CS540 Introduction to Artificial Intelligence
Lecture 2

Young Wu
Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

June 22, 2021

Two-thirds of the Average Game

Quiz

Socrative
CS540
wisc 2D

Q1

- Pick the number that is the closest to two-thirds of the average of the numbers other people picked.
- A: 0
- B: 1
- C: 2
- D: 3
- E: 4

avg = 1.2

$\frac{2}{3}$ avg = 0.8

Remind Me to Start Recording

Admin

- The annotated slides are posted on Q1, Q2, etc.
- The lecture recordings are shared on Canvas, are they ✓ visible?
- The messages you send in chat will be recorded: you can change your Zoom name now before I start recording.

Two-thirds of the Average Game

Quiz

A I

- Pick an integer between 0 and 100 (including 0 and 100) that is the closest to two-thirds of the average of the numbers other people picked.

Supervised Learning Example

Motivation

Data	images of cats and dogs
Features (Input)	height, length, eye color, ...
Output	cat or dog

Data	emails
Features (Input)	word count, capitalization, ...
Output	spam or ham

Supervised Learning

Motivation

x_i

0 ham
1 spam

Data	Features (Input)	Output	-
Training	$\{(x_{i1}, ..., x_{im})\}_{i=1}^{n'}$	$\{y_i\}_{i=1}^{n'}$	find "best" $\hat{f}$
-	observable	known	-
Test	$(x'_1, ..., x'_m)$	y'	guess $\hat{y} = \hat{f}(x')$
-	observable	unknown	-

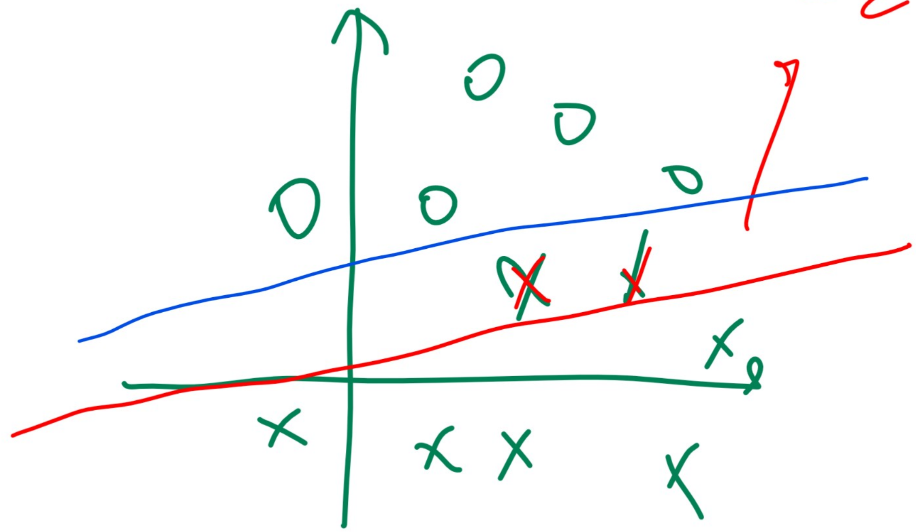
Loss Function Diagram

Motivation

best → objective

find $\hat{f}$ minimize # mistake on training $= 2$

cost
loss



Zero-One Loss Function

Motivation

- An objective function is needed to select the "best" $\hat{f}$. An example is the zero-one loss.

Loss

$$\mathbb{1}_E = \begin{cases} 1 & E \text{ true} \\ 0 & E \text{ false} \end{cases}$$

$$\hat{f} = \arg\min_f \sum_{i=1}^{n} \mathbb{1}_{\{f(x_i)\neq y_i\}}$$

$$\begin{cases} 1 & \text{mistake} \\ 0 & \text{correct prediction} \end{cases}$$

- $\arg\min_f$ objective (f) outputs the function that minimizes the objective.

C

- The objective function is called the cost function (or the loss function), and the objective is to minimize the cost.

