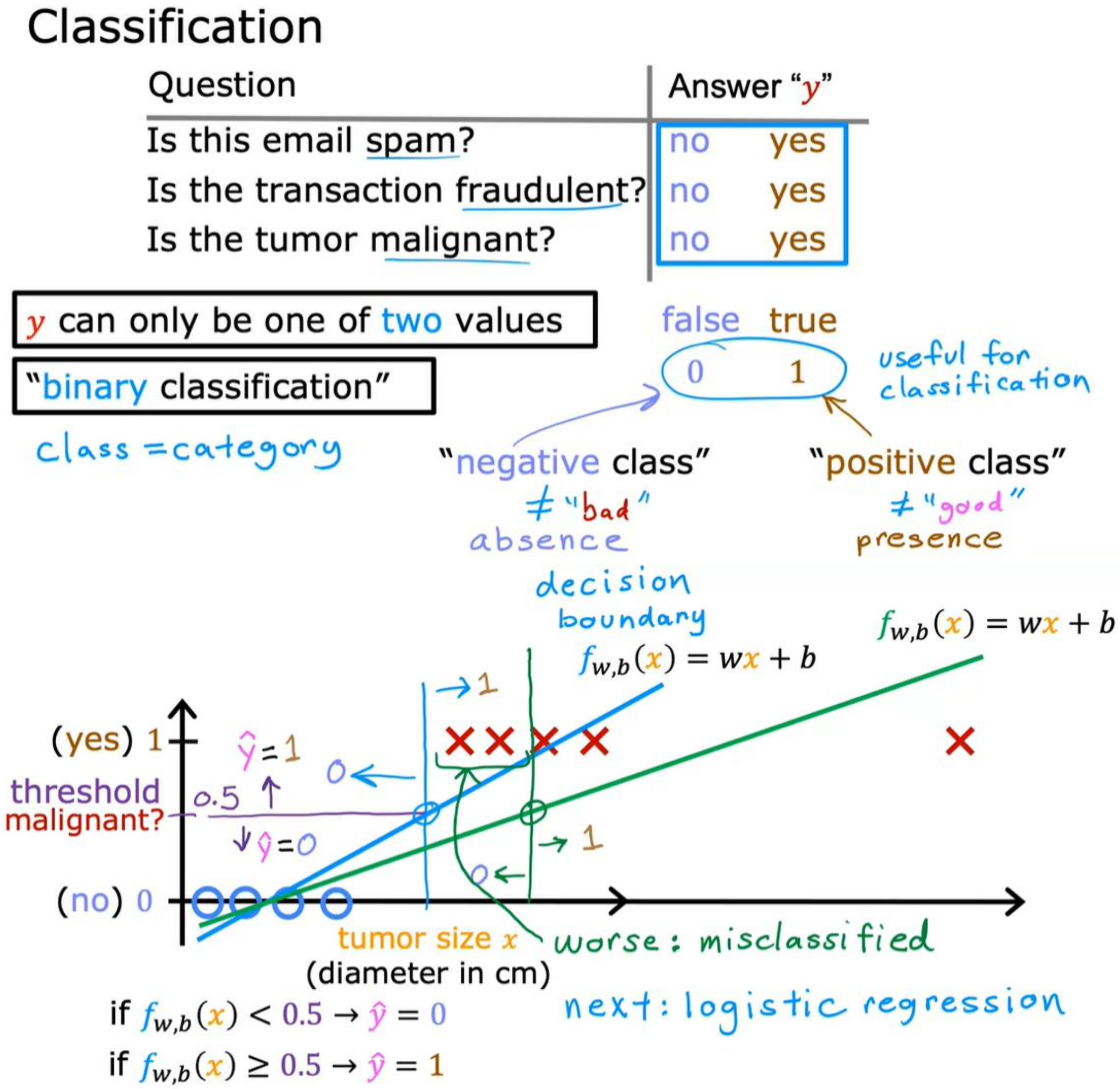
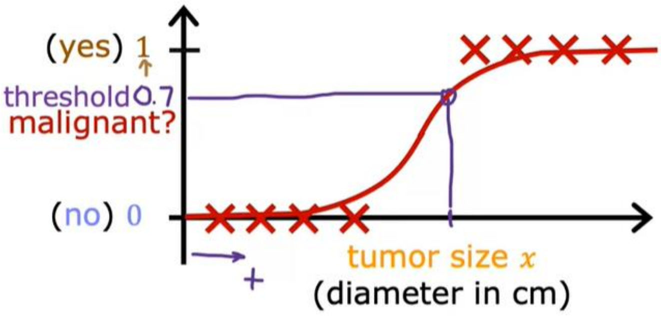
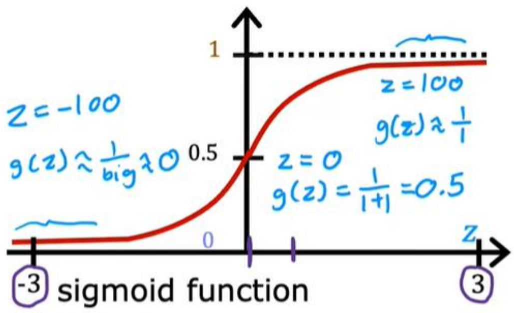
1. Classification motivation



