# 24: Linear Regression and Gradient Ascent 

Lisa Yan<br>June 1, 2020

## Quick slide reference

3 Linear Regression ..... 24a_linreg
7
Linear Regression: MSE ..... 24b_linreg_mse12 Linear Regression: MLE24c_linreg_mle
19 Gradient Ascent24d_gradient_ascent
24 Linear Regression with Gradient AscentLIVE

Extra: Derivations

## Linear Regression

## Today's goals

We are going to learn linear regression.

- Also known as "fit a straight line to data"
- However, linear models are too simple for more complex datasets.
- Furthermore, many tasks in CS deal with classification (categorical data), not regression.

The reason we cover this topic is to teach us important skills that will help us design and understand more complicated ML algorithms:

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

Training data: $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right),\left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots,\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$


## Linear Regression

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

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

## Training

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

Two approaches:

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


# Linear <br> Regression: MSE 

## Mean Squared Error (MSE)

For regression tasks, we usually want a $g(X)$ that minimizes MSE:

$$
\theta_{M S E}=\underset{\theta}{\arg \min } E\left[(Y-\hat{Y})^{2}\right]=\underset{\theta}{\arg \min } E\left[(Y-g(X))^{2}\right]
$$

- $Y$ and $\hat{Y}=g(X)$ are both random variables
- Intuitively: Choose parameter $\theta$ that minimizes the expected squared deviation ("error") of your prediction $\hat{Y}$ from the true $Y$

For linear regression, where $\theta=(a, b)$ and $\hat{Y}=a X+b$ :

$$
E\left[(Y-a X-b)^{2}\right]
$$

## Don't make me get non-linear!

$$
\begin{gathered}
\theta_{M S E}=\underset{\theta=(a, b)}{\arg \min } E\left[(Y-a X-b)^{2}\right] \\
a_{M S E}=\rho(X, Y) \frac{\sigma_{Y}}{\sigma_{X}}, \quad b_{M S E}=\mu_{Y}-a_{M S E} \mu_{X}
\end{gathered}
$$

(Derivation included at the end of this lecture)

Can we find these statistics on $X$ and $Y$ from our training data?
Training data: $\quad\left(x^{(1)}, y^{(1)}\right),\left(x^{(2)}, y^{(2)}\right), \ldots,\left(x^{(n)}, y^{(n)}\right)$

## Don't make me get non-linear!

$$
\begin{gathered}
\theta_{M S E}=\underset{\theta=(a, b)}{\arg \min } E\left[(Y-a X-b)^{2}\right] \\
a_{M S E}=\rho(X, Y) \frac{\sigma_{Y}}{\sigma_{X}}, \quad b_{M S E}=\mu_{Y}-a_{M S E} \mu_{X}
\end{gathered}
$$

(Derivation included at the end of this lecture)

Can we find these statistics on $X$ and $Y$ from our training data?
Training data: $\quad\left(x^{(1)}, y^{(1)}\right),\left(x^{(2)}, y^{(2)}\right), \ldots,\left(x^{(n)}, y^{(n)}\right)$

Estimate parameters based on observed training data:

$$
\hat{a}_{M S E}=\hat{\rho}(X, Y) \frac{S_{Y}}{S_{X}}, \quad \hat{b}_{M S E}=\bar{Y}-\hat{a}_{M S E} \bar{X}
$$

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

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

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

## Training

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

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

$$
\hat{a}_{M S E}=\hat{\rho}(X, Y) \frac{S_{Y}}{S_{X}}, \quad \hat{b}_{M S E}=\bar{Y}-\hat{a}_{M S E} \bar{X}
$$

# Linear <br> Regression: MLE 

## Linear Regression

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

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

$$
\begin{array}{ll}
\text { Training } & \text { Learn parameters } \theta=(a, b) \\
& \text { Training data: }\left(x^{(1)}, y^{(1)}\right),\left(x^{(2)}, y^{(2)}\right), \ldots,\left(x^{(n)}, y^{(n)}\right)
\end{array}
$$

We've seen which parameters minimize mean squared error.

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

Consider a sample of $n$ i.i.d. random variables $X_{1}, X_{2}, \ldots, X_{n}$.

