CS540 Introduction to Artificial Intelligence
Lecture 2

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

May 10, 2020

Feedback

Admin

- Please give me feedback on lectures, homework, exams on Socrative, room CS540.
- Please report bugs in homework, lecture examples and quizzes on Piazza.
- Please do NOT leave comments on YouTube.
- Email me (Young Wu) for personal issues.
- Email department chair (Dieter van Melkebeek) for issues with me.

Feedback from Last Year

Admin

- Too much math.
- Time spent on math.
- Cannot understand my handwriting.
- Mistake on slides.
- More examples.

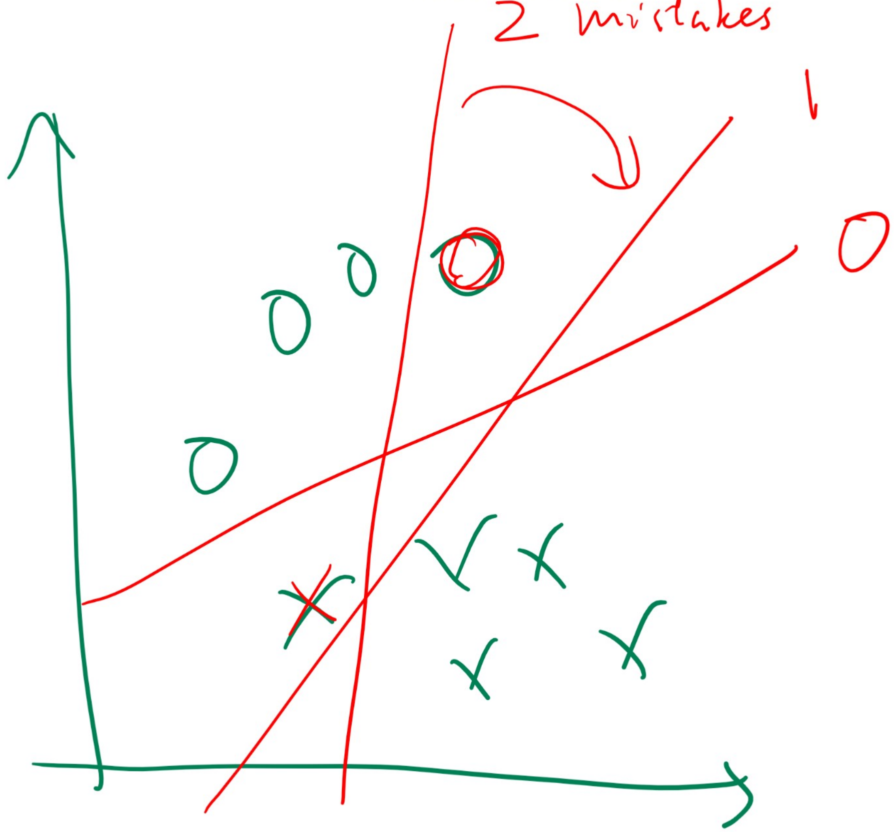
Supervised Learning

Motivation

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

