

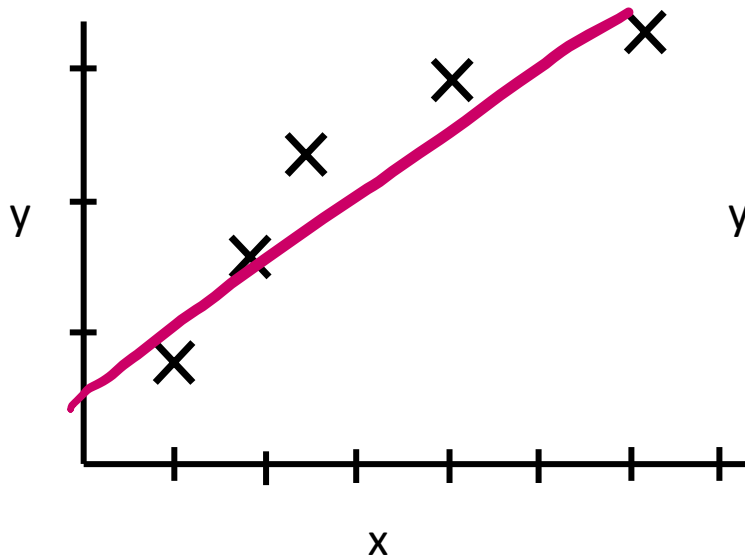
# The Problem of Overfitting

Prof. Dr. Christina Bauer

[christina.bauer@th-deg.de](mailto:christina.bauer@th-deg.de)

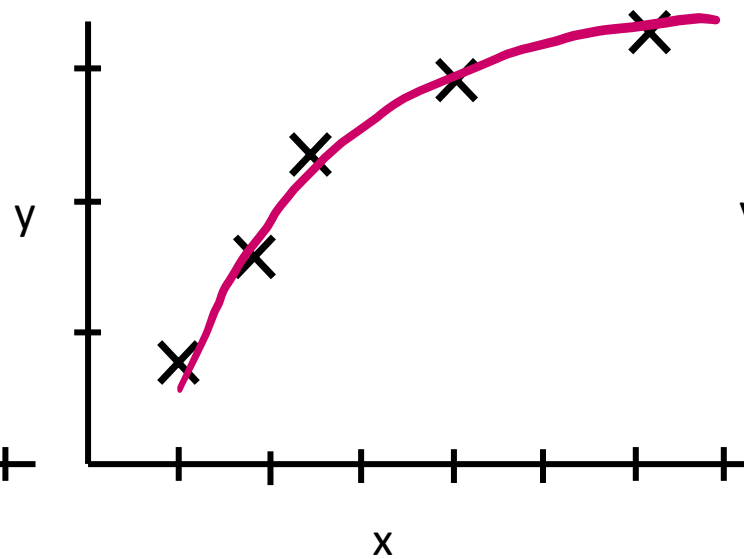
Faculty of Computer Science

# EXAMPLE – LINEAR REGRESSION

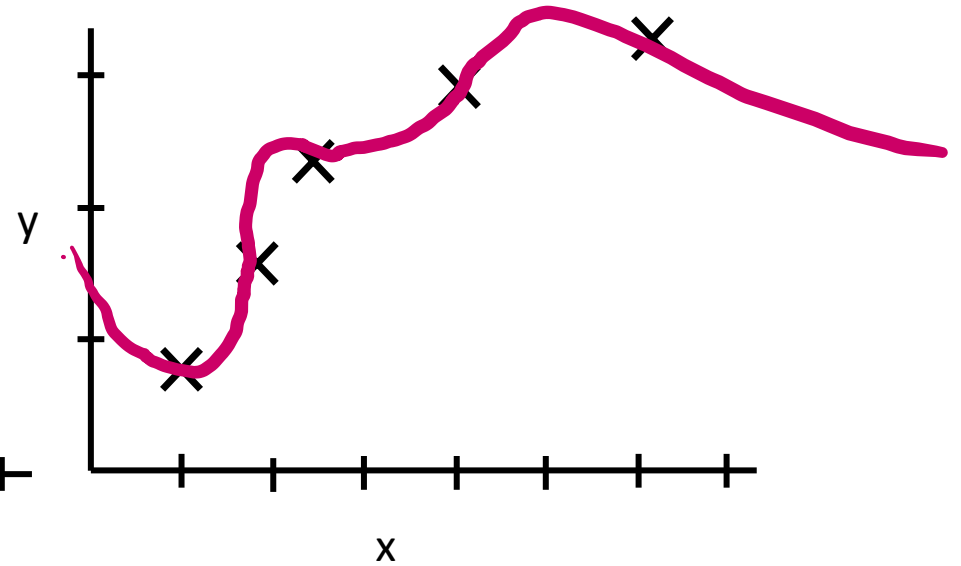


$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

→ Underfit + high bias



$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



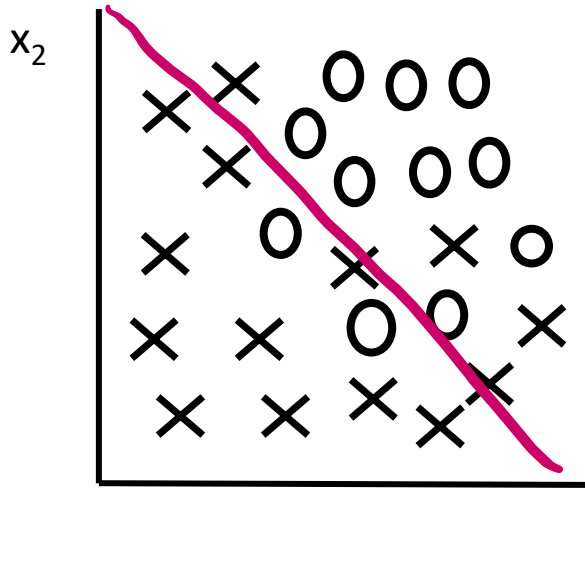
$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