Squared Loss Function

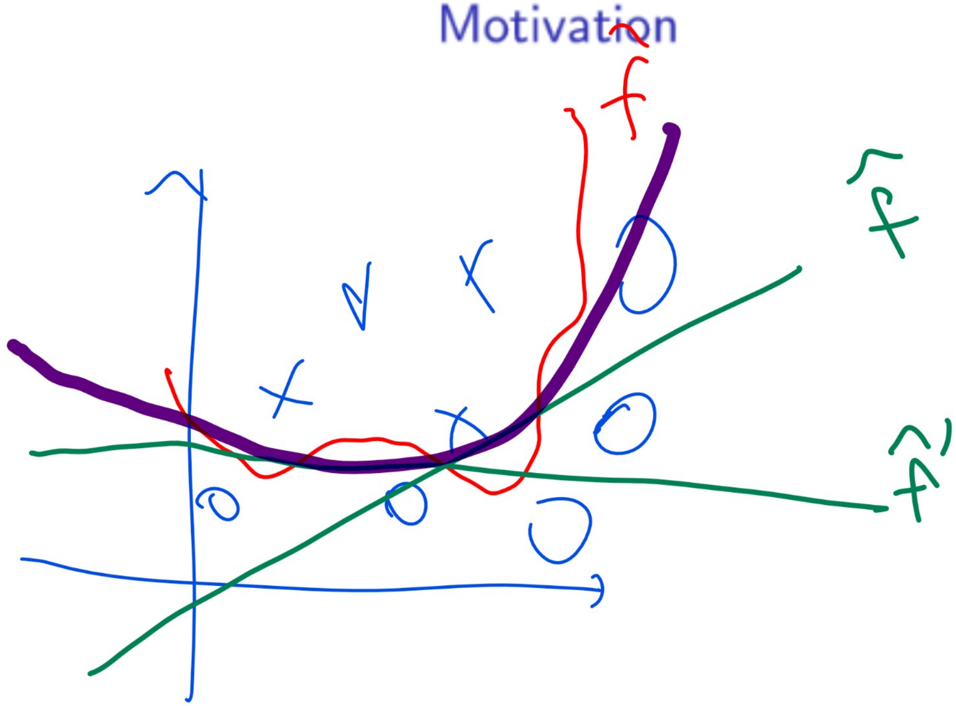
Motivation

- Zero-one loss counts the number of mistakes made by the classifier. The best classifier is the one that makes the fewest mistakes.
- Another example is the squared distance between the predicted and the actual y value:

$$\hat{f} = \arg\min_{f} \frac{1}{2} \sum_{i=1}^{n} \left(f\left(x_i\right) - y_i\right)^2$$

Function Space Diagram

Motivation



Hypothesis Space

Motivation

- There are too many functions to choose from.
- There should be a smaller set of functions to choose $\hat{f}$ from.

min = total # mistake

$$\hat{f} = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(f\left(x_i\right) - y_i\right)^2$$

the f that minimize cost.

- The set $\mathcal{H}$ is called the hypothesis space.

Activation Function

Motivation

- Suppose $\mathcal{H}$ is the set of functions that are compositions between another function g and linear functions.

$$\left(\hat{w}, \hat{b}\right) = \arg\min_{w,b} \frac{1}{2} \sum_{i=1}^{n} (a_i - y_i)^2$$

$f(x_i)$

$$\text{where } a_i = g\left(w^T x_i + b\right)$$

- g is called the activation function.

Linear Threshold Unit

Motivation

$w^T x + b$

- One simple choice is to use the step function as the activation function:

$$g\left(\boxed{\cdot}\right) = \mathbb{1}_{\{\boxed{\cdot} \geqslant 0\}} = \begin{cases} 1 & \text{if } \boxed{\cdot} \geqslant 0 \\ 0 & \text{if } \boxed{\cdot} < 0 \end{cases} \tag{1}$$

- This activation function is called linear threshold unit (LTU).

