ECE595 / STAT598: Machine Learning I

Lecture 16.1: Perceptron 1 - From Logistic to Perceptron

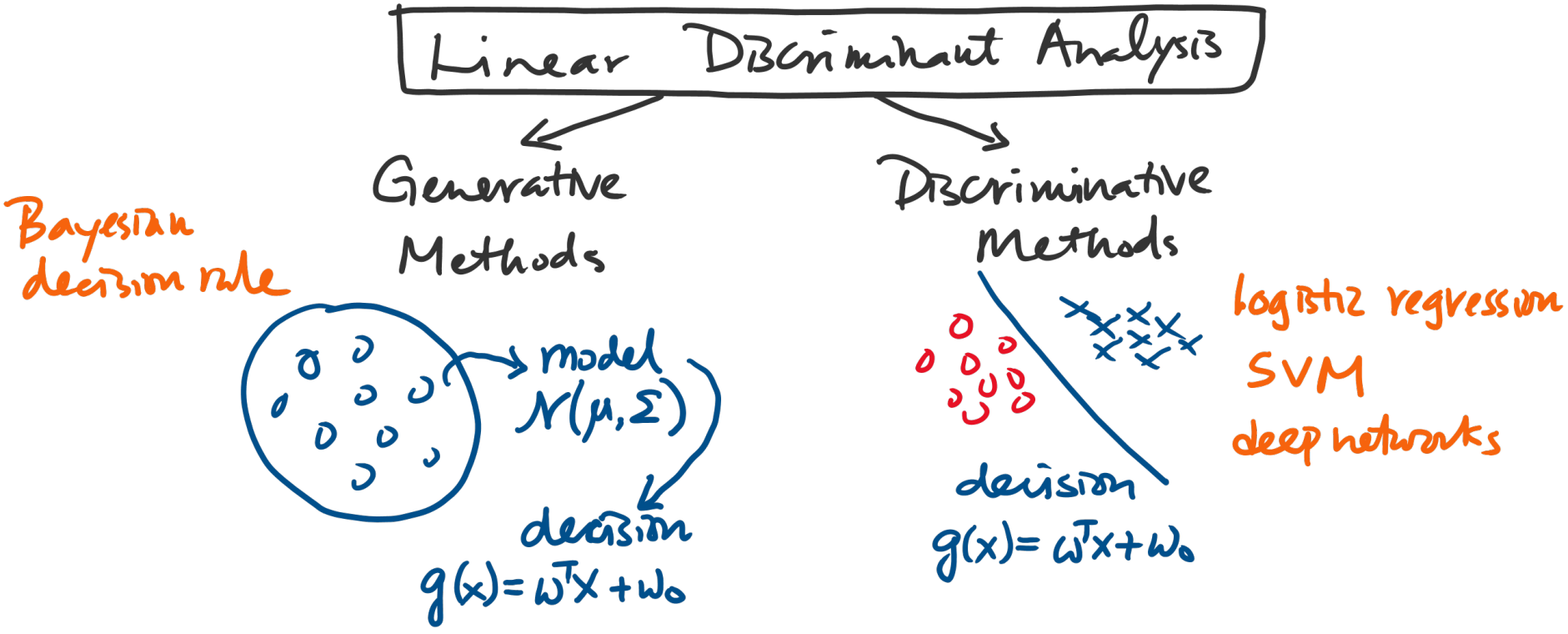
Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

PURDUE
UNIVERSITY

Overview



- In linear discriminant analysis (LDA), there are generally two types of approaches
- **Generative approach**: Estimate model, then define the classifier
- **Discriminative approach**: Directly define the classifier

