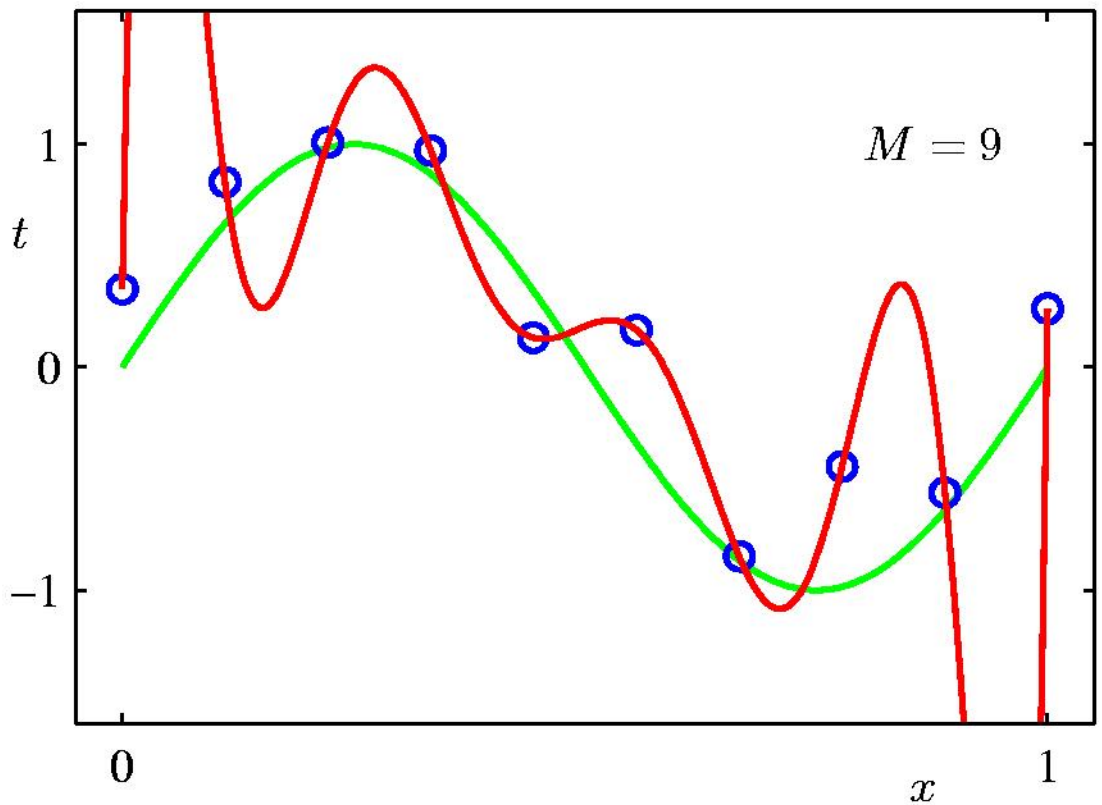
1ST ORDER POLYNOMIAL



3RD ORDER POLYNOMIAL



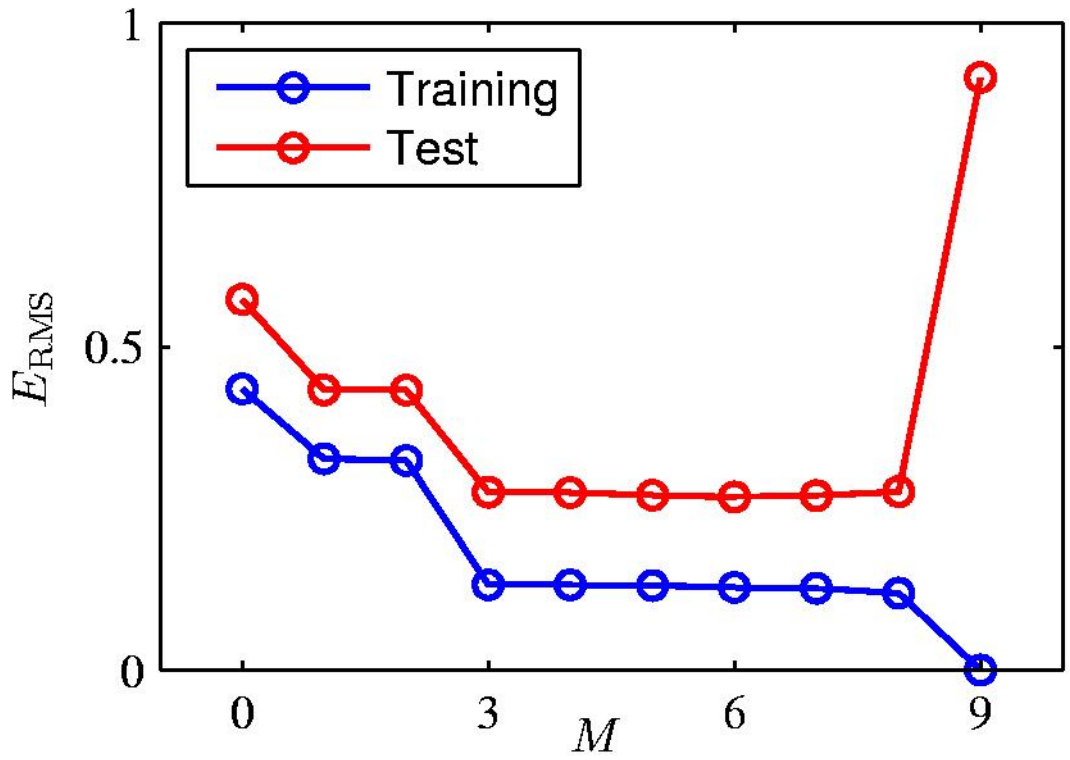
9TH ORDER POLYNOMIAL



POLYNOMIAL COEFFICIENTS

	$M=0$	$M=1$	$M=3$	$M=9$
θ_0	0.19	0.82	0.31	0.35
θ_1		-1.27	7.99	232.37
θ_2			-25.43	-5321.83
θ_3			17.37	48568.31
θ_4				-231639.30
θ_5				640042.26
θ_6				-1061800.52
θ_7				1042400.18
θ_8				-557682.99
θ_9				125201.43

OVER-FITTING



QUESTIONS?