Sigmoid Activation Function

Motivation

- When the activation function g is the sigmoid function, the problem is called logistic regression.

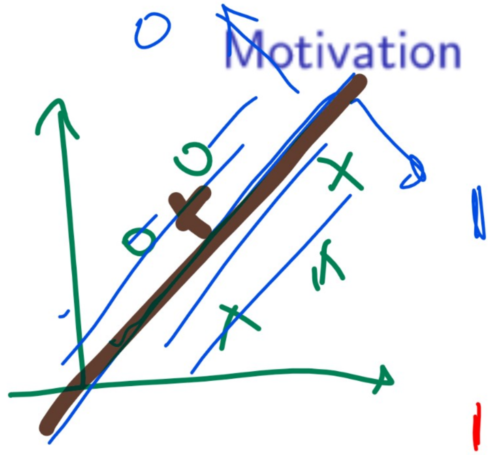
$$g\left(\boxed{\cdot}\right) = \frac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$$

$w^T x + b$

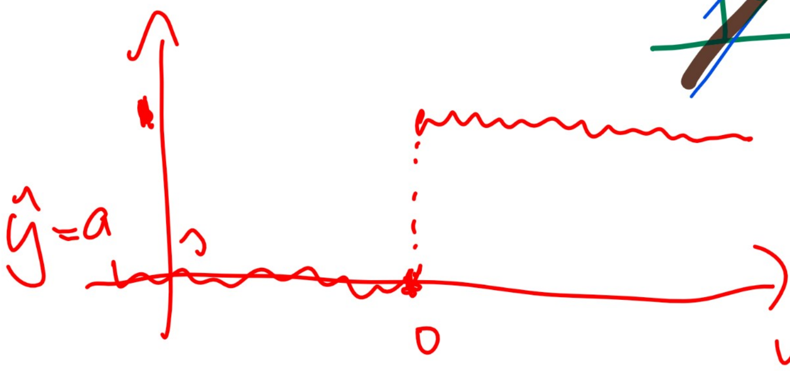
- This g is also called the logistic function.

Sigmoid Function Diagram

Motivation



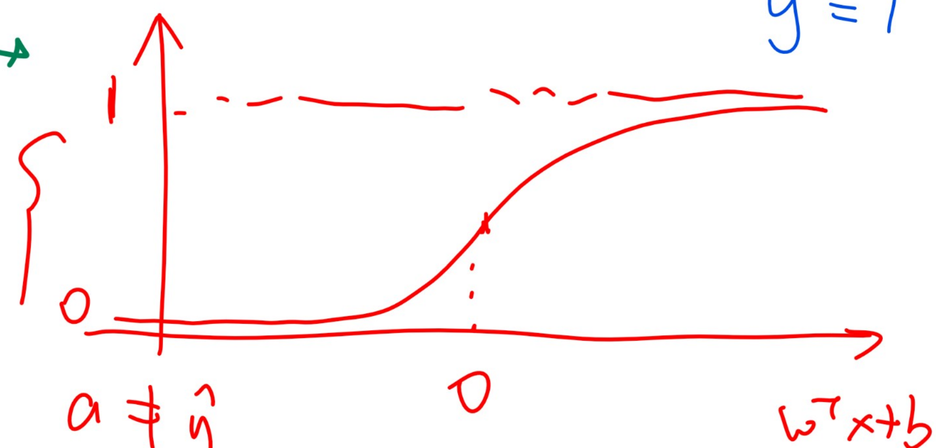
LTU



Perceptron Algorithm

logistic

$a_i = 0.99$
$\hat{y} = 1$



$a \neq \hat{y}$

Prob that label is 1

Gradient Descent

Cross-Entropy Loss Function

Motivation

- The cost function used for logistic regression is usually the log cost function.

$$C(f) = -\sum_{i=1}^{n} \left(y_i \log(f(x_i)) + (1 - y_i)\log(1 - f(x_i))\right)$$

Convex

- It is also called the cross-entropy loss function.