Outline

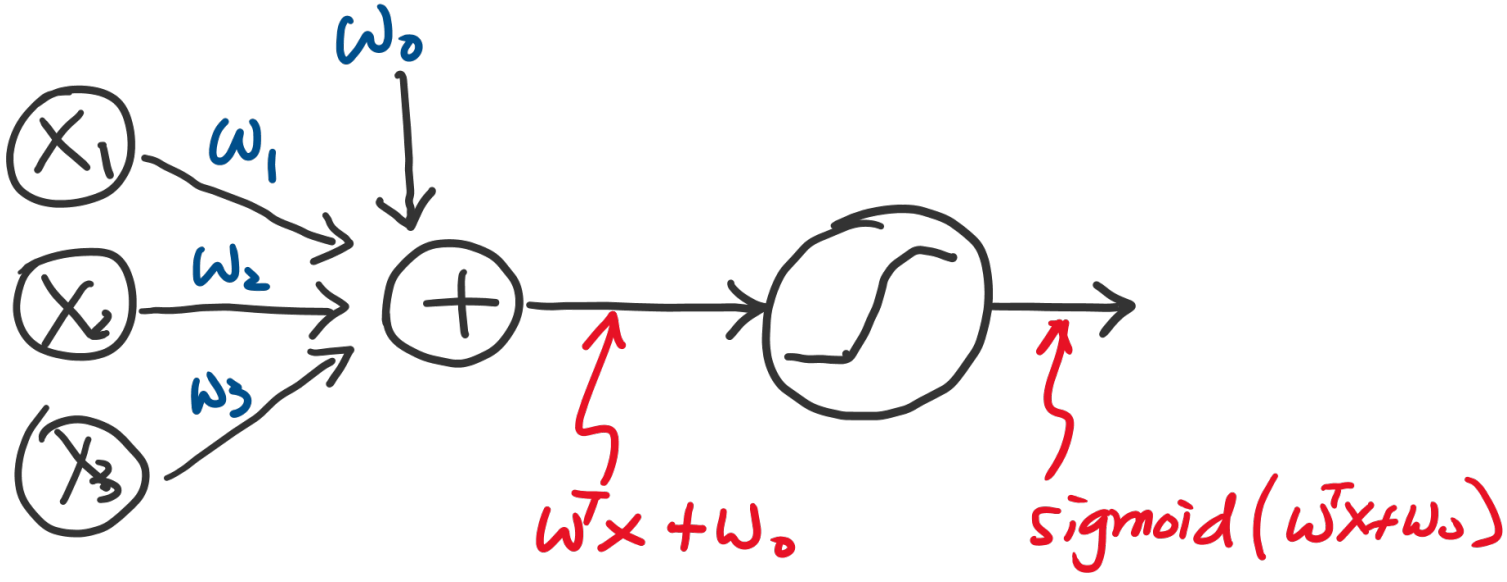
Discriminative Approaches

- Lecture 16 Perceptron 1: Definition and Basic Concepts
- Lecture 17 Perceptron 2: Algorithm and Property

This lecture: Perceptron 1

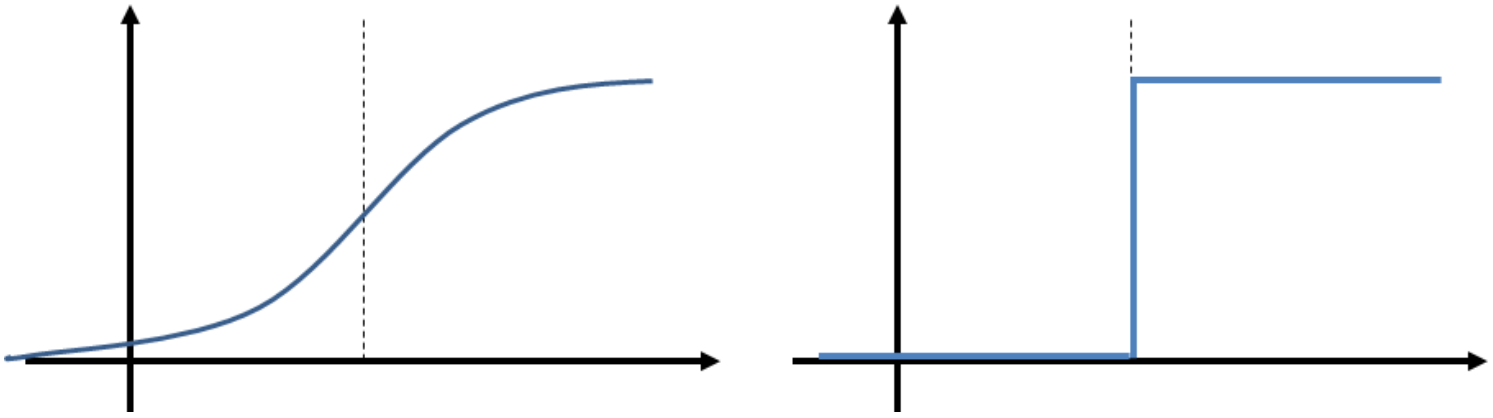
- From Logistic to Perceptron
  - What is Perceptron? Why study it?
  - Perceptron Loss
  - Connection with other losses
- Properties of Perceptron Loss
  - Convexity
  - Comparing with Bayesian Oracle
  - Preview of Perceptron Algorithm

