NYU

Introduction to Machine Learning
[Fall 2022]

Perceptron (Part 2)

October 25, 2022

Lerrel Pinto

Logistics

- Great to see many students doing well on the HWs and the blog posts!
- We are introducing 'extra credit' questions on the next HWs to help if you have a low-score.

Logistics

- Project (30%) will be announced today!
  - Needs to be done individually – no collaboration allowed.
  - Only discussion of the project is on chatroom project-discussion.
  - We have setup a Kaggle system for submitting solutions.
  - You will be evaluated on your final score (taken from Kaggle leaderboard) and creativity (taken from project report).

Topics for today

- Bringing gradients to Perceptron.

Recap: A general-purpose recipe for ML

- Step 1: Collect a dataset $D \equiv \{x^i, y^i\}_{i=1}^{N}$
- Step 2: Choose a decision function $\hat{y} = f_\theta(x)$
- Step 3: Construct a loss function $l(\hat{y}^i, y^i)$
- Step 4: Define goal:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} l(f_\theta(x^i), y^i)$$

- Step 5: Train with SGD (or variants of GD).

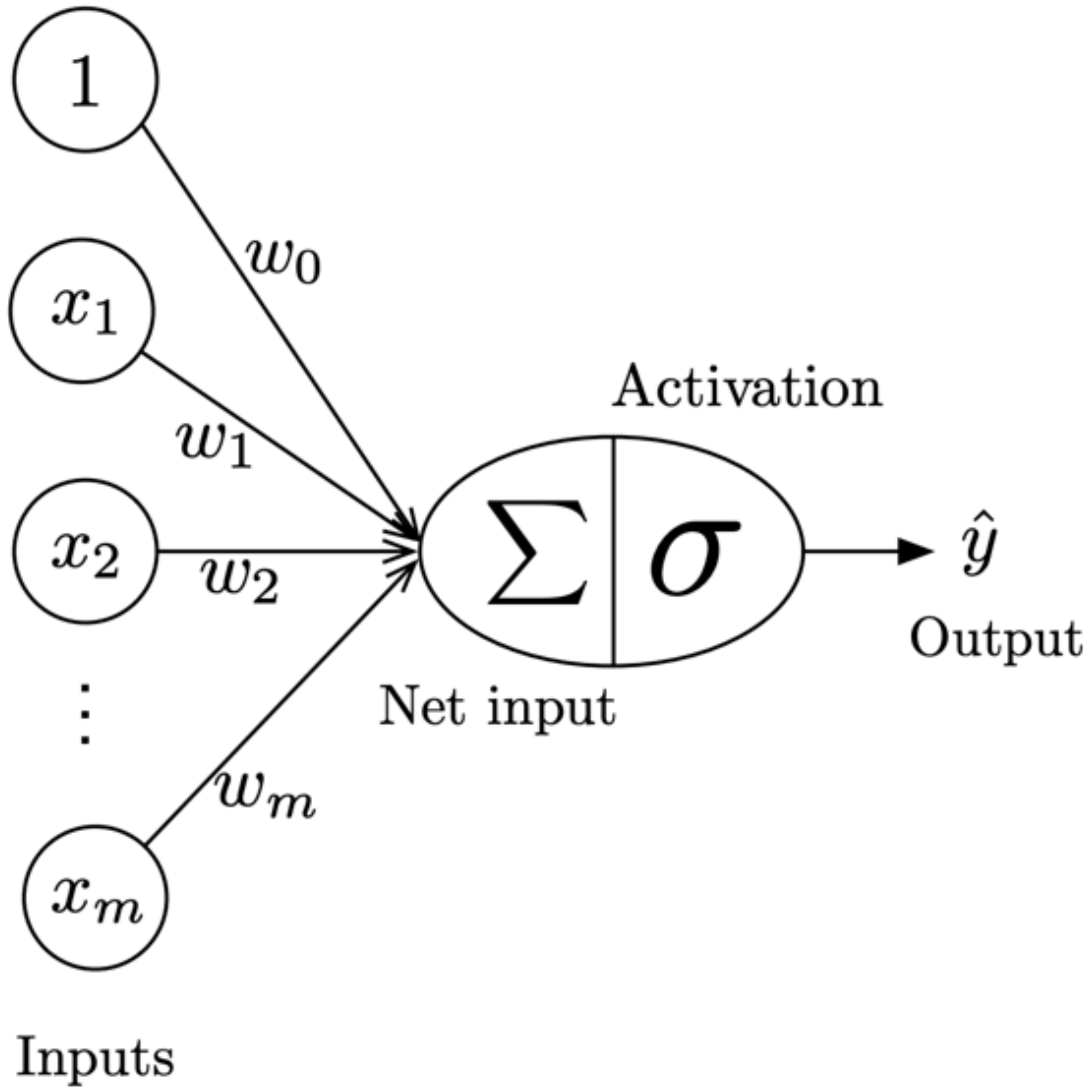
Recap: Rosenblatt's Perceptron

A learning rule for the computational/mathematical neuron model

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton. Project Para*. Cornell Aeronautical Laboratory.