→ Overfit + high variance

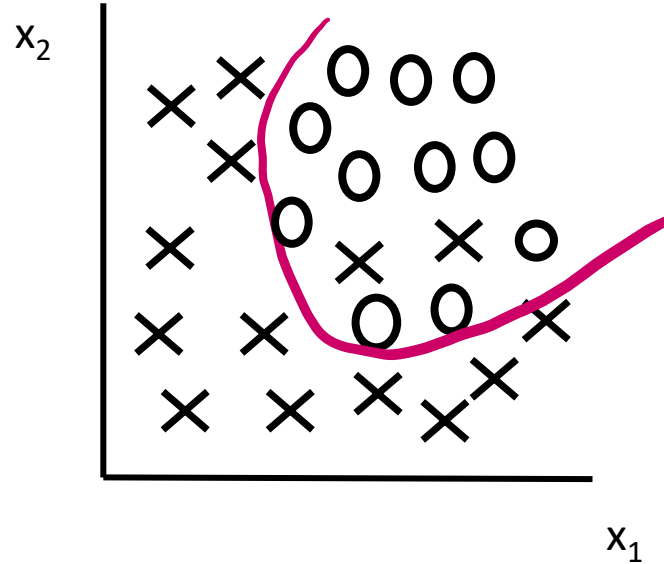
Overfitting: If we have too many features, the learned hypothesis may fit the training set very well

→  $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 \sim 0$ , but fail to generalize to new examples.

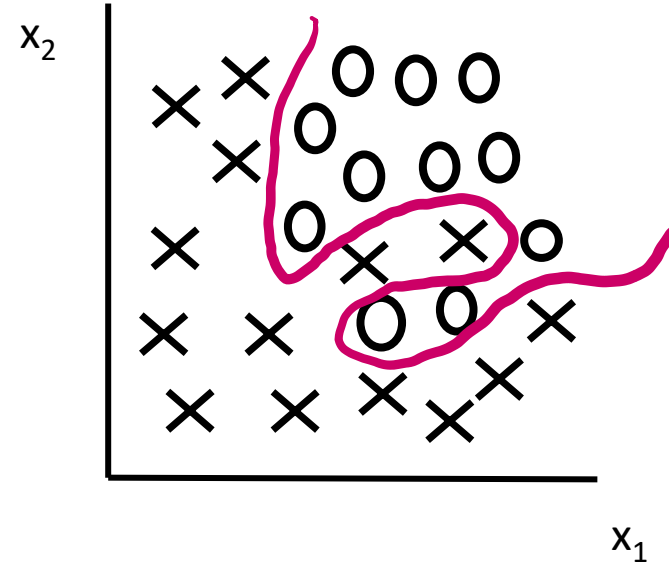
# EXAMPLE — LOGISTIC REGRESSION



$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$   
→  $G$  = sigmoid function  
→ Underfit



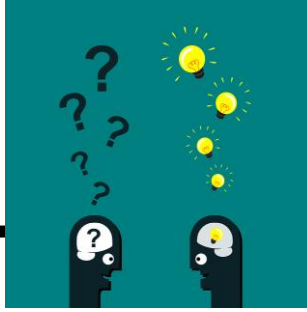
$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$



$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$   
→ Overfit

# QUESTION

---



Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis  $h_\theta(x)$  has overfit the training set, it means that:

A: It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.

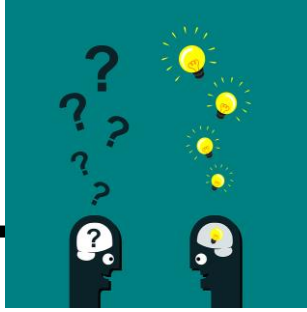
B: It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.

C: It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.

D: It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

# QUESTION

---



Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis  $h_\theta(x)$  has overfit the training set, it means that:

A: It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.

B: It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.