Logistic Regression Objective

Motivation

- The logistic regression problem can be summarized as the following.

Cost / Loss

Convex

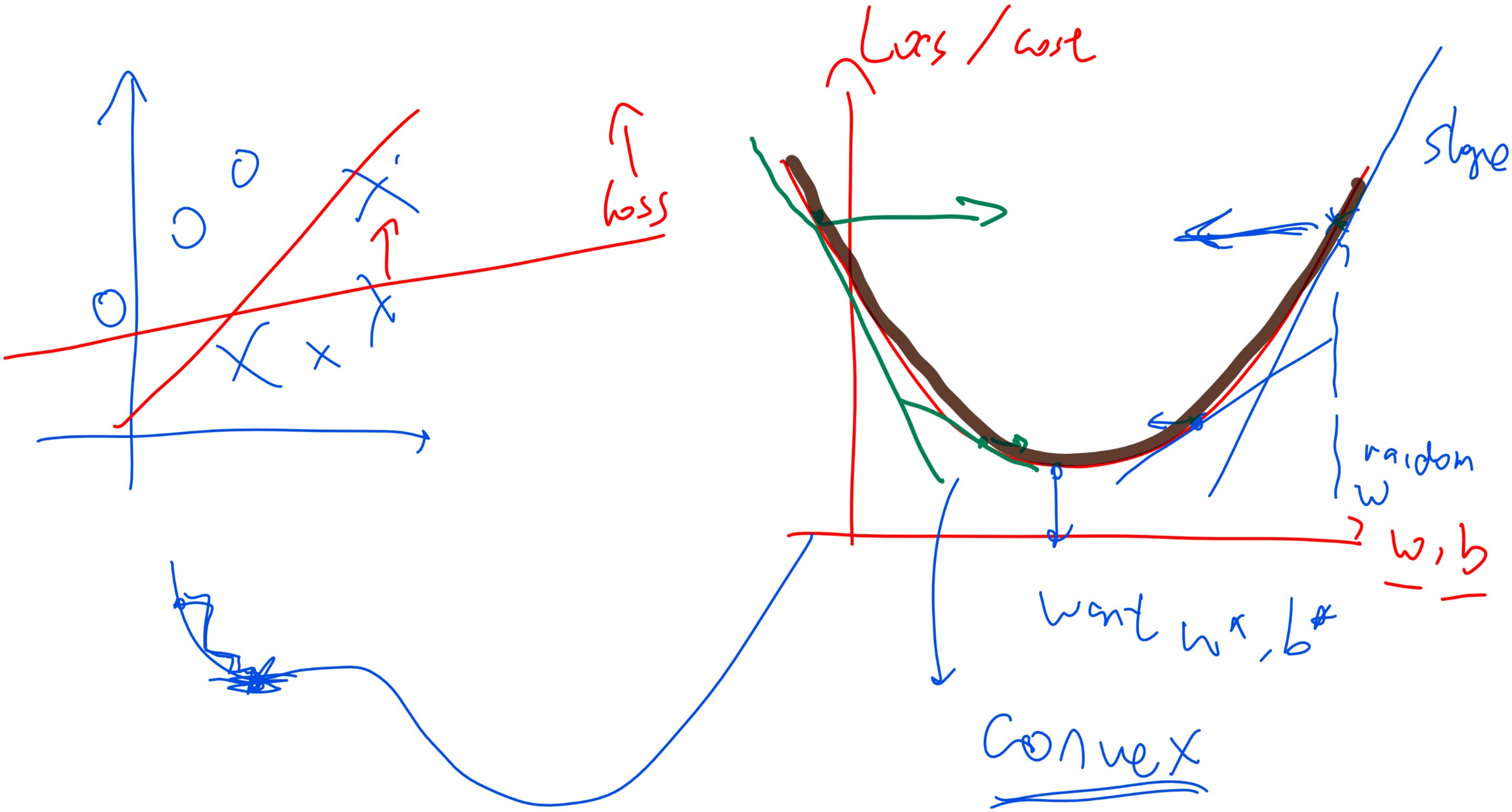
$$\left(\hat{w}, \hat{b}\right) = \arg\min_{w,b} - \sum_{i=1}^{n} \left(y_i \log\left(a_i\right) + \left(1 - y_i\right) \log\left(1 - a_i\right)\right)$$