Credits: Sebastian Raschka

Recap: Perceptron



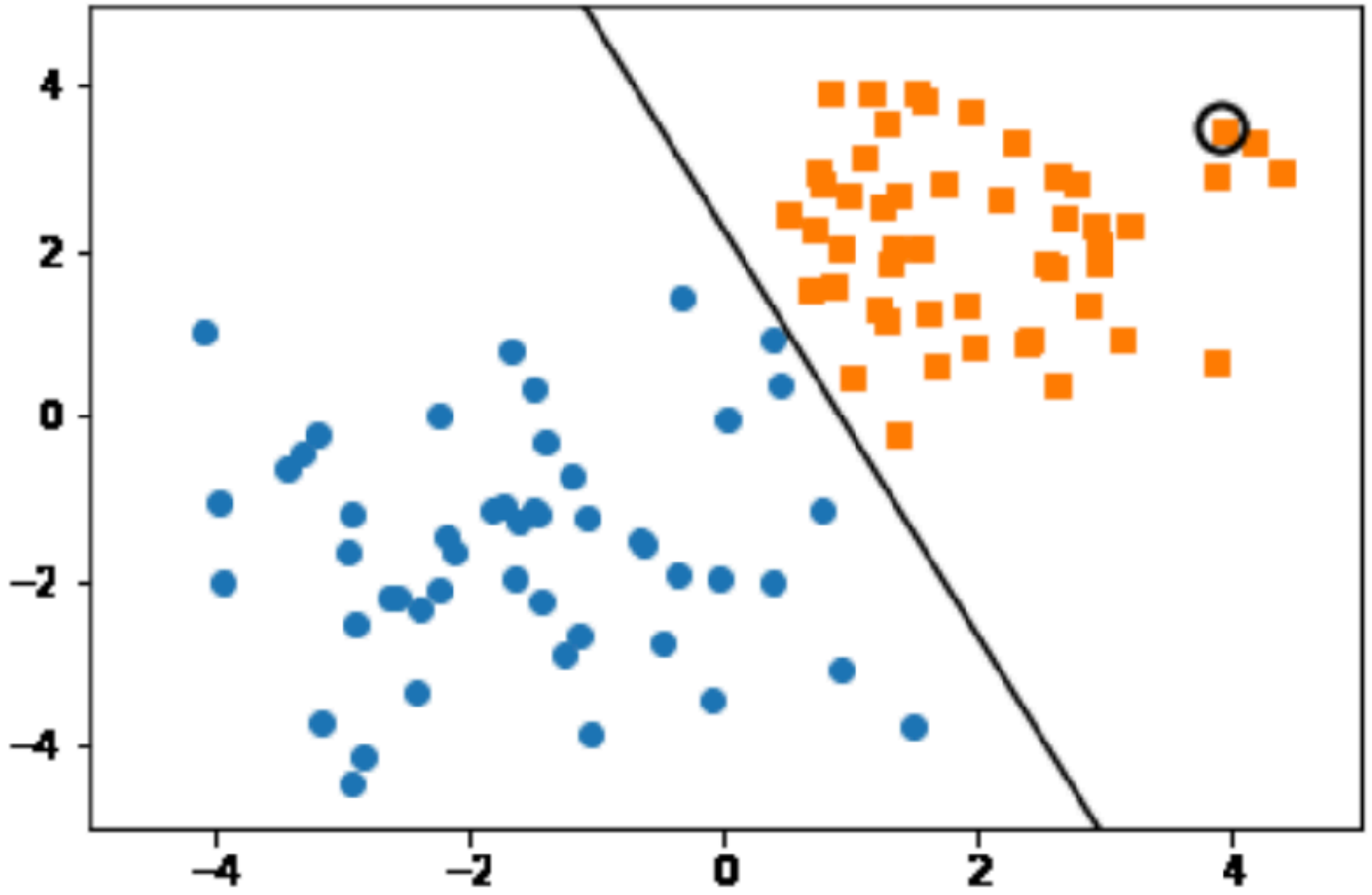
$$\sigma\left(\sum_{i=0}^{m} x_i w_i\right) = \sigma\left(\mathbf{x}^T\mathbf{w}\right) = \hat{y}$$

$$\sigma(z) = \begin{cases} 0, & z \le 0 \\ 1, & z > 0 \end{cases}$$

$$w_0 = -\theta$$

Credits: Sebastian Raschka

Recap: Perceptron



Credits: Sebastian Raschka

Recap: Perceptron

- Let: $D \equiv \{x^i, y^i\}_{i=1}^{N}$
- Initialize $\overrightarrow{w}^0 = 0^d$
- For every training 'epoch':
  - For every $(x^i, y^i) \in D$:
    - $\hat{y}^i = \sigma(\overrightarrow{w}^T x^i)$
    - $e = (y^i - \hat{y}^i)$
    - $\overrightarrow{w}^{t+1} \leftarrow \overrightarrow{w}^t + e \times x^i$

Credits: Sebastian Raschka

Recap: Perceptron

- Let: $D \equiv \{x^i, y^i\}_{i=1}^N$
- Initialize $\vec{w}^0 = 0^d$
- For every training 'epoch':
  - For every $(x^i, y^i) \in D$:
    - $\hat{y}^i = \sigma(\vec{w}^T x^i)$
    - $e = (y^i - \hat{y}^i)$
    - $\vec{w}^{t+1} \leftarrow \vec{w}^t + e \times x^i$

Principle:

- If there is no error, do not update.
- If output is 0 and target is 1, add input to weight vector.
- If output is 1 and target is 0, subtract input from weight vector.