Loss Function Diagram

Motivation



$$\boxed{\min_f \ \#\text{mistakes}}$$

Zero-One Loss Function

Motivation

min → # mistake

argmin ⇒ f that makes min # mistakes

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

training examples

1 mistake

or

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

$\hat{y}$

mistakes

1 $\hat{y} \neq y$

0 $\hat{y} = y$

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

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

5.5 → 25

0.6 → 0.5 → 0.01

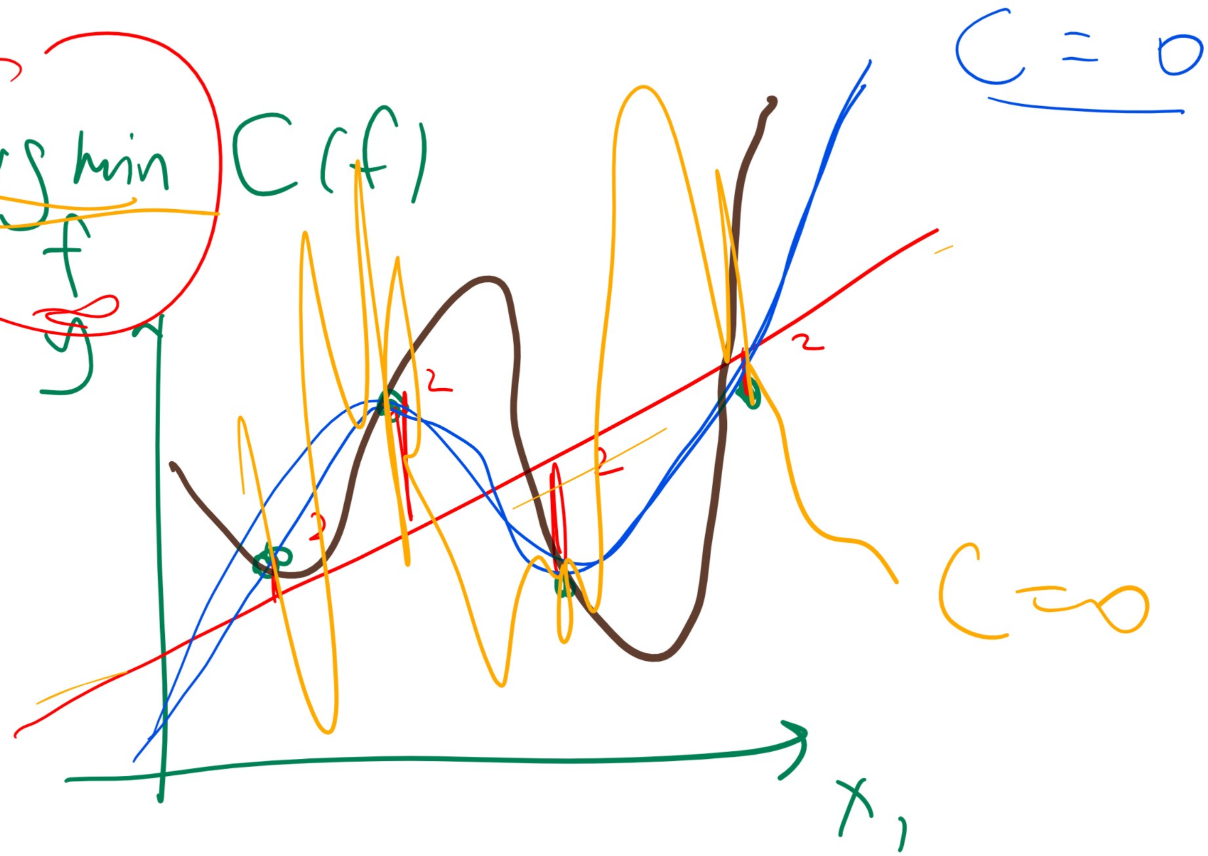
$\mathbb{1}\{\hat{y}_i \neq y_i\}$

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

Loss Functions Equivalence

Quiz

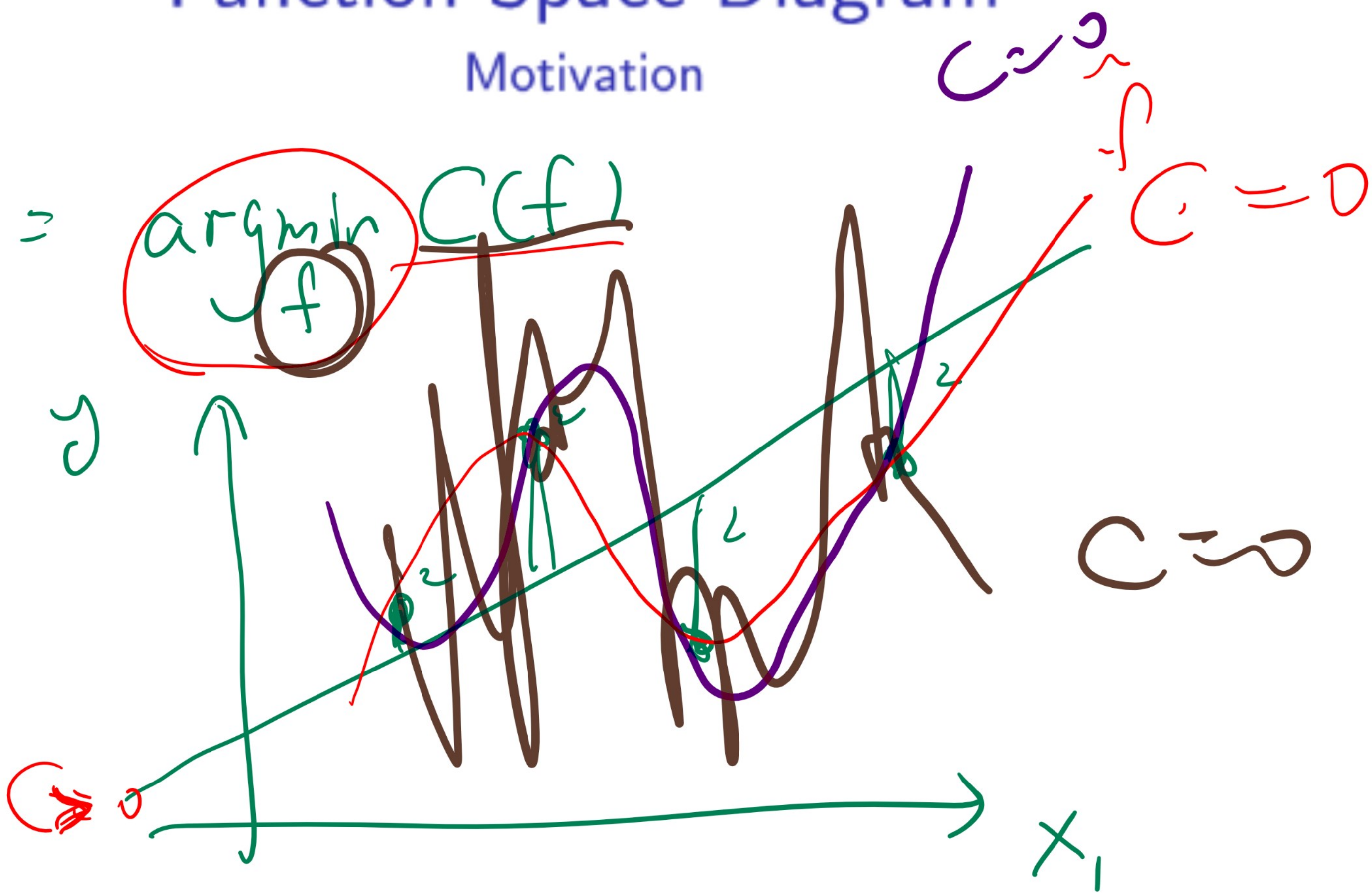
$$\hat{f} = \arg\min_{f} C(f)$$



Function Space Diagram

Motivation

$$\hat{f} = \underset{f}{\operatorname{argmin}}\ C(f)$$



Hypothesis Space

Motivation

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

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

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

Linear Regression

Motivation

- For example, $\mathcal{H}$ can be the set of linear functions. Then the problem can be rewritten in terms of the weights.

β

$$\left(\hat{w}_1, \ldots, \hat{w}_m, \hat{b}\right) = \arg \min_{w_1, \ldots, w_m, b} \frac{1}{2} \sum_{i=1}^{n} (a_i - y_i)^2$$

$\hat{f}$

where $a_i = w_1 x_{i1} + w_2 x_{i2} + \ldots + w_m x_{im} + b$

- The problem is called (least squares) linear regression.

