

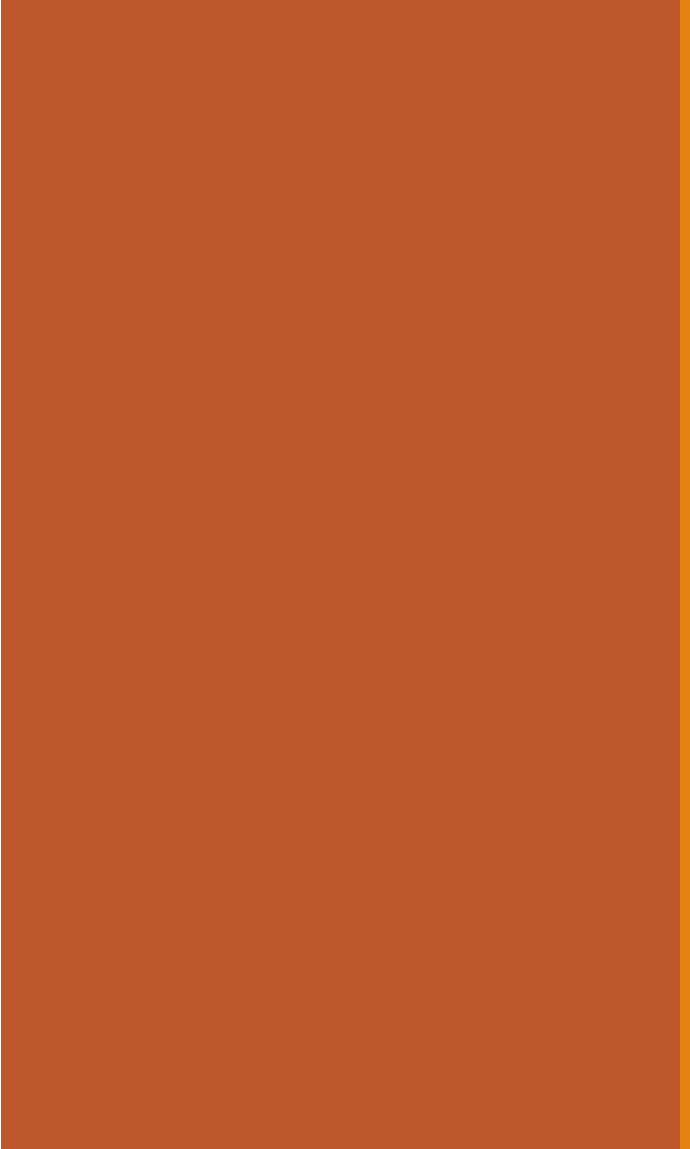
Table of Contents

2 Linear Regression
6 Linear Regression: MSE
11 Linear Regression: MLE
18 Gradient Ascent
44 Extra: Derivations

24: Linear Regression and Gradient Ascent

Jerry Cain
March 6, 2024

[Lecture Discussion on Ed](#)

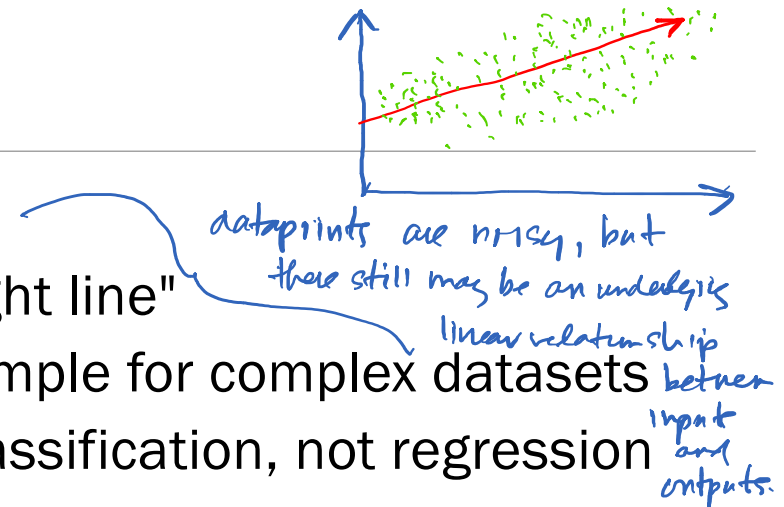


Linear Regression

Today's goals

We are going to learn linear regression.

- Informally known as "fitting data to a straight line"
- Linear models, admittedly, are often too simple for complex datasets
- Many tasks in computer science call for classification, not regression



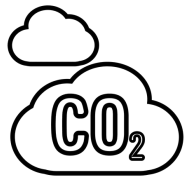
We still cover this topic so we can learn **compelling techniques** that will help us design and understand more complicated learning algorithms:

1. How to model likelihood of training data $(\mathbf{x}^{(i)}, y^{(i)})$
2. What rules of calculus and **argmax** are important to remember
3. What **gradient ascent** is and why it is useful

Regression: Predicting real numbers

Review

Training data: $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$



CO2 levels

Year 1	338.8
Year 2	340.0
...	
Year n	340.76

$\mathbf{X} = (X_1)$

(assume one feature)



Global Land-Ocean temperature

Output

0.26
0.32
$\vdots$
0.14

$Y \in \mathbb{R}$

Model:

*this is our predictor
 $\hat{Y}$ for unknown data*

$\hat{Y} = g(\mathbf{X}),$
for some parametric
function g

Linear Regression

Assume linear model
(and $\mathbf{X}$ is 1-D):

$$\hat{Y} = g(\mathbf{X}) = aX + b$$

these are unknowns, need to infer the best choices for them.

