Week 3

Classification

Question	Answer "y"
Is this email spam?	no yes
Is the transaction fraudulent?	no yes
Is the tumor malignant?	no yes

y can only be one of two values

"binary classification"

class = category

false true

0 1

useful for classification

"negative class"

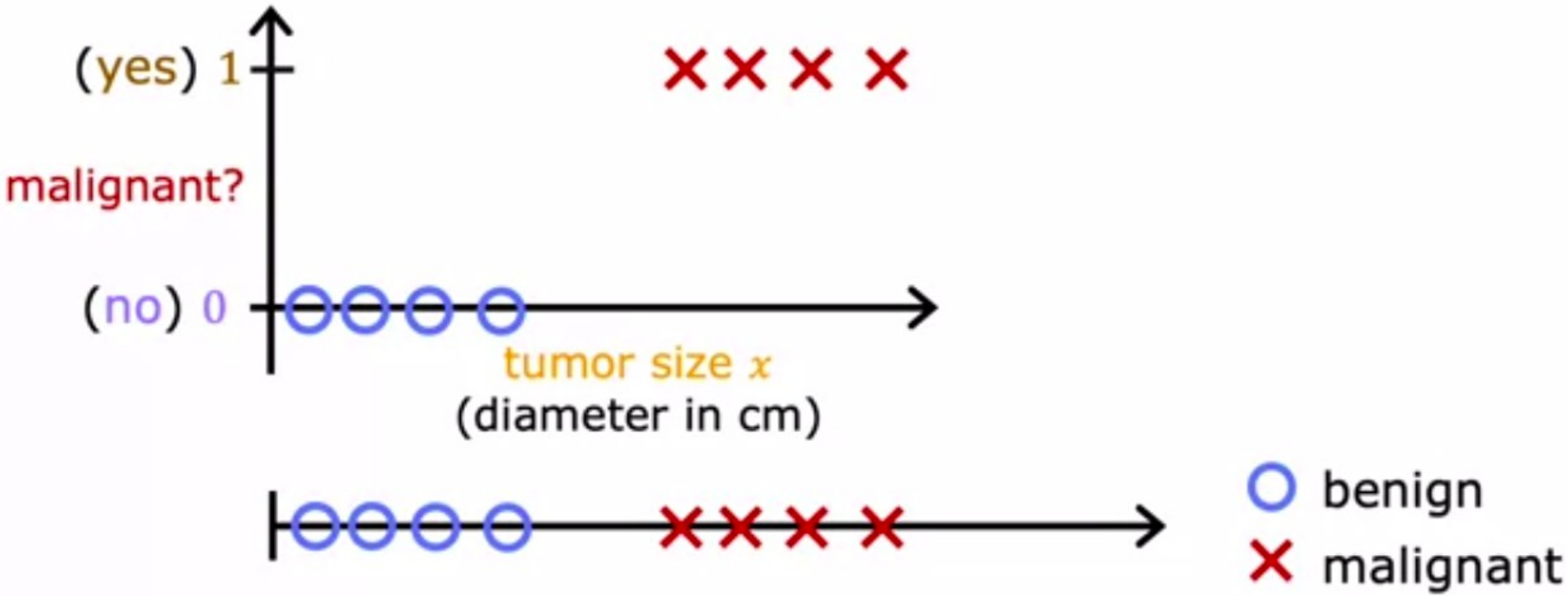
≠ "bad"

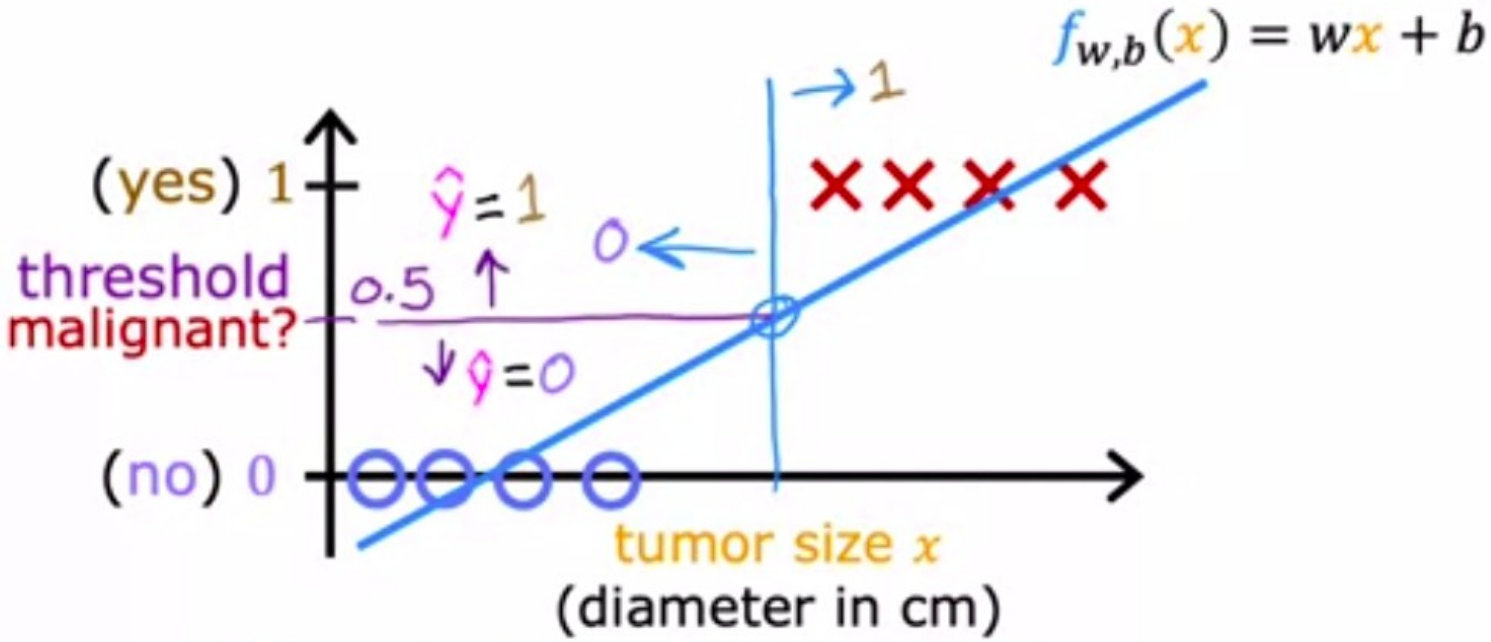
absence

"positive class"

≠ "good"

