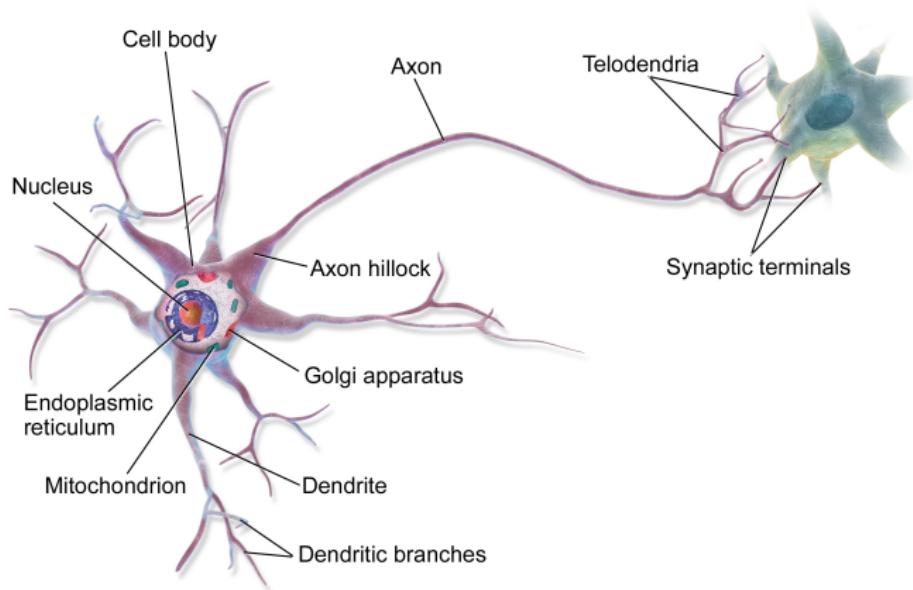
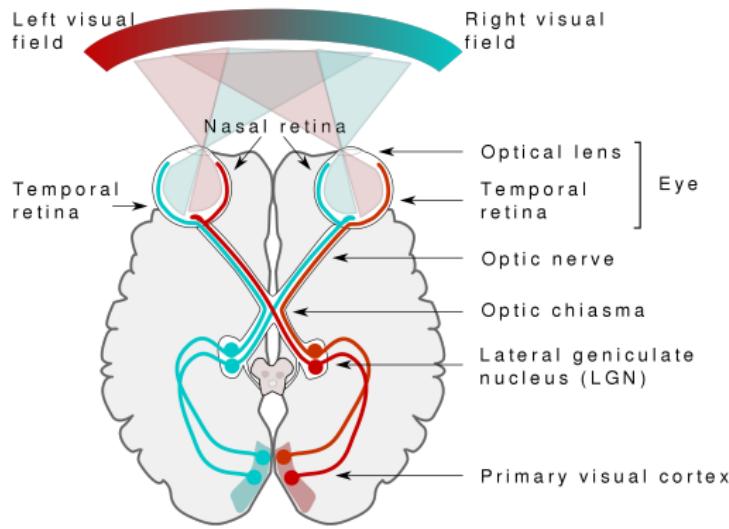
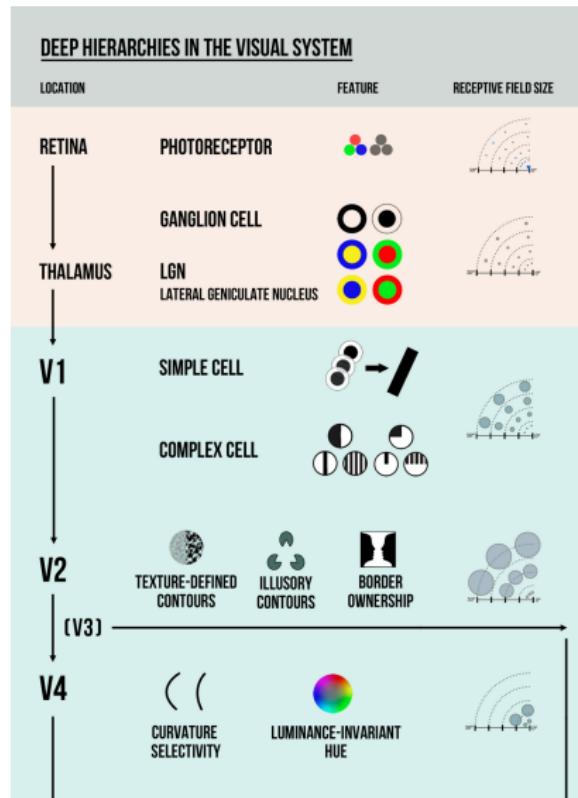