$y = 0$
1

$0 \to$ cat
$1 \to$ dog
$2 \to$ fox

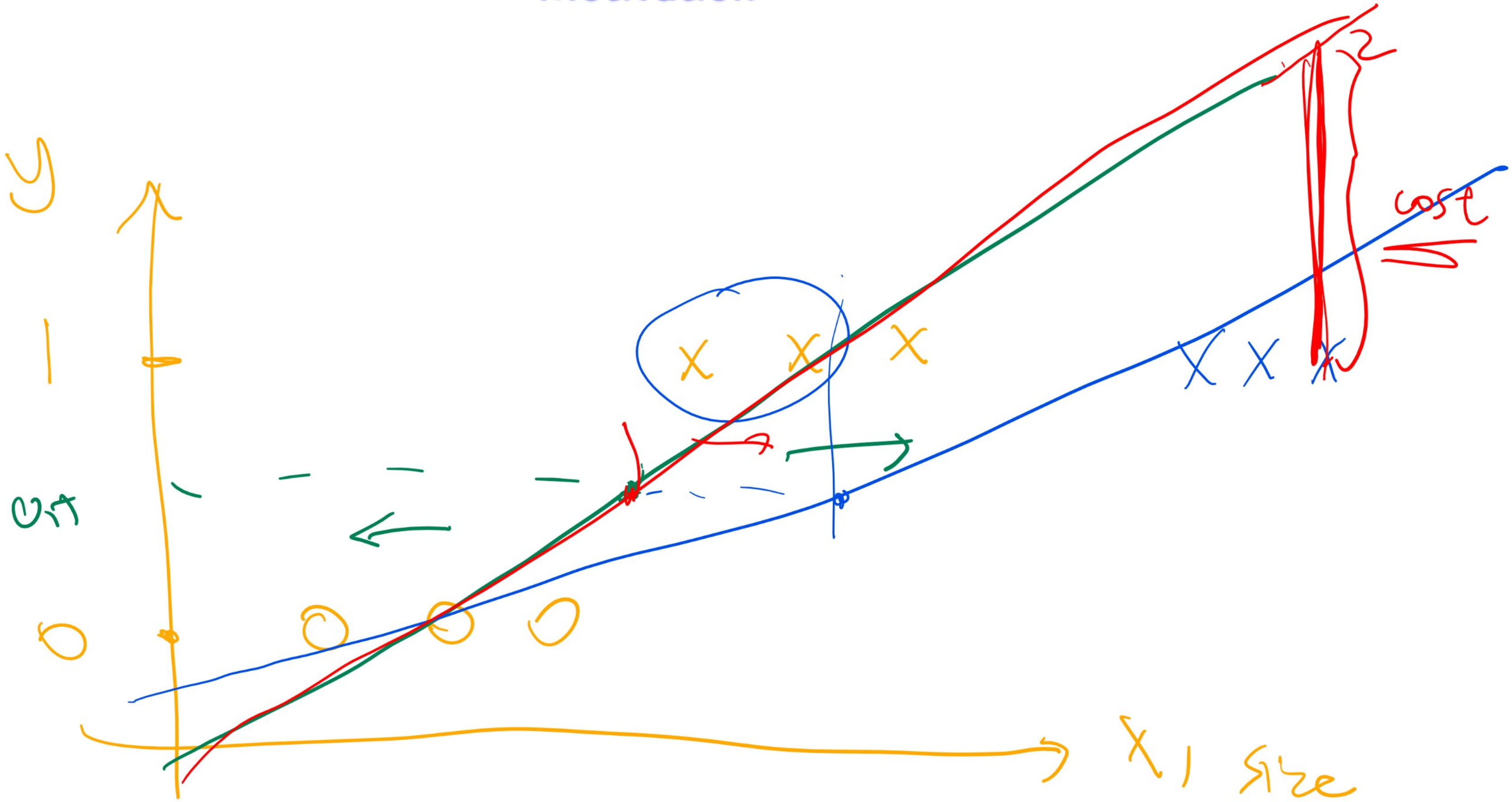
Binary Classification

Motivation

- If the problem is binary classification, y is either 0 or 1, and linear regression is not a great choice.
- This is because if the prediction is either too large or too small, the prediction is correct, but the cost is large.