- $X_{i}$ was drawn from a distribution with density function $f\left(X_{i} \mid \theta\right)$.
- Observed data: $\left(X_{1}, X_{2}, \ldots, X_{n}\right)$

Likelihood question:
How likely is the observed data ( $X_{1}, X_{2}, \ldots, X_{n}$ ) given parameter $\theta$ ?
Likelihood function, $L(\theta)$ :

$$
L(\theta)=f\left(X_{1}, X_{2}, \ldots, X_{n} \mid \theta\right)=\prod_{i=1}^{n} f\left(X_{i} \mid \theta\right)
$$

This is just a product, since $X_{i}$ are i.i.d.

## Likelihood of the training data

Training data ( $n$ datapoints):

- $\left(x^{(i)}, y^{(i)}\right)$ drawn i.i.d. from a distribution $f\left(X=x^{(i)}, Y=y^{(i)} \mid \theta\right)=f\left(x^{(i)}, y^{(i)} \mid \theta\right)$
- $\hat{Y}=g(X)$, where $g(\cdot)$ is a function with parameter $\theta$

We can show that $\theta_{M L E}$ maximizes the log conditional likelihood function:

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

(This derivation is included at the end of this video)

## Linear Regression, MLE

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

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

2. Define maximum likelihood estimator:

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

4 Issue: We have a model of the prediction $\hat{Y}($ and $\operatorname{not} Y)$

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

$$
\begin{array}{rlr}
Y & =\hat{Y}+Z \quad \quad \text { error/noise } \\
& =a X+b+Z \quad \text { (also random) }
\end{array}
$$

## Comparison: MSE vs MLE

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

Minimum Mean Squared Error

$$
\theta_{M S E}=\underset{\theta}{\arg \min } E\left[(Y-g(X))^{2}\right]
$$

- Do not directly model $Y$ (nor error)
- Parameters are estimates of statistics from training data:

$$
\begin{aligned}
& \hat{a}_{M S E}=\hat{\rho}(X, Y) \frac{S_{Y}}{S_{X}} \\
& \hat{b}_{M S E}=\bar{Y}-\hat{a}_{M S E} \bar{X}
\end{aligned}
$$

Maximum Likelihood Estimation

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

- Directly model error between predicted $\hat{Y}$ and $Y$

$$
Y=\hat{Y}+Z=a X+b+Z
$$

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

$$
\theta_{M S E}=\theta_{M L E}!
$$

## Linear Regression, MLE (next steps)

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

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

2. Define maximum likelihood estimator:

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

3. Model error, Z:

$$
Y=a X+b+Z, \text { where } \mathrm{Z} \sim \mathcal{N}\left(0, \sigma^{2}\right)
$$

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

We will not analytically find a solution. Instead, we are going to use gradient ascent, an iterative optimization algorithm.

## Gradient Ascent

## Computing the MLE

General approach for finding $\theta_{M L E}=\underset{\theta}{\arg \max } L L(\theta)$ :

## 1. Determine <br> formula for $L L(\theta)$

$L L(\theta)=\sum_{i=1}^{n} \log f\left(X_{i} \mid \theta\right)$
2. Differentiate $L L(\theta)$ w.r.t. (each) $\theta$

$$
\frac{\partial L L(\theta)}{\partial \theta}
$$

To maximize:

$$
\frac{\partial L L(\theta)}{\partial \theta}=0
$$

3. Solve resulting (simultaneous) equations
(algebra or computer)

If algebra is intractable, we can still find a maximum using gradient ascent!

## Multiple ways to calculate argmax

$$
\begin{aligned}
& \text { Let } f(x)=-x^{2}+4 \text {, } \\
& \text { where }-2<x<2 .
\end{aligned}
$$

What is $\arg \max \underbrace{f(x)}_{\text {objective function }}$
A. Graph and guess

B. Differentiate, set to 0 , and solve

$$
\begin{aligned}
\frac{d f}{d x} & =-2 x=0 \\
x & =0
\end{aligned}
$$

C. Gradient ascent: educated guess \& check


## Gradient ascent

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



If your function is concave,
Local maxima = global maxima

## Gradient ascent algorithm

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

$$
\begin{aligned}
& \text { Let } f(x)=-x^{2}+4, \\
& \text { where }-2<x<2 .
\end{aligned}
$$