DSC 40A

Lecture 11
Perceptrons



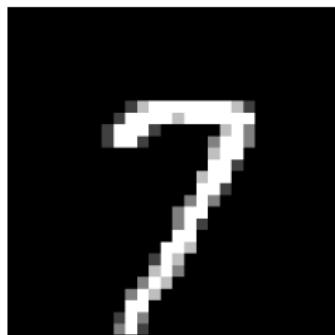




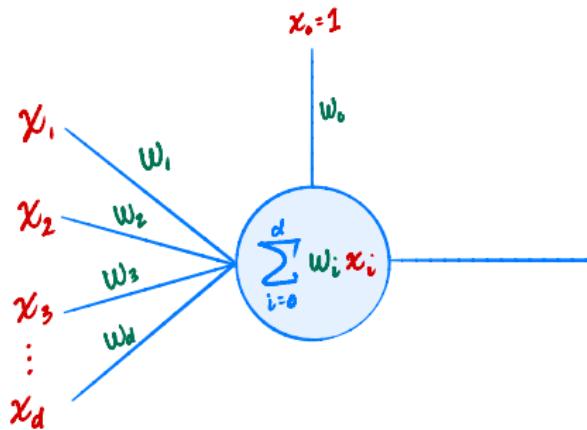


Today

- ▶ Design an artificial “neuron”, a **perceptron**.
- ▶ Train it to recognize images (handwritten digits).

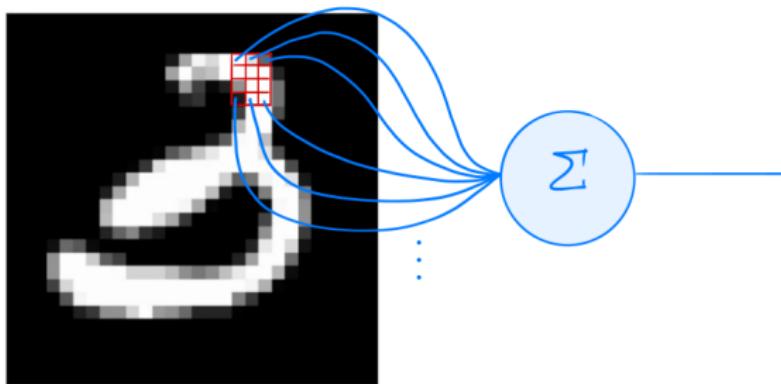


Modeling a Neuron



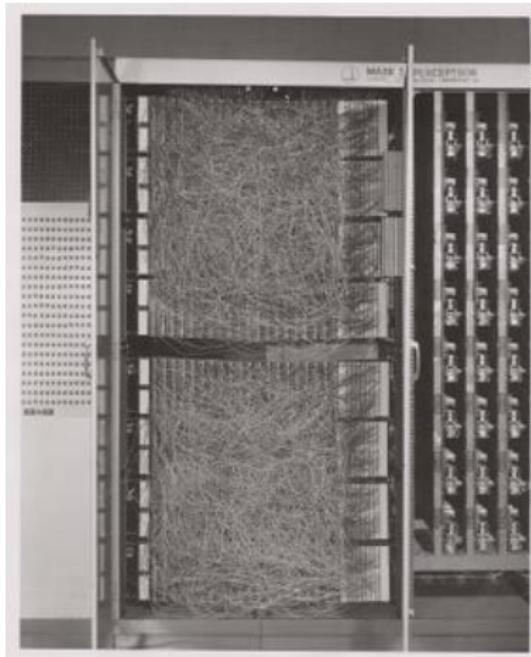
- ▶ Input: $\vec{x} = (x_1, \dots, x_d)^T$
- ▶ **Synapse weights:** $\vec{w} = (w_0, w_1, \dots, w_d)^T$
- ▶ Output: $\sum_{i=0}^d w_i x_i = \text{Aug}(\vec{x}) \cdot \vec{w}$
- ▶ This model is called a **perceptron**.