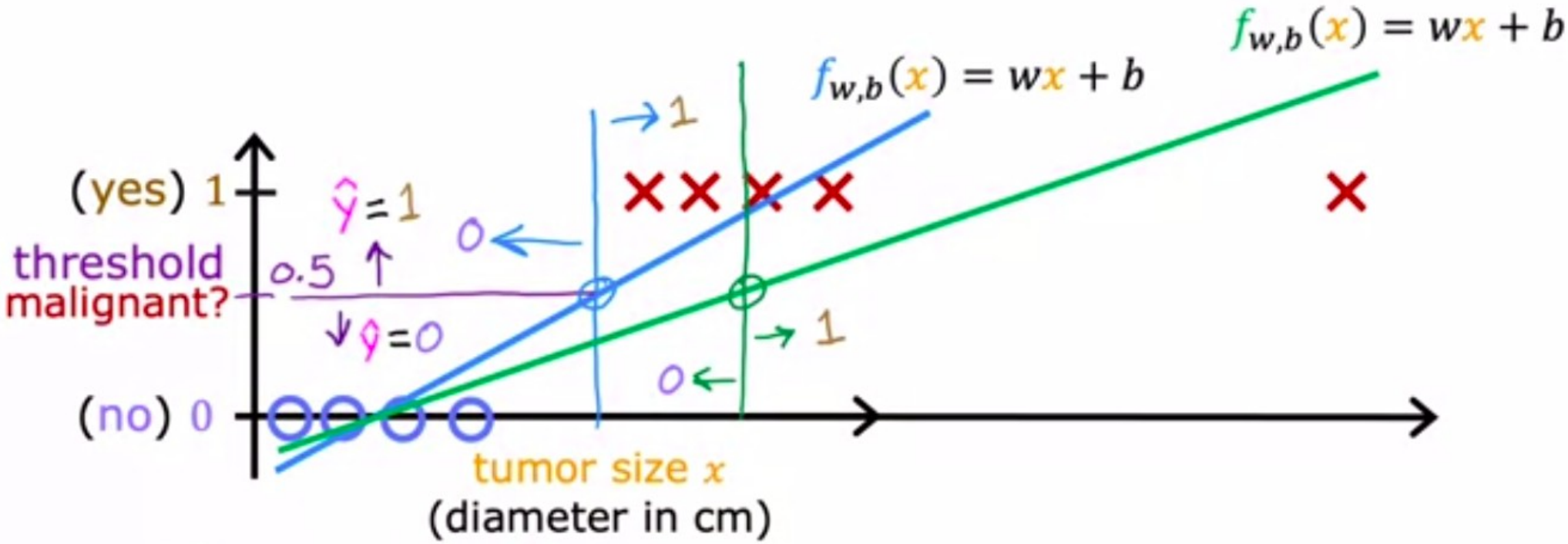
presence


(yes) 1
malignant?
(no) 0
tumor size x
(diameter in cm)
benign
malignant



if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$

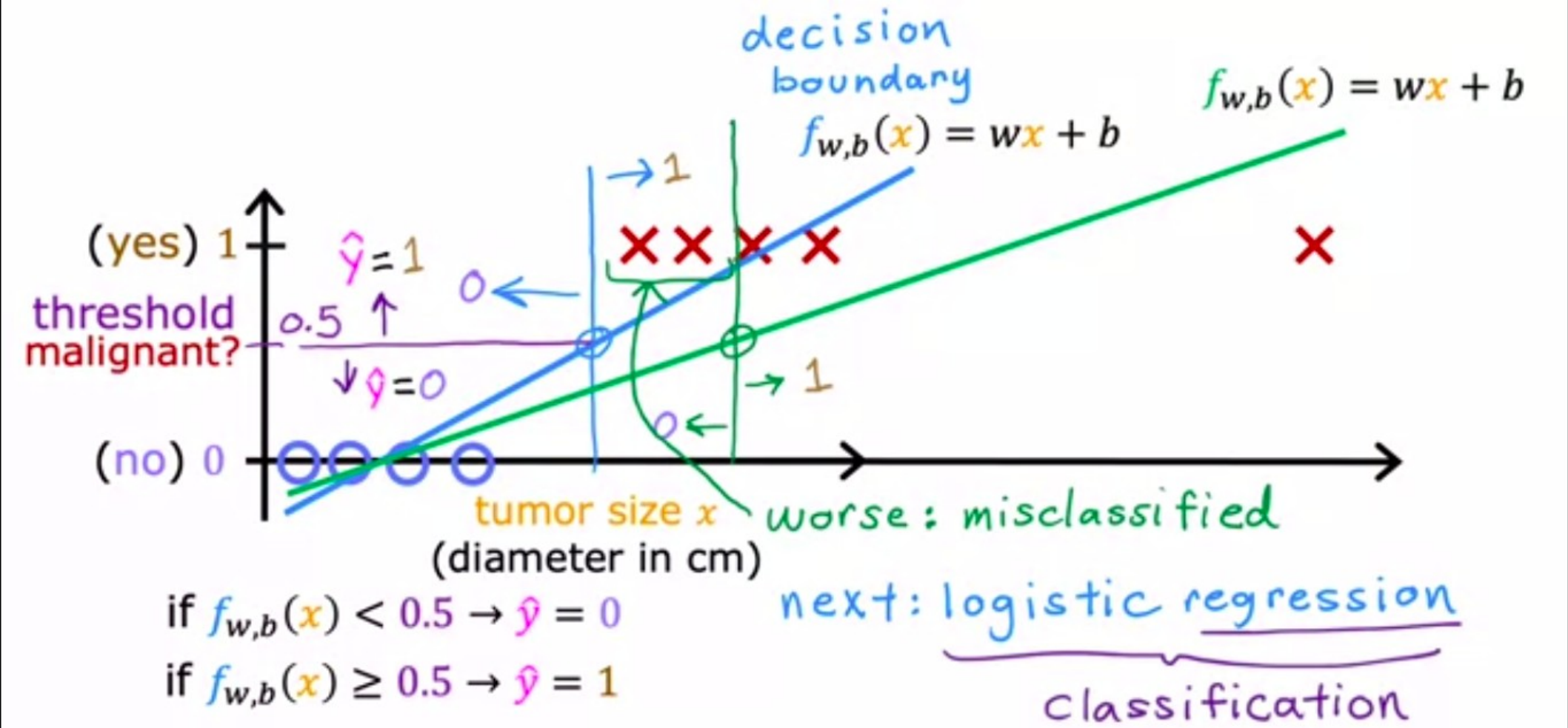
if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$



if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$

if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

This isn't what we want because adding that example way to the right shouldn't



Don't be confused by the name which was given for historical reasons.

Question

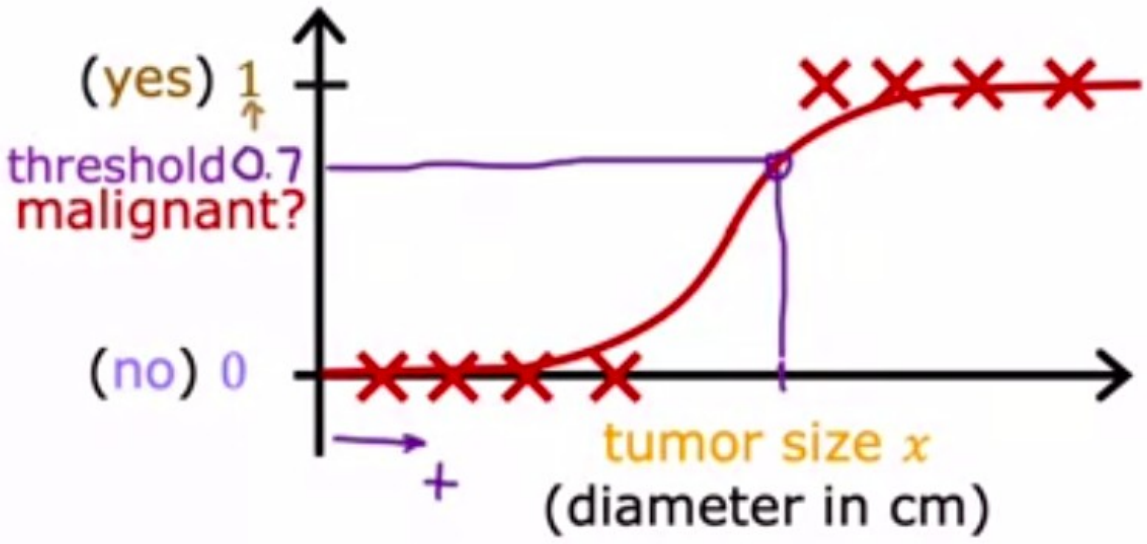
Which of the following is an example of a classification task?

- ( ) Estimate the weight of a cat based on its height.
- (•) Decide if an animal is a cat or not a cat.

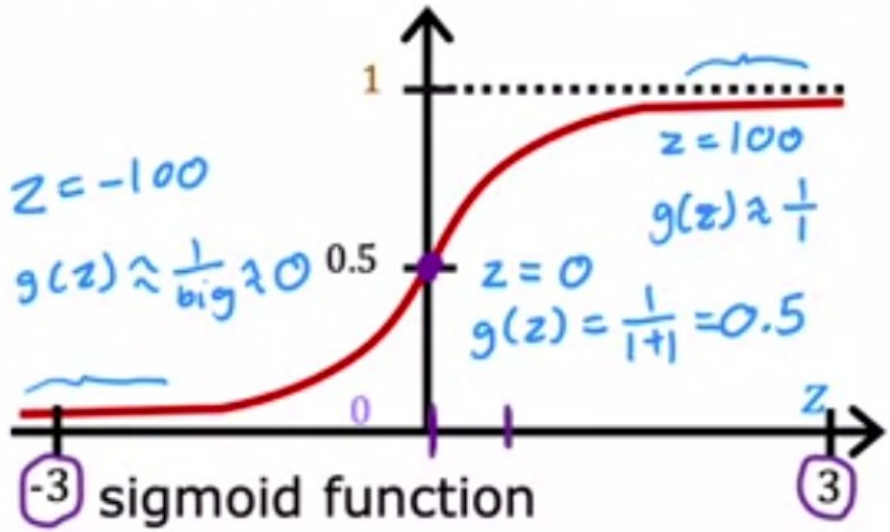
✓ **Correct**

Correct: This is an example of *binary classification* where there are two possible classes (True/False or Yes/No or 1/0).

Skip Continue



Want outputs between 0 and 1



sigmoid function

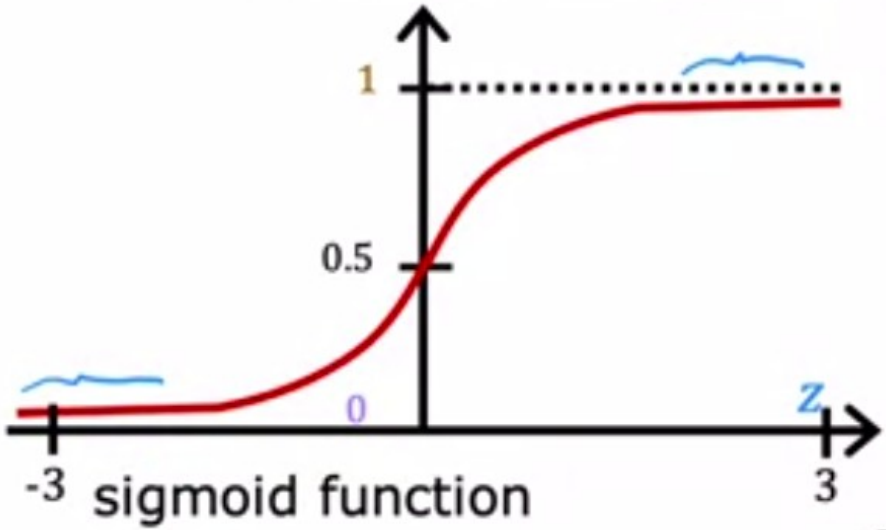
logistic function

outputs between 0 and 1

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

so that's why it passes the vertical axis at 0.5.

Want outputs between 0 and 1



sigmoid function

logistic function

outputs between 0 and 1

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

$$f_{\vec{w},b}(\vec{x})$$

$$z = \vec{w} \cdot \vec{x} + b$$

$$z$$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w} \cdot \vec{x} + b}_{z}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

"logistic regression"

Interpretation of logistic regression output

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}\cdot\vec{x}+b)}}$$

"probability" that class is 1

$$f_{\vec{w},b}(\vec{x}) = P(y = 1|\vec{x}; \vec{w}, b)$$

Probability that y is 1,
given input $\vec{x}$, parameters $\vec{w}, b$

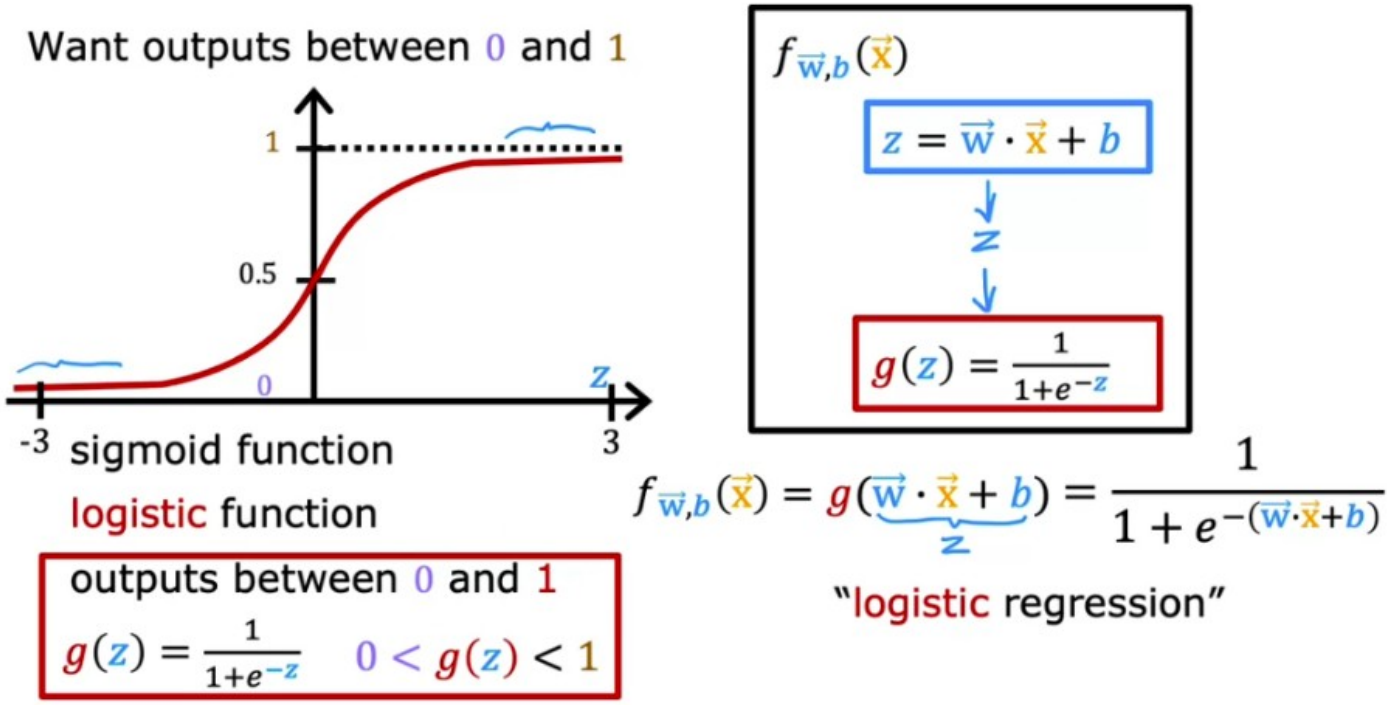
Example:
x is "tumor size"
y is 0 (not malignant)
or 1 (malignant)