Credits: Sebastian Raschka

Recap: Perceptron

- Let: $D \equiv \{x^i, y^i\}_{i=1}^N$
- Initialize $\overrightarrow{w}^0 = 0^d$
- For every training 'epoch':
  - For every $(x^i, y^i) \in D$:
    - $\hat{y}^i = \sigma(\overrightarrow{w}^T x^i)$
    - $e = (y^i - \hat{y}^i)$
    - $\overrightarrow{w}^{t+1} \leftarrow \overrightarrow{w}^t + e \times x^i$

Credits: Sebastian Raschka

Principle:

- If there is no error, do not update.
- If output is 0 and target is 1, add input to weight vector.
- If output is 1 and target is 0, subtract input from weight vector.

Guaranteed to converge if solution exists!

Recap: Perceptron

- No non-linear boundaries possible with classical perceptron
- Does not converge when classes are non-separable
- In its current form not compatible with gradient descent

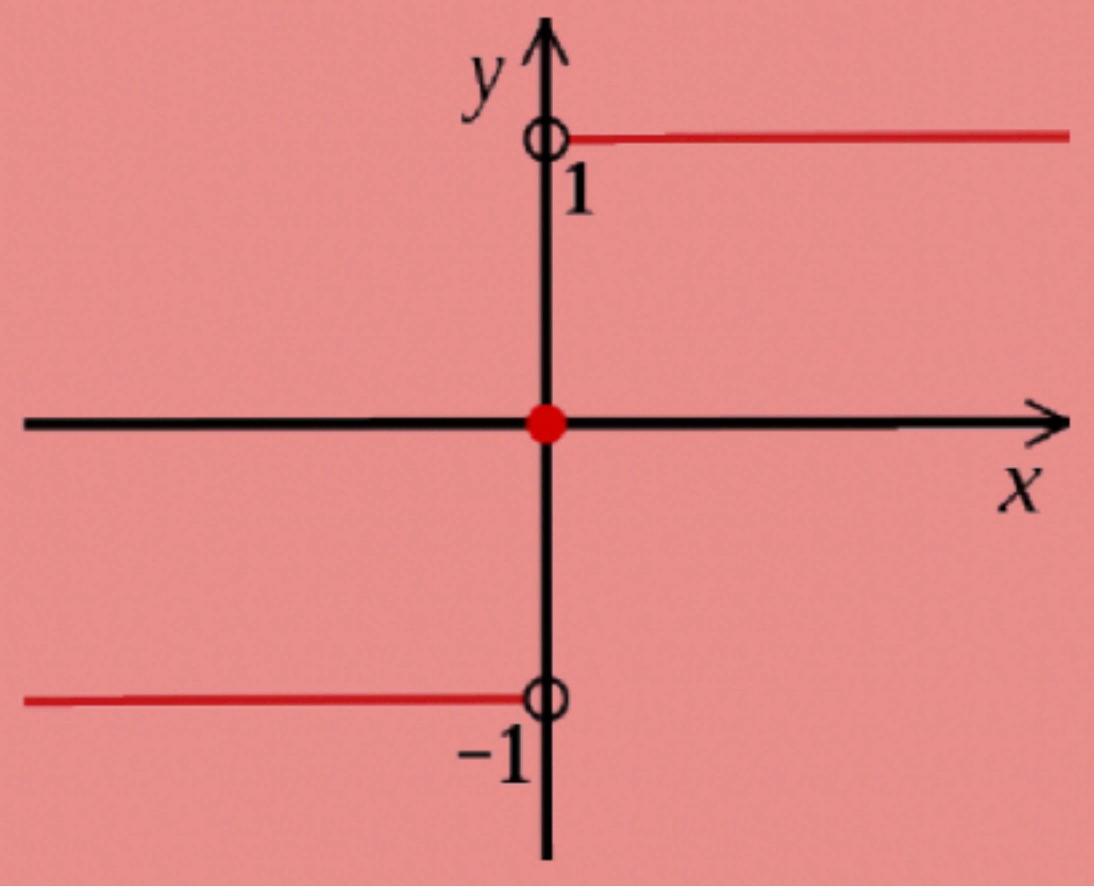
How to make perceptron differentiable?

Credits: Matt Gormley

How to make perceptron differentiable?