Example: Image Recognition



- ▶ Binary classifier:
 - ▶ If output > 0 , predicted digit is a three
 - ▶ If output < 0 , predicted digit is not a three

Rosenblatt's Perceptron (1958)



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human be-

ings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

1958 New York Times...

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

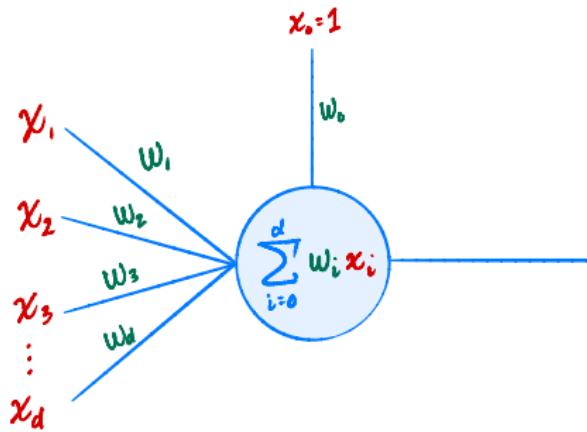
Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Training a Perceptron



- ▶ A perceptron is trained by adjusting its weights, w_0, \dots, w_d .

Example: Predicting Survival

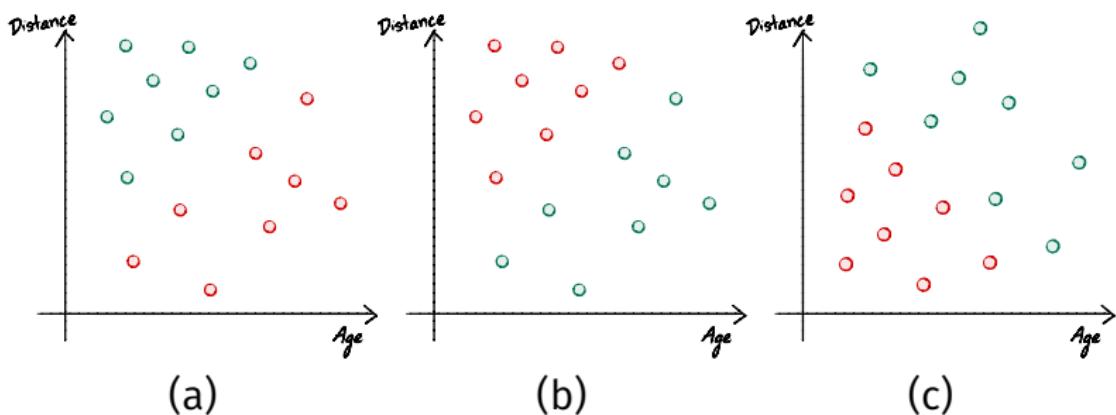
- ▶ We have a data set of hurricane survivors:

Age	Distance From Coast	Survived
21	30	1 (Yes)
84	2	-1 (No)
47	10	-1 (No)
30	15	1 (Yes)
:	:	:

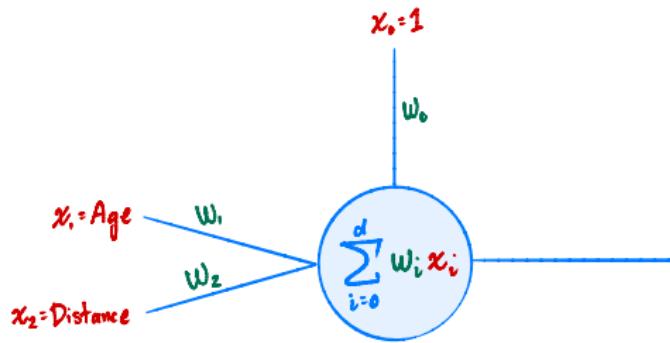
- ▶ Goal: train perceptron to predict if someone will survive.
 - ▶ If output is positive, prediction is “yes”
 - ▶ If output is negative, prediction is “no”

Discussion Question

Suppose we plot the data, with green points for people who survived, and red points for those who did not. Which do we see?