$= ax_i + b$ } only one x input for this example

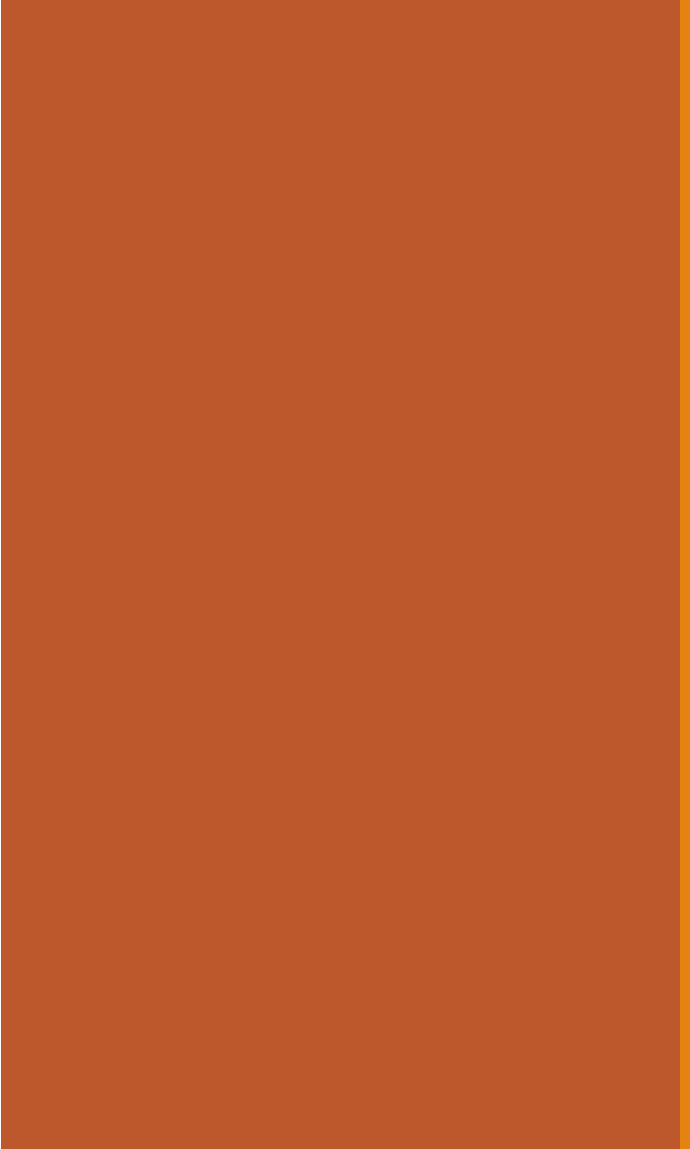
Training

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$
Learn parameters $\theta = (a, b)$

Two approaches:

- Analytical solution via mean squared error
- Iterative solution via MLE and gradient ascent

divable because linear and one dimensional



Linear Regression: MSE

Mean Squared Error (MSE)

} also called Least Squares Error as well.

For most regressions, we generally choose a $g(X)$ that minimizes MSE: Y is the true value we observe, $\hat{Y}$ is a prediction of Y based on x , given by $g(x)$

$$\theta_{MSE} = \arg \min_{\theta} E[(Y - \hat{Y})^2] = \arg \min_{\theta} E[(Y - g(X))^2]$$

could have been $E[|\hat{Y} - Y|]$, but absolute value function is more difficult to manipulate because of behavior at 0.

- Y and $\hat{Y} = g(X)$ are both random variables
- Intuitively: Choose parameter θ that minimizes the expected squared deviation—what can be called an error—of your $\hat{Y}$ predictor from true Y

For **linear** regression, where $\hat{Y} = aX + b$ (so that $\theta = (a, b)$):

$$E[(Y - aX - b)^2]$$

} we choose a and b so that the expected value of this quantity is as low as possible.

a_{MSE} and b_{MSE} are the best choices.

Don't make me go nonlinear!

$$\theta_{MSE} = \arg \min_{\theta=(a,b)} E[(Y - aX - b)^2]$$

$$a_{MSE} = \rho(X, Y) \frac{\sigma_Y}{\sigma_X}, \quad b_{MSE} = \mu_Y - a_{MSE} \mu_X$$

(derivation included at the end of slides)

note that b_{MSE} is defined in terms of a_{MSE} , and we choose to not expand it, just to save space.

Can we compute these statistics for X and Y from our training data?

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$

Technically no, but **we can estimate them!**



Don't make me go nonlinear!

$$\theta_{MSE} = \arg \min_{\theta=(a,b)} E[(Y - aX - b)^2]$$

$$a_{MSE} = \rho(X, Y) \frac{\sigma_Y}{\sigma_X}, \quad b_{MSE} = \mu_Y - a_{MSE} \mu_X$$

(derivation included at the end of slides)

Can we compute these statistics for X and Y from our training data?