$$P(y = 0) + P(y = 1) = 1$$

$$f_{\vec{w},b}(\vec{x}) = 0.7$$

70% chance that y is 1

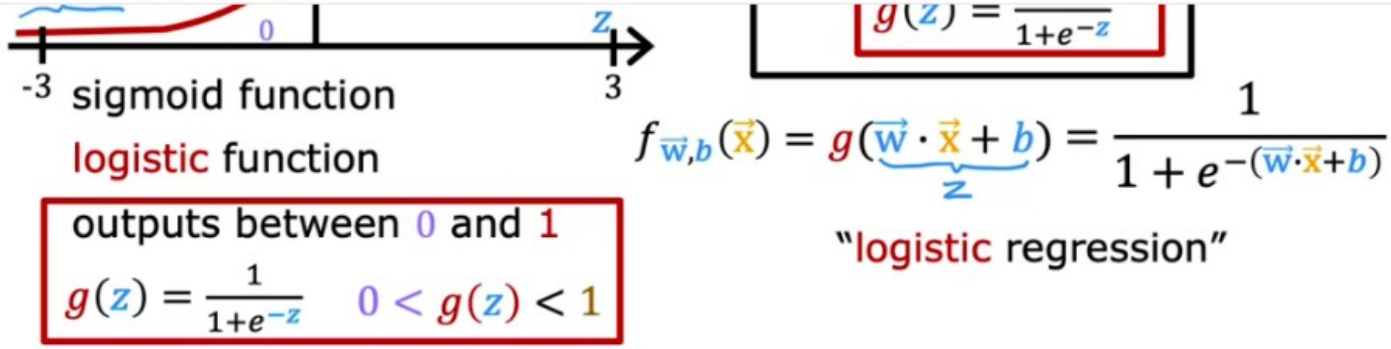
Question



Recall the sigmoid function is $g(z) = \frac{1}{1+e^{-z}}$

Skip Continue

Question



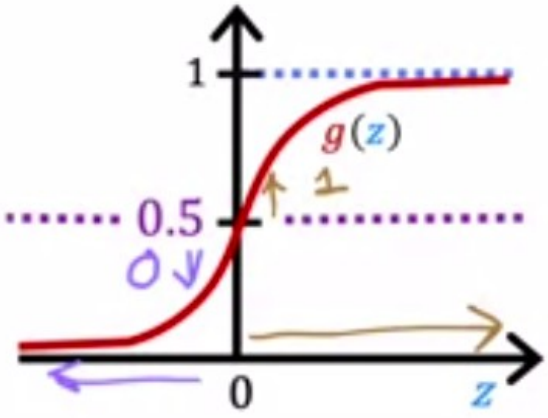
Recall the sigmoid function is $g(z) = \frac{1}{1+e^{-z}}$

If z is a large negative number then:

- $g(z)$ is near negative one (-1)
- ◉ $g(z)$ is near zero

✓ **Correct**
Say z = -100. e^{-z} is then e^{100}, a really big positive number. So, $g(z) = \frac{1}{1+\text{a big positive number}}$ or about 0

Skip Continue



$f_{\vec{w},b}(\vec{x})$

$z = \vec{w} \cdot \vec{x} + b$

z

$g(z) = \frac{1}{1+e^{-z}}$

$$f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w} \cdot \vec{x} + b}_{z}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

$= P(y = 1 | x; \vec{w}, b)$ 0.7 0.3

0 or 1?

Is $f_{\vec{w},b}(\vec{x}) \geq \overbrace{0.5}^{\text{threshold}}$?

Yes: $\hat{y} = 1$ No: $\hat{y} = 0$

When is $f_{\vec{w},b}(\vec{x}) \geq 0.5$?

$g(z) \geq 0.5$

$z \geq 0$

$\vec{w} \cdot \vec{x} + b \geq 0$ $\vec{w} \cdot \vec{x} + b < 0$

$\hat{y} = 1$ $\hat{y} = 0$

Decision boundary

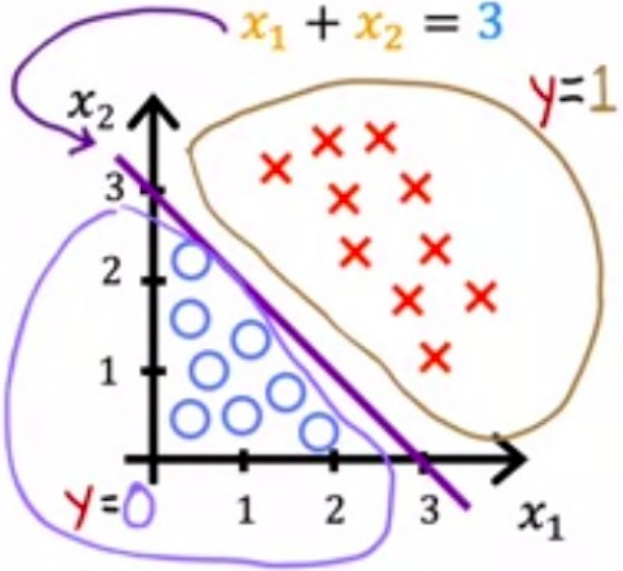
$$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1 x_1 + w_2 x_2 + b)$$

$w_1 = 1$, $w_2 = 1$, $b = -3$

Decision boundary $z = \vec{w} \cdot \vec{x} + b = 0$

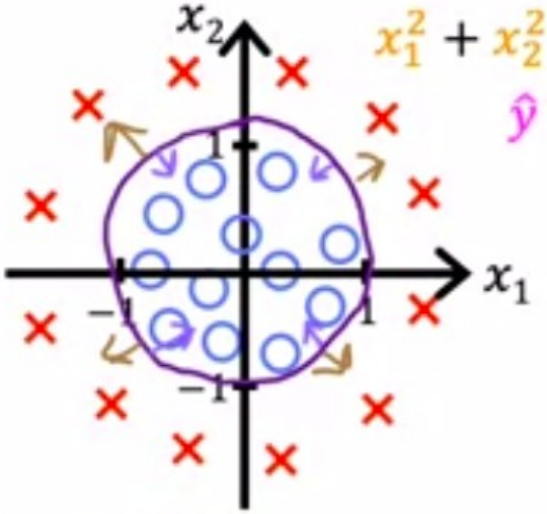
$z = x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$



the decision boundary would be a different line.

Non-linear decision boundaries



$x_1^2 + x_2^2 \geq 1$

$\hat{y} = 1$

$f_{\vec{w},b}(\vec{x}) = g(z) = g(\overbrace{w_1 x_1^2 + w_2 x_2^2 + b}^{z})$

$w_1 = 1$, $w_2 = 1$, $b = -1$

decision boundary $z = x_1^2 + x_2^2 - 1 = 0$

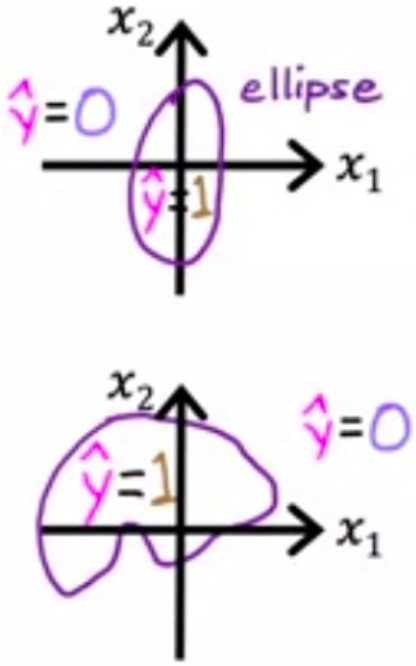
$x_1^2 + x_2^2 = 1$

$x_1^2 + x_2^2 < 1$

$\hat{y} = 0$

the circle and that's when you predict y to be 0.

Non-linear decision boundaries



$$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 + w_6x_1^3 + \cdots + b)$$

will always be a straight line.

Question

Let's say you are creating a tumor detection algorithm. Your algorithm will be used to flag potential tumors for future inspection by a specialist. What value should you use for a threshold?

- ○ High, say a threshold of 0.9?
- ◉ Low, say a threshold of 0.2?

✓ **Correct**

Correct: You would not want to miss a potential tumor, so you will want a low threshold. A specialist will review the output of the algorithm which reduces the possibility of a 'false positive'. The key point of this question is to note that the threshold value does not need to be 0.5.

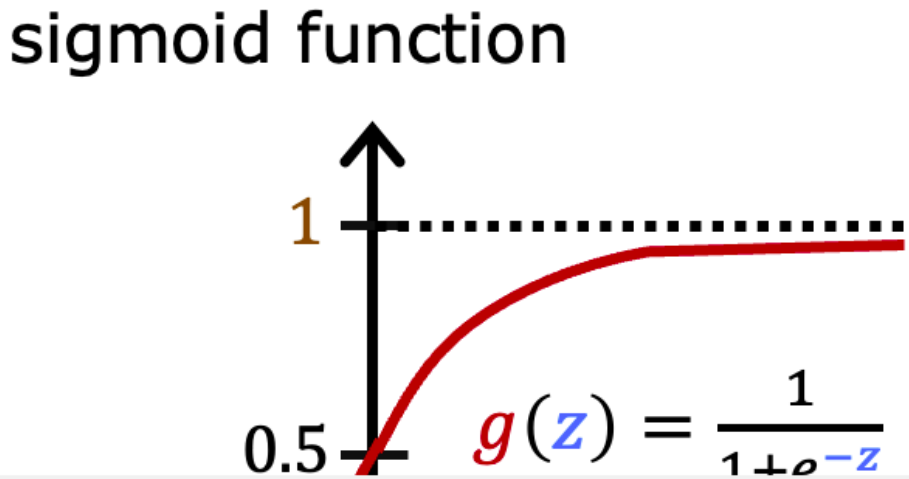
Skip Continue

1. Which is an example of a classification task? **1 / 1 point**

- ○ Based on a patient's age and blood pressure, determine how much blood pressure medication (measured in milligrams) the patient should be prescribed.
- ○ Based on a patient's blood pressure, determine how much blood pressure medication (a dosage measured in milligrams) the patient should be prescribed.
- ◉ Based on the size of each tumor, determine if each tumor is malignant (cancerous) or not.