Example: Predicting Survival

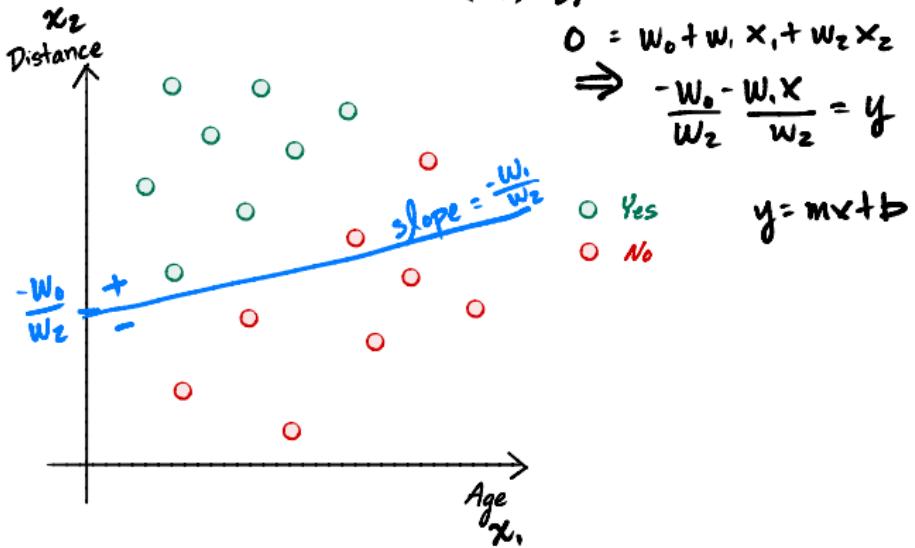


- ▶ Prediction rule:

$$H(\text{Age}, \text{Distance}) = w_0 + w_1 \times \text{Age} + w_2 \times \text{Distance}$$

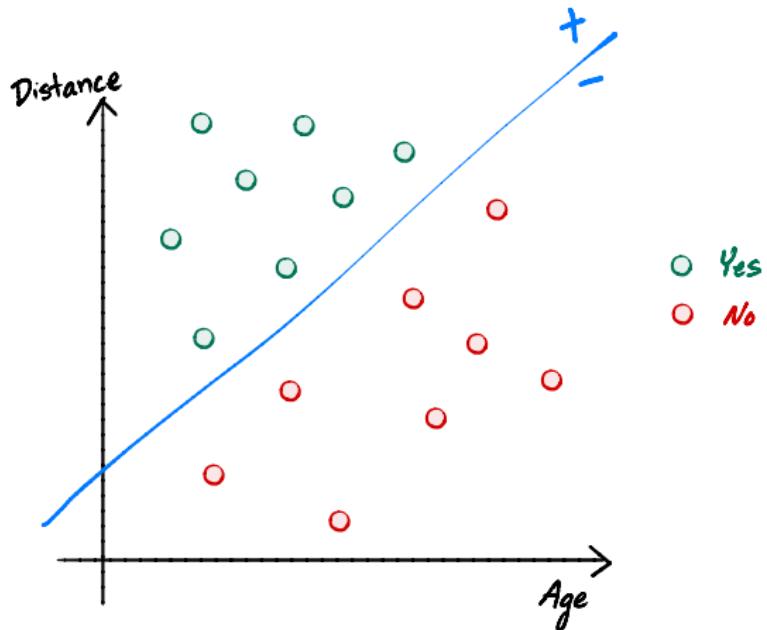
- ▶ If $H(x_1, x_2) > 0$, prediction is “yes”
- ▶ If $H(x_1, x_2) < 0$, prediction is “no”

The Decision Boundary



- ▶ The **decision boundary** is where $H = 0$.
- ▶ On one side of boundary, $H > 0$; prediction is **yes**.
- ▶ On other side, $H < 0$; prediction is **no**.
- ▶ **Important:** $|H(\vec{x})| \propto$ distance from boundary.