~~C: It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.~~

D: It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

# ADDRESSING OVERFITTING

---

$X_1$  = size of the house

$X_2$  = no. of bedrooms

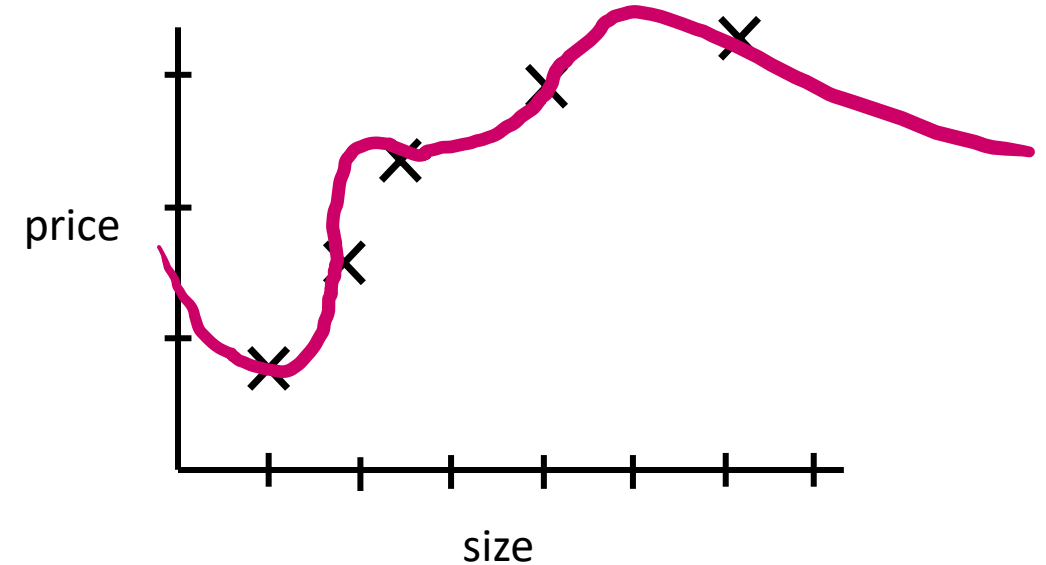
$X_3$  = no. of floors

$X_4$  = age of the house

$X_5$  = average income in neighbourhood

$X_6$  = kitchen size

....  
 $X_{100}$



→ Plotting the data may help but is not possible with a lot features

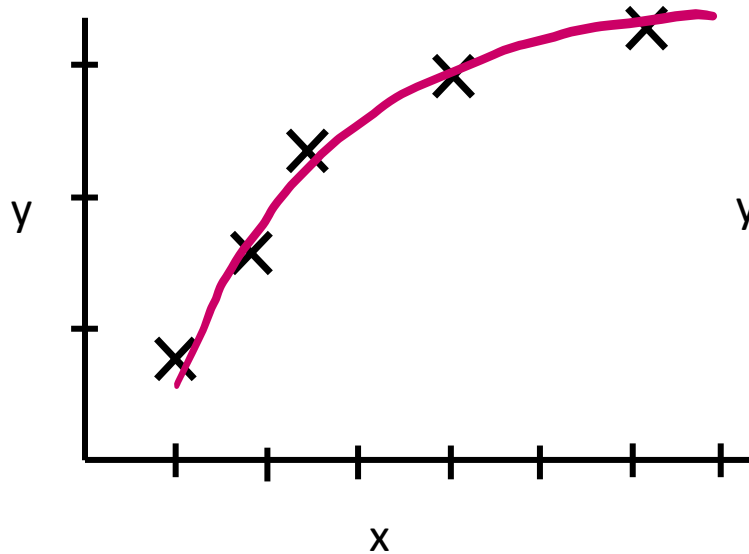
# ADDRESSING OVERFITTING

---

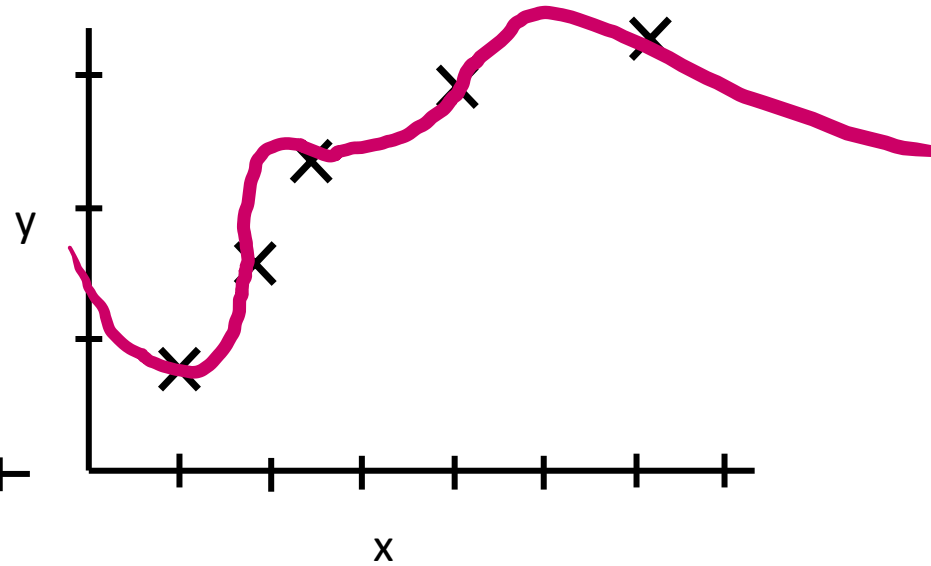
## Options:

- Reduce the number of features
  - Manually select which features to keep
  - Model selection algorithm
- Regularization
  - Keep all the features, but reduce magnitude/value of Parameters  $\theta_j$
  - Works well if we have a lot of features, each of which contributes to predicting  $y$

# REGULARIZATION – COST FUNCTION - IDEA



$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



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

Suppose we penalize and make  $\theta_3, \theta_4$  very small.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \frac{1000 * \theta_3^2 + 1000 * \theta_4^2}{2} \rightarrow \theta_3 \sim 0 ; \theta_4 \sim 0$$



# REGULARIZATION

---

- Small values for parameters  $\theta_0, \theta_1, \dots, \theta_n$ 
  - Simpler hypothesis
  - Less prone to overfitting
- Example “House prize prediction”:
  - Features:  $x_1, x_2, \dots, x_{100}$
  - Parameters:  $\theta_0, \theta_1, \dots, \theta_{100}$

$$\min_{\theta} \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

regularization term to shrink every parameter  $\theta_1, \dots, \theta_{100}$

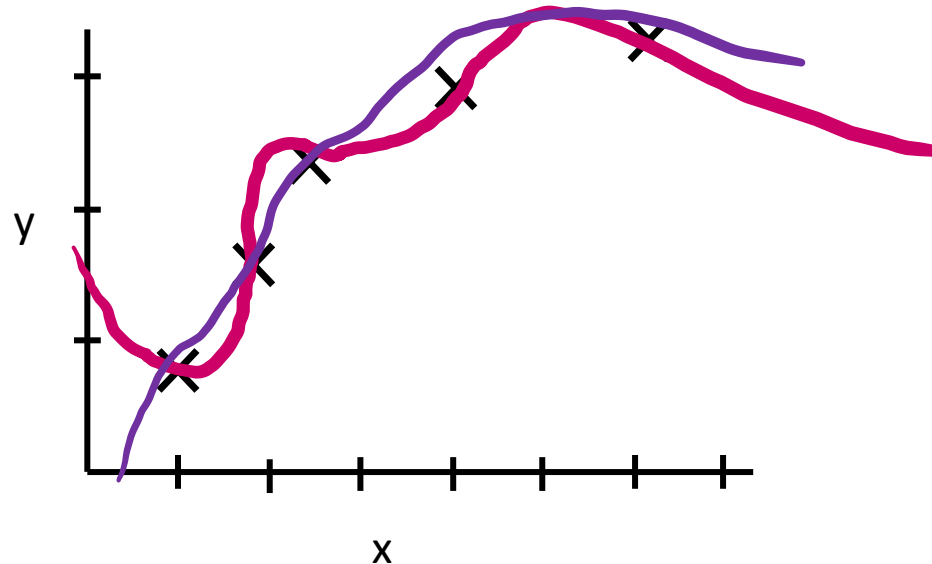
# REGULARIZATION

---

$$J(\theta) = \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

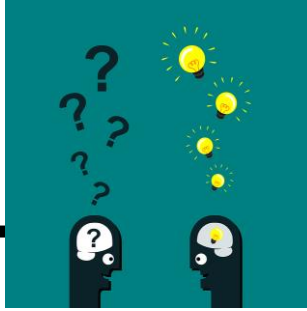
regularization parameter

$\min_{\theta} J(\theta)$



# QUESTION

---



In regularized linear regression, we choose  $\theta$  to minimize:

$$J(\theta) \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

What if  $\lambda$  is set to an extremely large value (perhaps too large for our problem, say  $\lambda=10^{10}$ )?

A: Algorithm works fine; setting  $\lambda$  to be very large can't hurt it.

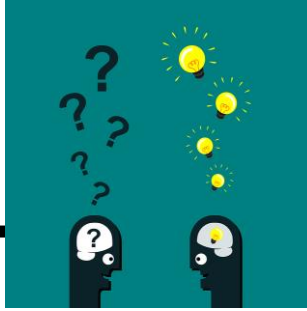
B: Algorithm fails to eliminate overfitting.

C: Algorithm results in underfitting (fails to fit even the training set).

D: Gradient descent will fail to converge.

# QUESTION

---



In regularized linear regression, we choose  $\theta$  to minimize:

$$J(\theta) \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

What if  $\lambda$  is set to an extremely large value (perhaps too large for our problem, say  $\lambda=10^{10}$ )?

A: Algorithm works fine; setting  $\lambda$  to be very large can't hurt it.

B: Algorithm fails to eliminate overfitting.

~~C: Algorithm results in underfitting (fails to fit even the training set).~~

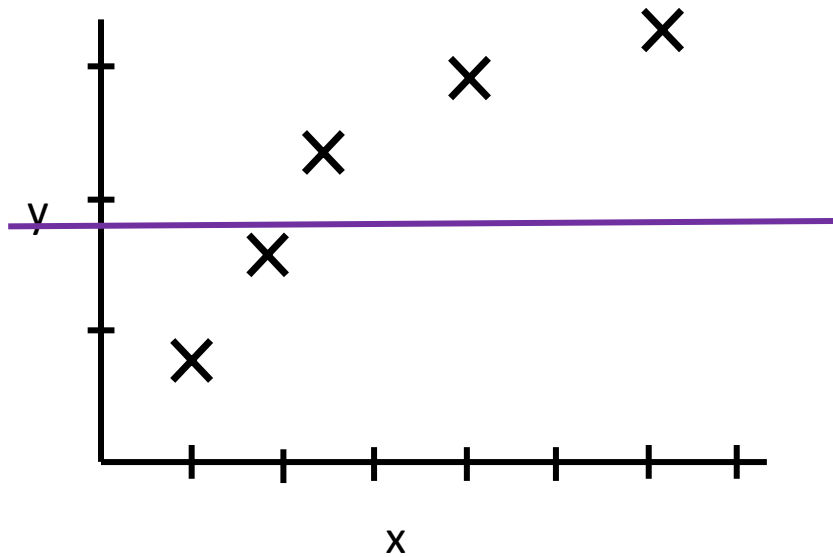
D: Gradient descent will fail to converge.

# COST FUNCTION

In regularized linear regression, we choose  $\theta$  to minimize:

$$J(\theta) = \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

What if  $\lambda$  is set to an extremely large value (perhaps too large for our problem, say  $\lambda=10^{10}$ )?



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

$$\theta_2, \theta_3, \theta_4 \sim 0$$

→ Underfit

# REGULARIZED LINEAR REGRESSION

---

$$J(\theta) \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

$$\min_{\theta} J(\theta)$$

# REGULARIZED LINEAR REGRESSION

---

Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_0^i \quad \text{Do not penalize } \theta_0$$
$$\theta_j := \theta_j - \alpha \left[ \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i + \frac{\lambda}{m} \theta_j \right] \quad \frac{\partial}{\partial \theta_j} J(\theta) \text{ (regularized)}$$

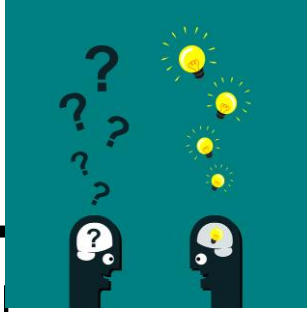
(j = 1, 2, ..., n)

}

Alternative:  $\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i$

# QUESTION

---



Suppose you are doing gradient descent on a training set of  $m > 0$  examples, using a fairly small learning rate  $\alpha > 0$  and some regularization parameter  $\lambda > 0$ . Consider the update rule:

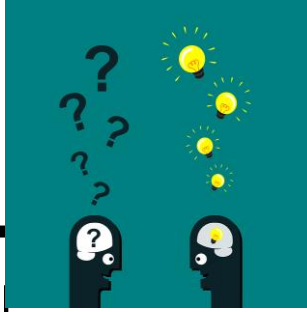
$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i$$

Which of the following statements about the term  $1 - \alpha \frac{\lambda}{m}$  must be true?

- A:  $1 - \alpha \frac{\lambda}{m} > 1$
- B:  $1 - \alpha \frac{\lambda}{m} = 1$
- C:  $1 - \alpha \frac{\lambda}{m} < 1$
- D: None of the above.



# QUESTION



Suppose you are doing gradient descent on a training set of  $m > 0$  examples, using a fairly small learning rate  $\alpha > 0$  and some regularization parameter  $\lambda > 0$ . Consider the update rule:

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i$$

Which of the following statements about the term  $1 - \alpha \frac{\lambda}{m}$  must be true?