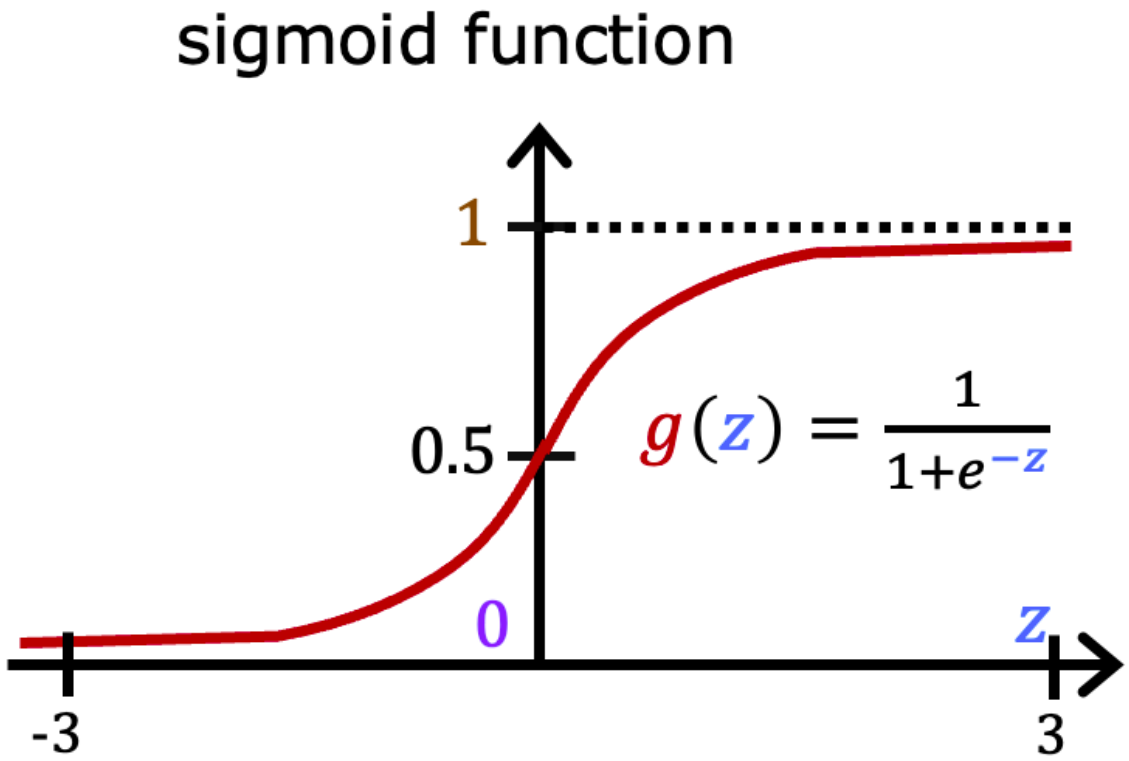
Correct
This task predicts one of two classes, malignant or not malignant.

2. Recall the sigmoid function is $g(z) = \frac{1}{1+e^{-z}}$ **1 / 1 point**

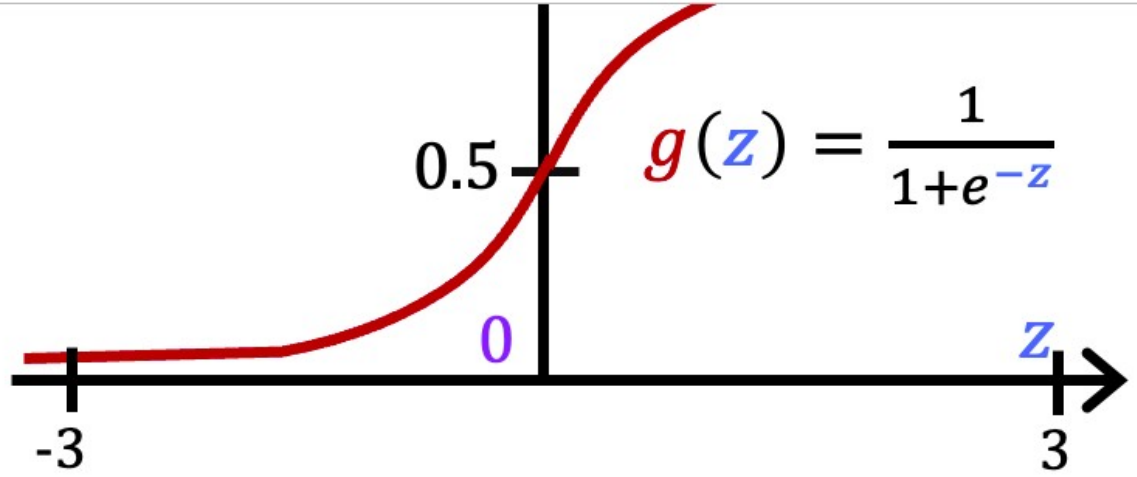


2. Recall the sigmoid function is $g(z) = \frac{1}{1+e^{-z}}$

1 / 1 point



If z is a large positive number, then:



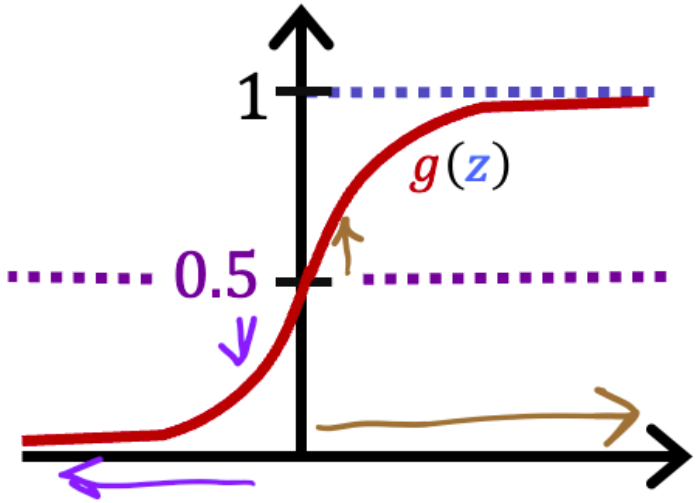
If z is a large positive number, then:

- ○ $g(z)$ will be near 0.5
- ○ $g(z)$ will be near zero (0)
- ○ $g(z)$ is near negative one (-1)
- ◉ $g(z)$ is near one (1)

Correct
Say z = $+100$. So e^{-z} is then e^{-100}, a really small positive number. So, $g(z) = \frac{1}{1+\text{a small positive number}}$ which is close to 1

3.

1 / 1 point



A cat photo classification model predicts 1 if it's a cat, and 0 if it's not a cat. For a particular photograph, the logistic regression model outputs $g(z)$ (a number between 0 and 1). Which of these would be a reasonable criteria to decide whether to predict if it's a cat?

- (●) Predict it is a cat if g(z) >= 0.5
- ( ) Predict it is a cat if g(z) = 0.5
- ( ) Predict it is a cat if g(z) < 0.7
- ( ) Predict it is a cat if g(z) < 0.5

Correct

Think of g(z) as the probability that the photo is of a cat. When this number is at or above the threshold of 0.5, predict that it is a cat.

A cat photo classification model predicts 1 if it's a cat, and 0 if it's not a cat. For a particular photograph, the logistic regression model outputs $g(z)$ (a number between 0 and 1). Which of these would be a reasonable criteria to decide whether to predict if it's a cat?

- (●) Predict it is a cat if g(z) >= 0.5
- ( ) Predict it is a cat if g(z) = 0.5
- ( ) Predict it is a cat if g(z) < 0.7
- ( ) Predict it is a cat if g(z) < 0.5

Correct
Think of g(z) as the probability that the photo is of a cat. When this number is at or above the threshold of 0.5, predict that it is a cat.

4. **1 / 1 point**

True/False? No matter what features you use (including if you use polynomial features), the decision boundary learned by logistic regression will be a linear decision boundary.

- (●) False
- ( ) True

Correct
The decision boundary can also be non-linear, as described in the lectures.

Training set

	tumor size (cm) x_1	...	patient's age x_n	malignant? y
$i=1$	10		52	1
	2		73	0
	5		55	0
	12		49	1
$i=m$	...		...	...

$i = 1, \ldots, m$ ← training examples

$j = 1, \ldots, n$ ← features

target y is 0 or 1

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

How to choose $\vec{w} = [w_1 \quad w_2 \quad \cdots \quad w_n]$ and b?

how can you choose parameters w and b?

Squared error cost

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2$$

loss

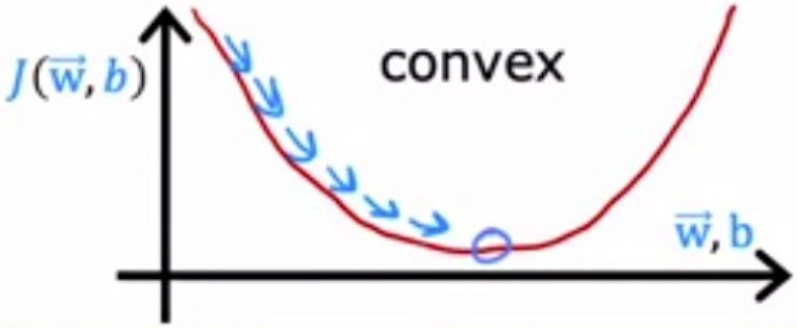
$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)$$