2. Logistic regression



Want outputs between 0 and 1



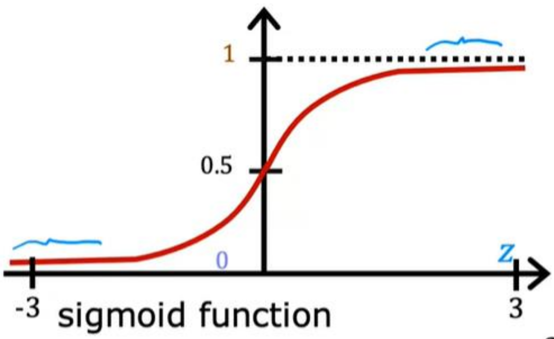
sigmoid function

logistic function

outputs between 0 and 1

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

Want outputs between 0 and 1



sigmoid function

logistic function

outputs between 0 and 1

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

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

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

$\downarrow$ z $\downarrow$

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

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

"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

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

$f_{\vec{w},b}(\vec{x}) = 0.7$
70% chance that y is 1

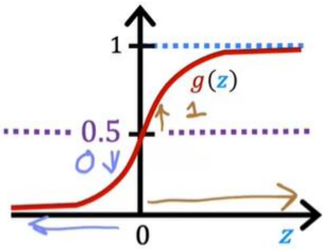
P(1)

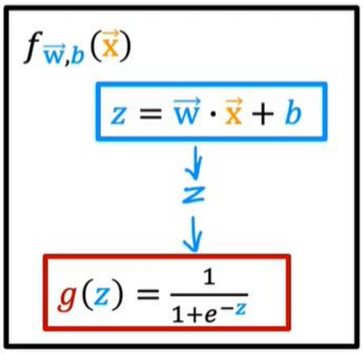
$$f_{\vec{w},b}(\vec{x}) = P(y = 1 | \vec{x}; \vec{w}, b)$$

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

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

3. Decision boundary





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

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

0 or 1? threshold

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

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

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

$g(z) \geq 0.5$

$z \geq 0$

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

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

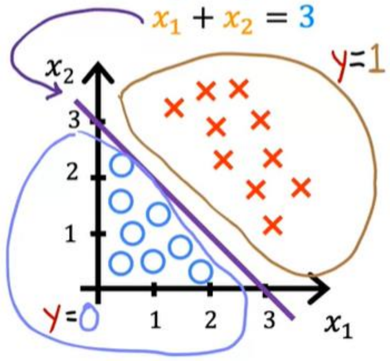
Decision boundary

$$f_{\vec{w},b}(\vec{x}) = g(z) = g(\underset{1}{w_1}x_1 + \underset{1}{w_2}x_2 + \underset{-3}{b})$$

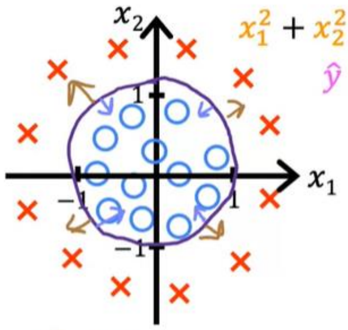
Decision boundary $z = \vec{w} \cdot \vec{x} + b = 0$

$z = x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$



Non-linear decision boundaries



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

$\hat{y} = 1$

$$f_{\vec{w},b}(\vec{x}) = g(z) = g(\overbrace{\underset{1}{w_1}x_1^2 + \underset{1}{w_2}x_2^2 + \underset{-1}{b}}^{z})$$