$$\text{where } a_i = \underbrace{\frac{1}{1 + \exp\left(-z_i\right)}}_{\text{activation}} \text{ and } z_i = \boxed{w^T x_i + b}$$

Cost

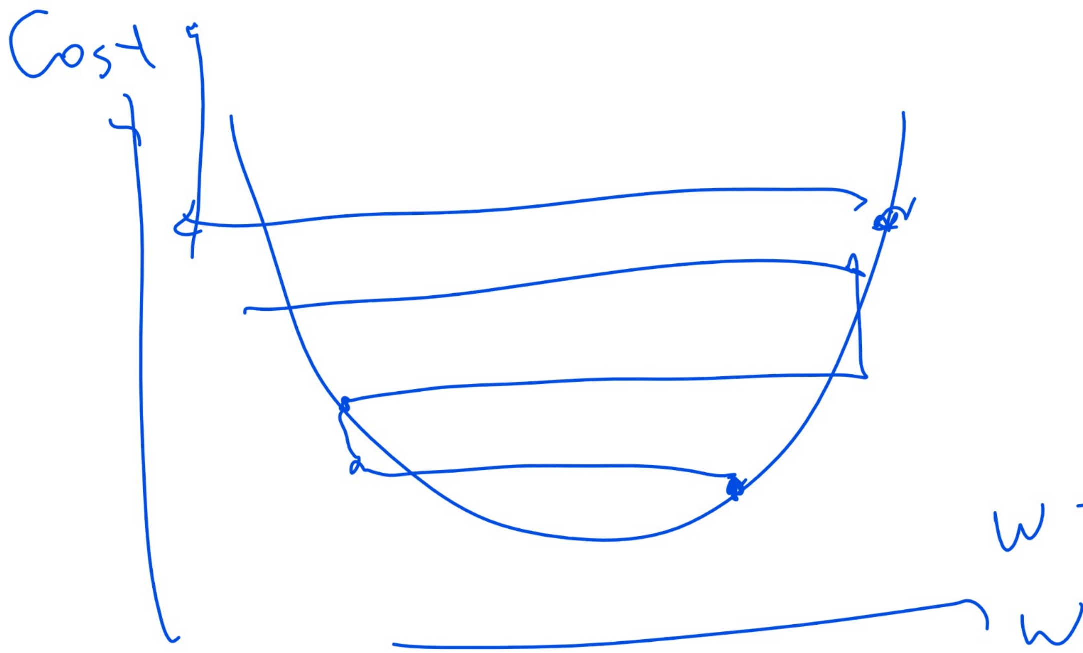
Optimization Diagram

Motivation



Learning Rate Demo

Motivation



negative direction of gradient

slope

$$w = w - \boxed{\alpha}(a_i - y_i)\, x_i$$

fraction

Logistic Regression

Description

- Initialize random weights.
- Evaluate the activation function.
- Compute the gradient of the cost function with respect to each weight and bias.
- Update the weights and biases using gradient descent.
- Repeat until convergent.

Logistic Gradient Derivation 1

CE Cost

Definition natural log ← ln

$$C = -\sum_{i=1}^{n} y_i \log(a_i) + (1-y_i)\log(1-a_i)$$

$$a_i = \frac{1}{1+e^{-(w^T x_i + b)}}$$

logistic activation

$j = 1, 2 \ldots m$

$$\frac{\partial C}{\partial w_j} = \sum_{i=1}^{n} \frac{\partial C}{\partial a_i}\frac{\partial a_i}{\partial w_j}$$

Chain rule.