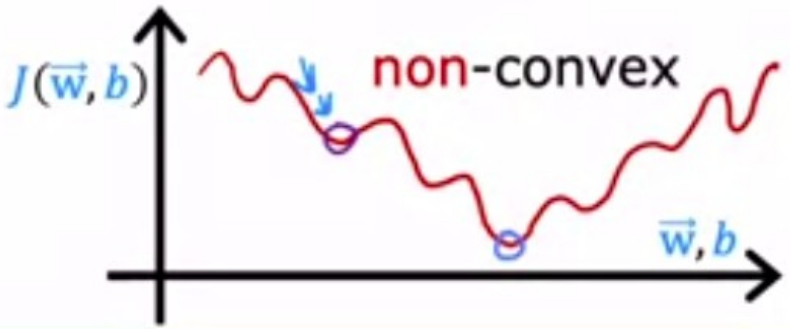
linear regression

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

logistic regression

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}\cdot\vec{x}+b)}}$$



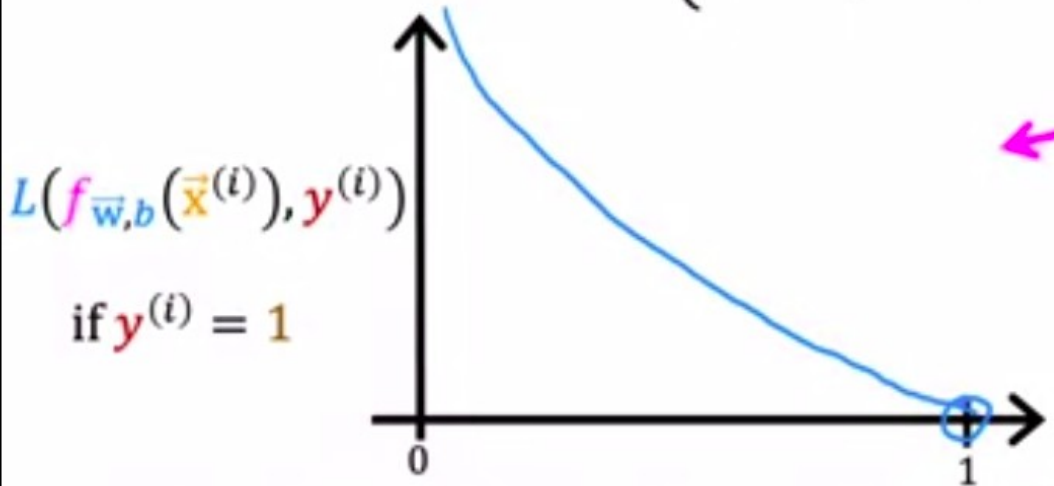


Logistic loss function

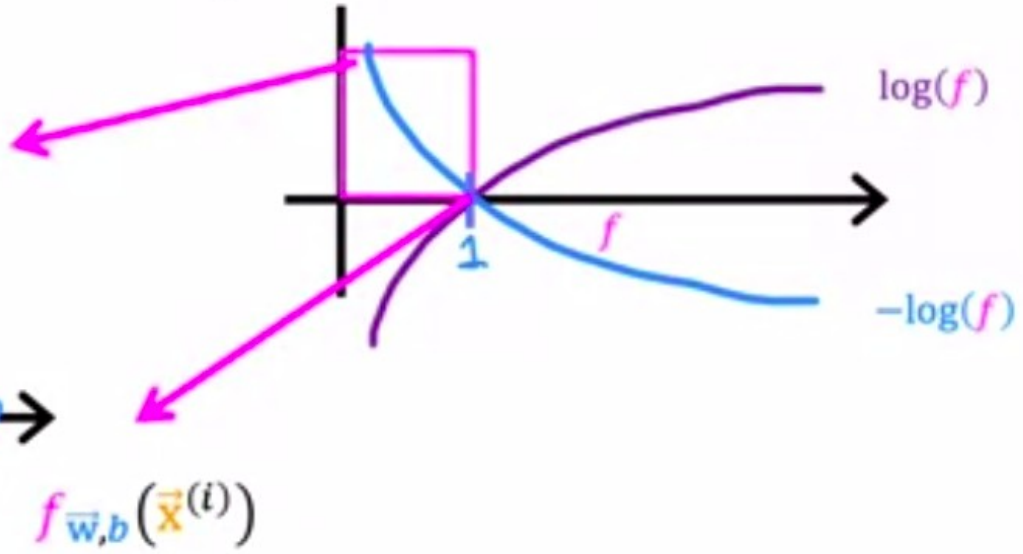
$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

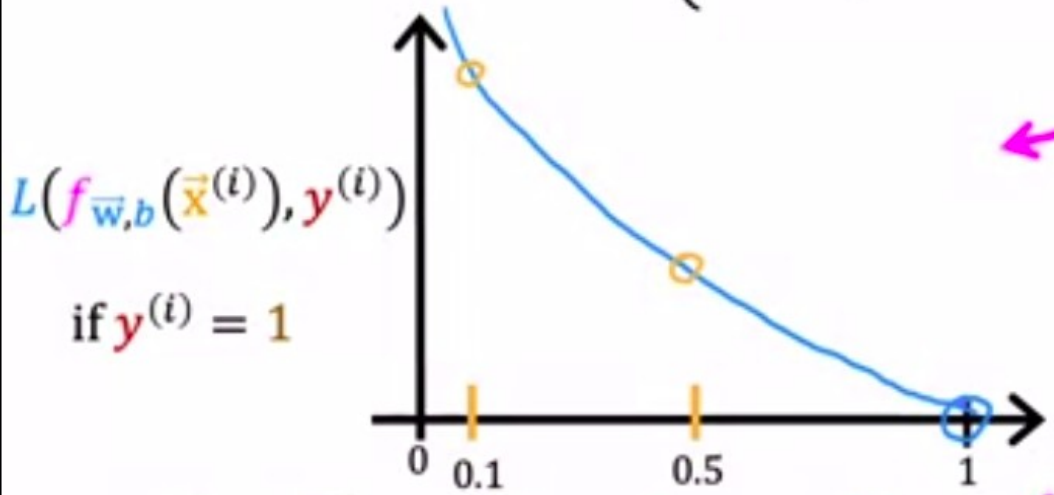


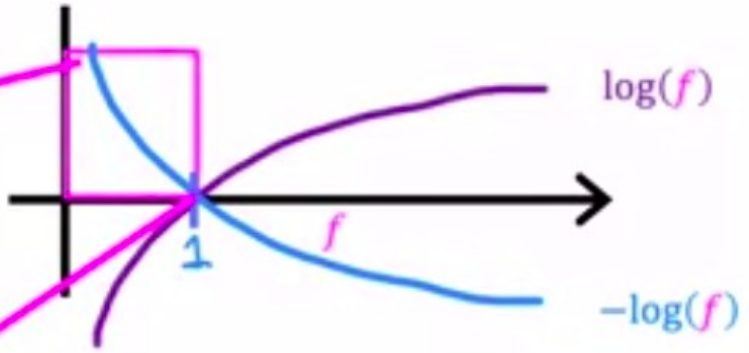
As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$ then loss $\to 0$



Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$





As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$ then loss $\to 0$

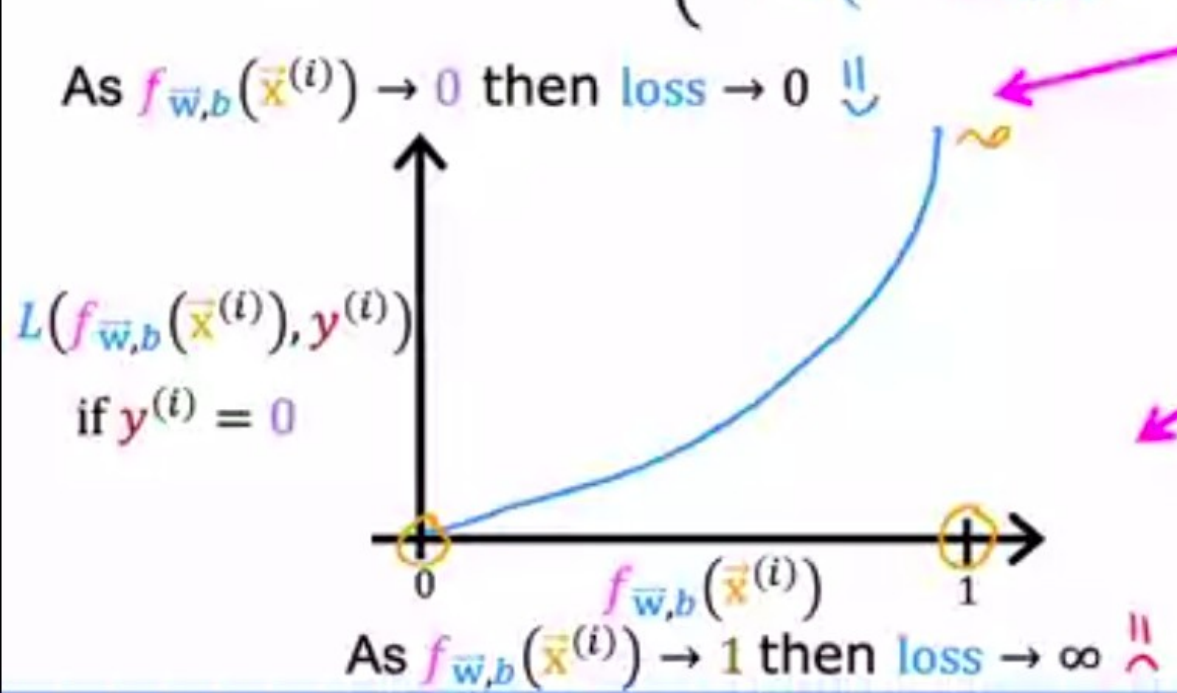
As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$ then loss $\to \infty$

Loss is lowest when $f_{\vec{w},b}\left(\vec{x}^{(i)}\right)$ predicts close to true label $y^{(i)}$.

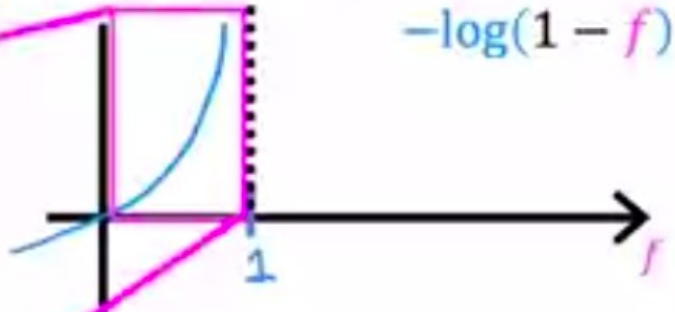
Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$ then loss $\to 0$

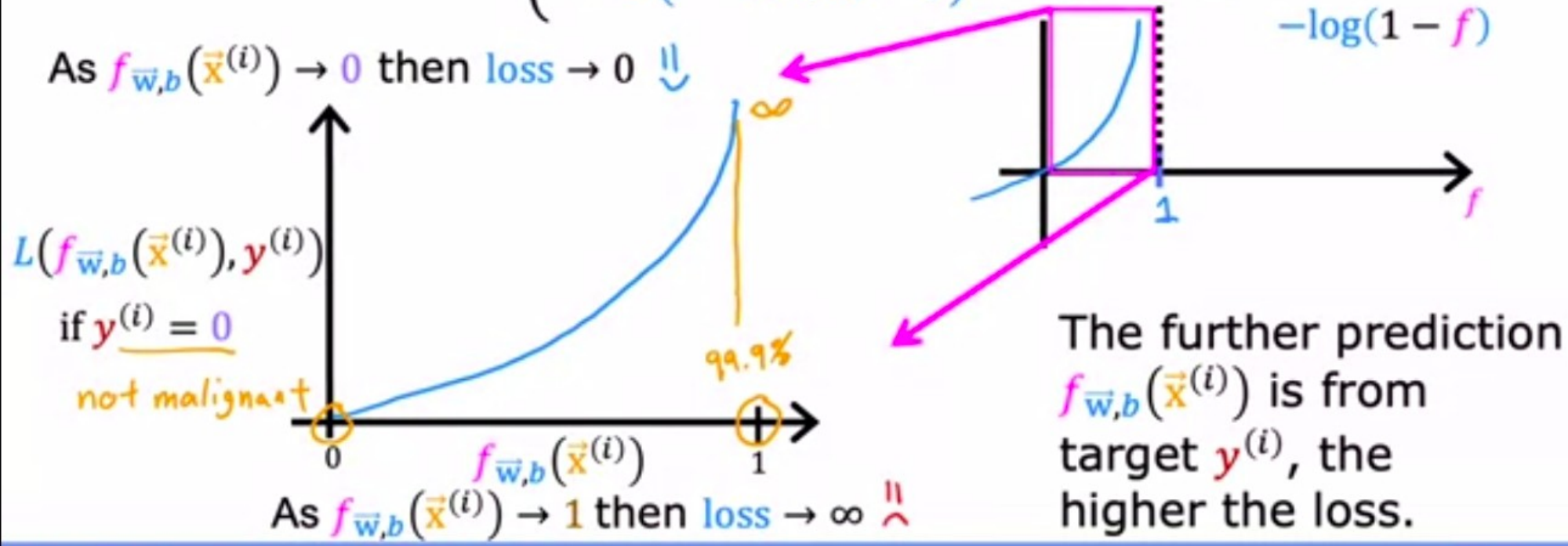


As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$ then loss $\to \infty$



Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$



The further prediction $f_{\vec{w},b}\left(\vec{x}^{(i)}\right)$ is from target $y^{(i)}$, the higher the loss.