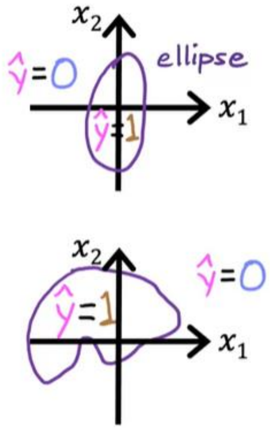
decision boundary $z = x_1^2 + x_2^2 - 1 = 0$

$x_1^2 + x_2^2 = 1$

$x_1^2 + x_2^2 < 1$

$\hat{y} = 0$

Non-linear decision boundaries



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

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.

4. Practice quiz 1

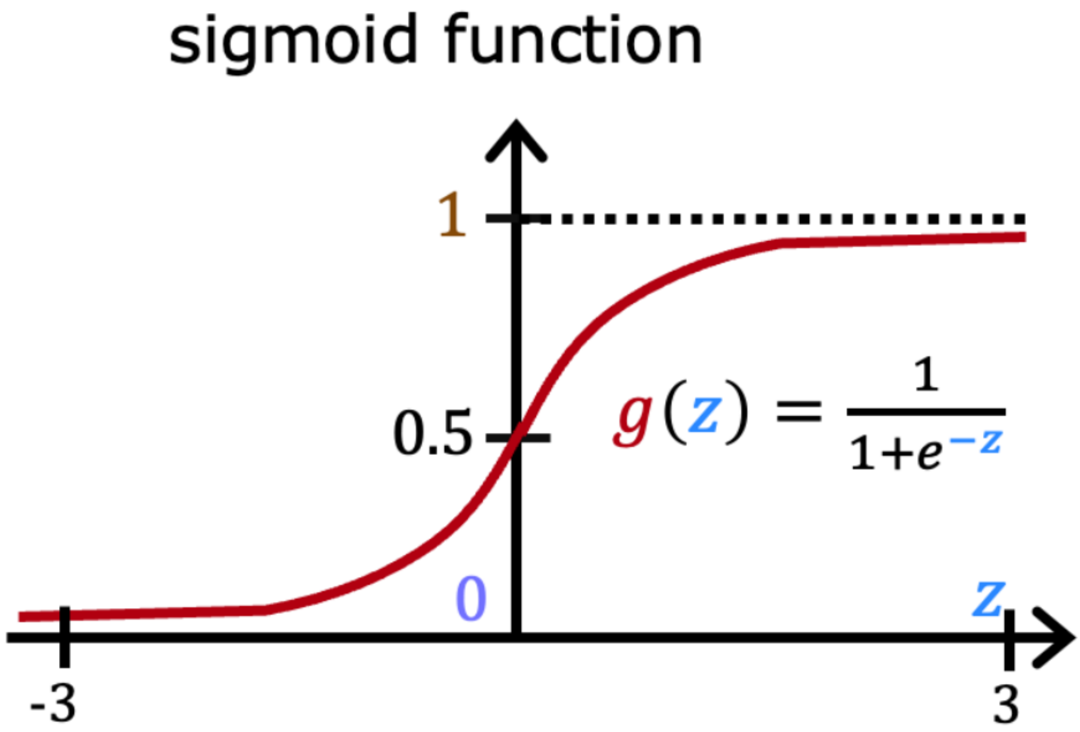
1. Which is an example of a classification task?

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

Correct

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

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



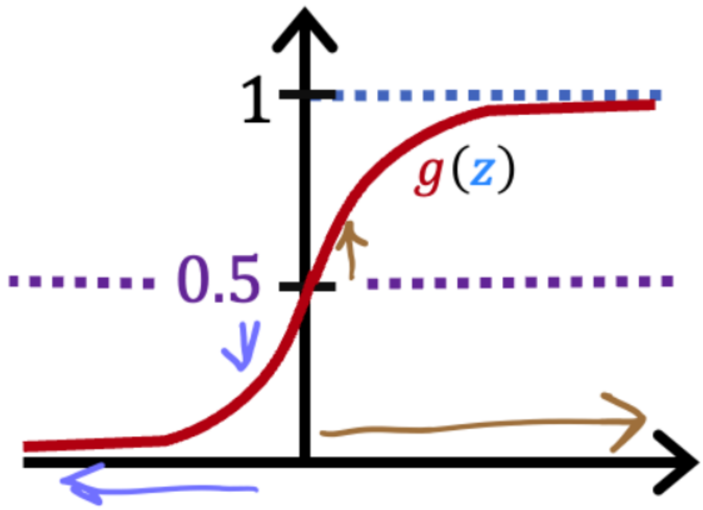
If z is a large positive number, then:

- [ ] $g(z)$ will be near 0.5
- [ ] $g(z)$ is near negative one (-1)
- [ ] $g(z)$ will be near zero (0)
- [x] $g(z)$ 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.