A Good Decision Boundary



Learning a Linear Decision Boundary

- ▶ Given feature vectors $\vec{x}^{(1)}, \dots, \vec{x}^{(n)}$ and labels y_1, \dots, y_n .
- ▶ Goal: find \vec{w} such that each point is **classified** correctly; i.e., it is on the correct side of boundary.
- ▶ We'll use empirical risk minimization.
- ▶ What is an appropriate loss function?

Loss #1: The 0-1 Loss

- ▶ How do we measure the error of a prediction?
- ▶ Four possibilities:
 - ▶ Correct label is -1 , and $H(\vec{x}) < 0$, classified **correctly**.
 - ▶ Correct label is -1 , and $H(\vec{x}) > 0$, classified **incorrectly**.
 - ▶ Correct label is 1 , and $H(\vec{x}) < 0$, classified **incorrectly**.
 - ▶ Correct label is 1 , and $H(\vec{x}) > 0$, classified **correctly**.
- ▶ The **0-1 loss**:

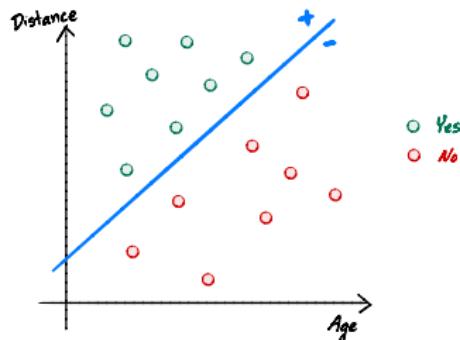
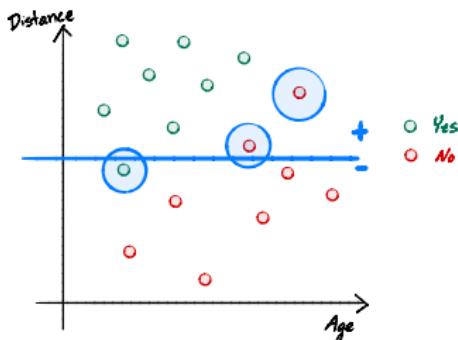
$$L_{01}(H(\vec{x}), y) = \begin{cases} 1, & \text{sign}(H(\vec{x})) \neq \text{sign}(y) \\ 0, & \text{sign}(H(\vec{x})) = \text{sign}(y) \end{cases}$$

Loss #1: The 0-1 Loss

- The **0-1 risk** is then:

$$R_{01}(\vec{w}) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1, & \text{sign}(H(\vec{x})) \neq \text{sign}(y) \\ 0, & \text{sign}(H(\vec{x})) = \text{sign}(y) \end{cases}$$

- Counts proportion of points which are misclassified.
- Example:



Goal: Minimize the 0-1 Risk

- ▶ Find \vec{w} which results in fewest # of misclassified points.
- ▶ That is, minimize

$$R_{01}(\vec{w}) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1, & \text{sign}(H(\vec{x}^{(i)})) \neq \text{sign}(y_i) \\ 0, & \text{sign}(H(\vec{x}^{(i)})) = \text{sign}(y_i) \end{cases}$$

- ▶ Gradient descent?

Gradient Descent for Functions of a Vector

- ▶ Pick α to be a positive number. It is the **learning rate**.
- ▶ Pick a starting parameter vector, $\vec{w}^{(0)}$.
- ▶ On step i , perform update $\vec{w}^{(i)} = \vec{w}^{(i-1)} - \alpha \cdot \frac{dR}{d\vec{w}}(\vec{w}^{(i-1)})$
- ▶ Repeat until convergence (when $\vec{w}^{(i)}$ doesn't change much).