Question

Why is the squared error cost not used in logistic regression?

- (●) The non-linear nature of the model results in a “wiggly”, non-convex cost function with many potential local minima.
- ( ) The mean squared error is used for logistic regression.

✓ **Correct**
If using the mean squared error for logistic regression, the cost function is "non-convex", so it's more difficult for gradient descent to find an optimal value for the parameters w and b.

Skip Continue

Cost

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m} L\underbrace{\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)}_{\text{loss}}$$

$$= \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 0 \end{cases}$$

convex
↳ can reach a global minimum

find w, b that minimize cost J

Simplified loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

1 $\qquad$ $(1-1)$ $\qquad$ 0

if $y^{(i)} = 1$:

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -1 \log\left(f(\vec{x})\right)$$

Simplified loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

$y^{(i)}$ → 0 ; $(1 - y^{(i)})$ → $(1 - 0)$

if $y^{(i)} = 1$:

$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -\log\left(f(\vec{x})\right)$

if $y^{(i)} = 0$:

$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) =$ $-(1-0)\log(1 - f(\vec{x}))$

Simplified cost function

loss

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

convex (single global minimum)

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left[L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right]$$

maximum likelihood
(don't worry about it!)

Question

For the simplified loss function:

$$L(f_{\mathbf{w},b}(\mathbf{x}^{(i)}), y^{(i)}) = -y^{(i)} log(f_{\mathbf{w},b}(\mathbf{x}^{(i)})) - (1 - y^{(i)}) log(1 - f_{\mathbf{w},b}(\mathbf{x}^{(i)}))$$

if the target $y^{(i)} = 1$, then what does this expression simplify to?

- (●) $-log(f_{\vec{w},b}(\mathbf{x}^{(i)})$
- ( ) $-log(1 - f_{\vec{\mathbf{w}},b}(\mathbf{x}^{(i)}))$

✓ **Correct**
The second term of the expression is reduced to zero when the target equals 1.

Skip Continue

Practice quiz: Cost function for logistic regression

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

1.

1 / 1 point

$$\overbrace{J(\vec{w}, b)}^{?} = \frac{1}{m}\sum_{i=1}^{m} \underbrace{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)}_{?}$$

In this lecture series, "cost" and "loss" have distinct meanings. Which one applies to a single training example?

- [x] Loss

Correct

In these lectures, loss is calculated on a single training example. It is worth noting that this definition is not universal. Other lecture series may have a different definition.

- [ ] Cost
- [ ] Both Loss and Cost
- [ ] Neither Loss nor Cost

Practice quiz: Cost function for logistic regression

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

Simplified loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

For the simplified loss function, if the label $y^{(i)} = 0$, then what does this expression simplify to?

- $\log(1 - f_{\vec{w},b}(\mathbf{x}^{(i)})) + log(1 - f_{\vec{w},b}(\mathbf{x}^{(i)}))$
- $\log(f_{\vec{w},b}(\mathbf{x}^{(i)})$
- $-\log(1 - f_{\vec{w},b}(\mathbf{x}^{(i)})) - log(1 - f_{\vec{w},b}(\mathbf{x}^{(i)}))$
- $-\log(1 - f_{\vec{w},b}(\mathbf{x}^{(i)}))$

Correct

When $y^{(i)} = 0$, the first term reduces to zero.

Training logistic regression

Find $\vec{w}, b$

Given new $\vec{x}$, output $f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

$$P(y = 1|\vec{x}; \vec{w}, b)$$

the probability that
the label y is one.

Gradient descent

cost

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

repeat {

$j = 1 \ldots n$

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

$$\frac{\partial}{\partial b} J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

} simultaneous updates

This gives you gradient descent for logistic regression.

Gradient descent for logistic regression

repeat {

looks like linear regression!

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \right]$$

} simultaneous updates

Same concepts:
- Monitor gradient descent (learning curve)
- Vectorized implementation
- Feature scaling

Linear regression $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Logistic regression $f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$

Gradient descent for logistic regression

repeat {

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \right]$$

} simultaneous updates

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Which of the following two statements is a more accurate statement about gradient descent for logistic regression?

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \right]$$

} simultaneous updates

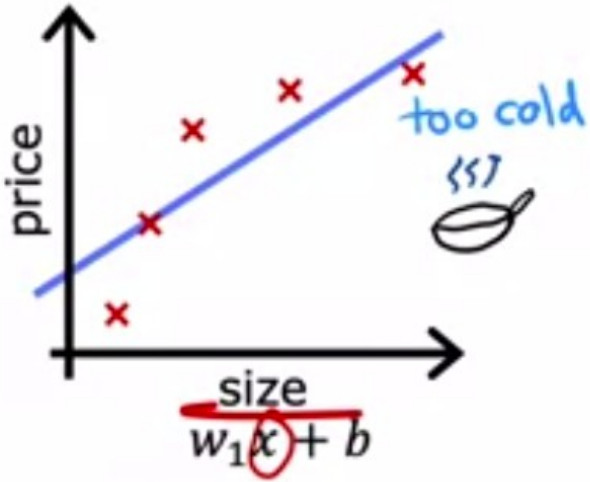
$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Which of the following two statements is a more accurate statement about gradient descent for logistic regression?

- The update steps are identical to the update steps for linear regression.
- The update steps look like the update steps for linear regression, but the definition of $f_{\vec{w},b}(\mathbf{x}^{(i)})$ is different.

Correct
For logistic regression, $f_{\vec{w},b}(\mathbf{x}^{(i)})$ is the sigmoid function instead of a straight line.

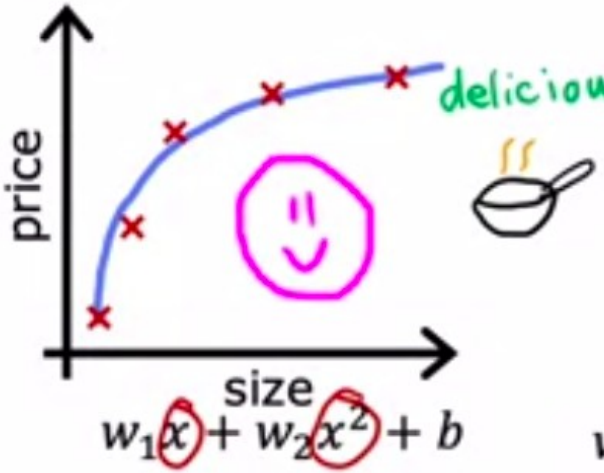
Regression example



underfit

- Does not fit the training set well

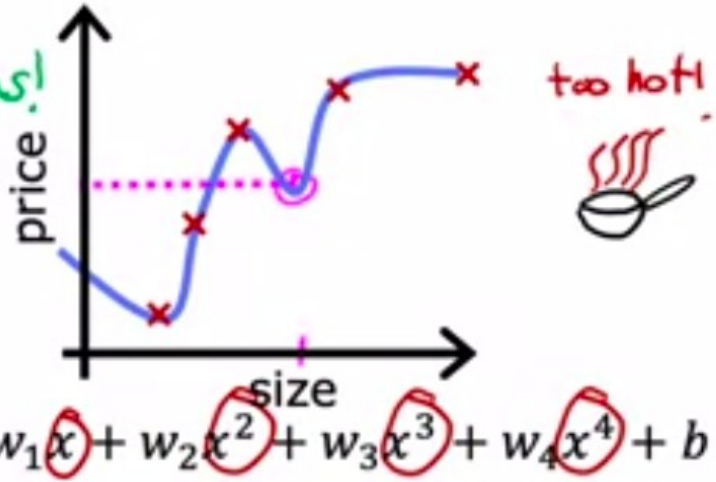
high bias



just right

- Fits training set pretty well

generalization



overfit

- Fits the training set extremely well

high variance

too many features like

Question

Our goal when creating a model is to be able to use the model to predict outcomes correctly for **new examples**. A model which does this is said to **generalize** well.

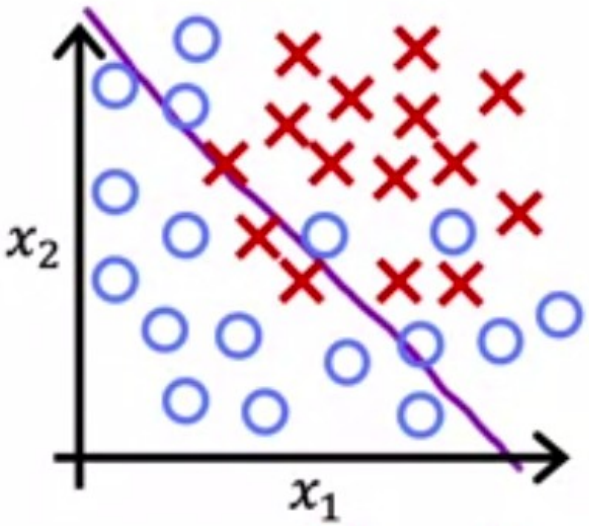
When a model fits the training data well but does not work well with new examples that are not in the training set, this is an example of:

- (•) Overfitting (high variance)
- ( ) None of the above
- ( ) Underfitting (high bias)
- ( ) A model that generalizes well (neither high variance nor high bias)

✓ **Correct**
This is when the model does not generalize well to new examples.