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

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.

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.

5. Cost function for logistic regression

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

$\boxed{\text{target } y \text{ is } 0 \text{ or } 1}$

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

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

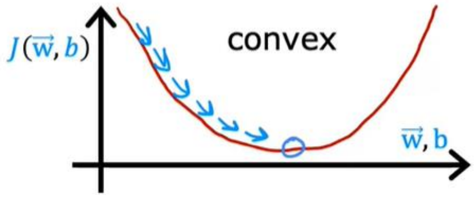
Squared error cost

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

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

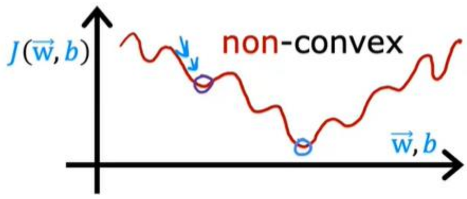
linear regression

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



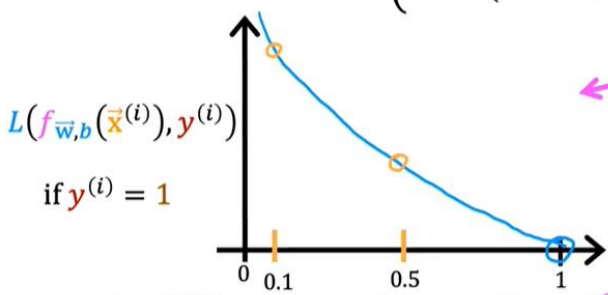
logistic regression

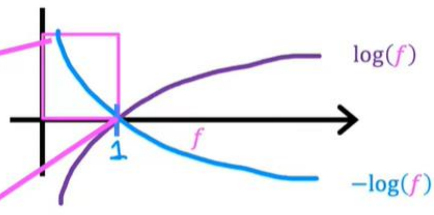
$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$



Logistic loss function

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





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

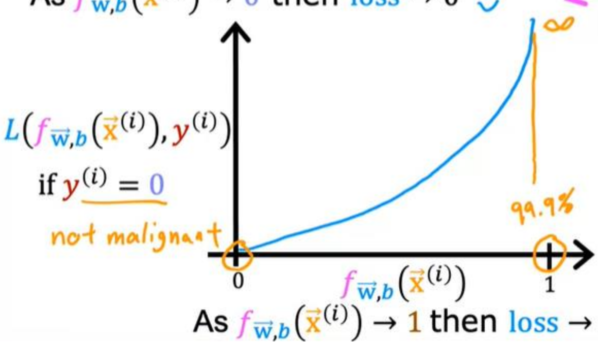
As $f_{\vec{w},b}(\vec{x}^{(i)}) \to 0$ then loss $\to \infty$

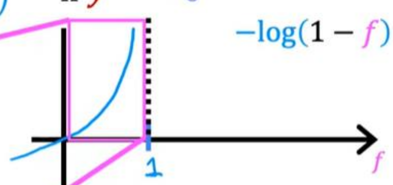
Loss is lowest when $f_{\vec{w},b}(\vec{x}^{(i)})$ predicts close to true label $y^{(i)}$.

Logistic loss function

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

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





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

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

Cost

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

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

convex
↳ can reach a global minimum

find w, b that minimize cost J

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

- [x] 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.

6. Simplified cost function for logistic regression

Simplified loss function

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

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

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

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

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

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

Simplified cost function

loss

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

cost

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

convex
(single global minimum)

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

maximum likelihood
(don't worry about it!)

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?

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

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

7. Practice quiz 2

1.

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

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

2.

Simplified loss function

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

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

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

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

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

8. Gradient descent implementation

Training logistic regression

Find $\vec{w}, b$

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

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

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 {

$j=1\ldots n$

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

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

} simultaneous updates

Gradient descent for logistic regression

repeat { looks like linear regression!