This decision function isn't differentiable:

$$y = sign(\overrightarrow{w}^T x)$$

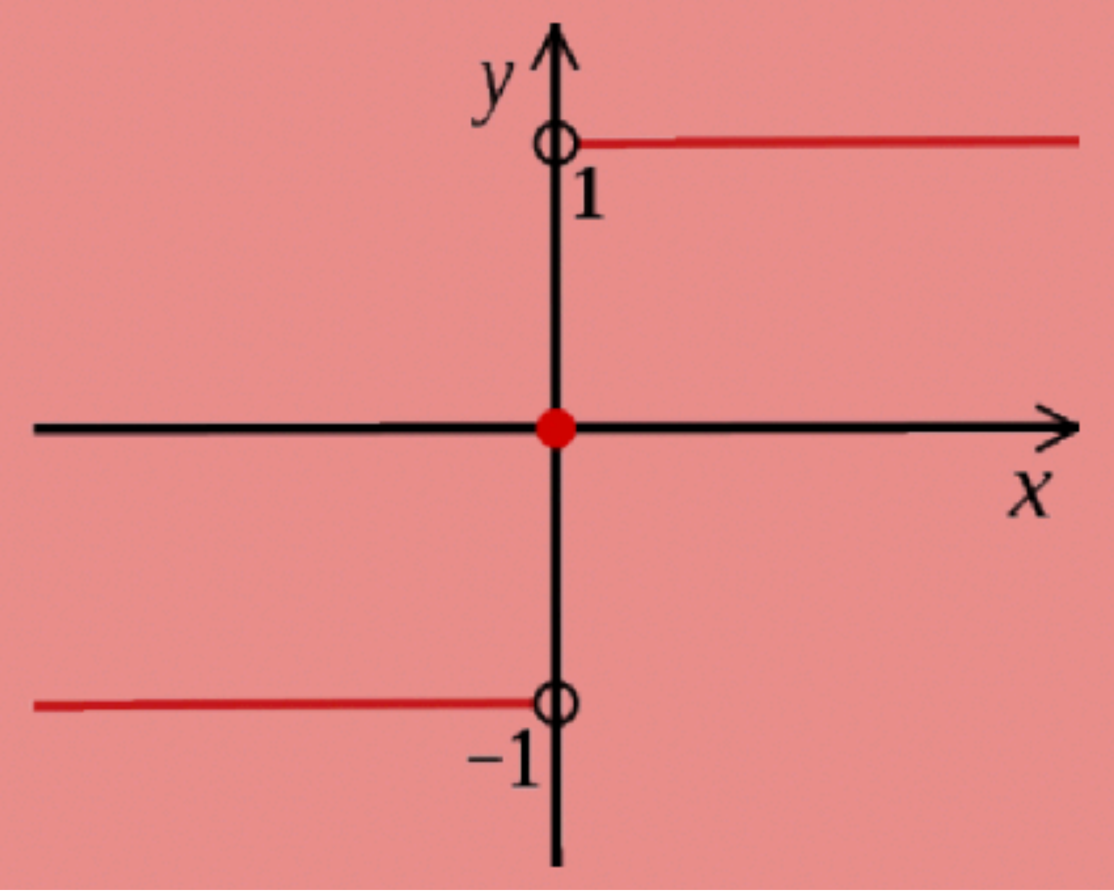


Credits: Matt Gormley

How to make perceptron differentiable?

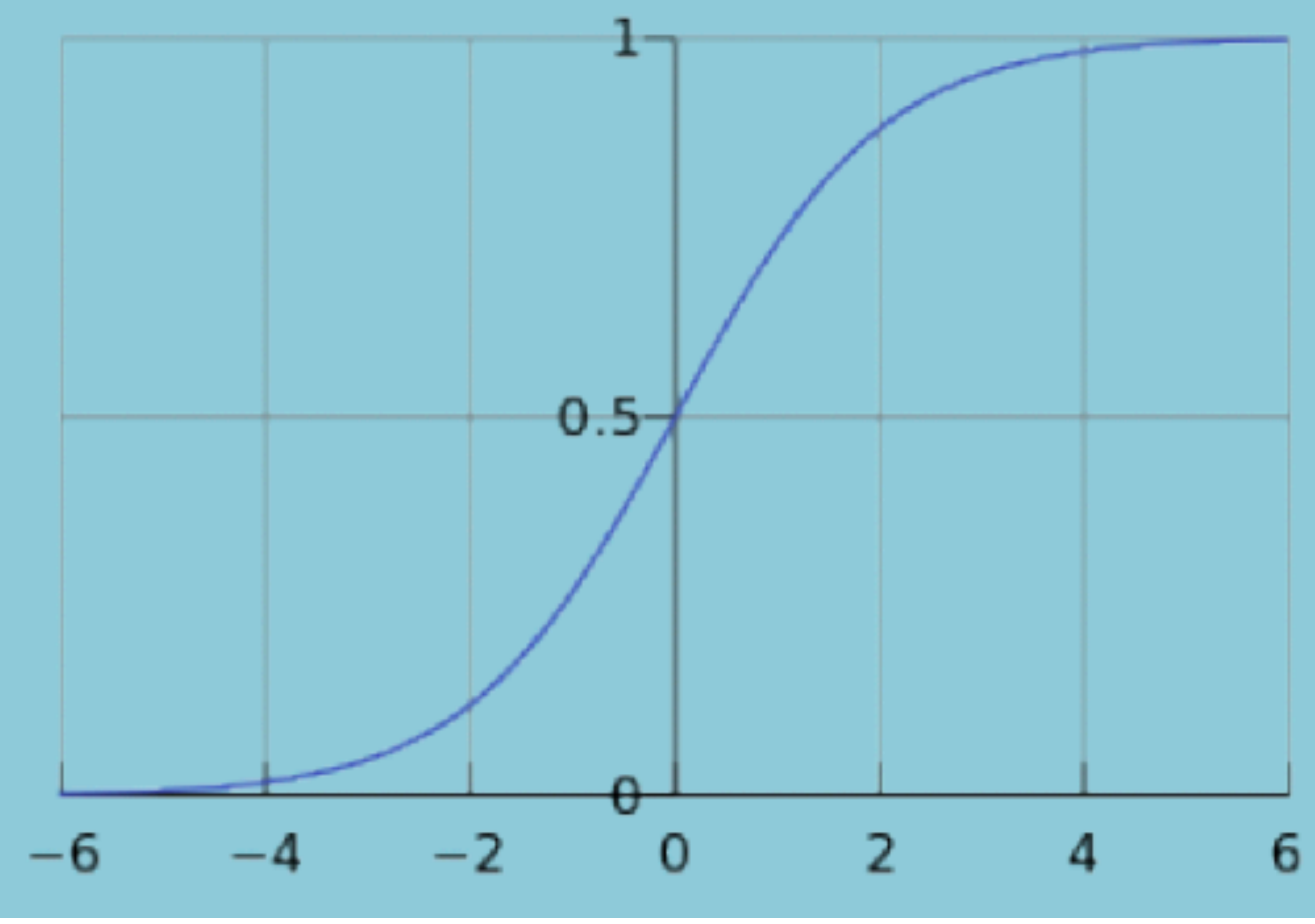
This decision function isn't differentiable:

$$y = sign(\overrightarrow{w}^T x)$$



Use a differentiable function instead:

$$y = \frac{1}{1 + e^{-\overrightarrow{w}^T x}}$$



Credits: Matt Gormley

Logistic regression with gradients

- Input data: $X \in \mathbb{R}^{d \times n}$, $Y \in \{0,1\}^n$, where $(\vec{x} \in \mathbb{R}^d, y \in \{0,1\})$ corresponds to a data point.

Credits: Matt Gormley

Logistic regression with gradients

- Input data: $X \in \mathbb{R}^{d \times n}$, $Y \in \{0,1\}^n$, where $(\vec{x} \in \mathbb{R}^d, y \in \{0,1\})$ corresponds to a data point.

- Decision function: $p(y = 1 | x) = \dfrac{1}{1 + e^{-\vec{w}x}}$

Credits: Matt Gormley

Logistic regression with gradients

- Input data: $X \in \mathbb{R}^{d \times n}$, $Y \in \{0,1\}^n$, where $(\vec{x} \in \mathbb{R}^d, y \in \{0,1\})$ corresponds to a data point.

- Decision function: $p(y = 1 \mid x) = \dfrac{1}{1 + e^{-\vec{w}x}}$

- Objective: $\arg\min_{\vec{w}} L(\vec{w}; D)$

Credits: Matt Gormley

Logistic regression with gradients

- Input data: $X \in \mathbb{R}^{d \times n}$, $Y \in \{0,1\}^n$, where $(\vec{x} \in \mathbb{R}^d, y \in \{0,1\})$ corresponds to a data point.

- Decision function: $p(y = 1 \mid x) = \dfrac{1}{1 + e^{-\vec{w}x}}$

- Objective: $\arg\min_{\vec{w}} L(\vec{w}; D)$

- What should the Loss function be?