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$

Estimate parameters based on observed training data:

$$\hat{a}_{MSE} = \hat{\rho}(X, Y) \frac{S_Y}{S_X}, \quad \hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE} \bar{X}$$

sample correlation (pointing to $\hat{\rho}(X, Y)$)
sample stdev (pointing to S_Y)
mean of all Y values in the sample (pointing to $\bar{Y}$)
mean of all input values (pointing to $\bar{X}$)

$\hat{\rho}(X, Y)$: Sample correlation ([Wikipedia](#))

Summary: Linear Regression

Assume linear model (and X is 1-D):

$$\hat{Y} = g(X) = aX + b$$

predictor

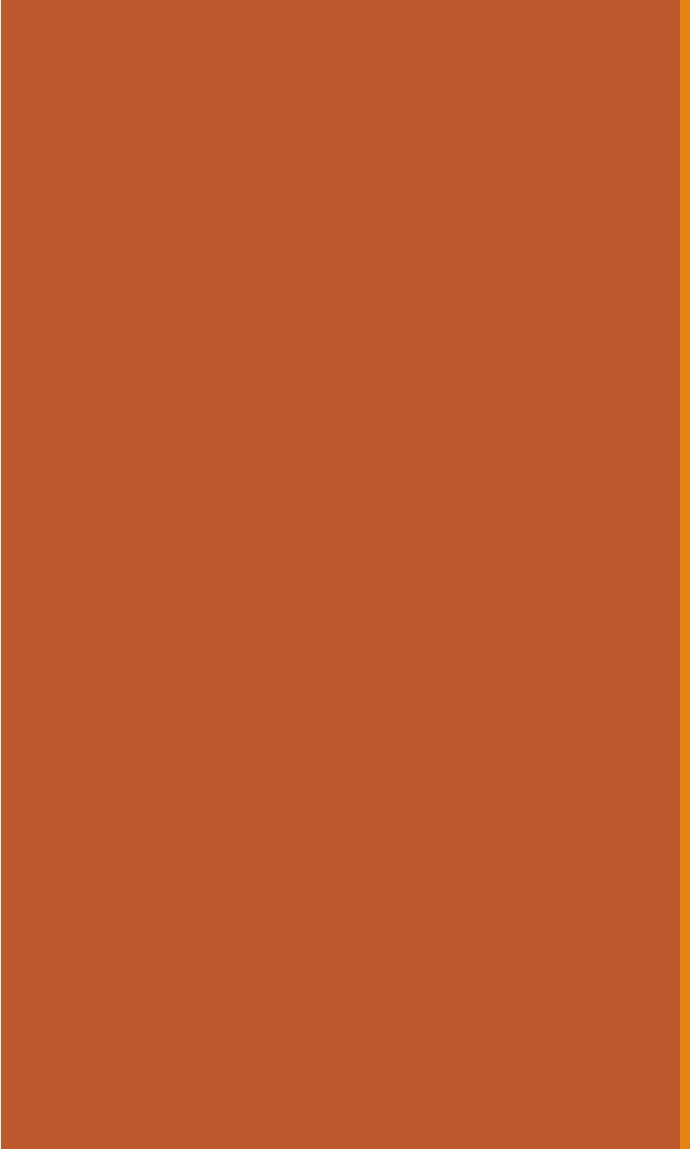
Training

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$
Learn parameters $\theta = (a, b)$

If we want to minimize the mean squared error of our prediction,

*general MSE theory
assumes we'll
compute these sample
statistics*

$$\hat{a}_{MSE} = \hat{\rho}(X, Y) \frac{S_Y}{S_X}, \quad \hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE} \bar{X}$$



Linear Regression: MLE

Linear Regression

Review

Assume linear model
(and X is 1-D, i.e., $X = X$):

$$\hat{Y} = g(X) = aX + b$$

Training

Learn parameters $\theta = (a, b)$

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})$

We've seen which parameters—that is, what choices of a and b —minimize mean squared error: a_{MSE} and b_{MSE} , estimated by $\hat{a}_{MSE}$ and $\hat{b}_{MSE}$.

What if we want parameters that maximize the **likelihood of the training data**?

Note: Maximizing likelihood is typically an objective for classification models.

Likelihood, it's been a minute

Review

Consider a sample of n iid random variables $X_1, X_2, \dots, X_n$. *for example, $X_i \sim \text{Poi}(\theta)$*

- X_i was drawn from some distribution with density function $f(X_i|\theta)$.
- Observed sample: $(X_1, X_2, \dots, X_n)$

Likelihood question:

How likely is the observed sample $(X_1, X_2, \dots, X_n)$ given parameter θ ?

Likelihood function, $L(\theta)$:

$$L(\theta) = f(X_1, X_2, \dots, X_n|\theta) = \prod_{i=1}^n f(X_i|\theta)$$

*you've seen this
before for MLE
and MAP*

This is just a product, since X_i are iid

Likelihood of the training data

Training data (n datapoints):

- $(x^{(i)}, y^{(i)})$ drawn iid from a distribution $f(X = x^{(i)}, Y = y^{(i)} | \theta) = f(x^{(i)}, y^{(i)} | \theta)$ (shorthand)
- $\hat{Y} = g(X)$, where g is a function on (X) and parameter θ