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

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

} simultaneous updates

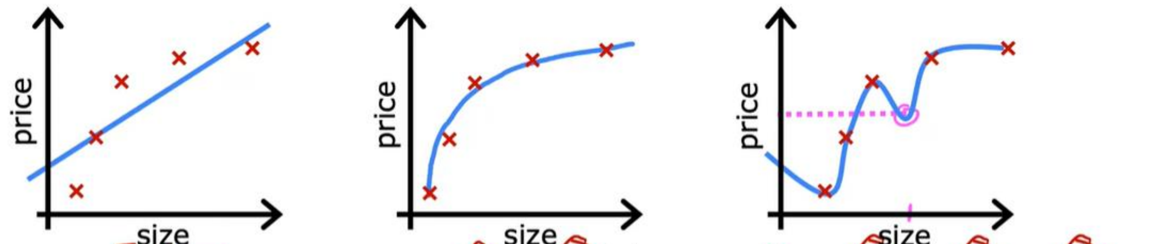
Same concepts:
- Monitor gradient descent (learning curve)
- Vectorized implementation
- Feature scaling

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

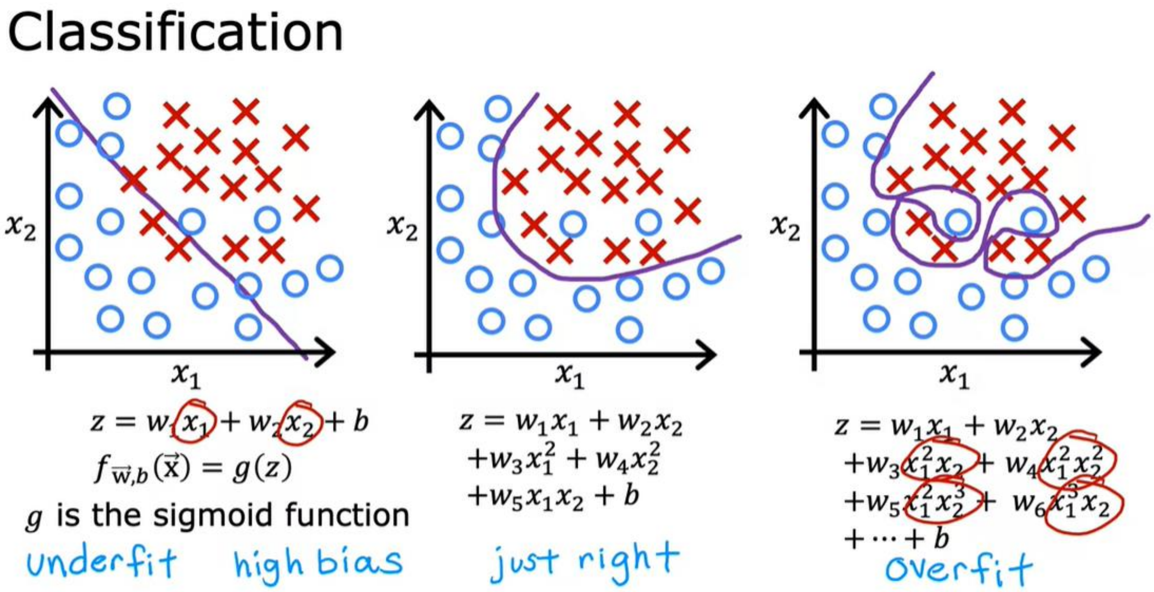
Logistic regression $\quad f_{\vec{w},b}(\vec{x}) = \dfrac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

9. The problem of overfitting

Regression example



$w_1x + b$	$w_1x + w_2x^2 + b$	$w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$
underfit	just right	overfit
• Does not fit the training set well	• Fits training set pretty well	• Fits the training set extremely well
high bias	generalization	high variance



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:

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

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

10. Addressing overfitting

Select features to include/exclude

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

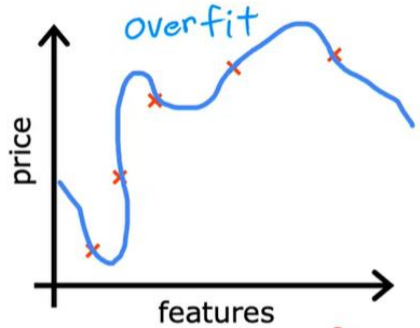
all features
+
insufficient data
↓
overfit

selected features
size
bedrooms
age
just right
feature selection
course 2

disadvantage
↓
useful features could be lost

Regularization

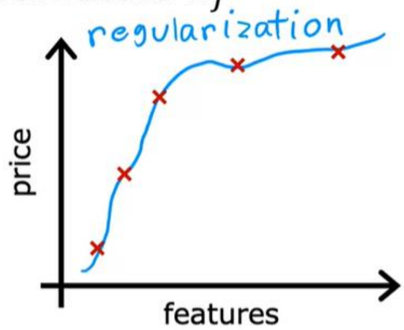
Reduce the size of parameters w_j



$$f(x) = 28x - 385x^2 + 39x^3 - \underbrace{\cancel{174}x^4}_{0} + 100$$

eliminate feature

large values for w_j



$$f(x) = 13x - 0.23x^2 + \underline{0.000014}x^3 - \underline{0.0001}\,x^4 + 10$$

small values for w_j

Addressing overfitting

Options

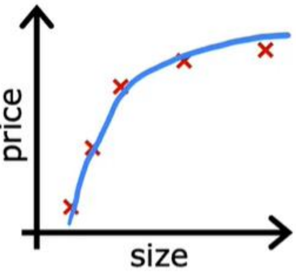
1. Collect more data

2. Select features
   - Feature selection in course 2

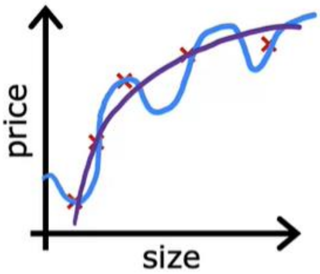
3. Reduce size of parameters
   - "Regularization" next videos!

11. Cost function with regularization

Intuition



$w_1 x + w_2 x^2 + b$



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

make w_3, w_4 really small (≈ 0)

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

Regularization

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

simpler model $w_3 \approx 0$

less likely to overfit $w_4 \approx 0$

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

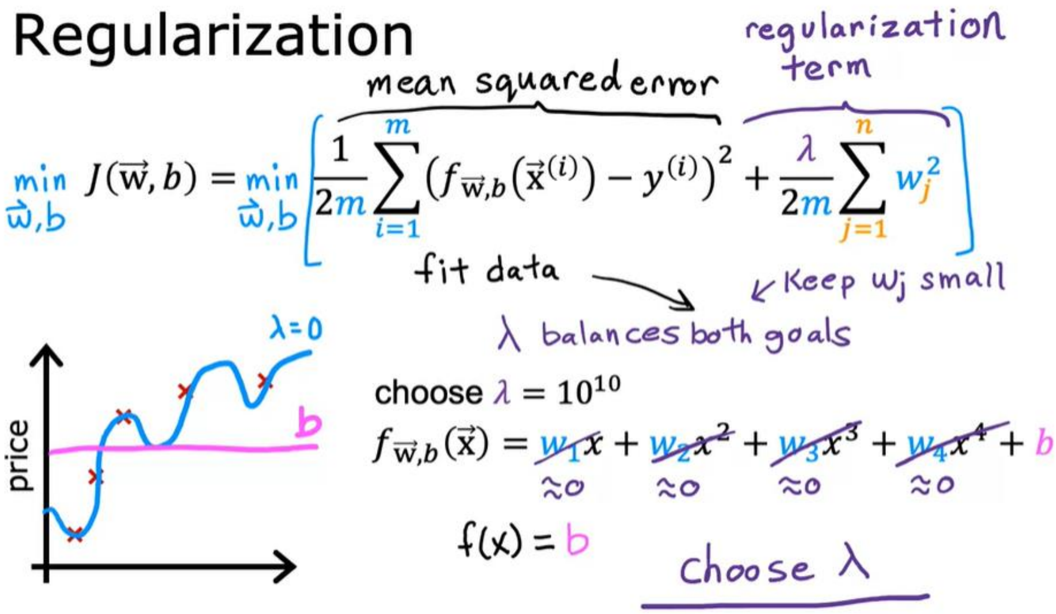
$w_1, w_1, w_2, \cdots, w_{100}, b$ n features $n = 100$