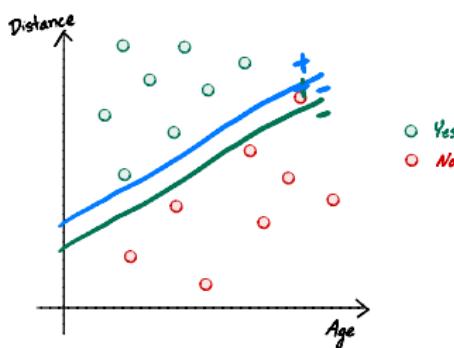
Discussion Question

Is

$$R_{01}(\vec{w}) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1, & \text{sign}(H(\vec{x}^{(i)})) \neq \text{sign}(y_i) \\ 0, & \text{sign}(H(\vec{x}^{(i)})) = \text{sign}(y_i) \end{cases}$$

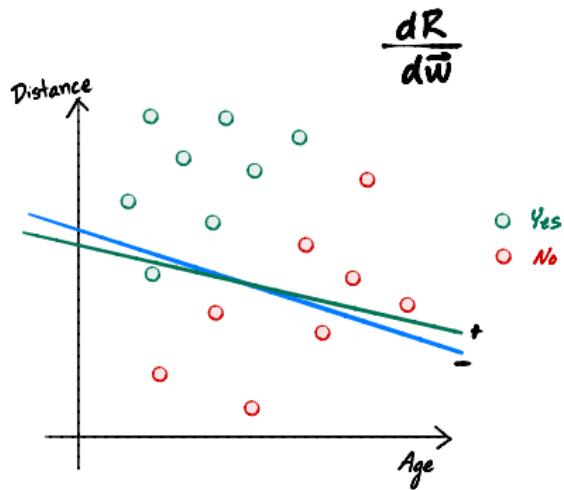
continuous? Differentiable?

- 15 A) Continuous, not differentiable
22 B) Continuous, differentiable
34 C) Neither



The Problem

- ▶ R_{01} is flat.
- ▶ The gradient gives no information.



Loss #2: The Perceptron Loss

- ▶ We need a loss function that isn't flat.
- ▶ If prediction is wrong, size of $|H(\vec{x})|$ measures how wrong.
- ▶ The **perceptron loss**:

$$L_{\text{tron}}(H(\vec{x}), y) = \begin{cases} 0, & \text{sign}(H(\vec{x})) = \text{sign}(y) \\ |H(\vec{x})|, & \text{sign}(H(\vec{x})) \neq \text{sign}(y) \end{cases}$$

Loss #2: The Perceptron Loss

- The perceptron risk:

$$R_{\text{tron}}(\vec{w}) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 0, & \text{sign}(H(\vec{x}^{(i)})) = \text{sign}(y_i) \\ |H(\vec{x}^{(i)})|, & \text{sign}(H(\vec{x}^{(i)})) \neq \text{sign}(y_i) \end{cases}$$
$$= \frac{1}{n} \sum_{i \in M} |H(\vec{x}^{(i)})|$$

correctly classified *misclassified*

where M is the set of misclassified points.

- Continuous, not differentiable, not flat.

Gradient Descent for Perceptron Learning

- ▶ Compute gradient of:

$$H(\vec{x}) = \text{Aug}(\vec{x}) \cdot \vec{w}$$

$$R_{\text{tron}}(\vec{w}) = \frac{1}{n} \sum_{i \in M} |H(\vec{x}^{(i)})|$$

$$\frac{dR}{d\vec{w}} = \frac{d}{d\vec{w}} \left[\frac{1}{n} \sum_{i \in M} |H(\vec{x}^{(i)})| \right] = \frac{1}{n} \sum_{i \in M} \frac{d}{d\vec{w}} |H(\vec{x}^{(i)})|$$