1. $d f$
$\frac{d f}{d x}=-2 x \quad$ Gradient at $x$
2. Gradient ascent algorithm: initialize x repeat many times: compute gradient x += $\eta$ * gradient

## (live)

24: Linear Regression
and Gradient Ascent

Lisa Yan<br>June 1, 2020

## Three goals today

1. How to model likelihood of training data $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

( $\theta_{\text {MLE }}$ maximizes log conditional likelihood)
2. What rules of argmax/calculus are important to remember
3. What gradient ascent is, why it is useful, and how to use it

1. Compute gradient.
2. initialize $x$ repeat many times: compute gradient

## Linear Regression, MLE (so far)

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

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

2. Define maximum likelihood estimator:

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

3. Model error, Z:
$Y=a X+b+Z$, where $Z \sim \mathcal{N}\left(0, \sigma^{2}\right)$
4. Pick $\theta=(a, b)$ that maximize likelihood of training data

Let's get started!

## Computing the MLE with gradient ascent

General approach for finding $\theta_{M L E}$, the MLE of $\theta$ :

## 1. Determine formula for $L L(\theta)$


$\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$
Now: optimize log conditional likelihood
2. Differentiate $L L(\theta)$
w.r.t. (each) $\theta$


$$
\begin{aligned}
& \text { To maximize: } \\
& \frac{\partial L L(\theta)}{\partial \theta}=0
\end{aligned}
$$

$$
\frac{\partial}{\partial \theta_{j}} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

3. Solve resulting (simultaneous) equations
(algebra or computer)
(computer)
Gradient Ascent

## 1. Determine formula for log conditional likelihood

$$
\begin{aligned}
\text { Model: } & \theta=(a, b) \\
& Y=a X+b+Z \\
& Z \sim \mathcal{N}\left(0, \sigma^{2}\right)
\end{aligned}
$$

1. What is the conditional distribution, $Y \mid X, \theta$ ?
2. Rewrite the objective:
$\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

## 1. Determine formula for log conditional likelihood

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

$$
Y=a X+b+Z
$$

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

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

$$
\begin{aligned}
& Y \mid X, \theta \sim \mathcal{N}\left(a X+b, \sigma^{2}\right) \\
& f\left(y^{(i)} \mid x^{(i)}, \theta\right)=\frac{1}{\sqrt{2 \pi} \sigma} e^{-\left(y^{(i)}-\left(a x^{(i)}+b\right)\right)^{2} /\left(2 \sigma^{2}\right)}
\end{aligned}
$$

2. Rewrite the objective:

$$
\begin{aligned}
\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) & =\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log \left[\frac{1}{\sqrt{2 \pi} \sigma} e^{-\left(y^{(i)}-a x^{(i)}-b\right)^{2} /\left(2 \sigma^{2}\right)}\right] \\
\text { using } & =\underset{\theta}{\arg \max }\left[\sum_{i=1}^{n}-\log \sqrt{2 \pi} \sigma-\frac{1}{2 \sigma^{2}} \sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
\end{aligned}
$$

## 1. Determine formula for log conditional likelihood

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

$$
\begin{aligned}
& Y=a X+b+Z \\
& Z \sim \mathcal{N}\left(0, \sigma^{2}\right)
\end{aligned}
$$

3. Use argmax properties to get rid of constants

$$
\begin{aligned}
& \underset{\theta}{\arg \max }\left[\sum_{i=1}^{n}-\log \sqrt{2 \pi} \sigma-\frac{1}{2 \sigma^{2}} \sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right] \quad \text { (from previous slide) } \\
& \quad=\underset{\theta}{\arg \max }\left[-\frac{1}{2 \sigma^{2}} \sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right] \\
& \quad \begin{array}{l}
\text { Argmax refresher \#1: } \\
\text { Invariant to additive constants }
\end{array} \\
& \quad=\underset{\theta}{\arg \max \left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]_{\text {Argmax refresher \#2: }}^{\text {Invariant to positive constant scalars }}} \begin{array}{l}
\text { Stanford University } 30
\end{array}
\end{aligned}
$$