We can show that θ_{MLE} maximizes the **log conditional likelihood** function:

$$\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

$$\theta_{MLE} = \arg \max_{\theta} \prod_{i=1}^n f(x^{(i)}, y^{(i)} | \theta)$$

this is more difficult to manipulate, so we work with this instead



Linear Regression, MLE

1. Assume linear model for $\hat{Y}$
(and $\mathbf{X}$ is 1-D):

$$\hat{Y} = g(\mathbf{X}) = aX + b$$

2. Define maximum likelihood estimator:

$$\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

! Drama: We have a model for $\hat{Y}$, not Y

- Remember the MSE approach, where we minimize the squared **error** between $\hat{Y}$ and Y ?
- Here we **model this error** directly!

$$\begin{aligned} Y &= \hat{Y} + \mathbf{Z} && \text{error/noise} \\ &= aX + b + \mathbf{Z} && \text{(also random)} \end{aligned}$$

Comparison: MSE vs MLE

$$\hat{Y} = g(X) = aX + b$$

Minimum Mean Squared Error

$$\theta_{MSE} = \arg \min_{\theta} E \left[(Y - g(X))^2 \right]$$

- Don't directly model Y (or any errors)
- Parameters are estimates of statistics from training data:

$$\hat{a}_{MSE} = \hat{\rho}(X, Y) \frac{S_Y}{S_X}$$
$$\hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE} \bar{X}$$

Maximum Likelihood Estimation

$$\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

- Directly model error between predicted $\hat{Y}$ and Y as an RV Z

$$Y = \hat{Y} + Z = aX + b + Z$$

If we assume error $Z \sim \mathcal{N}(0, \sigma^2)$, then these two estimators are **equivalent**.

$$\theta_{MSE} = \theta_{MLE}!$$

Linear Regression, MLE (next steps)

1. Assume linear model
(and $\mathbf{X}$ is 1-D):

$$\hat{Y} = g(\mathbf{X}) = aX + b$$

2. Define maximum likelihood estimator:

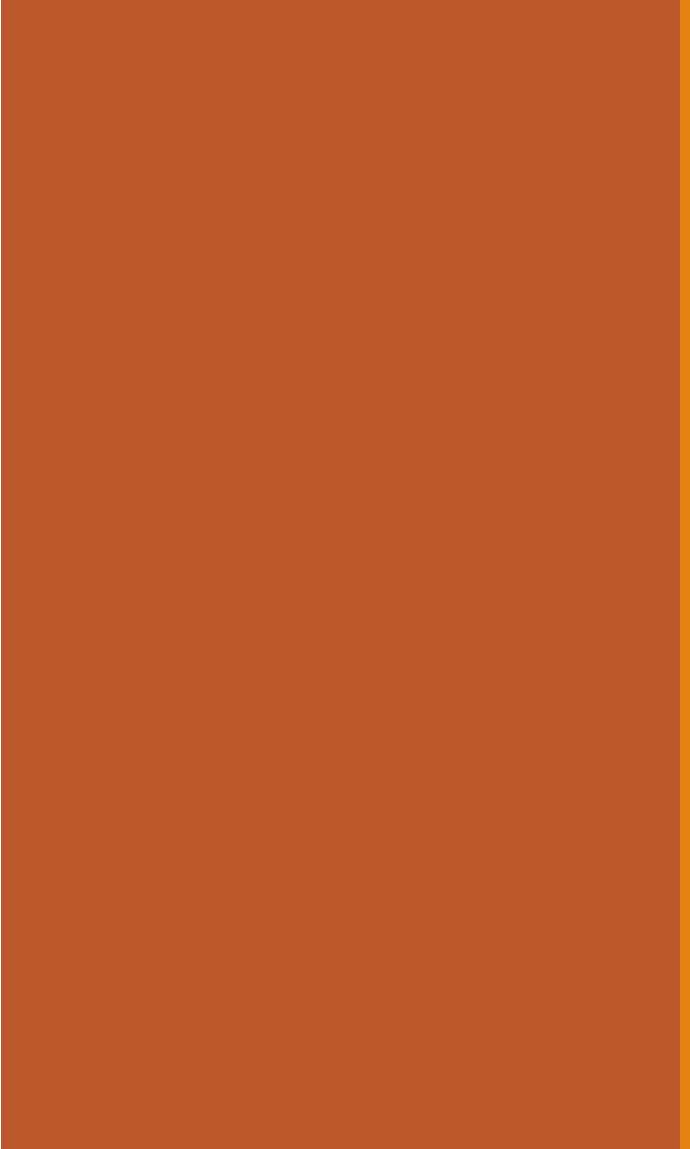
$$\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

3. Model error, Z :
and also model Y in terms of $\hat{Y}$ and Z .

$$Y = aX + b + Z, \text{ where } Z \sim \mathcal{N}(0, \sigma^2)$$

4. Pick $\theta = (a, b)$ that maximizes likelihood of training data

We won't find a solution analytically. Instead, we'll leverage **gradient ascent**, an **iterative optimization algorithm**.



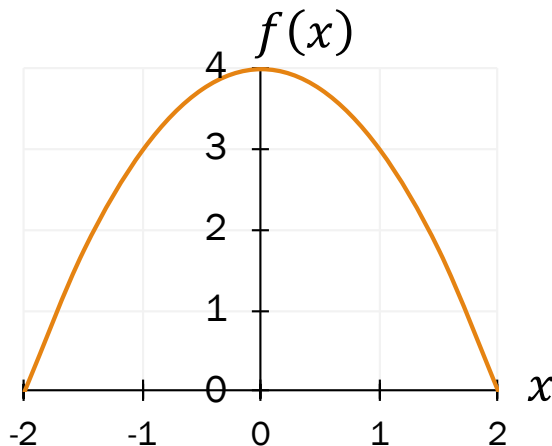
Gradient Ascent

Multiple ways to calculate argmax

Let $f(x) = -x^2 + 4$,
where $-2 < x < 2$.