Skip Continue

Classification

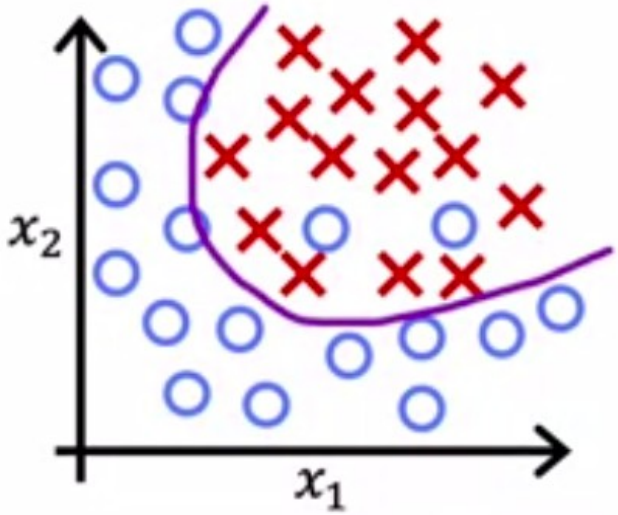


$$z = w_1 x_1 + w_2 x_2 + b$$

$$f_{\vec{w},b}(\vec{x}) = g(z)$$

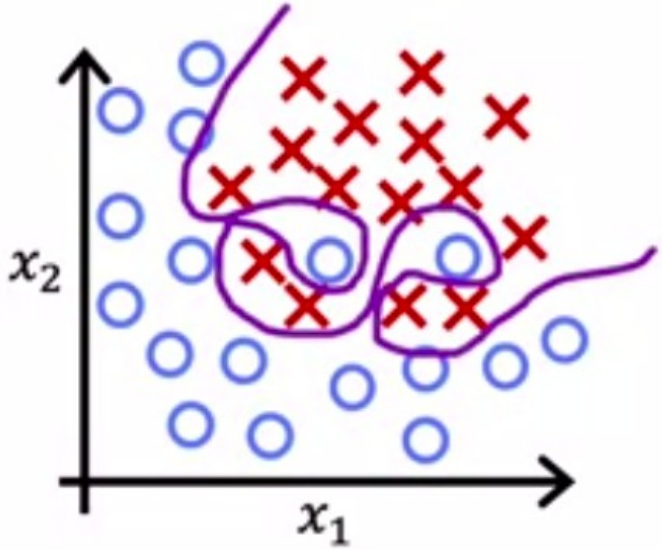
g is the sigmoid function

underfit high bias



$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + b$$

just right

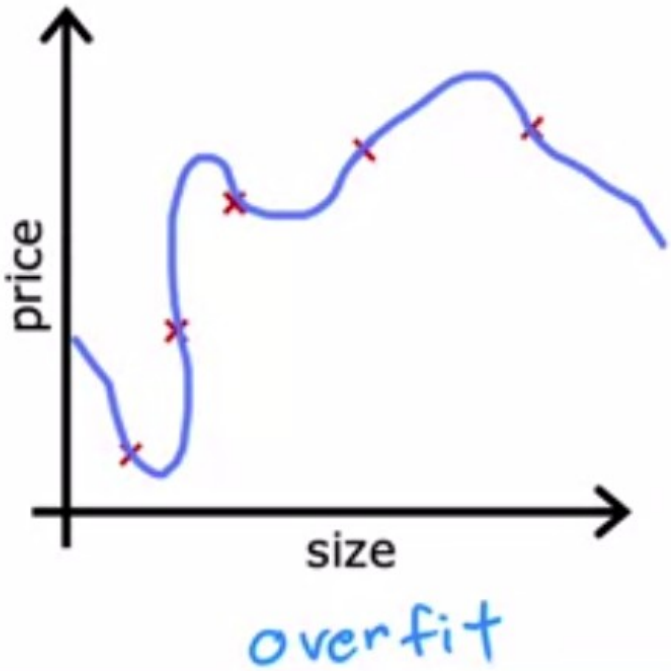


$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + w_6 x_1^3 x_2 + \cdots + b$$

overfit

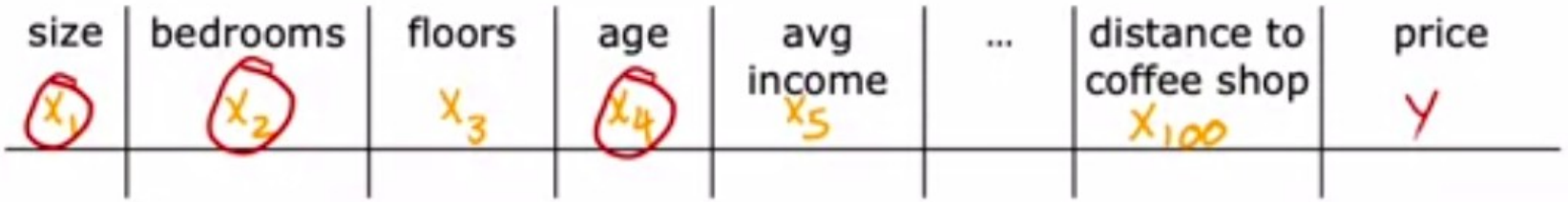
doesn't look like it'll
generalize well to new examples.

Collect more training examples



this can work really well.

Select features to include/exclude



all features

+

insufficient data

↓

overfit

selected features

size
bedrooms
age
just right

feature selection

course 2

disadvantage

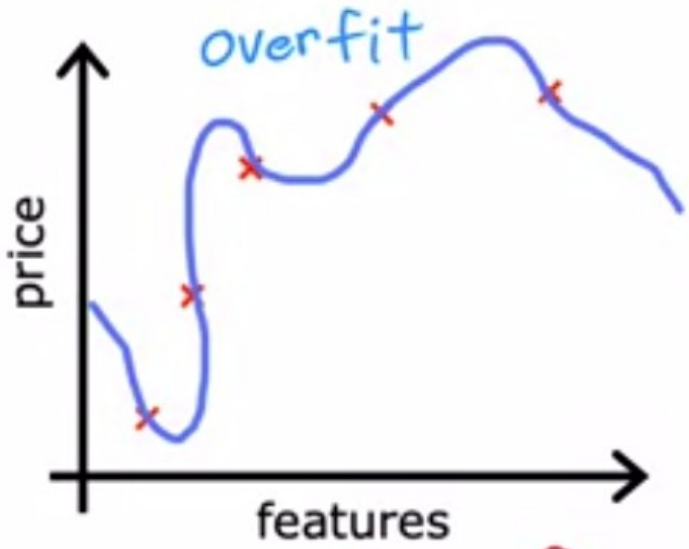
↓

useful features could be lost

features to use for our prediction task.

Regularization

Reduce the size of parameters w_j



$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$

0

eliminate feature

large values for w_j

$f(x) = 13x - 0.23x^2 + 0.000014x^3 - 0.0001x^4 + 10$

small values for w_j

I usually don't and it's just

Addressing overfitting

Options

1. Collect more data

2. Select features
   - Feature selection in course 2

3. Reduce size of parameters
   - "Regularization" next videos!

including neural
networks specifically,

Question

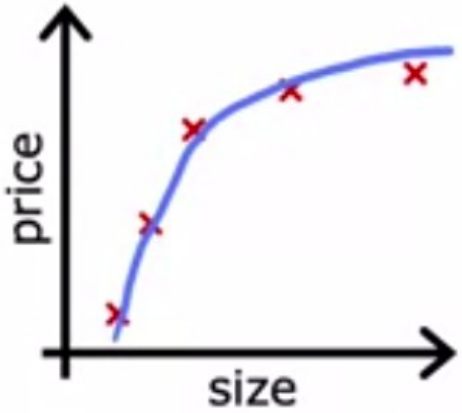
Applying regularization, increasing the number of training examples, or selecting a subset of the most relevant features are methods for…

- (•) Addressing overfitting (high variance)
- ( ) Addressing underfitting (high bias)

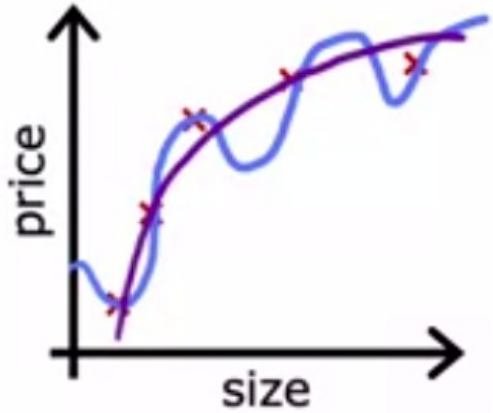
✓ **Correct**
These methods can help the model generalize better to new examples that are not in the training set.

Skip Continue

Intuition



$$w_1 x + w_2 x^2 + b$$



$$w_1 x + w_2 x^2 + \underbrace{w_3 x^3}_{\approx 0} + \underbrace{w_4 x^4}_{\approx 0} + b$$

make w_3, w_4 really small (≈ 0)

$$\min_{\vec{w},b} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + 1000\, \underbrace{w_3^2}_{0.001} + 1000\, \underbrace{w_4^2}_{0.002}$$

the parameters could be large and you end up with this weekly quadratic function

Regularization

small values $w_1, w_2, \cdots, w_n, b$

simpler model

less likely to overfit

$w_3 \approx 0$

$w_4 \approx 0$

size	bedrooms	floors	age	avg income	...	distance to coffee shop	price
x_1	x_2	x_3	x_4	x_5		x_{100}	y

$n = 100$

n features

$w_1, w_1, w_2, \cdots, w_{100}, b$

regularization term

$$J(\vec{w}, b) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2 + \frac{\lambda}{2m} b^2\right]$$

can include or exclude b

"lambda"

regularization parameter

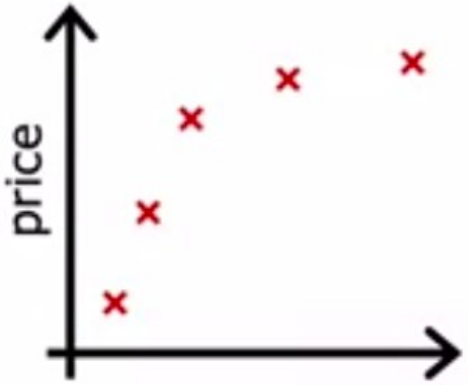
$\lambda > 0$

Regularization

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\overbrace{\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2}^{\text{mean squared error}} + \overbrace{\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2}^{\text{regularization term}} \right]$$

fit data

Keep w_j small



$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

Regularization

mean squared error

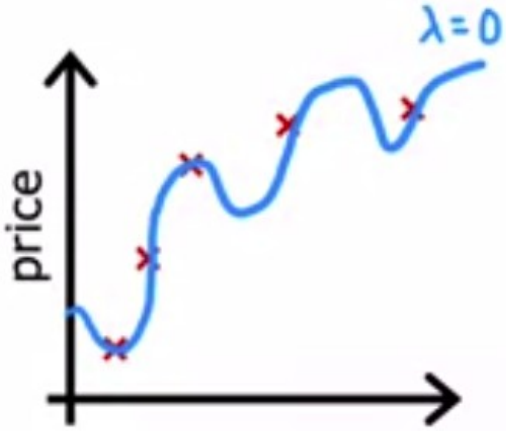
regularization term

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right]$$

fit data

Keep w_j small

λ balances both goals



$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

Regularization

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right]$$

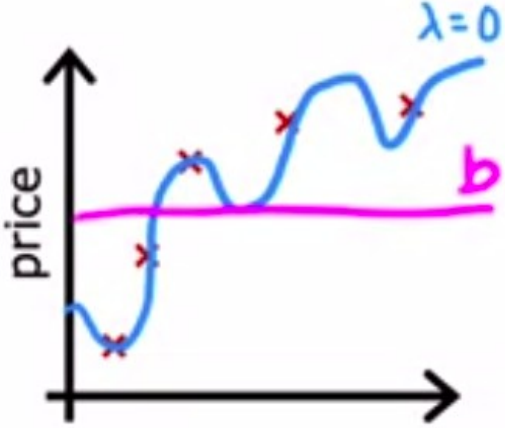
mean squared error

regularization term

fit data

Keep w_j small

λ balances both goals



choose $\lambda = 10^{10}$

$$f_{\vec{w},b}(\vec{x}) = \underset{\approx 0}{w_1 x} + \underset{\approx 0}{w_2 x^2} + \underset{\approx 0}{w_3 x^3} + \underset{\approx 0}{w_4 x^4} + b$$

$$f(x) = b$$

Question

For a model that includes the regularization parameter λ (lambda), increasing λ will tend to…

- Increase the size of parameter b.
- Decrease the size of parameters $w_1, w_2, \ldots w_n$.
- Increases the size of the parameters $w_1, w_2, \ldots w_n$
- Decrease the size of the parameter b.