## 1. Determine formula for log conditional likelihood

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

## Optimization problem: $\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

4. Celebrate!

$$
\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
$$

## 2. Compute gradient

Model: $\quad \theta=(a, b)$
$Y=a X+b+Z$
$Z \sim \mathcal{N}\left(0, \sigma^{2}\right)$
Optimization problem:

$$
\begin{aligned}
\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log & f\left(y^{(i)} \mid x^{(i)}, \theta\right) \\
& =\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
\end{aligned}
$$

1. What is the derivative of the (w.r.t. - "with respect to") objective function w.r.t. $a$ ?

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

Calculus refresher \#1:
Derivative(sum) = sum(derivative)

$$
\begin{array}{ll}
=-\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(-x^{(i)}\right) & \text { Calculu } \\
=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) & \text { (rewrite) }
\end{array}
$$

Calculus refresher \#2: Chain rule

## 2. Compute gradient

Model: $\quad \theta=(a, b)$
$Y=a X+b+Z$
$Z \sim \mathcal{N}\left(0, \sigma^{2}\right)$
Optimization problem:

$$
\begin{aligned}
& \underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) \\
&= \underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
\end{aligned}
$$

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

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

2. What is the derivative of the objective function w.r.t. $b$ ?

## 2. Compute gradient

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

$$
\begin{array}{r}
\text { Optimization } \quad \underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) \\
=\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
\end{array}
$$

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

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

2. What is the derivative of the objective function w.r.t. $b$ ?

## 2. Compute gradient

Model: $\quad \theta=(a, b)$
$Y=a X+b+Z$
$Z \sim \mathcal{N}\left(0, \sigma^{2}\right)$
Optimization problem:

$$
\begin{aligned}
\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log & f\left(y^{(i)} \mid x^{(i)}, \theta\right) \\
= & \underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]
\end{aligned}
$$

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

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

2. What is the derivative of the objective function w.r.t. $b$ ?

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

If we set to 0 and solve, we will get an analytical solution for $a_{M L E}, b_{M L E}$.
We will reach the same solution with gradient ascent.

Interlude for
jokes/announcements

## Announcements

## Extra Office Hours

Check out the OH calendar for extra help on p-set 6 and the final!

## Interesting probability news

## Astronomer Uses Bayesian Statistics to Weigh Likelihood of Complex Life and Intelligence beyond Earth

"In Bayesian inference, prior probability distributions always need to be selected," [the astronomer] said.
"But a key result here is that when one compares the rare-life versus common-life scenarios, the commonlife scenario is always at least nine times more likely than the rare one."

## 3. Gradient ascent with multiple parameters

$$
\begin{aligned}
\text { Optimization } \underset{\text { problem: }}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right] & \text { Gradient: } \frac{\partial h(\theta)}{\partial a}=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
=\underset{\theta}{\arg \max h(\theta)} & \frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

```
initialize 0
repeat many times:
    compute gradient
    0 += \eta * gradient
```


## 3. Gradient ascent with multiple parameters

$$
\begin{aligned}
\text { Optimization } \underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right] & \text { Gradient: } \frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
& =\underset{\theta}{\arg \max } h(\theta) & \frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

$a, b=0,0$
\# initialize $\theta$
repeat many times:
gradient_a, gradient_b $=0,0$
\# TODO: fill in

How do we pseudocode the gradient computation?

$$
\begin{aligned}
& \mathrm{a}+=\eta * \text { gradient_a } \\
& \mathrm{b}+=\eta * \text { gradient_b }
\end{aligned}
$$

## 3. Gradient ascent with multiple parameters

Optimization
$\quad$ problem:
$\arg \max$$\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]$
Gradient: $\frac{\partial h(\theta)}{\partial a}=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right)$

$$
=\underset{\theta}{\arg \max } h(\theta)
$$

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

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

$$
\begin{aligned}
& \text { gradient_a, gradient_b = 0, 0 } \\
& \text { for each training example }(x, y) \text { : } \\
& \text { diff }=y-(a * x+b) \\
& \text { gradient_a += } 2 \text { * diff * x } \\
& \text { gradient_b += } 2 \text { * diff } \\
& \text { a += } \eta \text { * gradient_a \# } \theta \text { += } \eta \text { * gradient } \\
& \text { b += } \eta \text { * gradient_b }
\end{aligned}
$$

Finish computing gradient before updating any part of $\theta$.