Binary Classification Linear Regression Diagram

Motivation



Activation Function

Motivation

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

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

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

- g is called the activation function.

Linear Threshold Unit

Motivation

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

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

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

Sigmoid Activation Function

Motivation

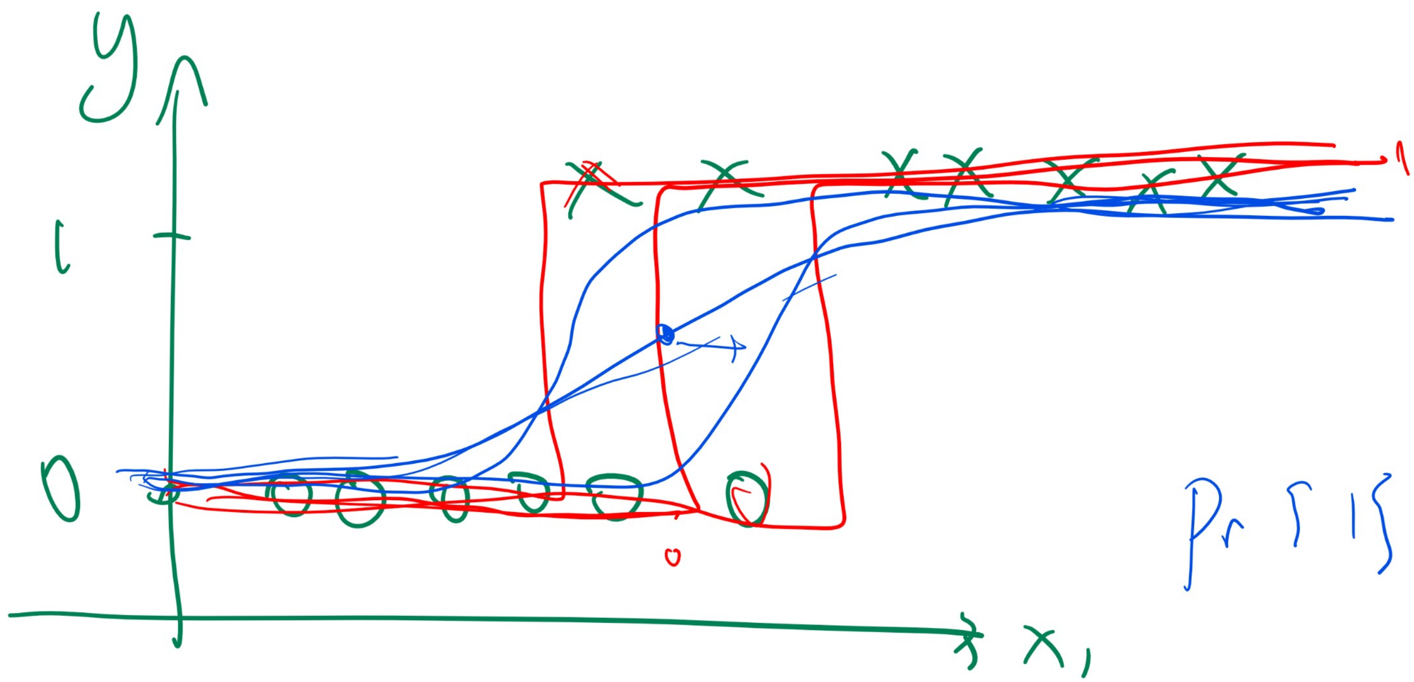
- When the activation function g is the sigmoid function, the problem is called logistic regression.