Correct

Increasing the regularization parameter $lambda$ reduces overfitting by reducing the size of the parameters. For some parameters that are near zero, this reduces the effect of the associated features.

Skip Continue

Regularized linear regression

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right]$$

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w},b)$$

$j = 1, \ldots, n$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right) x_j^{(i)} + \frac{\lambda}{m} w_j$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w},b)$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

don't have to regularize b

} simultaneous update

Implementing gradient descent

repeat {

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left[\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} \right] + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

} simultaneous update

Implementing gradient descent

repeat {

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left[\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} \right] + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

} simultaneous update $j = 1 \ldots n$

$$w_j = \underbrace{1 w_j - \alpha \frac{\lambda}{m} w_j}_{w_j \left(1 - \alpha \frac{\lambda}{m}\right)} - \underbrace{\alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)}}_{\text{usual update}}$$

shrink w_j

$\alpha \frac{\lambda}{m}$

$0.01 \frac{1}{50} = 0.0002$

$w_j (1 - 0.0002)$

0.9998

How we get the derivative term (optional)

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{d}{dw_j}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(\underbrace{f(\vec{x}^{(i)})}_{\vec{w}\cdot\vec{x}^{(i)}+b} - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right]$$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left[\left(\vec{w}\cdot\vec{x}^{(i)} + b - y^{(i)}\right) 2x_j^{(i)}\right] + \frac{\lambda}{2m} 2w_j \quad \text{No } \sum_{j=1}^{n}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[\left(\underbrace{\vec{w}\cdot\vec{x}^{(i)} + b}_{f(\vec{x})} - y^{(i)}\right) x_j^{(i)}\right] + \frac{\lambda}{m} w_j$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}\right] + \frac{\lambda}{m} w_j$$

Question

Recall the gradient descent algorithm utilizes the gradient calculation:

$$\begin{aligned}&\text{repeat until convergence: } \{\\ &\quad w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} w_j \right] \qquad \text{for j} = 1..\text{n}\\ &\quad b = b - \alpha \frac{1}{m} \sum_{i=0}^{m-1} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})\\ &\}\end{aligned}$$

Where each iteration performs simultaneous updates on w_j for all j.

In lecture, this was rearranged to emphasize the impact of regularization:

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} w_j \right] \quad \text{for j} = 1..\text{n}$$

is rearranged to be:

$$w_j = \underbrace{w_j \left(1 - \alpha \frac{\lambda}{m}\right)}_{NewPart} - \underbrace{\alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}}_{OriginalPart} \quad \text{for j} = 1..\text{n}$$

Assuming α, the learning rate, is a small number like 0.001, λ is 1, and $m = 50$, what is the effect of the 'new part' on updating w_j?

Skip Continue

Question

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} w_j \right] \quad \text{for j} = 1..\text{n}$$

is rearranged to be:

$$w_j = w_j \underbrace{\left(1 - \alpha \frac{\lambda}{m}\right)}_{NewPart} - \underbrace{\alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}}_{OriginalPart} \quad \text{for j} = 1..\text{n}$$

Assuming α, the learning rate, is a small number like 0.001, λ is 1, and $m = 50$, what is the effect of the 'new part' on updating w_j?

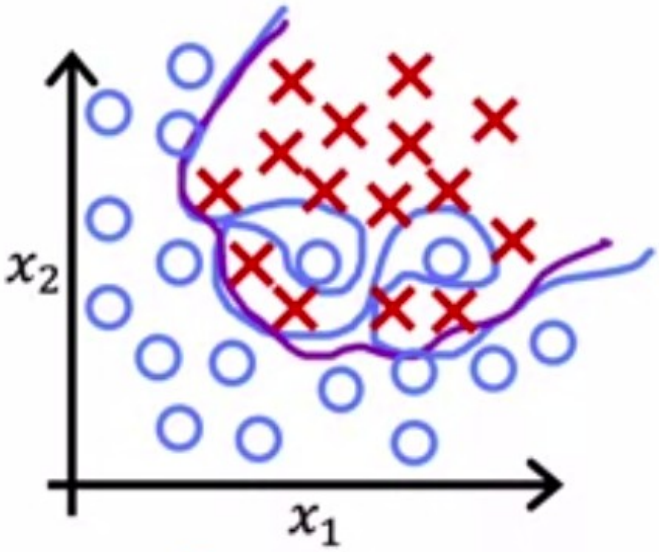
- (●) The new part decreases the value of w_j each iteration by a little bit.
- ( ) The new part increases the value of w_j each iteration by a little bit.
- ( ) The new parts impact varies each iteration.

✓ **Correct**

Correct: the new term decreases w_j each iteration.

Skip Continue

Regularized logistic regression



$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + \cdots + b$$

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-z}}$$

Cost function

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

$\min_{\vec{w},b} J(\vec{w},b) \rightarrow w_j \downarrow$

Regularized logistic regression

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

$\min_{\vec{w},b}$

Gradient descent

repeat {

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\vec{w}, b)$$

$j = 1 \ldots n$

$$b = b - \alpha\frac{\partial}{\partial b}J(\vec{w}, b)$$

}

Looks same as for linear regression!

$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}w_j$$

logistic regression

$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

don't have to regularize

Question

Regularized logistic regression

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

$\min_{\vec{w},b}$

Gradient descent

Looks same as for linear regression!

repeat {

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\vec{w}, b) \quad j = 1 \ldots n$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}w_j$$

logistic regression

$$b = b - \alpha\frac{\partial}{\partial b}J(\vec{w}, b)$$

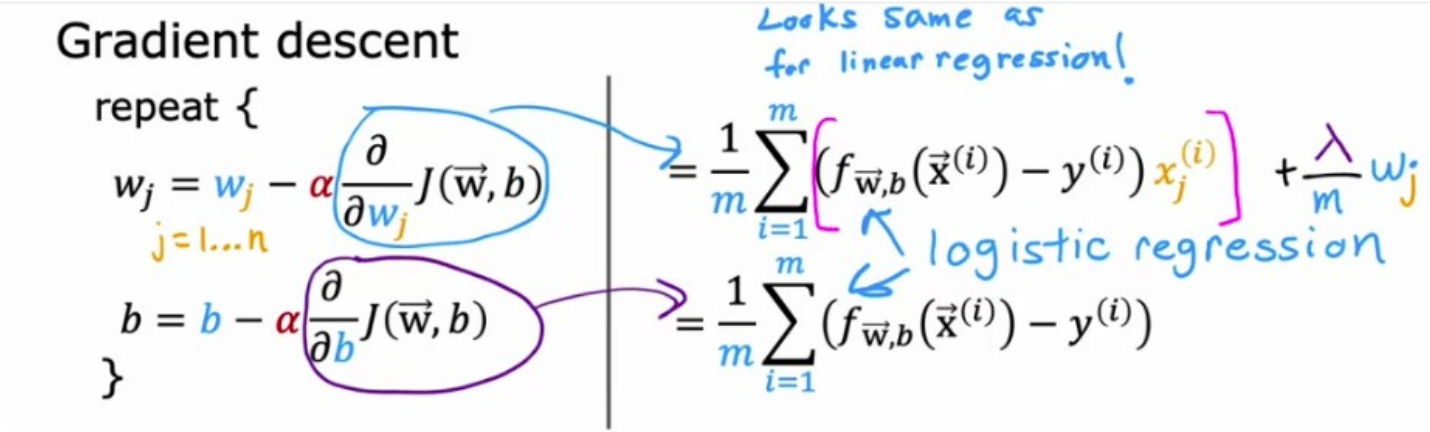
$$= \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

}

For regularized **logistic** regression, how do the gradient descent update steps compare to the steps for linear regression?

Skip Continue

Question



For regularized **logistic** regression, how do the gradient descent update steps compare to the steps for linear regression?

- (•) They look very similar, but the $f(x)$ is not the same.
- ( ) They are identical

✓ **Correct**
For logistic regression, $f(x)$ is the sigmoid (logistic) function, whereas for linear regression, $f(x)$ is a linear function.

Skip Continue

Back

Practice quiz: The problem of overfitting

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

1. Which of the following can address overfitting?

1 / 1 point

- [ ] Remove a random set of training examples
- [x] Select a subset of the more relevant features.

Correct
If the model trains on the more relevant features, and not on the less useful features, it may generalize better to new examples.

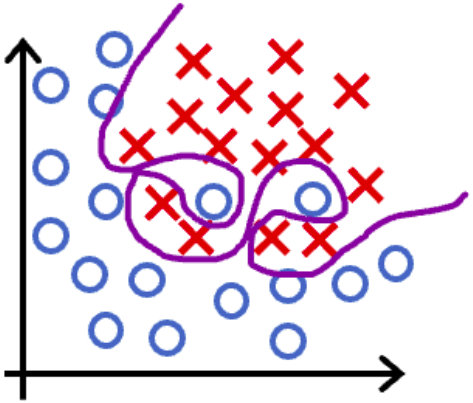
- [x] Apply regularization

Correct
Regularization is used to reduce overfitting.

- [x] Collect more training data

Correct
If the model trains on more data, it may generalize better to new examples.

2. You fit logistic regression with polynomial features to a dataset, and your model looks like this. **1 / 1 point**



What would you conclude? (Pick one)

- (●) The model has high variance (overfit). Thus, adding data is likely to help
- ( ) The model has high bias (underfit). Thus, adding data is, by itself, unlikely to help much.
- ( ) The model has high bias (underfit). Thus, adding data is likely to help
- ( ) The model has high variance (overfit). Thus, adding data is, by itself, unlikely to help much.

Correct
The model has high variance (it overfits the training data). Adding data (more training examples) can help.

Correct
The model has high variance (it overfits the training data). Adding data (more training examples) can help.

3.

Regularization

1 / 1 point

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\overbrace{\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2}^{\text{mean squared error}} + \overbrace{\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2}^{\text{regularization term}} \right]$$

Suppose you have a regularized linear regression model. If you increase the regularization parameter λ, what do you expect to happen to the parameters $w_1, w_2, \ldots, w_n$?

- ○ This will increase the size of the parameters $w_1, w_2, \ldots, w_n$
- ◉ This will reduce the size of the parameters $w_1, w_2, \ldots, w_n$

Correct
Regularization reduces overfitting by reducing the size of the parameters $w_1, w_2, \ldots w_n$.