What is $\arg \max_x \underbrace{f(x)}_{\text{objective function}}?$

A. Graph and guess

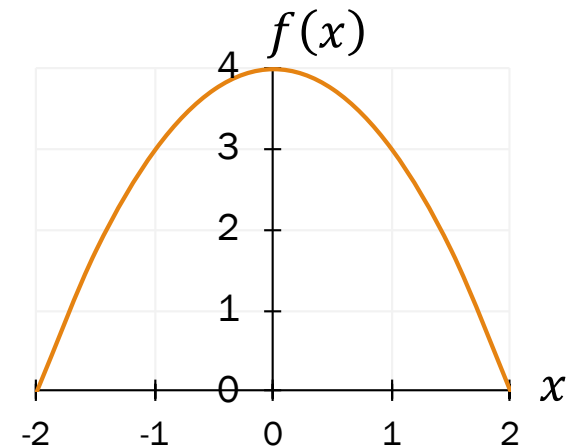


B. Differentiate,
set derivative to
0, and solve

$$\frac{df}{dx} = -2x = 0$$

$$x = 0$$

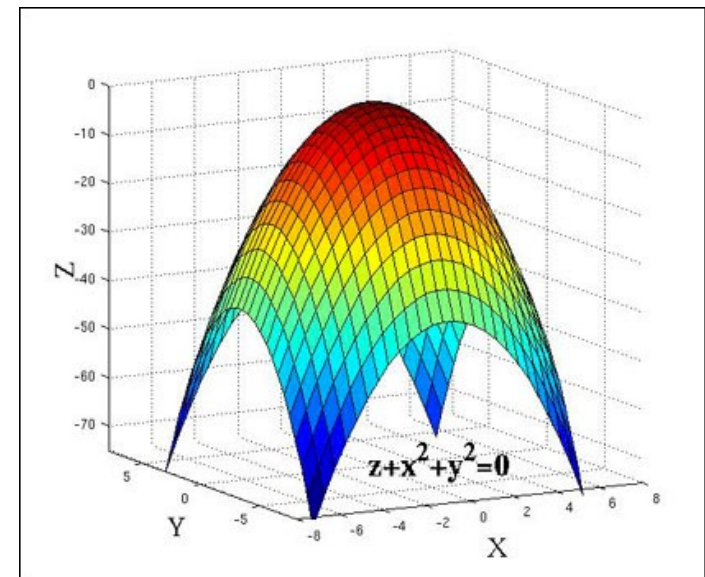
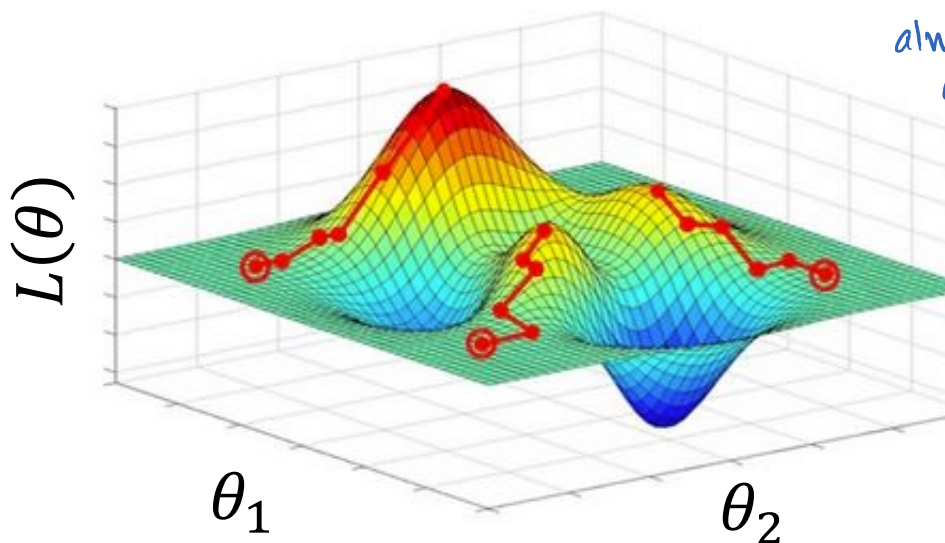
C. **Gradient ascent**: educated
guess & iteratively update



Gradient ascent

Walk uphill and you'll find a local maxima
(if your step is small enough).

*In CS109, $L(\theta)$
and $U(\theta)$ are
always
concave*

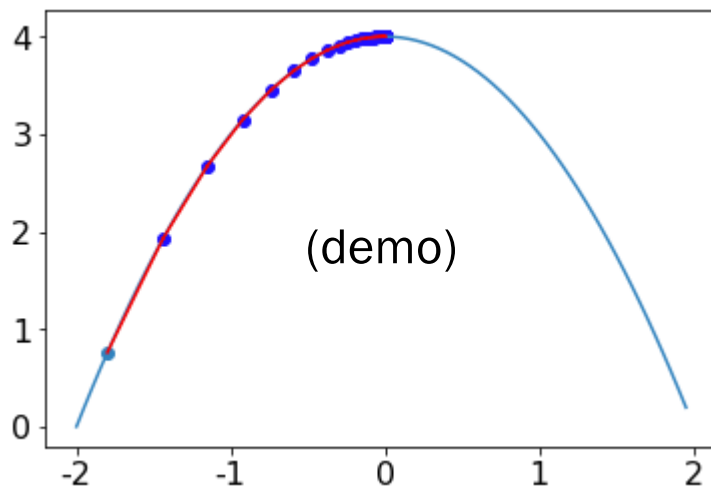


If your function is concave,
Local maxima = global maxima

Gradient ascent algorithm

Walk uphill and you'll find a local maxima
(if your step is small enough).