Same w for all i

$$= -\sum_{i=1}^{n} \left(\frac{y_i}{a_i} - \frac{1-y_i}{1-a_i}\right)\left(\frac{e^{-(w^T x_i + b)}}{(1+e^{-(w^T x_i+b)})^2} \cdot x_{ij}\right)$$

$$= -\sum_{i=1}^{n} \frac{y_i - \cancel{a_i y_i} - a_i + \cancel{a_i y_i}}{\cancel{a_i(1-a_i)}}\, \cancel{a_i(1-a_i)}\, x_{ij}$$

$$\frac{\partial C}{\partial w_j} = \sum_{i=1}^{n} (a_i - y_i)\, x_{ij}$$

Logistic Gradient Derivation 2

Definition

$$\nabla_w C = \begin{bmatrix} \frac{\partial C}{\partial w_1} \\ \frac{\partial C}{\partial w_2} \\ \vdots \\ \frac{\partial C}{\partial w_m} \end{bmatrix}$$

$$= \sum_{i=1}^{n} (a_i - y_i) \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{bmatrix}$$

$$= \sum_{i=1}^{n} (a_i - y_i) X_i$$

$$w^T x_i = w_1 x_{i1} + w_2 x_{i2} + \dots$$

$$\frac{\partial w^T x_i}{\partial w_j} = x_{ij}$$

$$a_i (1 - a_i)$$

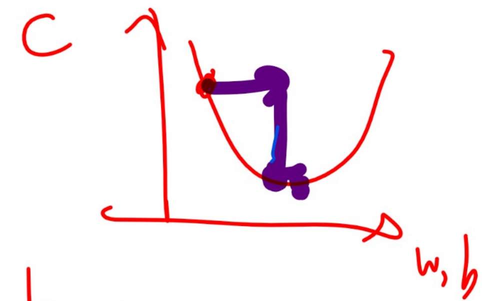
$$\frac{1}{1+e^{-(w^T x + b)}} \cdot \underbrace{\left(1 - \frac{1}{1+e^{-(w^T x + b)}}\right)}_{\frac{e^{-(w^T x + b)}}{1+e^{-(w^T x + b)}}}$$

$$\frac{e^{-(w^T x + b)}}{1+e^{-(w^T x + b)}}$$

Gradient Descent Step

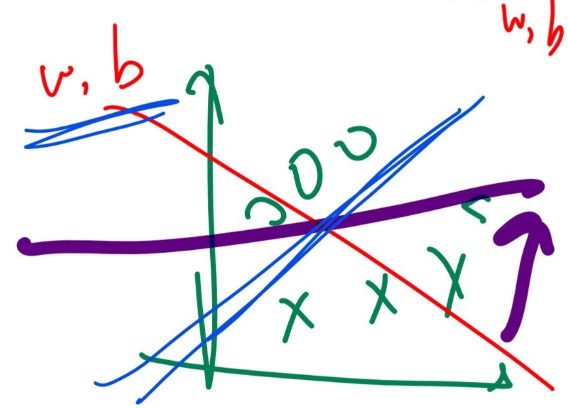
Definition

- For logistic regression, use chain rule twice.



$$w = w - \alpha \sum_{i=1}^{n} \left(a_i - y_i\right) x_i$$

$$b = b - \alpha \sum_{i=1}^{n} \left(a_i - y_i\right)$$



$$a_i = g\left(w^T x_i + b\right), g\left(\boxed{\cdot}\right) = \frac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$$

- α is the learning rate. It is the step size for each step of gradient descent.