$$= \frac{1}{n} \sum_{i \in M} \frac{d}{d\vec{w}} \begin{cases} \text{Aug}(\vec{x}^{(i)}) \cdot \vec{w}, & \text{Aug}(\vec{x}^{(i)}) \cdot \vec{w} > 0 \\ -\text{Aug}(\vec{x}^{(i)}) \cdot \vec{w}, & \text{Aug}(\vec{x}^{(i)} \cdot \vec{w}) < 0 \end{cases}$$

$$= \frac{1}{n} \sum_{i \in M} \begin{cases} \text{Aug}(\vec{x}^{(i)}), & \text{Aug}(\vec{x}^{(i)}) \cdot \vec{w} > 0 \\ -\text{Aug}(\vec{x}^{(i)}), & \text{Aug}(\vec{x}^{(i)} \cdot \vec{w}) < 0 \end{cases}$$

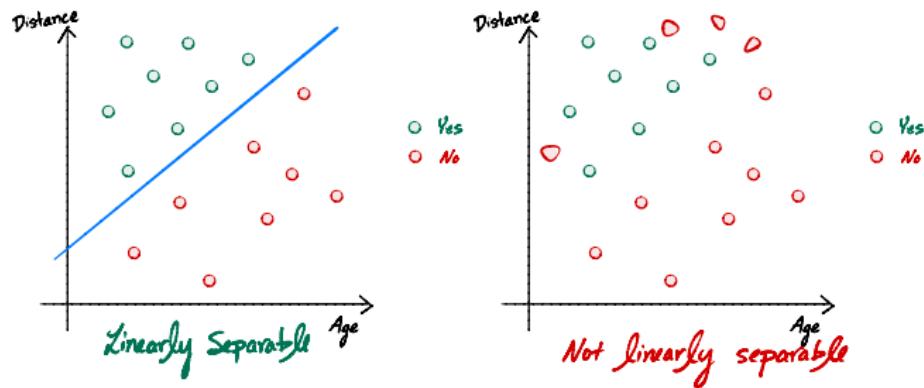
The Perceptron Algorithm

- ▶ Pick an initial $\vec{w}^{(0)}$.
- ▶ On iteration t :
 - ▶ Construct set M of misclassified points using $\vec{w}^{(t-1)}$
 - ▶ If M is empty, break (no points misclassified).
 - ▶ Otherwise, perform update:

$$\vec{w}^{(t)} = \vec{w}^{(t-1)} - \frac{\alpha}{n} \sum_{i \in M} \begin{cases} \text{Aug}(\vec{x}^{(i)}), & \vec{w}^{(t-1)} \cdot \text{Aug}(\vec{x}^{(i)}) \geq 0 \\ -\text{Aug}(\vec{x}^{(i)}), & \vec{w}^{(t-1)} \cdot \text{Aug}(\vec{x}^{(i)}) < 0 \end{cases}$$

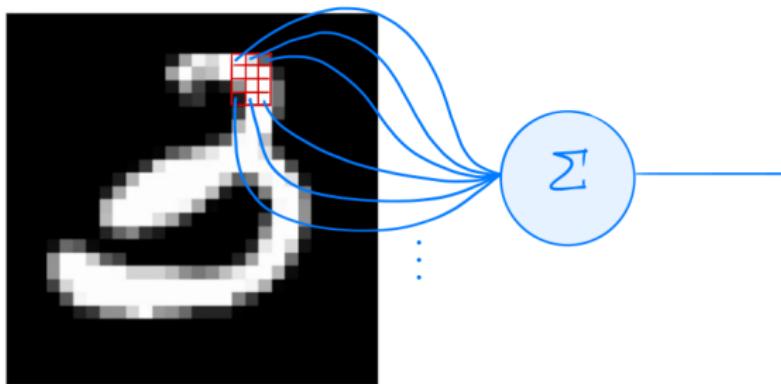
The Perceptron Algorithm

- ▶ Data is **linearly separable** if classes can be separated by a line (plane):



- ▶ If linearly separable, perceptron algorithm will terminate; classify all points correctly.

Example: Image Recognition



- ▶ Binary classifier:
 - ▶ If output > 0 , predicted digit is a three
 - ▶ If output < 0 , predicted digit is not a three

(DEMO)