Let $f(x) = -x^2 + 4$,
where $-2 < x < 2$.



Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2024

1. $\frac{df}{dx} = -2x$

2. Gradient ascent algorithm:

initialize x

repeat many times:

compute gradient

$x \leftarrow x + \eta * \text{gradient}$

η we call this a learning rate

pretend that gradient is so complicated that we can't just set equal to 0 and solve. Instead we discover solution numerically using

Gradient at x

Linear Regression, MLE (so far)

Assume linear model
(and $\mathbf{X}$ is 1-D):

$$\hat{Y} = g(\mathbf{X}) = aX + b$$

Model Y as $\hat{Y} + Z$:

$$Y = aX + b + Z, \text{ where } Z \sim \mathcal{N}(0, \sigma^2)$$

Pick $\theta = (a, b)$ that maximizes
likelihood of training data

$$\begin{aligned}\theta_{MLE} &= \arg \max_{\theta} LL(\theta) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log f(x^{(i)}, y^{(i)}, |\theta) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)\end{aligned}$$

(θ_{MLE} also maximizes
log conditional likelihood)

Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ :

1. Determine formula for $LL(\theta)$

log conditional likelihood

$$\sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$



2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

$\theta_1 \quad \theta_2$
 $\downarrow \quad \downarrow$
 $\theta = (a, b)$



3. Solve resulting equations

(computer)
Gradient Ascent

1. Determine formula for log conditional likelihood

Model:	$\theta = (a, b)$	Optimization	$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} x^{(i)}, \theta)$
	$Y = aX + b + Z$	problem:	
	$Z \sim \mathcal{N}(0, \sigma^2)$		

Over the next few slides, we will show that our MLE linear regression θ_{MLE} reduces to

$$\arg \max_{\theta} \left[- \underbrace{\sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2}_{\text{objective function}} \right]$$

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$

$$Y = aX + b + Z$$

$$Z \sim \mathcal{N}(0, \sigma^2)$$

Optimization
problem:

$$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

goal $\rightarrow$ $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

1. What is the conditional distribution, $Y|X, \theta$?
2. Substitute 1. into objective fn.
3. Use argmax properties to simplify objective fn.



1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$

$$Y = aX + b + Z$$

$$Z \sim \mathcal{N}(0, \sigma^2)$$

Optimization
problem:

$$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

$$Y | X=x, \theta=(a,b) \\ Y = aX + b + Z$$

1. What is the conditional distribution, $Y | X, \theta$?

$$Y | X, \theta \sim \mathcal{N}(aX + b, \sigma^2)$$

$$f(y^{(i)} | x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(y^{(i)} - (ax^{(i)} + b)\right)^2 / (2\sigma^2)}$$

2. Substitute 1. into objective fn.

$$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta) = \arg \max_{\theta} \sum_{i=1}^n \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\left(y^{(i)} - ax^{(i)} - b\right)^2 / (2\sigma^2)} \right]$$

$$\begin{aligned} \text{using} \\ \text{natural log} \end{aligned} \quad = \arg \max_{\theta} \left[\sum_{i=1}^n -\log \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n \left(y^{(i)} - ax^{(i)} - b\right)^2 \right]$$

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$

$$Y = aX + b + Z$$

$$Z \sim \mathcal{N}(0, \sigma^2)$$

Optimization
problem:

$$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

3. Use argmax properties
to simplify objective fn.

$$\arg \max_{\theta} \left[\sum_{i=1}^n \underbrace{-\log \sqrt{2\pi}\sigma}_{\substack{\text{doesn't} \\ \text{depend on } \theta = (a, b)}} - \frac{1}{2\sigma^2} \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right] \quad \text{(from previous slide)}$$

$$= \arg \max_{\theta} \left[\underbrace{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2}_{\substack{\text{irrelevant} \\ \text{constant with regard} \\ \text{to arg max}}} \right]$$

$$= \arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$$

Argmax refresher #1:

Invariant to additive constants

Argmax refresher #2:

Invariant to positive constant scalars

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$

$Y = aX + b + Z$

$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization
problem:

$$\arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

4. Celebrate!

$$\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$$



Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ :

1. Determine formula for $LL(\theta)$

log conditional likelihood

$$\sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

$$h(\theta) = - \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

2-D gradient:

$$\left(\frac{\partial h(\theta)}{\partial a}, \frac{\partial h(\theta)}{\partial b} \right)$$

3. Solve resulting (simultaneous) equations

(computer)
Gradient Ascent

2. Compute gradient

Model: $\theta = (a, b)$

$$Y = aX + b + Z$$

$$Z \sim \mathcal{N}(0, \sigma^2)$$

Optimization
problem:

$$\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$$

1. What is the derivative of the objective function w.r.t. a ?

$$\frac{\partial}{\partial a} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right] =$$