Perceptron Algorithm

Definition

- Update weights using the following rule.

$$w = w - \alpha \, (a_i - y_i) \, x_i$$

$$b = b - \alpha \, (a_i - y_i)$$

$$a_i = \mathbb{1}_{\{w^T x_i + b \geqslant 0\}}$$

Other Non-linear Activation Function

Discussion

- Activation function: $g\left(\boxed{\cdot}\right) = \tanh\left(\boxed{\cdot}\right) = \dfrac{e^{\boxed{\cdot}} - e^{-\boxed{\cdot}}}{e^{\boxed{\cdot}} + e^{-\boxed{\cdot}}}$
- Activation function: $g\left(\boxed{\cdot}\right) = \arctan\left(\boxed{\cdot}\right)$
- Activation function (rectified linear unit): $g\left(\boxed{\cdot}\right) = \boxed{\cdot}\,\mathbb{1}_{\left\{\boxed{\cdot} \geqslant 0\right\}}$

  ReLU
- All these functions lead to objective functions that are convex and differentiable (almost everywhere). Gradient descent can be used.

Loss Functions Equivalence

Quiz

- Which one of the following functions is not equivalent to the squared error for binary classification?

which not equivalent to

$$C = \sum_{i=1}^{n} (f(x_i) - y_i)^2, \; f(x_i) \in \{0,1\}, y_i \in \{0,1\}$$

chou4 choose

- ~~A~~: $\sum \mathbb{1}_{\{f(x_i) \neq y_i\}}$
- B: $\sum \mathbb{1}_{\{f(x_i) = y_i\}}$
- ~~C~~: $\sum |f(x_i) - y_i|$
- D: $\sum \max\{0, 1 - f(x_i)\, y_i\}$
- ~~E~~: $\sum \frac{1}{2} \max\{0, 1 - (2 \cdot f(x_i) - 1)(2 \cdot y_i - 1)\}$

D	for y	$f(x_i)$	y_i	$(f - y_i)^2$	$\mathbb{1}_{\{f \neq y_i\}}$
1	1	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
0	1	1	1	0	0

Q2

Loss Functions Equivalence, Answer

Quiz

Gradient Descent

Quiz

- What is the gradient descent step for w if the objective (cost) function is the squared error?

square loss

$$C = \frac{1}{2} \sum_{i=1}^{n} \boxed{\left(a_i - y_i\right)^2}, a_i = g\left(w^T x_i + b\right), g'(z) = g(z) \cdot (1 - g(z))$$

Q3

$$w = w - \alpha \nabla_w C = \begin{pmatrix} \frac{\partial C}{\partial w_1} \\ \vdots \\ \frac{\partial C}{\partial w_m} \end{pmatrix}$$

- A: $w = w - \alpha \sum (a_i - y_i)$
- ~~B~~: $w = w - \alpha \sum (a_i - y_i) x_i$ ← logistic with CE

cross entropy

- C: $w = w - \alpha \sum (a_i - y_i) a_i x_i$
- D: $w = w - \alpha \sum (a_i - y_i) (1 - a_i) x_i$
- E: $w = w - \alpha \sum \boxed{(a_i - y_i)} \boxed{a_i (1 - a_i) x_i}$

Gradient Descent, Answer

Quiz

Gradient Descent, Another One

Quiz

- What is the gradient descent step for w if the activation function is the identity function?

$$C = \frac{1}{2} \sum_{i=1}^{n} \left(a_i - y_i\right)^2, a_i = w^T x_i + b$$

- A: $w = w - \alpha \sum \left(a_i - y_i\right)$
- B: $w = w - \alpha \sum \left(a_i - y_i\right) x_i$
- C: $w = w - \alpha \sum \left(a_i - y_i\right) a_i x_i$
- D: $w = w - \alpha \sum \left(a_i - y_i\right) \left(1 - a_i\right) x_i$
- E: $w = w - \alpha \sum \left(a_i - y_i\right) a_i \left(1 - a_i\right) x_i$

Gradient Descent, Another One, Answer

Quiz

Remind Me to Stop Recording

Admin

- If you accidentally selected an obviously incorrect answer earlier, you can enter the question name and the correct answer here.