Perceptron as a Single-Layer Network



- Logistic regression: Soft threshold
- Perceptron: Hard threshold

From Logistic to Perceptron



- Logistic regression

$$h(x) = \frac{1}{1 + e^{-a(x-x_0)}}.$$

- Make $a \to \infty$, then $h(x) \to$ step function

$$\begin{aligned}\lim_{a\to\infty} h(x) &= \lim_{a\to\infty} \frac{1}{1 + e^{-a(x-x_0)}} \\ &= \text{sign}(a(x - x_0)).\end{aligned}$$

From Logistic to Perceptron

- Linear regression

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + w_0).$$

  - Stage 1: Training the discriminant function

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0.$$

  - Stage 2: Threshold to make decision

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{sign}(g_{\boldsymbol{\theta}}(\boldsymbol{x})).$$

- Logistic regression

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^T\boldsymbol{x} + w_0)}}.$$

- Perceptron algorithm

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + w_0).$$

How to Define Perceptron Loss Function

- Logistic regression

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} -\Big\{ y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1-y_n)\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) \Big\}$$

  - Okay if $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$ is soft-decision.
  - Not okay if $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$ is binary: Either all fit or none fit.

- "Candidate" perceptron loss function

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \max\Big\{ -y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0 \Big\}.$$

  - Does not have the log-term
  - Will not run into $\pm\infty$

Understanding the Perceptron Loss function

- "Candidate" perceptron loss function

$$J_{\text{hard}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \max\Big\{ -y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0 \Big\}.$$

- $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = \text{sign}(\boldsymbol{w}^T \boldsymbol{x}_n + w_0)$ is either $+1$ or -1.
- If the decision is correct, then must have
  - $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = +1$ and $y_n = +1$
  - $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = -1$ and $y_n = -1$
  - In both cases, $y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = +1$
  - So the loss is $\max\{-y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0\} = 0$.
- If the decision is wrong, then must have
  - $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = +1$ and $y_n = -1$
  - $h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = -1$ and $y_n = +1$
  - In both cases, $y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = -1$
  - So the loss is $\max\{-y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0\} = 1$.
- $J(\boldsymbol{\theta})$ is not differentiable in $\boldsymbol{\theta}$.

Perceptron Loss function

- Define the **perceptron loss** as

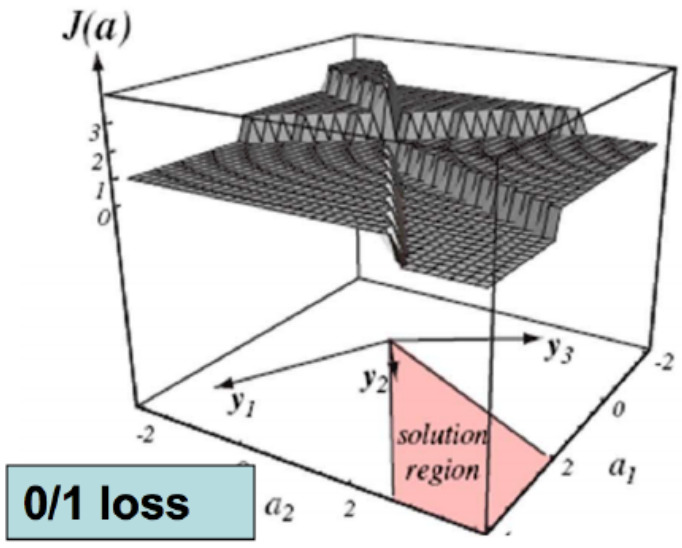
$$J_{\text{soft}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \max\Big\{ -y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0 \Big\}.$$

- $g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = \boldsymbol{w}^T \boldsymbol{x}_n + w_0$ is either +ve or -ve.
- If the decision is correct, then must have
  - $g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) > 0$ and $y_n = +1$
  - $g_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0$ and $y_n = -1$
  - In both cases, $y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) > 0$
  - So the loss is $\max\{-y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0\} = 0$.
- If the decision is wrong, then must have
  - $g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) > 0$ and $y_n = -1$
  - $g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) < 0$ and $y_n = +1$
  - In both cases, $y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) < 0$
  - So the loss is $\max\{-y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0\} > 0$.