Calculus refresher #1:

Derivative(sum) = sum(derivative)

**Calculus refresher
#2:**

Chain rule ★★ ★

2. What is the derivative of the objective function wrt b ?



2. Compute gradient

Model: $\theta = (a, b)$
 $Y = aX + b + Z$
 $Z \sim \mathcal{N}(0, \sigma^2)$

Optimization
problem:

$$\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$$

1. What is the derivative of the objective function w.r.t. a ?

$$\begin{aligned} \frac{\partial}{\partial a} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right] &= - \sum_{i=1}^n \frac{d}{da} (y^{(i)} - ax^{(i)} - b)^2 \\ &= -2 \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b) (-x^{(i)}) \\ &= 2 \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b) x^{(i)} \end{aligned}$$

Calculus refresher #1:

Derivative(sum) = sum(derivative)

Calculus refresher #2:

Chain rule ★★

2. Compute gradient

Model: $\theta = (a, b)$

$$Y = aX + b + Z$$

$$Z \sim \mathcal{N}(0, \sigma^2)$$

Optimization
problem:

$$\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$$

1. What is the derivative of the objective function wrt a ?

$$\sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$$

2. What is the derivative of the objective function wrt b ?

$$\sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

we could do this, but we're going to pretend it's impossible, as it often will be for more difficult objective functions.

analytical solution for a_{MLE}, b_{MLE} : Set to 0 and solve simultaneous equations

Next up: We will reach the same solution computationally with **gradient ascent**.

Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ :

1. Determine
formula for $LL(\theta)$

log conditional likelihood

$$\sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

2. Differentiate $LL(\theta)$
wrt (each) θ

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

$$h(\theta) = - \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2$$

$$\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

3. Solve resulting
(simultaneous)
equations

(computer)
Gradient Ascent

3. Gradient ascent with multiple parameters (if time)

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
initialize  $\theta$ 
repeat many times:
  compute gradient
   $\theta \ += \ \eta \ * \ \text{gradient}$ 
```

How does this
work for
multiple
parameters?

3. Gradient ascent with multiple parameters

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0          # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    # TODO: fill in
```

```
    a +=  $\eta$  * gradient_a    #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

How do we
pseudocode the
gradients we
derived?

3. Gradient ascent with multiple parameters

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0 # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        diff = y - (a * x + b)
        gradient_a += 2 * diff * x
        gradient_b += 2 * diff
```

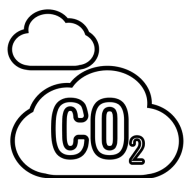
```
    a +=  $\eta$  * gradient_a #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

Finish computing gradient before updating any part of θ .

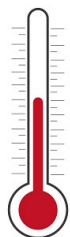
([demo](#))

Global land-ocean temperature prediction

Training data: $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$



CO2 levels



Output

Year 1	338.8
Year 2	340.0
...	
Year n	340.76

0.26
0.32
$\vdots$
0.14

$\mathbf{X} = (X_1)$
(assume one feature)

$Y \in \mathbb{R}$

Minimizing
Mean Square Error

Review

$$\theta_{MSE} = \arg \min_{\theta} E \left[(Y - g(X))^2 \right]$$

$$\hat{Y} = \hat{\rho}(X, Y) \frac{S_Y}{S_X} (X - \bar{X}) + \bar{Y}$$

$$a_{MSE} = 0.01452$$

$$b_{MSE} = 0.17511$$

3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0 # initialize  $\theta$   
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0  
    for each training example (x, y):  
        diff = y - (a * x + b)  
        gradient_a += 2 * diff * x  
        gradient_b += 2 * diff
```

```
a +=  $\eta$  * gradient_a #  $\theta$  +=  $\eta$  * gradient  
b +=  $\eta$  * gradient_b
```

Updates to a and b should include information from all n training datapoints

3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0 # initialize  $\theta$ 
repeat many times:
```

```
gradient_a, gradient_b = 0, 0
for each training example (x, y):
```

```
    diff = y - (a * x + b)
    gradient_a += 2 * diff * x
    gradient_b += 2 * diff
```