Credits: Matt Gormley

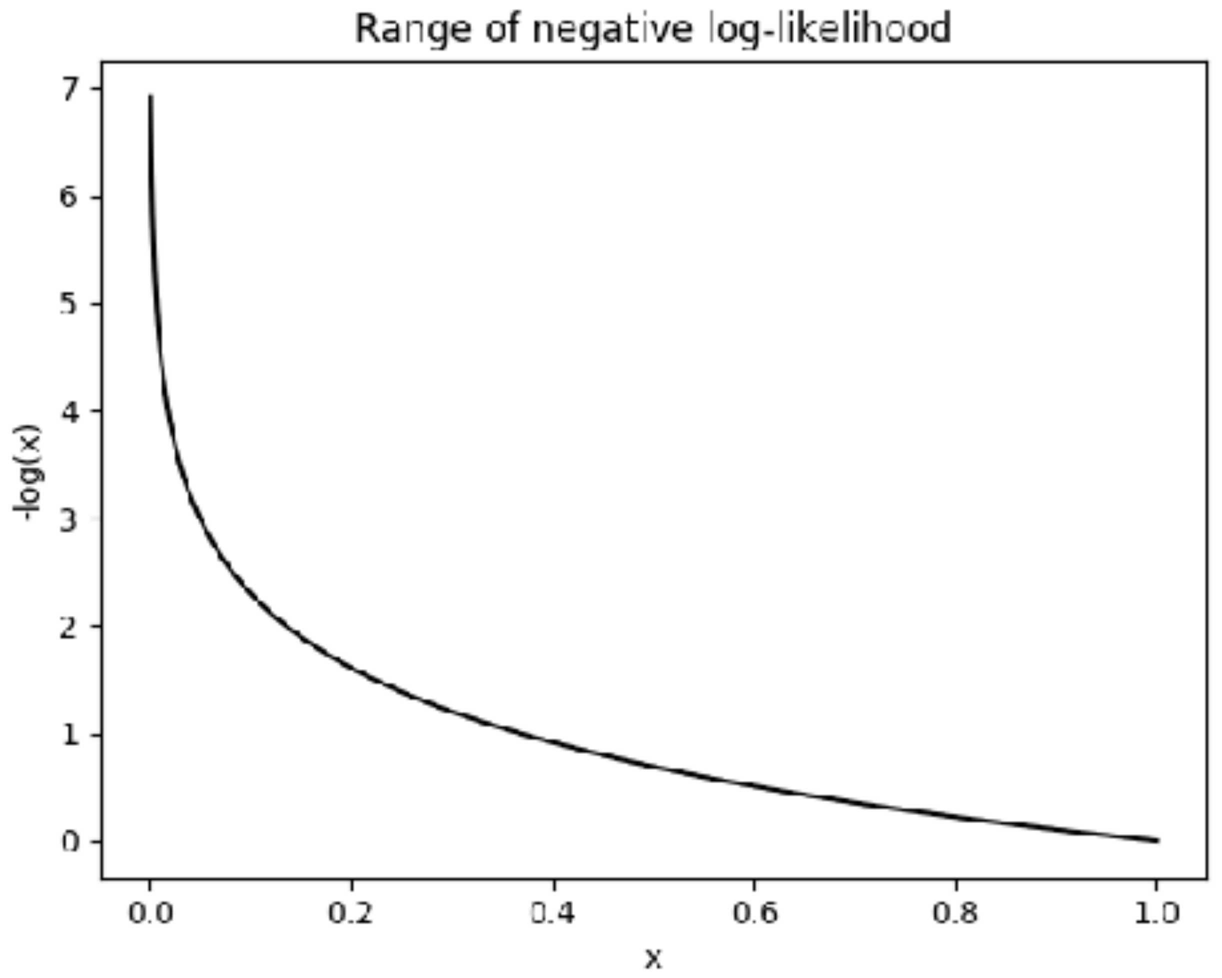
Logistic regression with gradients

- Loss function: Negative Log Likelihood (aka surprisal)

Credits: Matt Gormley

Logistic regression with gradients

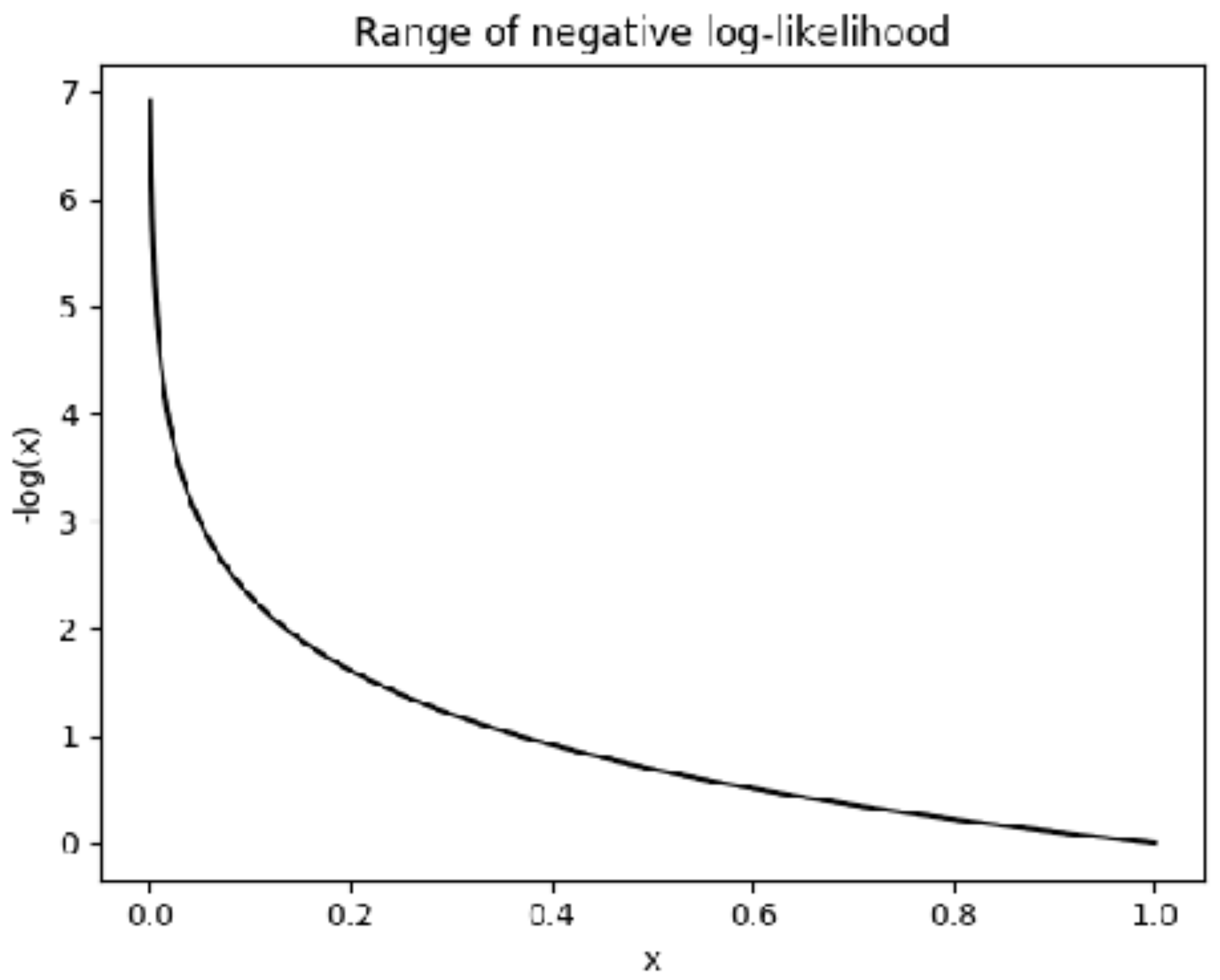
- Loss function: Negative Log Likelihood (aka surprisal)
    - If $y^i = 1, l(x^i, y^i) = -ln(p^i)$
    - If $y^i = 0, l(x^i, y^i) = -ln(1 - p^i)$