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

- This g is also called the logistic function.

Sigmoid Function Diagram

Motivation



Cross Entropy Loss Function

Motivation

y_i	$\hat{y}_i$	C
0	0	0
1	1	0
0	1	$+\infty$
1	0	

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

$y_i = 0$ $\quad \hat{y}_i = 0$ $\quad C = 0$

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

$1 \quad P \quad 1 \cdot \quad 0$

$0 \qquad 1 \cdot \log(0) = -\infty$

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

Logistic Regression Objective

Motivation

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

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

where $a_i = \dfrac{1}{1 + \exp\left(-z_i\right)}$ and $z_i = w^T x_i + b$

Optimization Diagram

Motivation

slope

large neg incre w ↑ by a lot

small neg incre w ↑ by a little

pos decr ↓ w

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.

Gradient Descent Intuition

Definition

Demo

- If a small increase in w_1 causes the distances from the points to the regression line to decrease: increase w_1.
- If a small increase in w_1 causes the distances from the points to the regression line to increase: decrease w_1.
- The change in distance due to change in w_1 is the derivative.
- The change in distance due to change in $\begin{bmatrix} w \\ b \end{bmatrix}$ is the gradient.

Gradient

Definition

- The gradient is the vector of derivatives.
- The gradient of $f(x_i) = w^T x_i + b = w_1 x_{i1} + w_2 x_{i1} + \ldots + w_m x_{im} + b$ is:

$$\nabla_w f = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \ldots \\ \frac{\partial f}{\partial w_m} \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \ldots \\ x_{im} \end{bmatrix} = x_i$$

$$\nabla_b f = 1$$

Chain Rule

Definition

- The gradient of $f(x_i) = g(w^T x_i + b) = g(w_1 x_{i1} + w_2 x_{i1} + \ldots + w_m x_{im} + b)$ can be found using the $\boxed{\text{chain rule.}}$

$\frac{\partial}{\partial w} w^T x + b$

$$\nabla_w f = g'(w^T x_i + b) x_i$$
$$\nabla_b f = g'(w^T x_i + b) \; 1$$

- In particular, for the logistic function g :

$$g(\boxdot) = \frac{1}{1 + \exp(-\boxdot)}$$
$$g'(\boxdot) = g(\boxdot)(1 - g(\boxdot))$$

Logistic Gradient Derivation 1

$-z \quad - \quad -1$

Definition

$$g(z) = \frac{1}{1+\exp(-z)} \qquad g'(z) = \frac{\cancel{-1}\, \boxed{\exp(-z)} \cdot \cancel{(-1)}}{(1+\exp(-z))^2}$$

$$\frac{1}{x} \rightarrow -\frac{1}{x^2}$$

$$g'(z) = \frac{1}{1+\exp(-z)} \cdot \frac{\exp(-z)}{1+\exp(-z)}$$

$$= \frac{1}{1+\exp(-z)}\left(1 - \frac{1}{1+\exp(-z)}\right)$$

$$\nabla_w f^{=a_i} = \underbrace{\boxed{g(z)}}_{a_i} \cdot \Big(1 - \underbrace{\boxed{g(z)}}_{a_i}\Big) \cdot X_i$$

Logistic Gradient Derivation 2

Definition

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

$$\nabla_w a_i = a_i(1-a_i)\cdot x_i$$

one term

$$\frac{\partial C}{\partial a_i} = y_i \frac{1}{a_i} - (1-y_i)\cdot\frac{1}{1-a_i}$$