A:  $1 - \alpha \frac{\lambda}{m} > 1$

B:  $1 - \alpha \frac{\lambda}{m} = 1$

~~C:  $1 - \alpha \frac{\lambda}{m} < 1$~~

D: None of the above.

$$\alpha \frac{\lambda}{m} \rightarrow \text{all positive} \rightarrow > 0$$

# REGULARIZED LINEAR REGRESSION

---

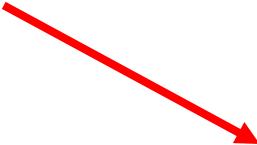
Repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_0^i$$

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i$$

(j = 1, 2, ..., n)

}



$\sim \theta_j * 0.99 \rightarrow$  shrinking  $\theta_j^2$

# REGULARIZED LINEAR REGRESSION

---

## Normal equation

$$X = \begin{bmatrix} (X^{(1)})^T \\ \dots \\ (X^{(m)})^T \end{bmatrix} \quad y = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix}$$

m x 1 vector

m x (n+1) matrix

$$\min_{\theta} J(\theta) \quad \rightarrow \quad \frac{\partial}{\partial \theta_j} J(\theta) \text{ set } 0$$

$$\theta = (X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix})^{-1} X^T y$$

(n+1) x (n+1) matrix  $\rightarrow$  Example is for n = 4

# REGULARIZED LINEAR REGRESSION

---

## Non-invertibility

Suppose  $m < n$  (#examples  $\leq$  #feature)

$\theta = (X^T X)^{-1} X^T y \rightarrow$  will be non-invertible/singular (may be non invertible if  $m = n$ )

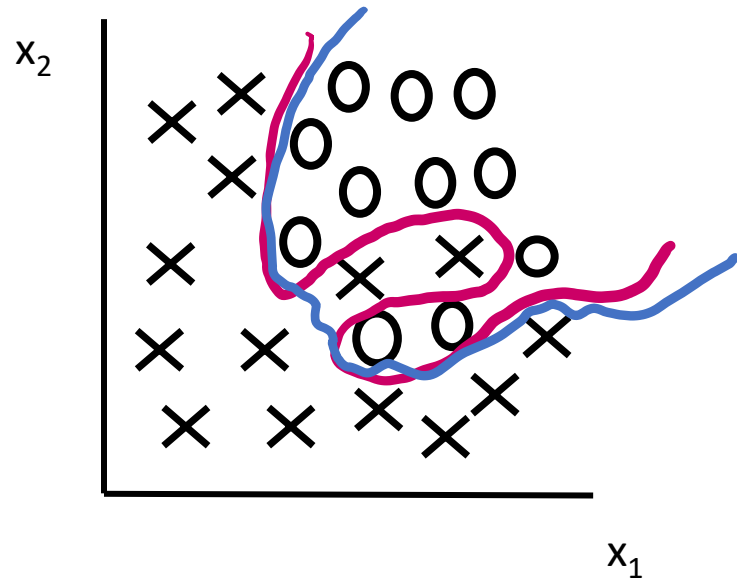
If  $\lambda > 0$

$$\theta = (X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix})^{-1} X^T y$$

$\rightarrow$  Is invertible

$\rightarrow$  Another benefit of regularization

# REGULARIZED LOGISTIC REGRESSION



$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

$$\text{Cost Function: } J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] \\ + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad \longrightarrow \quad \text{Penalizing } \theta_1, \theta_2, \dots, \theta_n \text{ for being too large}$$

# GRADIENT DESCENT

---

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$$

$$\theta_j := \theta_j - \alpha \left[ \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j + \frac{\lambda}{m} \theta_j \right]$$

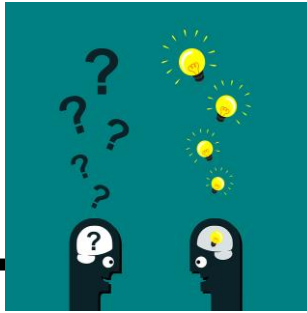
(j = 1, 2, 3, ..., n)

}

$\frac{\partial}{\partial \theta_j} J(\theta)$  (regularized)

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

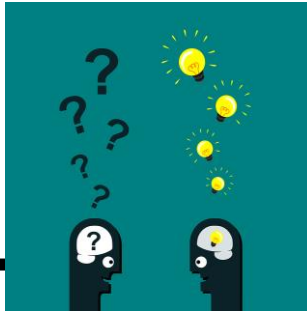
# QUESTION



When using regularized logistic regression, which of these is the best way to monitor whether gradient descent is working correctly?

- A: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right]$  as a function of the number of iterations and make sure it's decreasing.
- B: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] - \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.
- C: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.
- D: Plot  $\sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.

# QUESTION



When using regularized logistic regression, which of these is the best way to monitor whether gradient descent is working correctly?

A: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right]$  as a function of the number of iterations and make sure it's decreasing.

B: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] - \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.

~~C~~: Plot  $-\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.

D: Plot  $\sum_{j=1}^n \theta_j^2$  as a function of the number of iterations and make sure it's decreasing.



# ADVANCED OPTIMIZATION

---

```
function[jVal, gradient] = costFunction(theta)
```

```
jVal = [code to compute J(θ)] ;
```

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

```
gradient(1) = [code to compute  $\frac{\partial}{\partial \theta_0} J(\theta)$ ] ;
```

$$\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$$

```
gradient(2) = [code to compute  $\frac{\partial}{\partial \theta_1} J(\theta)$ ] ;
```

$$\left( \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_1 \right) + \frac{\lambda}{m} \theta_1$$

```
gradient(3) = [code to compute  $\frac{\partial}{\partial \theta_2} J(\theta)$ ] ;
```

$$\left( \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_2 \right) + \frac{\lambda}{m} \theta_2$$

...

```
gradient(n+1) = [code to compute  $\frac{\partial}{\partial \theta_n} J(\theta)$ ] ;
```

# WRAP-UP

---

## The Problem of Overfitting

- Consider the problem of predicting  $y$  from  $x \in \mathbb{R}$ .
- When talking about linear regression, the hypothesis  $y = \theta_0 + \theta_1 x_1$  always fits a linear line through our data. Often, we see that the data does not really lie on a straight line, and so the fit is not very good (Underfitting).
- Naively, it might seem that the more features we add, the better. However, there is also a danger in adding too many features: If our fitted curve passes through the data perfectly, we would not expect this to be a very good predictor in many cases (Overfitting) .
- Underfitting, or high bias, is when the form of our hypothesis function  $h$  maps poorly to the trend of the data. It is usually caused by a function that is too simple or uses too few features. At the other extreme, overfitting, or high variance, is caused by a hypothesis function that fits the available data but does not generalize well to predict new data. It is usually caused by a complicated function that creates a lot of unnecessary curves and angles unrelated to the data.