## Global land-ocean temperature prediction

Training data: $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right),\left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots,\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$

|  | $\xrightarrow{\left(\mathrm{CO}_{2}\right)}$ | 5 |
| :---: | :---: | :---: |
|  | CO2 levels | Output |
| Year 1 | 338.8 | 0.26 |
| Year 2 | 340.0 | 0.32 |
| ... |  | : |
| Year $n$ | 340.76 | 0.14 |
| $\boldsymbol{X}=\left(X_{1}\right)$ <br> (assume one feature) |  | $Y \in \mathbb{R}$ |

## Global land-ocean temperature prediction

Training data: $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right),\left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots,\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$


## 3b. Interpret

Optimization $\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]$

$$
=\underset{\theta}{\arg \max } h(\theta)
$$

$$
\text { Gradient: } \begin{aligned}
\frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
\frac{\partial h(\theta)}{\partial b} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

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

$$
\begin{aligned}
& \text { aradient } a, ~ a r a d i e n t ~ \\
& \text { for each training example }(x, y): \\
& \text { diff }=y-(a * x+b) \\
& \text { gradient_a }+=2 * \text { diff } * x \\
& \text { gradient_b += } 2 * \text { diff } \\
& \text { a }+=\eta * \text { gradient_a } \# \theta+=\eta * \text { gradient } \\
& b+=\eta * \text { gradient_b }
\end{aligned}
$$

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

## 3b. Interpret

Optimization $\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]$

$$
=\underset{\theta}{\arg \max } h(\theta)
$$

$$
\text { Gradient: } \begin{aligned}
\frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
\frac{\partial h(\theta)}{\partial b} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

$a, b=0,0$
\# 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

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

$$
\begin{aligned}
& \mathrm{a}+=\eta * \text { gradient_a } \\
& \mathrm{b}+=\eta * \ln +=\eta * \operatorname{gradient}
\end{aligned}
$$

## 3b. Interpret



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

$$
=\underset{\theta}{\arg \max } h(\theta) \quad \frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
$$

$a, b=0,0$
\# initialize $\theta$
repeat many times:
gradient_a, gradient_b $=0,0$
for each training example $(x, y)$ :
diff $=y-(a * x+b)$ gradient_a += $2 * \operatorname{diff} * x$ gradient_b += 2 * diff

Prediction error!

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

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

## 3b. Interpret



$$
=\underset{\theta}{\arg \max } h(\theta)
$$

$$
\text { Gradient: } \begin{aligned}
\frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
\frac{\partial h(\theta)}{\partial b} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

$a, b=0,0$
\# 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 $\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]$

$$
=\underset{\theta}{\arg \max } h(\theta)
$$

$$
\text { Gradient: } \begin{aligned}
\frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
\frac{\partial h(\theta)}{\partial b} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

$a, b=0,0$
\# 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
$\hat{Y}=a X+b$, so
update to $a$ should also scale by $x^{(i)}$

## 3b. Interpret

Optimization $\underset{\theta}{\arg \max }\left[-\sum_{i=1}^{n}\left(y^{(i)}-a x^{(i)}-b\right)^{2}\right]$

$$
=\underset{\theta}{\arg \max } h(\theta)
$$

$$
\text { Gradient: } \begin{aligned}
\frac{\partial h(\theta)}{\partial a} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)\left(x^{(i)}\right) \\
\frac{\partial h(\theta)}{\partial b} & =\sum_{i=1}^{n} 2\left(y^{(i)}-a x^{(i)}-b\right)
\end{aligned}
$$

$a, b=0,0$
\# initialize $\theta$
repeat many times:
gradient_a, gradient_b $=0,0$
for each training example $(x, y)$ :
prediction_error $=y-(a * x+b)$

$$
\hat{Y}=a X+b, \text { so }
$$

gradient_a += 2 * prediction_error * x update to $b$ just gradient_b += $2 *$ prediction_error * 1 scales by 1 , not $x^{(i)}$
a += $\eta$ * gradient_a \# $\theta+=\eta$ * gradient
b += $\eta$ * gradient_b

## Reflecting on today

We did a lot today!