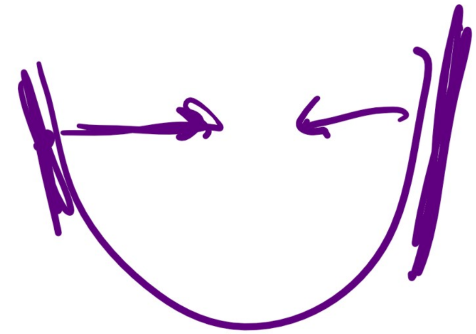
$$= \frac{y_i(1-a_i) - (1-y_i)a_i}{a_i(1-a_i)}$$

$$= \frac{y_i - a_i}{\cancel{a_i(1-a_i)}} \cdot \cancel{a_i(1-a_i)}\, x_i$$

$$\nabla_w C = \sum_{i=1}^{n} (a_i - y_i)\, x_i$$

Gradient Descent Step

Definition



- For logistic regression, use chain rule twice.

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

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

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

- 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.

$$w = w - \alpha (a_i - y_i) x_i$$
$$b = b - \alpha (a_i - y_i)$$
$$a_i = \mathbb{1}_{\{w^T x_i + b \geqslant 0\}}$$

0, 1

Learning Rate Diagram

Definition

Gradient Descent

Quiz

Gradient Descent, Another One

Quiz

Logistic Regression, Part 1

Algorithm

- Inputs: instances: $\{x_i\}_{i=1}^{n}$ and $\{y_i\}_{i=1}^{n}$
- Outputs: weights and biases: $w_1, w_2, ..., w_m$ and b
- Initialize the weights.

$$w_1, ..., w_m, b \sim \text{Unif}\ [\overset{-1}{\cancel{0}}, 1]$$

- Evaluate the activation function.

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

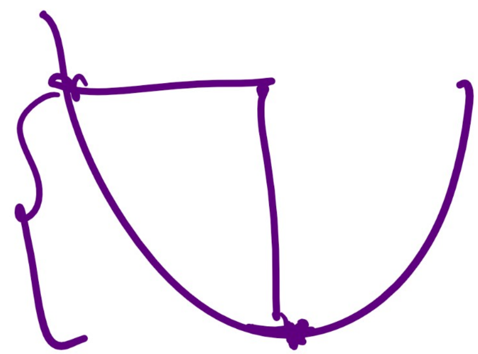
Logistic Regression, Part 2

Algorithm

- Update the weights and bias using gradient descent.

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

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



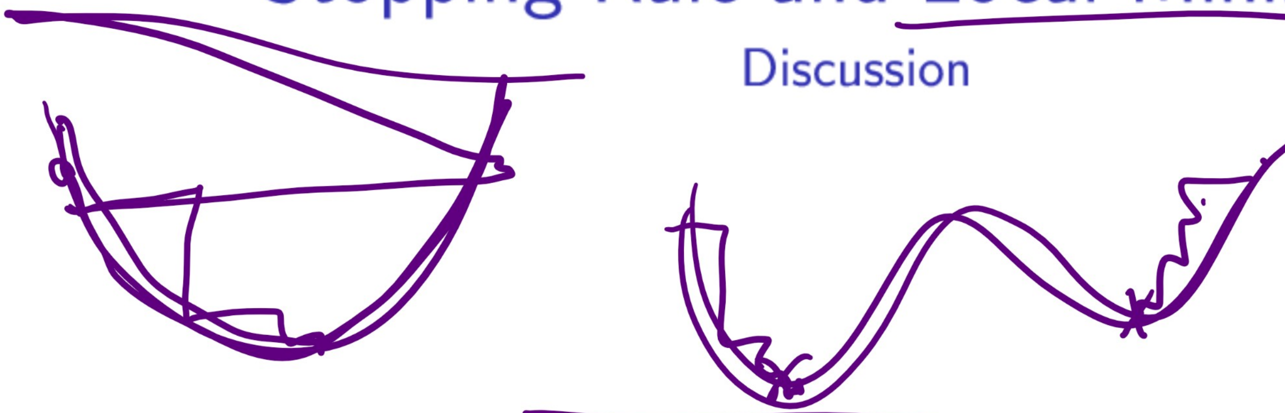
- Repeat the process until convergent.