```
a +=  $\eta$  * gradient_a #  $\theta$  +=  $\eta$  * gradient
b +=  $\eta$  * gradient_b
```

How do we interpret the contribution of the i-th training datapoint?



3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0 # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        diff = y - (a * x + b)
        gradient_a += 2 * diff * x
        gradient_b += 2 * diff
```

```
    a +=  $\eta$  * gradient_a #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

Prediction error!

$$y^{(i)} - \hat{y}^{(i)}$$

3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0 # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        prediction_error = y - (a * x + b)
        gradient_a += 2 * prediction_error * x
        gradient_b += 2 * prediction_error
```

```
    a +=  $\eta$  * gradient_a #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0          # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        prediction_error = y - (a * x + b)
        gradient_a += 2 * prediction_error * x
        gradient_b += 2 * prediction_error
```

```
    a +=  $\eta$  * gradient_a      #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

$\hat{Y} = aX + b$, so
update to a should
also scale by $x^{(i)}$

3b. Interpret

Optimization problem: $\arg \max_{\theta} \left[- \sum_{i=1}^n (y^{(i)} - ax^{(i)} - b)^2 \right]$

$$= \arg \max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)(x^{(i)})$

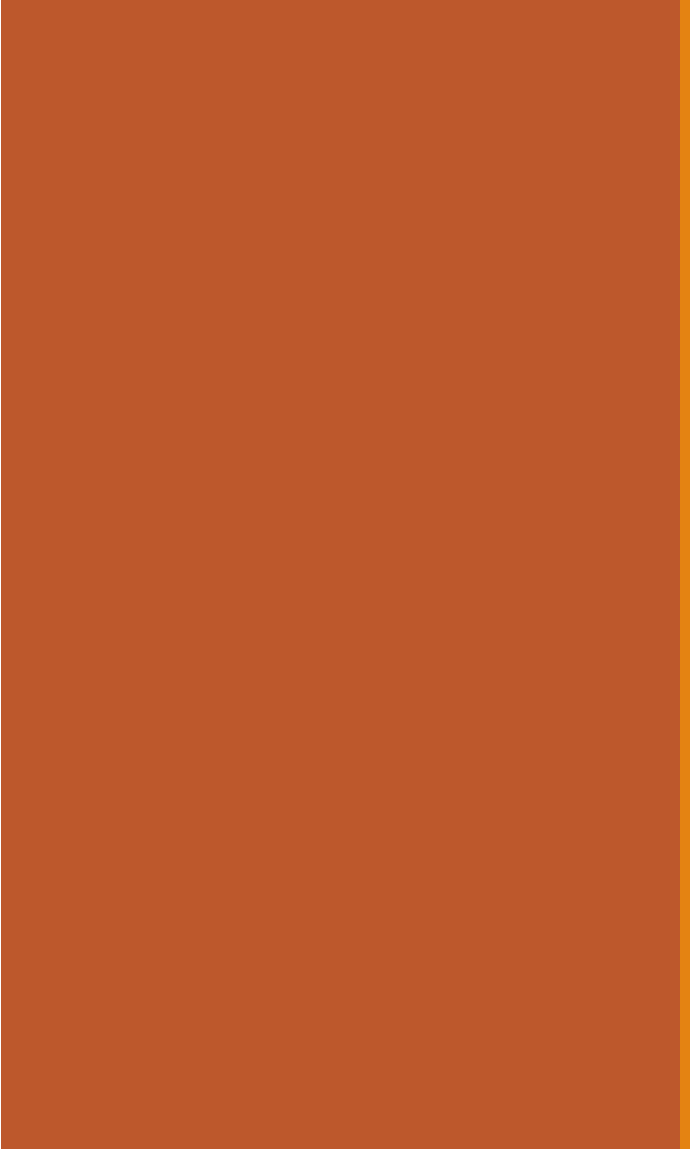
$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^n 2(y^{(i)} - ax^{(i)} - b)$$

```
a, b = 0, 0          # initialize  $\theta$ 
repeat many times:
```

```
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        prediction_error = y - (a * x + b)
        gradient_a += 2 * prediction_error * x
        gradient_b += 2 * prediction_error * 1
```

```
    a +=  $\eta$  * gradient_a      #  $\theta$  +=  $\eta$  * gradient
    b +=  $\eta$  * gradient_b
```

$\hat{Y} = aX + b$, so
update to b just
scales by 1, not $x^{(i)}$



Extra: Derivations

Don't make me get nonlinear!

$$\theta_{MSE} = \arg \min_{\theta=(a,b)} E[(Y - aX - b)^2]$$

1. Differentiate wrt (each) θ , set to 0

$$\begin{aligned} \frac{\partial}{\partial a} E[(Y - aX - b)^2] &= E \left[\frac{\partial}{\partial a} (Y - aX - b)^2 \right] && (E[\cdot] \text{ is a linear function w.r.t. } a) \\ &= E[-2(Y - aX - b)X] \\ &= -2E[XY] + 2aE[X^2] + 2bE[X] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial b} E[(Y - aX - b)^2] &= E[-2(Y - aX - b)] \\ &= -2E[Y] + 2aE[X] + 2b \end{aligned}$$

2. Solve resulting simultaneous equations

$$a_{MSE} = \frac{E[XY] - E[X]E[Y]}{E[X^2] - (E[X])^2} = \frac{\text{Cov}(X, Y)}{\text{Var}(X)} = \rho(X, Y) \frac{\sigma_Y}{\sigma_X}$$

$$b_{MSE} = E[Y] - a_{MSE}E[X] = \mu_Y - \rho(X, Y) \frac{\sigma_Y}{\sigma_X} \mu_X$$

Log conditional likelihood, a derivation

$\hat{Y} = g(X)$, where $g(\cdot)$ is a function with parameter θ

Show that θ_{MLE} maximizes the **log conditional likelihood** function:

$$\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta)$$

Proof:

$$\begin{aligned} \theta_{MLE} &= \arg \max_{\theta} \prod_{i=1}^n f(x^{(i)}, y^{(i)} | \theta) &&= \arg \max_{\theta} \sum_{i=1}^n \log f(x^{(i)}, y^{(i)} | \theta) && (\theta_{MLE} \text{ also maximizes } LL(\theta)) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log f(x^{(i)} | \theta) + \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta) && (\text{chain rule, log of product = sum of logs}) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log f(x^{(i)}) + \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta) && (x^{(i)} \text{ indep. of } \theta) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log f(y^{(i)} | x^{(i)}, \theta) && (f(x^{(i)}) \text{ constant w.r.t. } \theta) \end{aligned}$$