Credits: Matt Gormley

Logistic regression with gradients

- Loss function: Negative Log Likelihood (aka surprisal)
  - If $y^i = 1$, $l(x^i, y^i) = -ln(p^i)$
  - If $y^i = 0$, $l(x^i, y^i) = -ln(1 - p^i)$
  - Together: $l(x^i, y^i) = -y^i ln(p^i) - (1 - y^i) ln(1 - p^i)$



Credits: Matt Gormley

Logistic regression with gradients

- Input data: $X \in \mathbb{R}^{d \times n}$, $Y \in \{0,1\}^n$, where ($\overrightarrow{x} \in \mathbb{R}^d, y \in \{0,1\}$) corresponds to a data point.
- Decision function: $p(y = 1 | x) = \dfrac{1}{1 + e^{-\overrightarrow{w}x}}$
- Objective: $\arg\min_{\overrightarrow{w}} L(\overrightarrow{w}; D)$
- Loss function: $l(x^i, y^i) = -y^i ln(p^i) - (1 - y^i) ln(1 - p^i)$
- Solve with SGD or its variants.

Credits: Matt Gormley

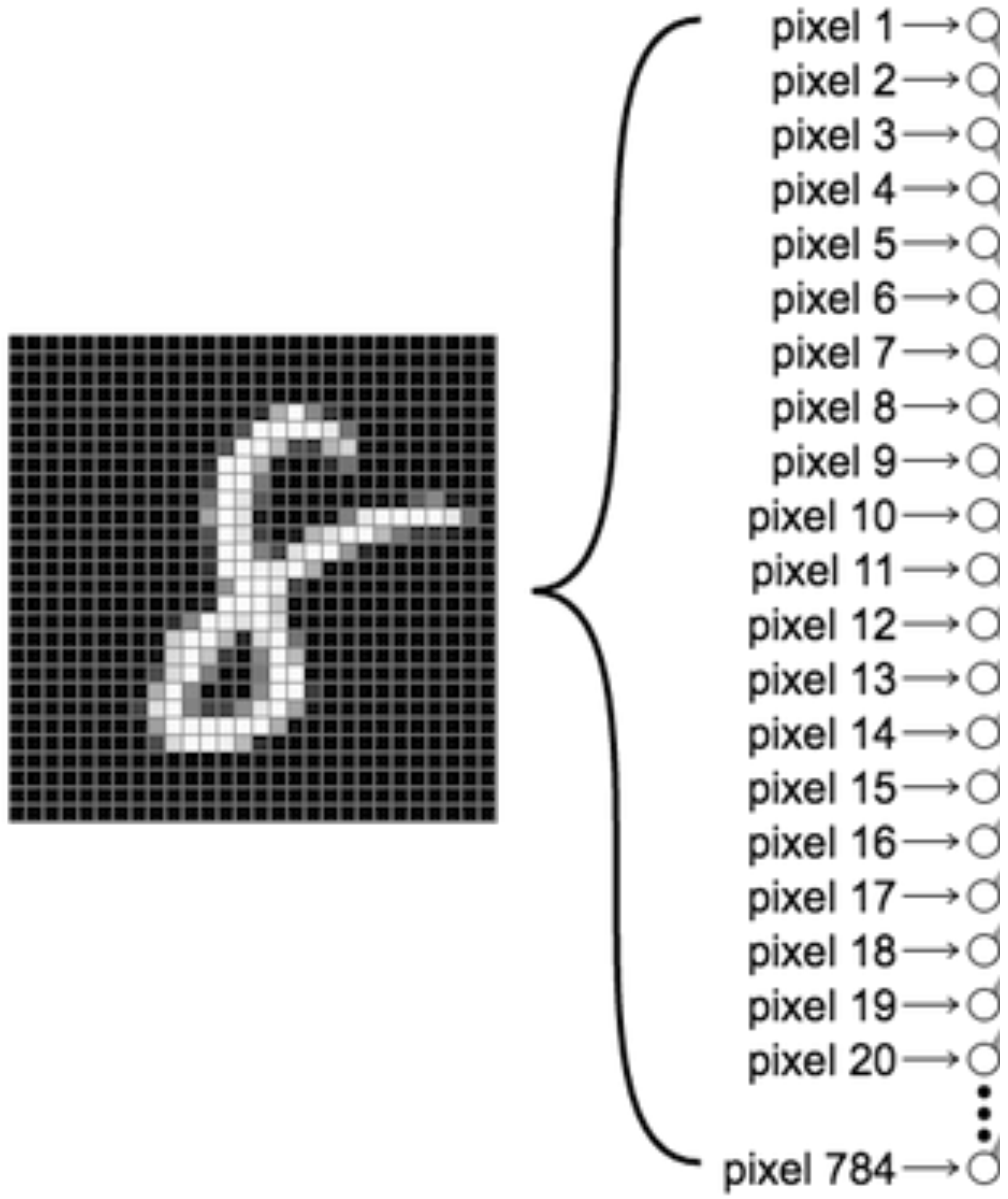
Example of data

Credits: MNIST

Example of data

Credits: MNIST

Example of data



Credits: MNIST

Better feature

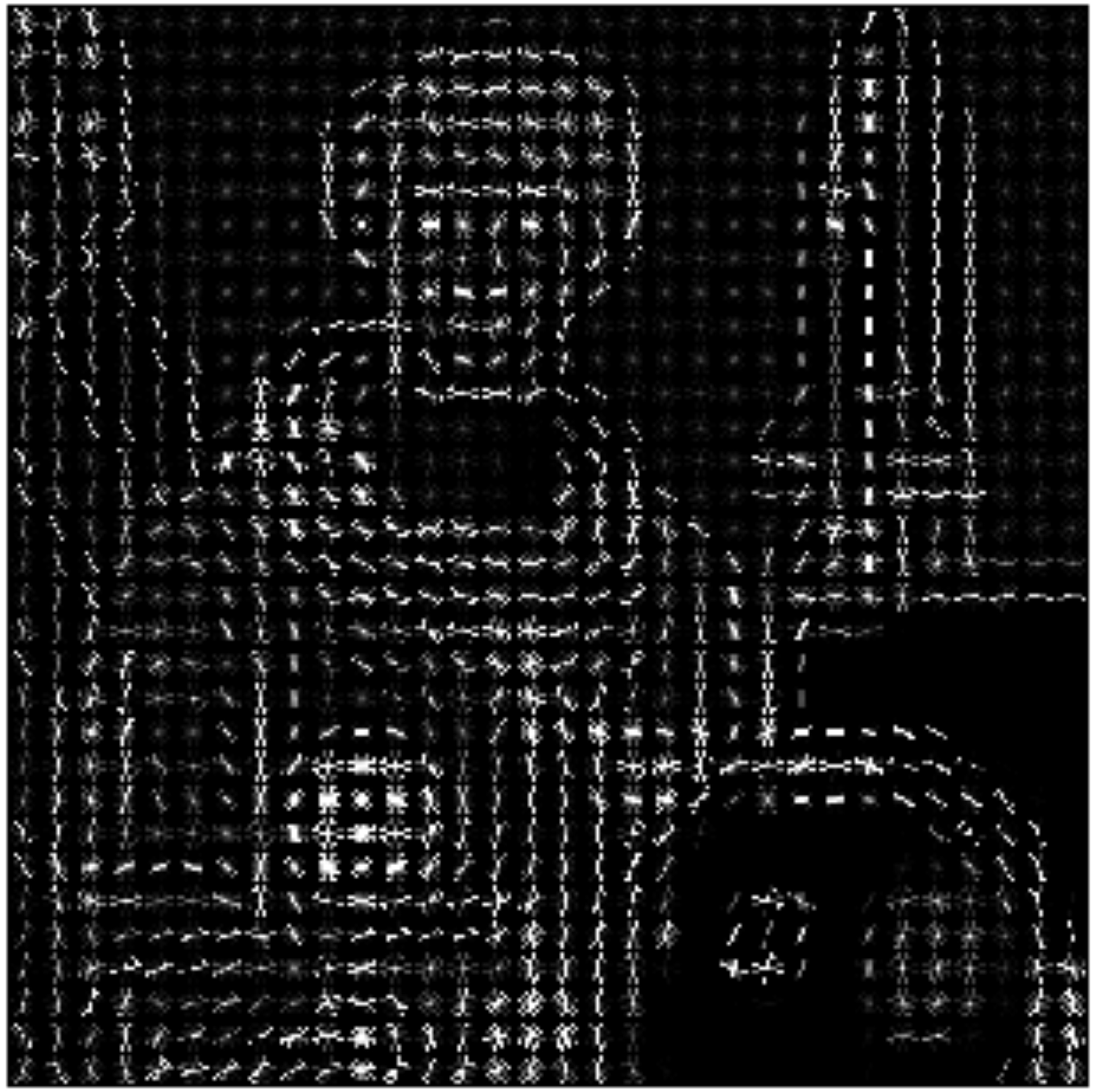
Input image

Credits: iq.opengenus.org

Better feature

Input image

Histogram of Oriented Gradients



Credits: iq.opengenus.org

Logistic regression with ‘features’

Credits: iq.opengenus.org

Additional Reading

- Book chapter and exercises: https://nhorton.people.amherst.edu/ips9/IPS_09_Ch14.pdf

- History of logistic regression: https://papers.tinbergen.nl/02119.pdf

Questions?