- Learned gradient ascent
- Modeled likelihood of training dataset
- Thanked argmax for its convenience
- Remembered calculus
- Implemented gradient ascent with multiple parameters to optimize for


## A General Approach to (Deep) Learning

initialize $\theta_{j}=0$ for $0 \leq j \leq m$ repeat many times:

$$
\operatorname{gradient}[j]=0 \text { for } 0 \leq j \leq m
$$

for each training example ( $x, y$ ):
for each $0 \leq j \leq m:$
// update gradient[j] using any objective function
$\theta_{j}+=\eta * \operatorname{gradient}[j]$ for all $0 \leq j \leq m$

We can plug in all sorts of prediction functions and objective functions!

As long as you know how each parameter impacts the final objective function, you can train a model with gradient ascent.

## Teaser: Logistic Regression

$\begin{array}{r}\text { Gradient } \\ \text { Ascent Step }\end{array} \theta_{j}^{\text {new }}=\theta_{j}^{\text {old }}+\eta \cdot \sum_{i=1}^{n}\left[y^{(i)}-\sigma\left(\theta^{\text {old }^{T}} \boldsymbol{x}^{(i)}\right)\right] x_{j}^{(i)}$
initialize $\theta_{j}=0$ for $0 \leq j \leq m$ repeat many times:
gradient[j] $=0$ for $0 \leq j \leq m$
for each training example ( $x, y$ ):
for each $0 \leq \mathrm{j} \leq \mathrm{m}$ :

$$
\text { gradient }[j]+=\left[y-\frac{1}{1+e^{-\theta^{T} x}}\right] x_{j}
$$

We'll discover a new prediction function and objective function for predicting probabilities. It's the same general approach, just with a tweak to the gradient update.

## Teaser: Logistic Regression


initialize $\theta_{j}=0$ for $0 \leq j \leq m$ repeat many times:
gradient[j] $=0$ for $0 \leq j \leq m$
for each training example ( $x, y$ ):
for each $0 \leq j \leq m:$

$$
\text { gradient }[j]+=\left[y-\frac{1}{1+e^{-\theta^{T} x}}\right] x_{j}
$$

$\theta_{j}+=\eta * \operatorname{gradient}[j]$ for all $0 \leq j \leq m$


#### Abstract

We'll discover a new prediction function and objective function for predicting probabilities. It's the same general approach, just with a tweak to the gradient update.


(Hint: this slide can help with p-set 6!)

## Extra: Derivations

## Don't make me get non-linear!

$$
\theta_{M S E}=\underset{\theta=(a, b)}{\arg \min } E\left[(Y-a X-b)^{2}\right]
$$

1. Differentiate w.r.t. (each) $\theta$,

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

$(E[\cdot]$ is a linear function w.r.t. a) set to 0
2. Solve resulting simultaneous

$$
\begin{aligned}
& a_{M S E}=\frac{E[X Y]-E[X] E[Y]}{E\left[X^{2}\right]-(E[X])^{2}}=\frac{\operatorname{Cov}(X, Y)}{\operatorname{Var}(X)} \\
& =\rho(X, Y) \frac{\sigma_{Y}}{\sigma_{X}} \\
& b_{M S E}=E[Y]-a_{M S E} E[X] \quad=\mu_{Y}-\rho(X, Y) \frac{\sigma_{Y}}{\sigma_{X}} \mu_{X}
\end{aligned}
$$ equations

## Log conditional likelihood, a derivation

Show that $\theta_{M L E}$ maximizes the Iog conditional likelihood function:

$$
\theta_{M L E}=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)
$$

Proof: $\quad \theta_{M L E}=\underset{\theta}{\arg \max } \prod_{i=1}^{n} f\left(x^{(i)}, y^{(i)} \mid \theta\right) \quad=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(x^{(i)}, y^{(i)} \mid \theta\right) \quad \begin{aligned} & \left(\theta_{M L E} \text { also }\right. \\ & \text { maximizes } L L(\theta))\end{aligned}$

$$
\begin{array}{ll}
=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(x^{(i)} \mid \theta\right)+\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) & \begin{array}{l}
\text { (chain rule, } \\
\text { log of product }=\text { sum of logs) }
\end{array} \\
=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(x^{(i)}\right)+\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) & \left(x^{(i)} \text { indep. of } \theta\right) \\
=\underset{\theta}{\arg \max } \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) & \left(f\left(x^{(i)}\right) \text { constant w.r.t. } \theta\right)
\end{array}
$$