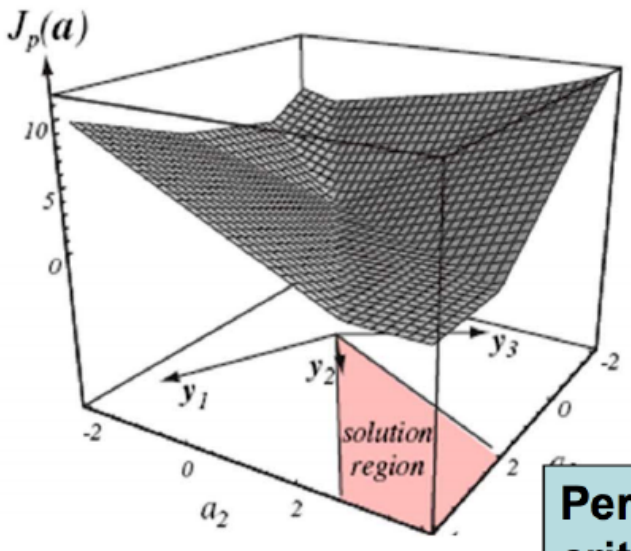
Comparing the loss function

- **Linear regression**
  - $J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (g_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - y_n)^2$
  - Convex, closed-form solution
  - Usually: Unique global minimizer
- **Logistic regression**
  - $J(\boldsymbol{\theta}) = \sum_{n=1}^{N} -\Big\{ y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1 - y_n) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) \Big\}$
  - Convex, no closed-form solution
  - Usually: Unique global minimizer
- **Perceptron (Hard)**
  - $J_{\text{hard}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \max \Big\{ - y_n h_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0 \Big\}$
  - Not convex, no closed-form solution
  - Usually: Many global minimizers
- **Perceptron (Soft)**
  - $J_{\text{soft}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \max \Big\{ - y_n g_{\boldsymbol{\theta}}(\boldsymbol{x}_n), 0 \Big\}$
  - Convex, no closed-form solution
  - Usually: Unique global minimizer