$$|C - C^{\text{prev}}| < \varepsilon$$

0.0001

Stopping Rule and Local Minimum

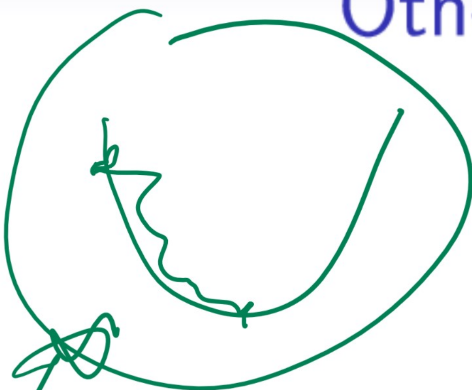
Discussion

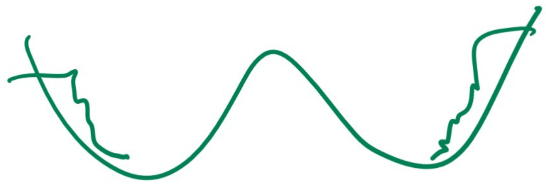


- Start with multiple random weights.
- Use smaller or decreasing learning rates. One popular choice is $\frac{\alpha}{\sqrt{t}}$, where t is the iteration count.
- Use the solution with the lowest C.

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): $g\left(\boxed{\cdot}\right) = \boxed{\cdot}\,\mathbb{1}_{\left\{\boxed{\cdot} \geq 0\right\}}$

ReLU

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

Convexity Diagram

Discussion

Convexity

Discussion

- If a function is convex, gradient descent with any initialization will converge to the global minimum.
- If a function is not convex, gradient descent with different initializations may converge to different local minima.
- A twice differentiable function is convex if and only its second derivative is non-negative.
- In the multivariate case, it means the Hessian matrix is positive semidefinite.

$f''(w) \geq 0$

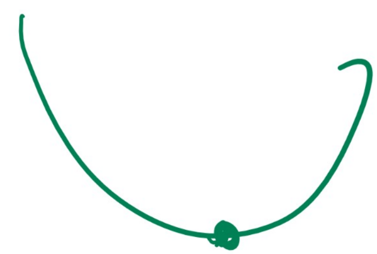
Positive Semidefinite

Discussion

$$f(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

- Hessian matrix is the matrix of second derivatives:

$$H : H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$



- A matrix H is positive semidefinite if $x^T H x \geq 0 \; \forall \; x \in \mathbb{R}^n$.
- A symmetric matrix is positive semidefinite if and only if all of its eigenvalues are non-negative.

$$\underbrace{H}_{\text{matrix}} \, \underbrace{x}_{\text{eigen vector}} = \underbrace{\lambda}_{\text{scalar}} \, x$$

eigen value

$$x^T H x$$

$$x^T \lambda x = \underbrace{\lambda}_{\geq 0} \, \underbrace{x^T x}_{\geq 0}$$

Convex Function Example 1

Discussion

$$f(x, y) = x^2 + xy + y^2$$

$$\text{Hess} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} 2x + y \\ 2y + x \end{pmatrix}$$

$$\lambda_1$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Convex Function Example 2

Discussion

λ_2

$$\begin{bmatrix}2&1\\1&2\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix} = \begin{bmatrix}1\\-1\end{bmatrix} = 1\begin{bmatrix}1\\-1\end{bmatrix}$$

$$\begin{bmatrix}2&1\\1&2\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} = \begin{bmatrix}-1\\1\end{bmatrix} = 1\begin{bmatrix}-1\\1\end{bmatrix}$$

$\lambda_1, \lambda_2 = 1, 3 > 0$

pos. semi. det

Convex Functions

Quiz

Definiteness

Quiz