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

can include or exclude b

"lambda" regularization parameter

$\lambda > 0$

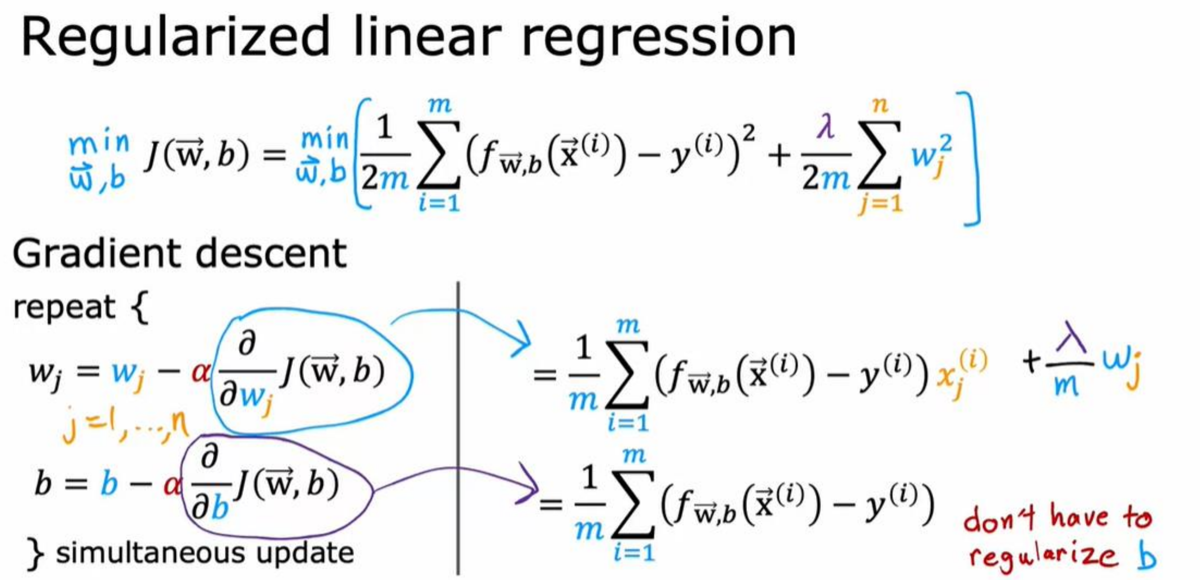


For a model that includes the regularization parameter λ (lambda), increasing λ will tend to...

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

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.

12. Regularized linear regression



How we get the derivative term (optional)

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

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

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

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

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 $j = 1 \ldots n$

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

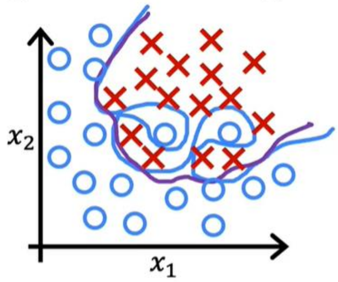
$$\alpha\frac{\lambda}{m}$$

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

$$w_j\,(\underbrace{1-0.0002}_{0.9998})$$

13. Regularized logistic regression

Regularized logistic regression



$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + \cdots + b$$

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

Cost function

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

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

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

Looks same as for linear regression!

repeat {

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

$j = 1 \ldots n$

logistic regression

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

}

don't have to regularize

14. Practice quiz

1. Which of the following can address overfitting?

- [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] Collect more training data

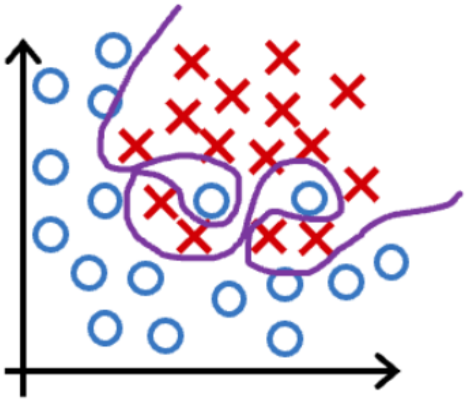
Correct
If the model trains on more data, it may generalize better to new examples.

- [x] Apply regularization

Correct
Regularization is used to reduce overfitting.

- [ ] Remove a random set of training examples

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



What would you conclude? (Pick one)

- 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.
- 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.

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

3.

Regularization

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

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

- This will increase the size of the parameters $w_1, w_2, ..., w_n$
- This will reduce the size of the parameters $w_1, w_2, ..., w_n$

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