# WRAP-UP

---

## The Problem of Overfitting

- This terminology is applied to both linear and logistic regression. There are two main options to address the issue of overfitting:
- 1) Reduce the number of features:
  - Manually select which features to keep.
  - Use a model selection algorithm
- 2) Regularization
  - Keep all the features, but reduce the magnitude of parameters  $\theta_j$
  - Regularization works well when we have a lot of slightly useful features.

# WRAP-UP

---

## Cost Function

- If we have overfitting from our hypothesis function, we can reduce the weight that some of the terms in our function carry by increasing their cost.
- Say we wanted to make the following function more quadratic:
- $h_{\theta}(x) = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4$
- We'll want to eliminate the influence of  $\theta_3x^3$  and  $\theta_4x^4$ . Without actually getting rid of these features or changing the form of our hypothesis, we can instead modify our cost function:
- $\min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + 1000 * \theta_3^2 + 1000 * \theta_4^2$
- We've added two extra terms at the end to inflate the cost of  $\theta_3$  and  $\theta_4$ . Now, in order for the cost function to get close to zero, we will have to reduce the values of  $\theta_3$  and  $\theta_4$  to near zero. This will in turn greatly reduce the values of  $\theta_3x^3$  and  $\theta_4x^4$  in our hypothesis function. As a result, we see that the new hypothesis looks like a quadratic function.

# WRAP-UP

---

- **Cost Function**

- We could also regularize all of our theta parameters in a single summation as:

- $$J(\theta) \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^i) - y^i)^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

- The  $\lambda$  is the regularization parameter. It determines how much the costs of our theta parameters are inflated.
- Using the above cost function with the extra summation, we can smooth the output of our hypothesis function to reduce overfitting. If lambda is chosen to be too large, it may smooth out the function too much and cause underfitting.

# WRAP-UP

---

## Regularized Linear Regression

### Gradient Descent

We will modify our gradient descent function to separate out  $\theta_0$  from the rest of the parameters because we do not want to penalize  $\theta_0$ .

Repeat until convergence {

$$\begin{aligned}\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_0^i \\ \theta_j &:= \theta_j - \alpha \left[ \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i + \frac{\lambda}{m} \theta_j \right] \\ &\quad (j = 1, 2, \dots, n)\end{aligned}$$

}

The term  $\frac{\lambda}{m} \theta_j$  performs the regularization. Our update rule can also be represented as:

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^i) - y^i) * x_j^i$$

The first term in the above equation,  $1 - \alpha \frac{\lambda}{m}$  will always be less than 1. Intuitively you can see it as reducing the value of  $\theta_j$  by some amount on every update. Notice that the second term is now exactly the same as it was before.

# WRAP-UP

---

## Regularized Linear Regression Normal Equation

Now let's approach regularization using the alternate method of the non-iterative normal equation.

To add in regularization, the equation is the same as our original, except that we add another term inside the parentheses:

$$\theta = (X^T X + \lambda * L)^{-1} X^T y$$

with  $L =$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

$L$  is a matrix with 0 at the top left and 1's down the diagonal, with 0's everywhere else. It should have the dimension  $(n+1) \times (n+1)$ . Intuitively, this is the identity matrix (though we are not including  $x_0$ ), multiplied with a single real number  $\lambda$ .

Recall that if  $m < n$ , then  $X^T X$  is non-invertible. However, when we add the term  $\lambda \cdot L$ , then  $X^T X + \lambda \cdot L$  becomes invertible.

# WRAP-UP

---

## Regularized Logistic Regression

We can regularize logistic regression in a similar way that we regularize linear regression. As a result, we can avoid overfitting.

Recall that our cost function for logistic regression was:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right]$$

We can regularize this equation by adding a term to the end:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

The second sum,  $\sum_{j=1}^n \theta_j^2$  **means to explicitly exclude** the bias term,  $\theta_0$ . Thus, when computing the equation, we should continuously update the two following equations:

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$$

$$\theta_j := \theta_j - \alpha \left[ \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j + \frac{\lambda}{m} \theta_j \right]$$

(j = 1,2,3,...,n)

}



# QUIZ - QUESTION 1

---

You are training a classification model with logistic regression. Which of the following statements are true? Check all that apply.

- A: Adding a new feature to the model always results in equal or better performance on the training set.
- B: Adding many new features to the model helps prevent overfitting on the training set.
- C: Introducing regularization to the model always results in equal or better performance on the training set.
- D: Introducing regularization to the model always results in equal or better performance on examples not in the training set.

# QUIZ - QUESTION 1

---

You are training a classification model with logistic regression. Which of the following statements are true? Check all that apply.

- ☒ A: Adding a new feature to the model always results in equal or better performance on the training set.
- ☐ B: Adding many new features to the model helps prevent overfitting on the training set.
- ☐ C: Introducing regularization to the model always results in equal or better performance on the training set.
- ☐ D: Introducing regularization to the model always results in equal or better performance on examples not in the training set.