Comparing the loss function



0/1 loss

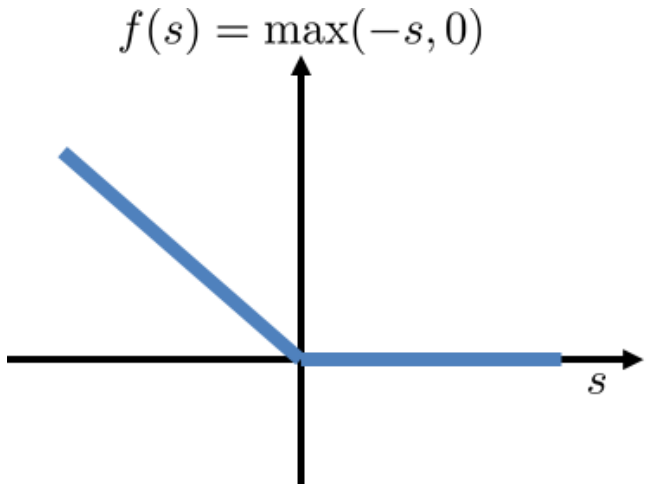


Perceptron criterion

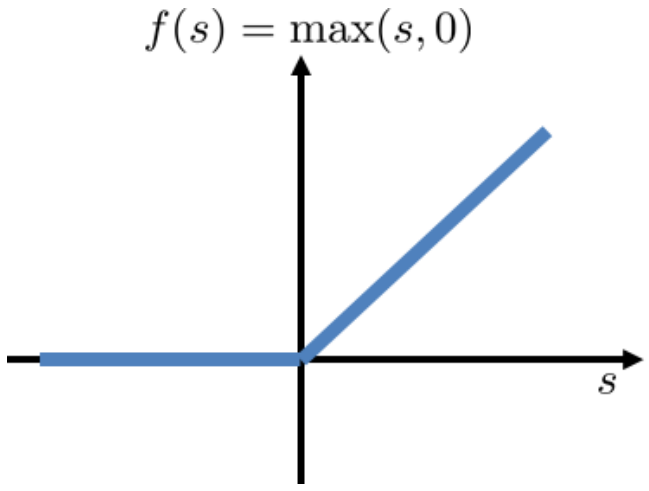
Nice gradient

https://www.cc.gatech.edu/~bboots3/CS4641-Fall2016/Lectures/Lecture5.pdf

Perceptron Loss, Hinge Loss and ReLU



Perceptron Loss

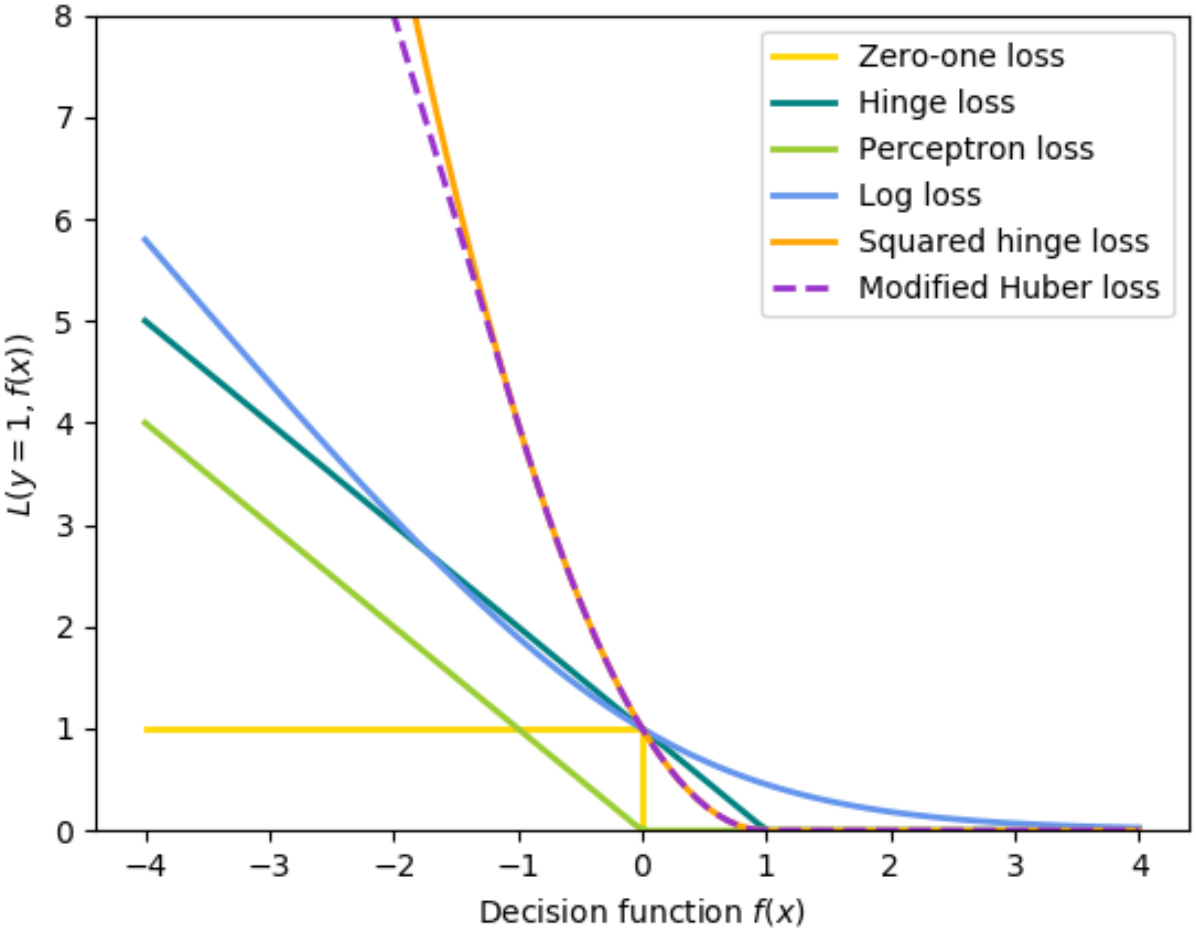


Rectified Linear Unit

- The function $f(s) = \max(-s, 0)$ is called the perceptron loss
- A variant $\max(1 - s, 0)$ is called Hinge Loss
- Another variant $\max(s, 0)$ is called ReLU
- We can prove that the gradient of $f(\boldsymbol{s}) = \max(-\boldsymbol{x}^T\boldsymbol{s}, 0)$ is

$$\nabla_{\boldsymbol{s}} \max(-\boldsymbol{x}^T\boldsymbol{s}, 0) = \begin{cases} -\boldsymbol{x}, & \text{if} \quad \boldsymbol{x}^T\boldsymbol{s} < 0, \\ 0, & \text{if} \quad \boldsymbol{x}^T\boldsymbol{s} \geq 0. \end{cases}$$

Comparing Loss functions



https://scikit-learn.org/dev/auto_examples/linear_model/plot_sgd_loss_functions.html