## QUIZ - QUESTION 2

---

Suppose you ran logistic regression twice, once with  $\lambda = 0$ , and once with  $\lambda = 1$ . One of the times, you got parameters  $\theta = [23.4; 37.9]$  and the other time you got  $\theta = [1.03; 0.28]$ . However, you forgot which value of  $\lambda$  corresponds to which value of  $\theta$ . Which one do you think corresponds to  $\lambda = 1$ ?

A:  $\theta = [23.4; 37.9]$

B:  $\theta = [1.03; 0.28]$

## QUIZ - QUESTION 2

---

Suppose you ran logistic regression twice, once with  $\lambda = 0$ , and once with  $\lambda = 1$ . One of the times, you got parameters  $\theta = [23.4; 37.9]$  and the other time you got  $\theta = [1.03; 0.28]$ . However, you forgot which value of  $\lambda$  corresponds to which value of  $\theta$ . Which one do you think corresponds to  $\lambda = 1$ ?

A:  $\theta = [23.4; 37.9]$

~~B:  $\theta = [1.03; 0.28]$~~

# QUIZ - QUESTION 3

---

Which of the following statements about regularization are true? Check all that apply.

A: Because regularization causes  $J(\theta)$  to no longer be convex, gradient descent may not always converge to the global minimum (when  $\lambda > 0$  and when using an appropriate learning rate  $\alpha$ ).

B: Using too large a value of  $\lambda$  can cause your hypothesis to underfit the data.

C: Using a very large value of  $\lambda$  cannot hurt the performance of your hypothesis; the only reason we do not set  $\lambda$  to be too large is to avoid numerical problems.

D: Because logistic regression outputs values  $0 \leq h_{\theta}(x) \leq 1$  its range of output values can only be "shrunk" slightly by regularization anyway, so regularization is generally not helpful for it.

# QUIZ - QUESTION 3

---

Which of the following statements about regularization are true? Check all that apply.

A: Because regularization causes  $J(\theta)$  to no longer be convex, gradient descent may not always converge to the global minimum (when  $\lambda > 0$  and when using an appropriate learning rate  $\alpha$ ).

☒ B: Using too large a value of  $\lambda$  can cause your hypothesis to underfit the data.

C: Using a very large value of  $\lambda$  cannot hurt the performance of your hypothesis; the only reason we do not set  $\lambda$  to be too large is to avoid numerical problems.

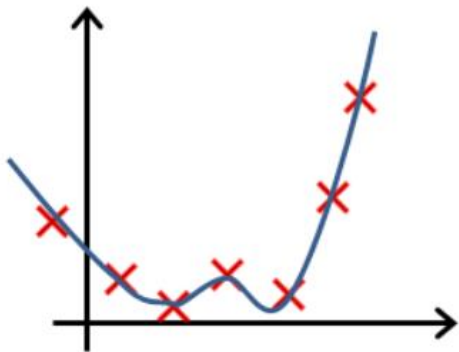
D: Because logistic regression outputs values  $0 \leq h_{\theta}(x) \leq 1$  its range of output values can only be "shrunk" slightly by regularization anyway, so regularization is generally not helpful for it.

# QUIZ - QUESTION 4

---

In which one of the following figures do you think the hypothesis has overfit the training set?

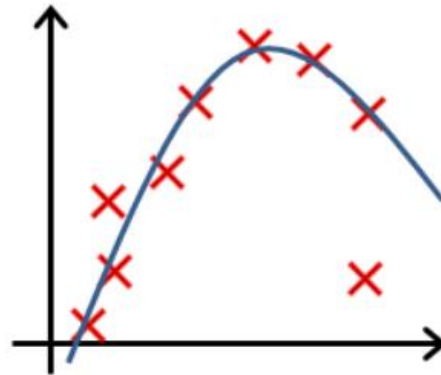
A:



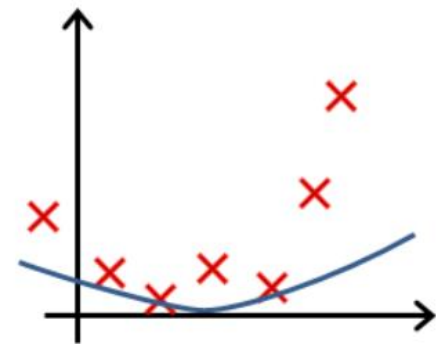
B:



C:



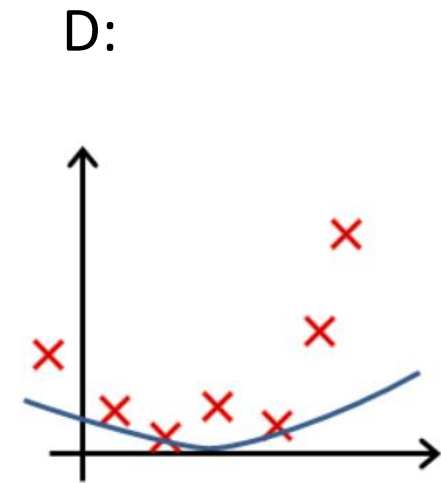
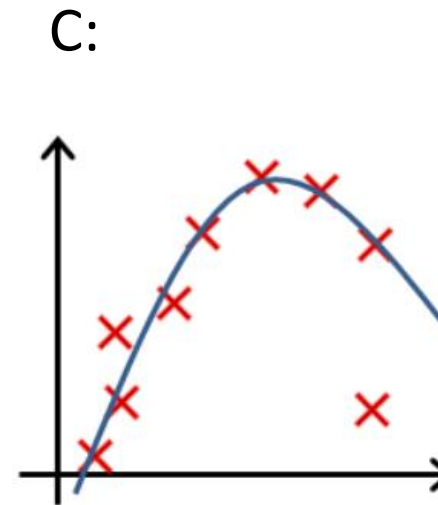
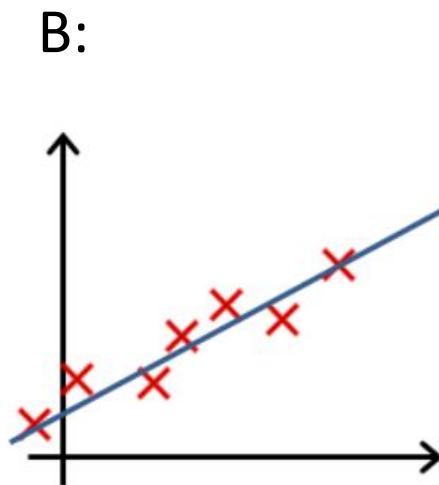
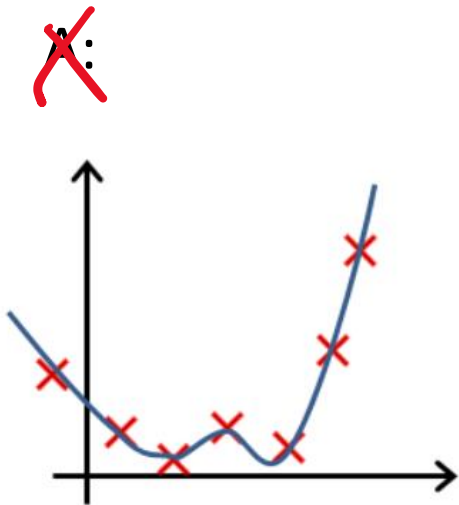
D:



# QUIZ - QUESTION 4

---

In which one of the following figures do you think the hypothesis has **overfit** the training set?



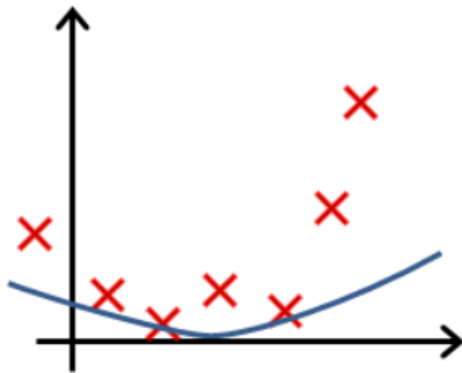


# QUIZ - QUESTION 5

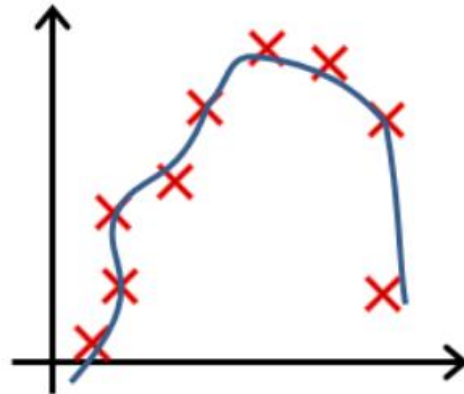
---

In which one of the following figures do you think the hypothesis has underfit the training set?

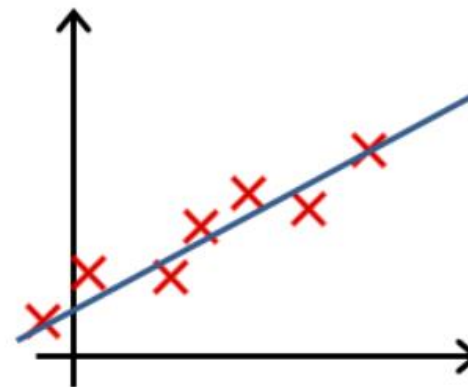
A:



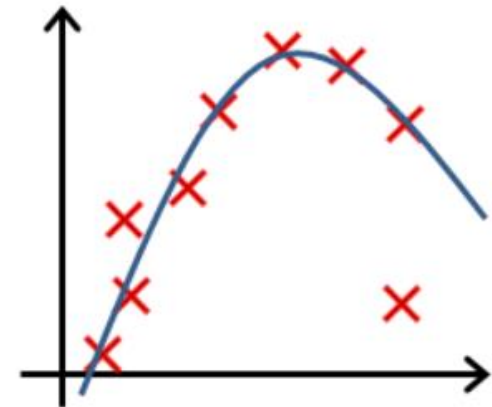
B:



C:



D:

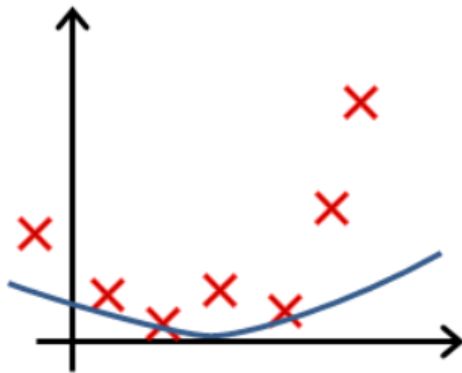


# QUIZ - QUESTION 5

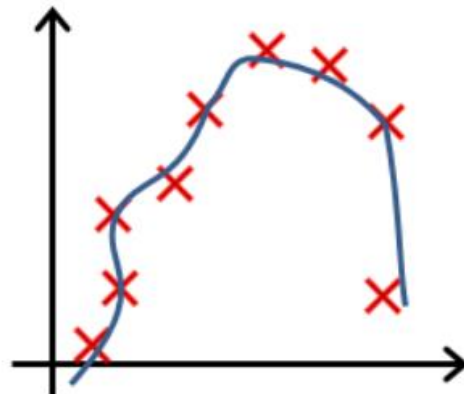
---

In which one of the following figures do you think the hypothesis has **underfit** the training set?

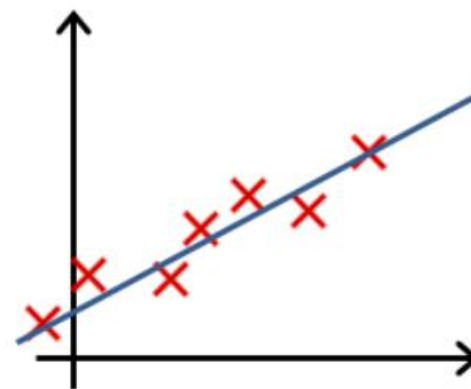
~~A:~~



B:



C:



D:

