CS 480/680
Introduction to Machine Learning

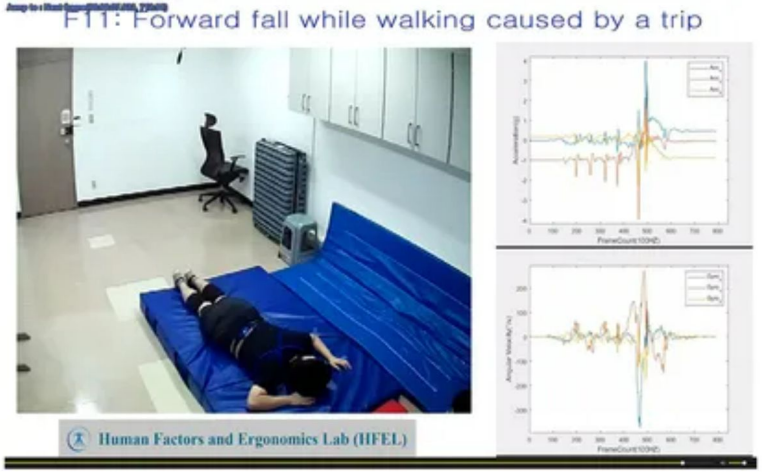
Lecture 1
Halfspaces and the Perceptron Algorithm

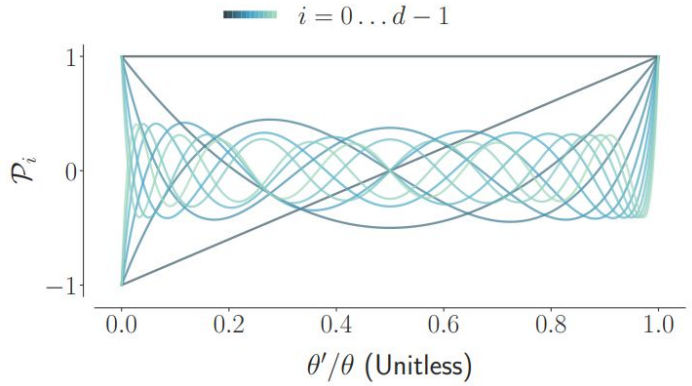
Kathryn Simone

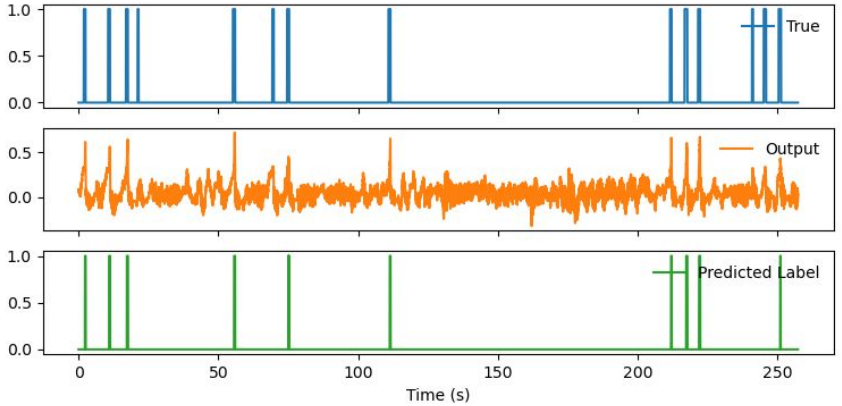
10 September 2024

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Classification: Fall detection from accelerometer data

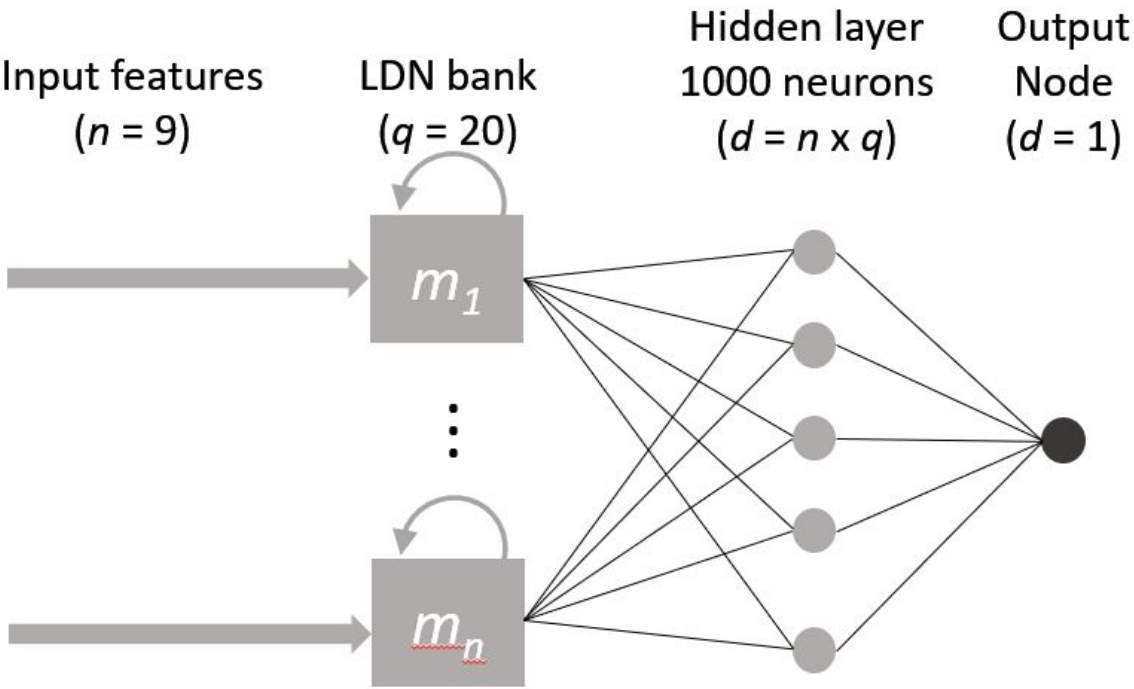






Clockwise from top:
Yu, Xiaoqun, Jaehyuk Jang, and Shuping Xiong. *Frontiers in Aging Neuroscience* 13 (2021): 692865.
Voelker, Aaron, Ivana Kajić, and Chris Eliasmith. *Advances in neural information processing systems* 32 (2019).
Barkley and Simone 2023, Unpublished

Most of ML makes use of linear methods



The perceptron algorithm learns a classifier using linear combinations of features

Aims

At the end of the lecture, we should be able to:

- Identify the components of a dataset required for supervised learning.
- Interpret the separating hyperplane hypothesis class geometrically.
- Implement the Perceptron algorithm and list its properties.
- Reproduce Novikoff's proof of the Perceptron convergence theorem.

Lecture Outline

I. **What is needed in order to learn a classifier?**

   The structure of observations and hypotheses

II. **How can we learn a hypothesis from data?**

   The Perceptron Algorithm

III. **Why does this work?**

   Convergence analysis and other properties

IV. **Summary + Housekeeping**

Lecture Outline

I. **What is needed in order to learn a classifier?**

The structure of observations and hypotheses

II. **How can we learn a hypothesis from data?**

The Perceptron Algorithm

III. **Why does this work?**

Convergence analysis and other properties

IV. **Summary + Housekeeping**

A motivating example: predicting whether you'll pass a class

A motivating example: predicting whether you'll pass a class

Divination effort dataset

Name	Total time	Proportion on Exams	Passed?
Draco Malfoy	20	0.8	No
Hannah Abbott	50	0.5	Yes
Padma Patil	90	0.1	Yes
Susan Bones	70	0.9	No
Terry Boot	80	0.2	Yes
Oliver Wood	120	0.3	No

The Binary Classification Problem

Given $(\vec{x_1}, y_1), (\vec{x_2}, y_2), ...(\vec{x_k}, y_k)$,

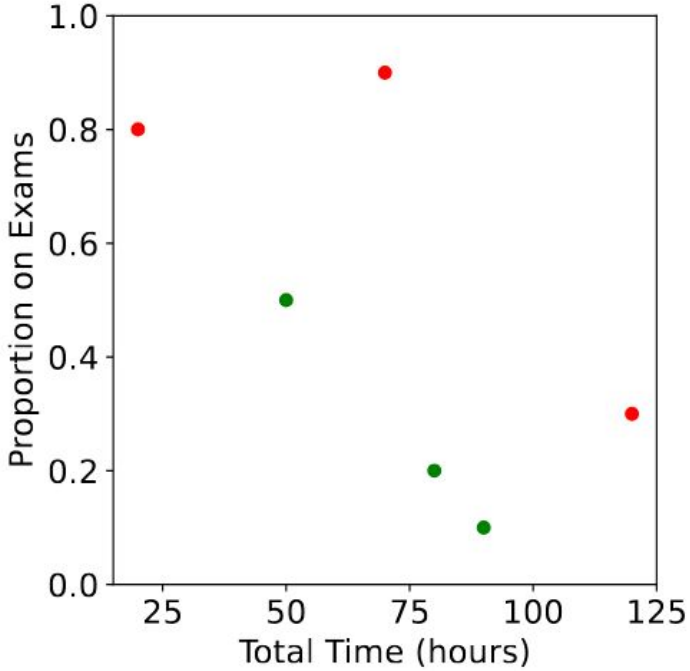
- $\vec{x_i}$: *feature vector*. Often, $\vec{x_i} \in \mathbb{R}^m$.
- y_i: *class label*. For binary classification, $y_i \in \{-1, +1\}$.

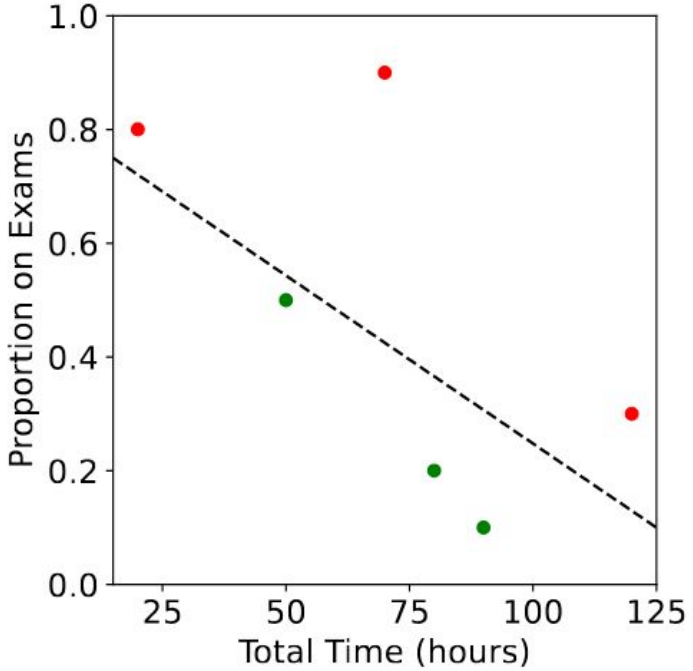
Observation Index	x_1	x_2	$\cdots$	x_m	y_i
1	x_{11}	x_{12}	$\cdots$	x_{1m}	y_1
2	x_{21}	x_{22}	$\cdots$	x_{2m}	y_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
n	x_{n1}	x_{n2}	$\cdots$	x_{nm}	y_n

... Learn a function h such that $h(\vec{x}) = y$, to predict the class label y_i for some arbitrary feature vector $\vec{x_i} \in \mathbb{R}^m$.

Because it allows us to make predictions, $h(x)$ is called a *hypothesis*.

Exploring the “divination effort” dataset

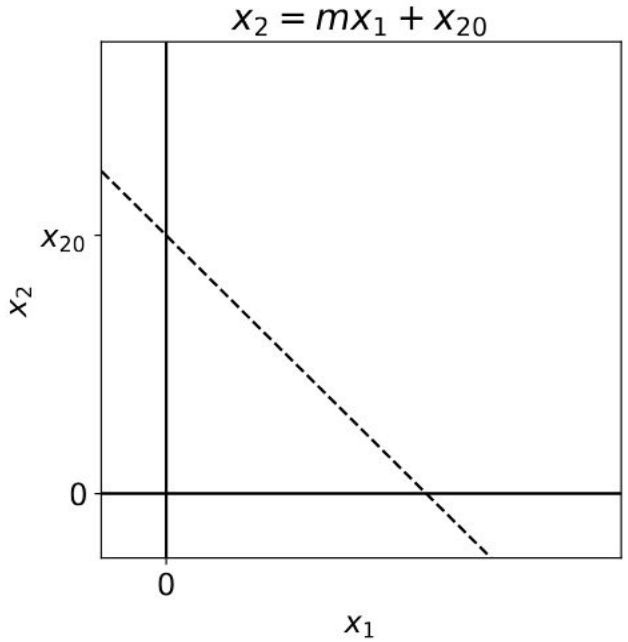




The slope-intercept form for a line is inconvenient

Slope-intercept form:

- $x_2 = mx_1 + x_{2o}$
- Implies x_2, a feature of the data, is the dependent variable
- Indirect process for computing class labels y



A line defines a hyperplane, or affine set, in $\mathbb{R}^2$

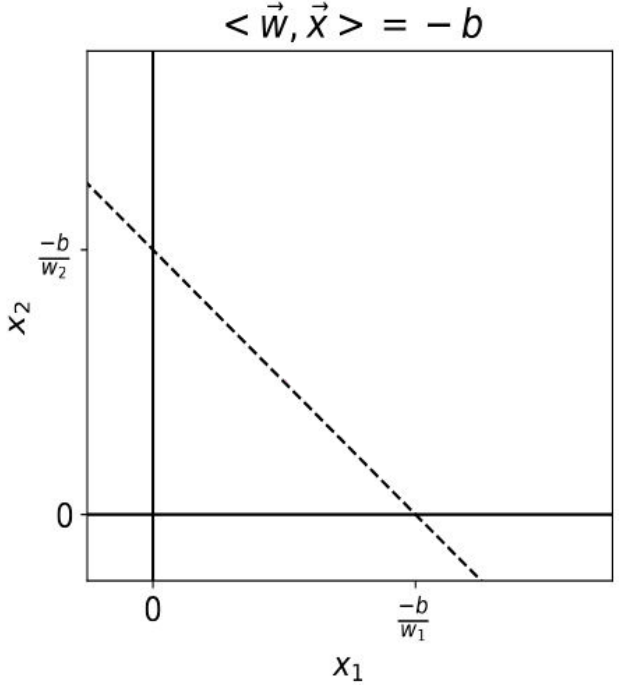
Affine set:

- The set of points $x \in \mathbb{R}^d$ for which the following holds

$$w^T x = -b,$$

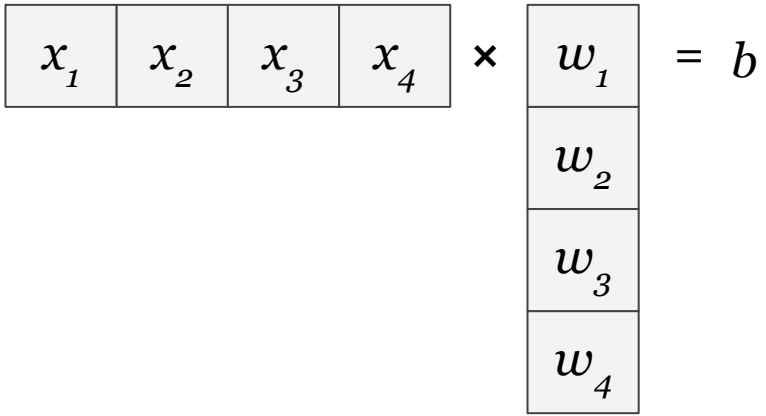
where w is a parameter vector, x is the feature vector, and b is a scalar.

If equality holds, some arbitrary x is on the line.

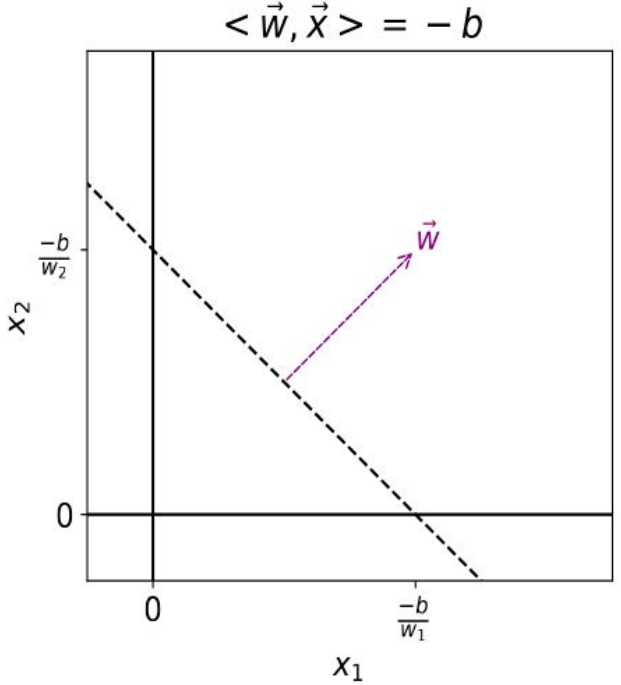


The notations and meaning of inner product

$w^T x = w_1 x_1 + w_2 x_2, \ldots, w_d x_d = \sum_{i=1}^{d} w_i x_i$. Referred to as the "inner product" and written as $< w, x >$.

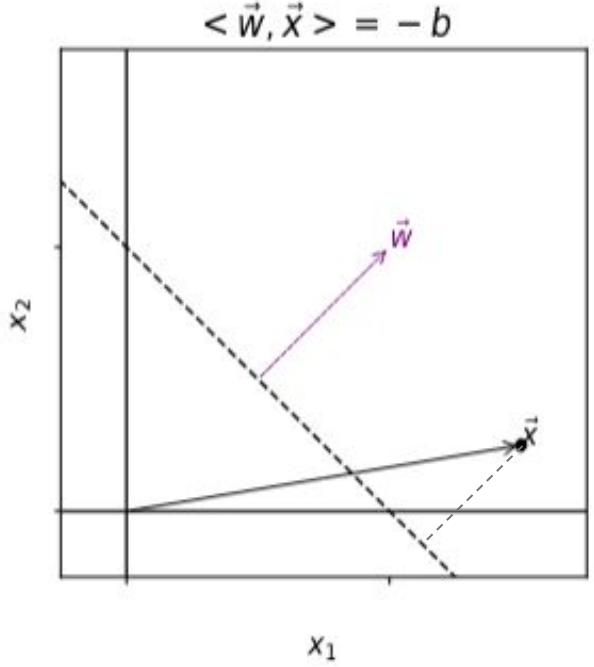


A line defines a hyperplane, or affine set, in $\mathbb{R}^2$



The separating hyperplane hypothesis class

- Signed distance of a point $\vec{x}$ to the decision boundary is segment that joins the point to the line and is perpendicular to the line.
- $h(x) = \text{sgn}(w^T + b)$
- w: "feature weights", b: "bias", sgn: return the sign.



Lecture Outline

I. **What is needed in order to learn a classifier?**

The structure of observations and hypotheses

II. **How can we learn a hypothesis from data?**

The Perceptron Algorithm

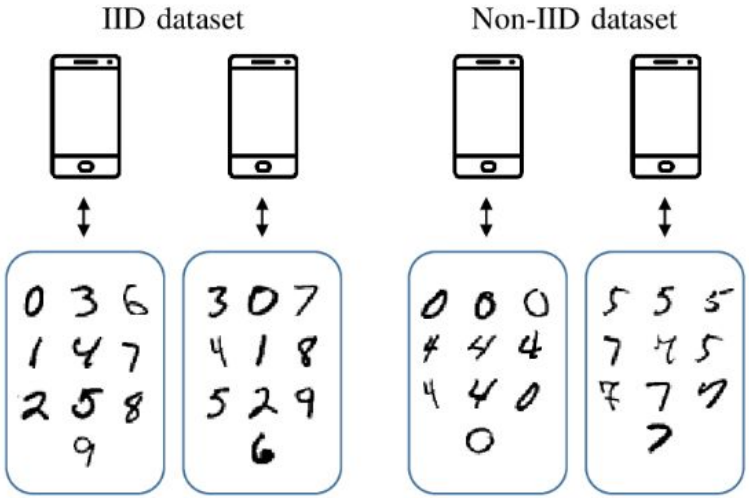
III. **Why does this work?**

Convergence analysis and other properties

IV. **Summary + Housekeeping**

Statistical (Batch) Learning

- Given $(\vec{x_1}, y_1), (\vec{x_2}, y_2), ...(\vec{x_k}, y_k) \sim_{i.i.d} P$, where
  - $i.i.d$: independently and identically distributed



  - P: some unknown distribution
- Goal: Learn $h : Rd \rightarrow \{\pm 1\}$ such that $\Pr_{(x,y)\sim P}[h(x) = y]$ is large.

Online Learning

- Streaming data
- Predict y
- Learn $y = h(x)$ to minimize the number of mistakes.

The Perceptron Algorithm

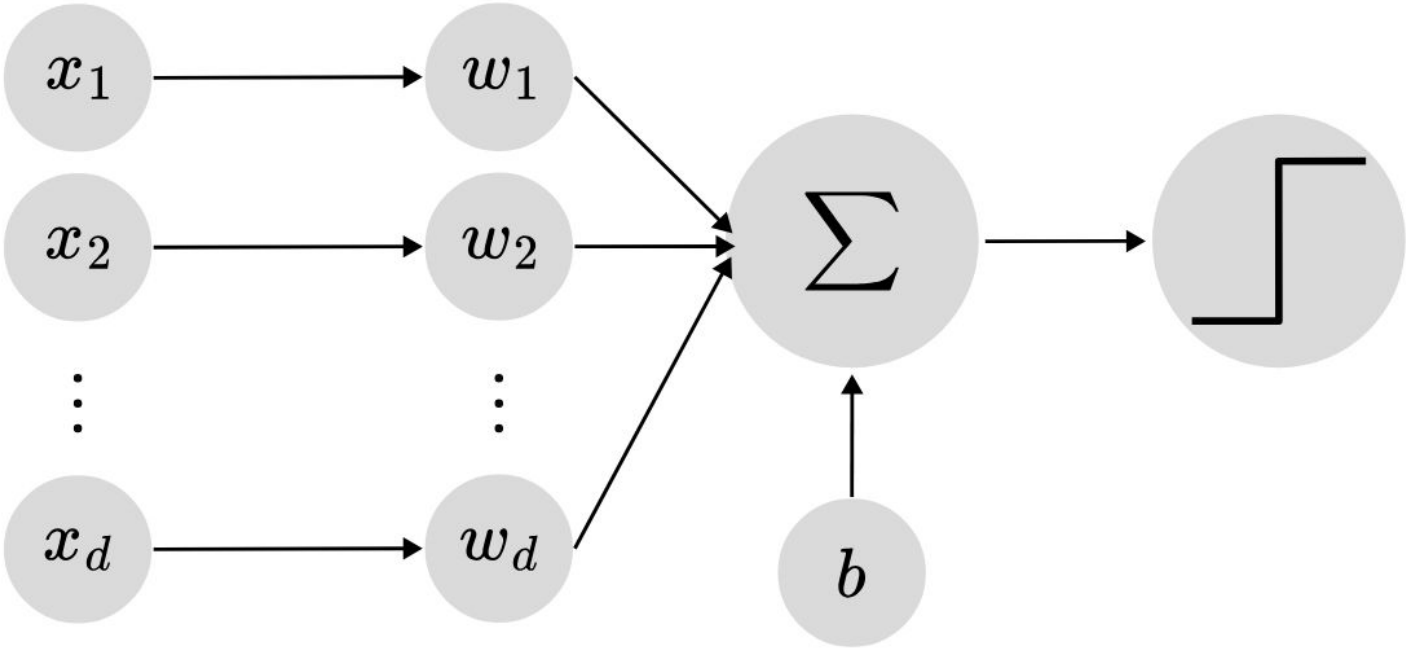
Algorithm 1 The Perceptron (Rosenblatt 1958).

Input: Dataset $D = \{(x_i, y_i) \in \mathbb{R}^d \times \{\pm 1\}$, initialization $w \in \mathbb{R}^d$ and $b \in R$, threshold $\delta \geq 0$
Output: Approximate solution w and b

for $t = 1,2,...,k$ **do**
  Receive training example (x_t, y_t)
  Compute prediction $\hat{y} = w^T x_t + b$
  if $y\hat{y} \leq \delta$ **then**
    $w \leftarrow w + yx_t$
    $b \leftarrow b + y$
  end if
end for

The Perceptron Algorithm in Action

Biological interpretation



Lecture Outline

I. **What is needed in order to learn?**

   The structure of observations and hypotheses

II. **How can we learn a hypothesis from data?**

   The Perceptron Algorithm

III. Why does this work?

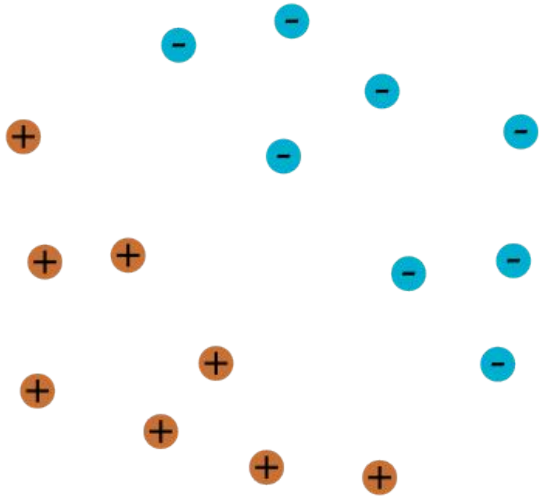
   Convergence analysis and other properties

IV. **Summary + Housekeeping**

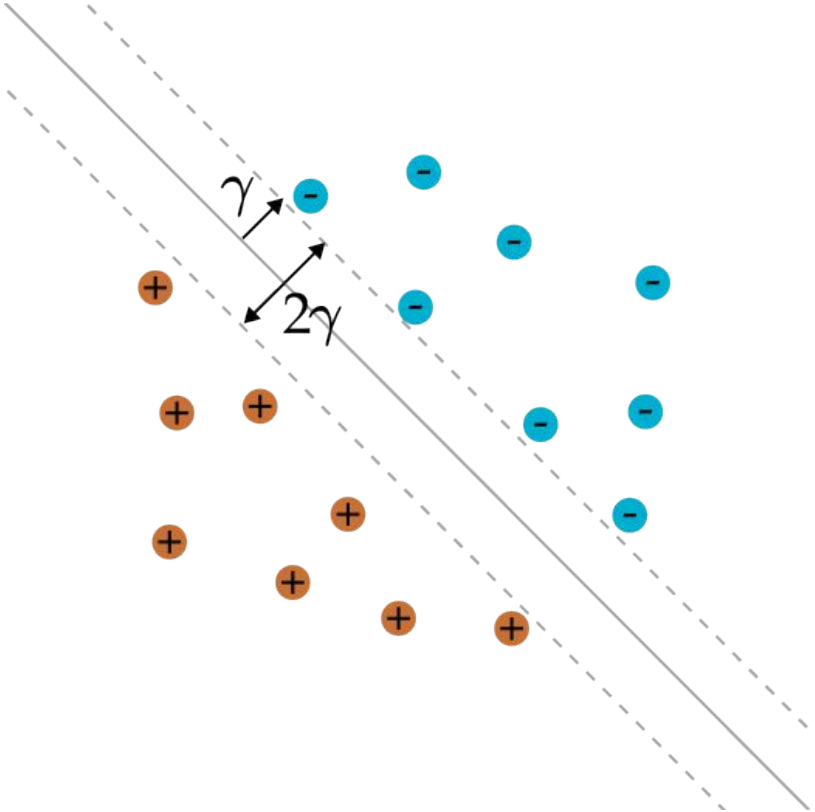
The Perceptron convergence theorem (informal)

Linearly separable  Perceptron converges

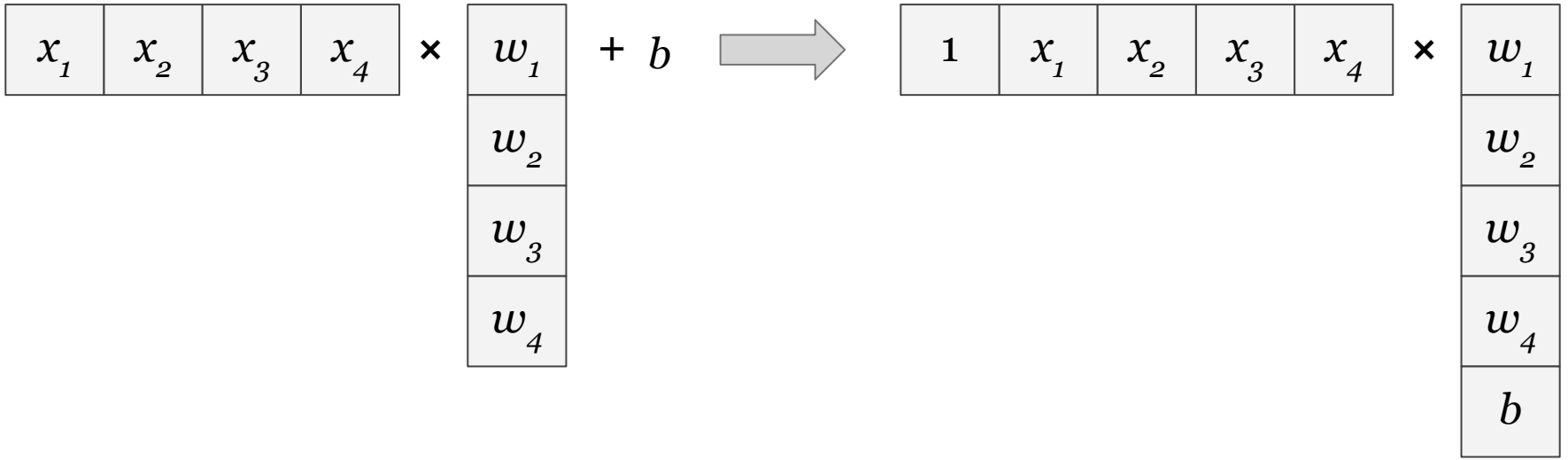
How can we define linear separability?



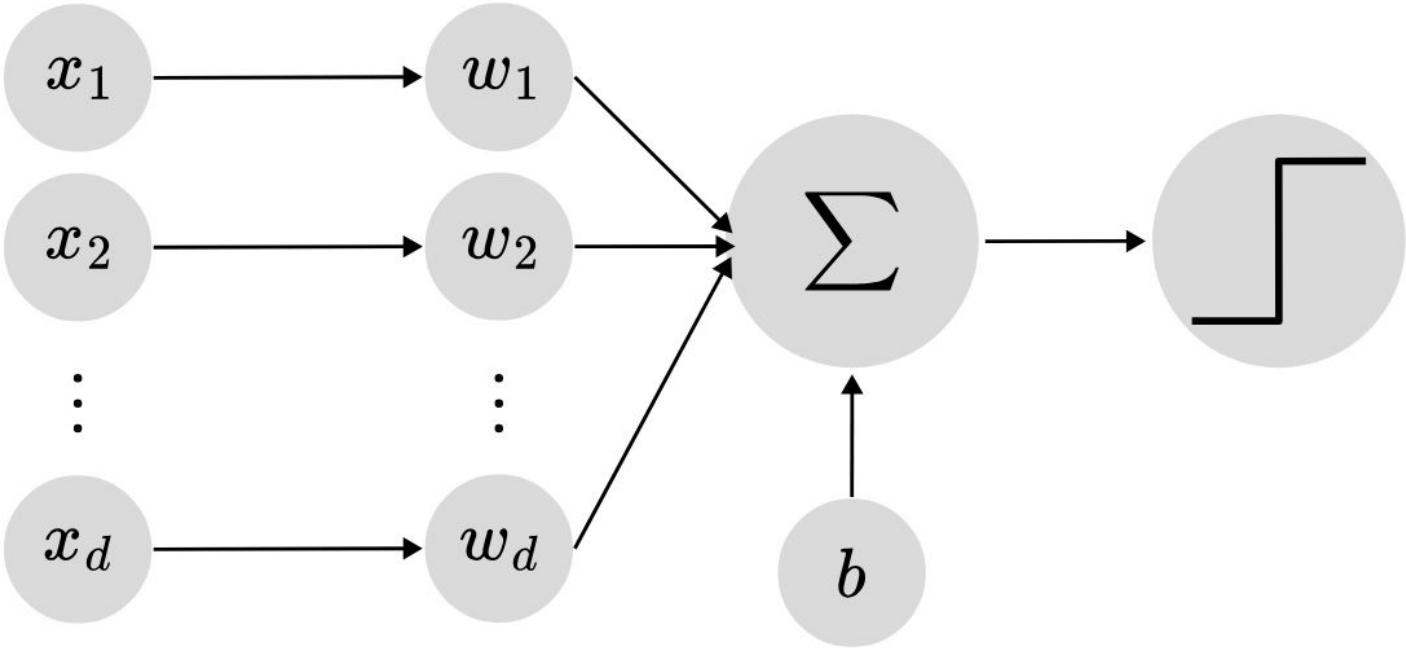
Linear separability and the margin, γ, of a dataset, D



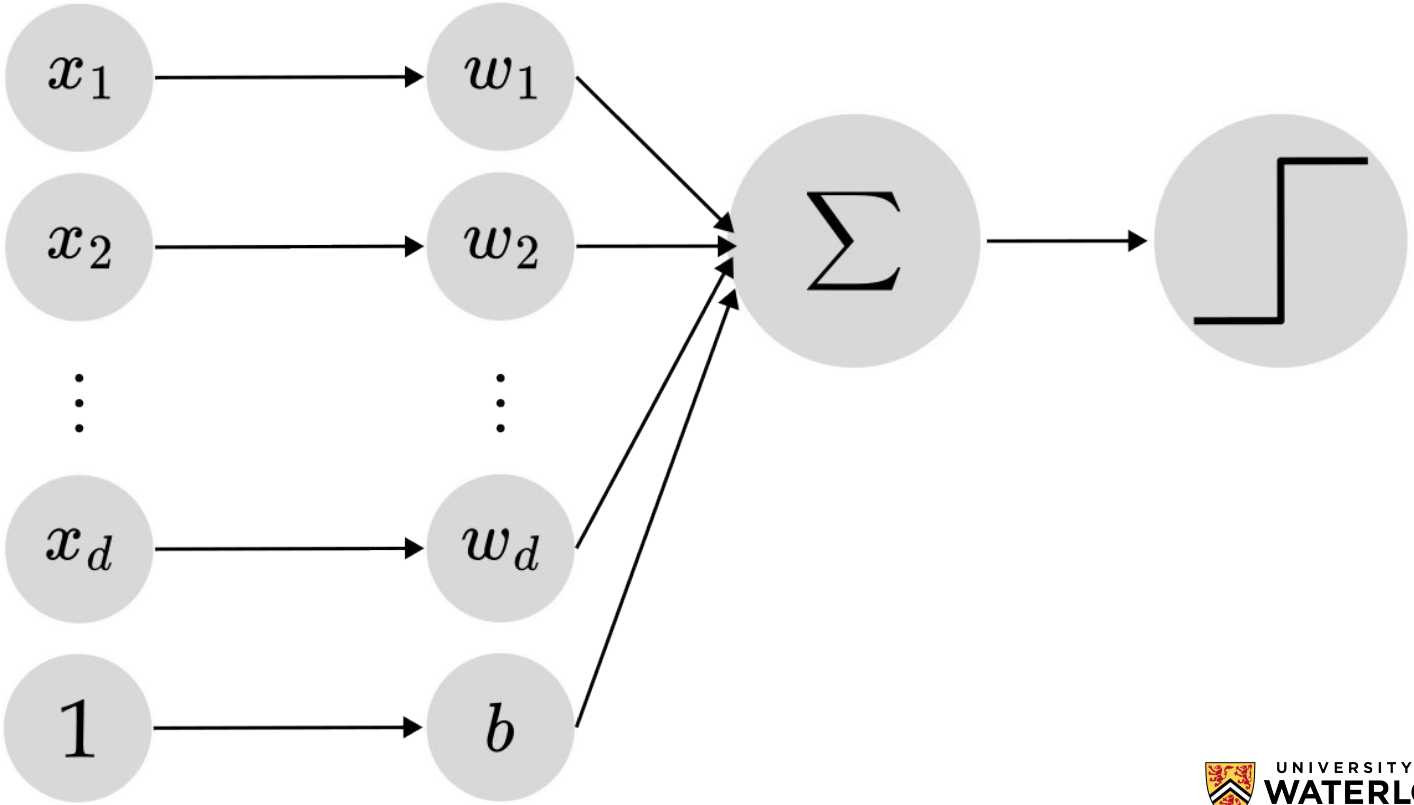
The padding trick simplifies analysis



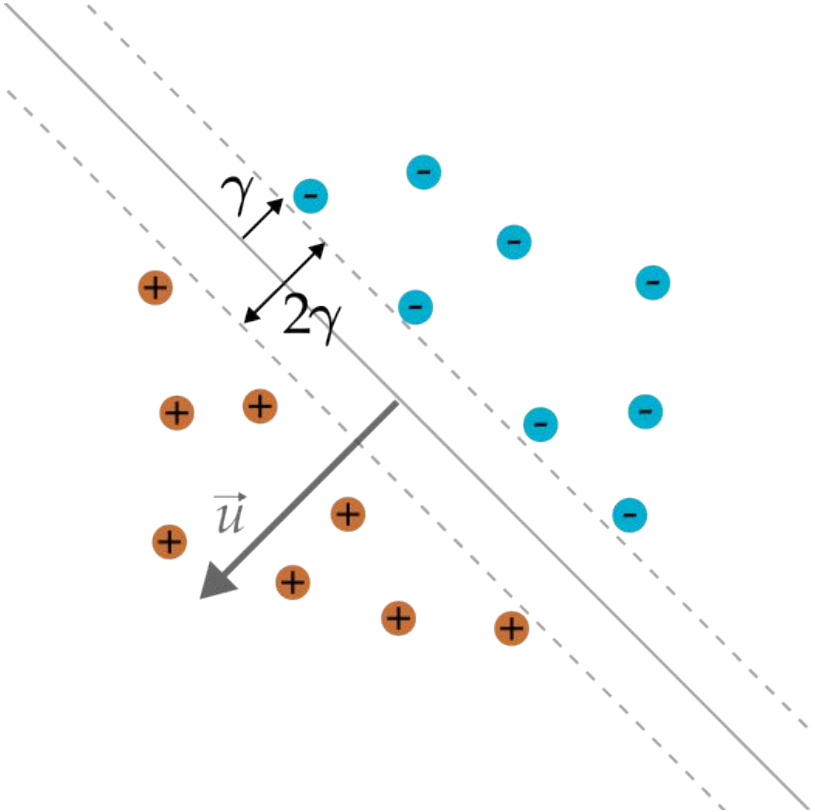
Biological interpretation



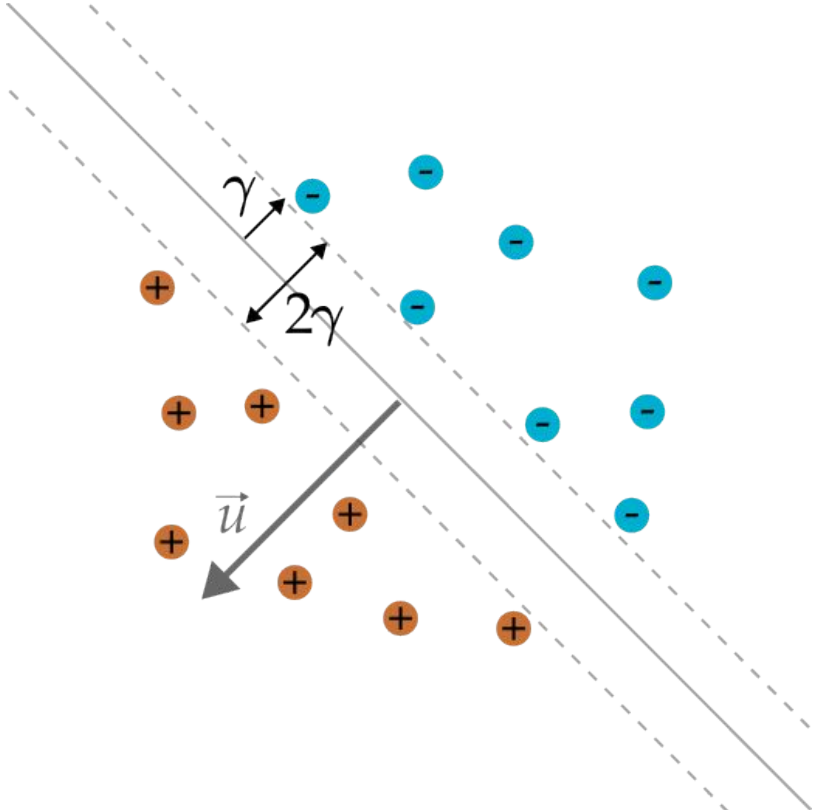
Biological interpretation with padding



The Oracle Vector

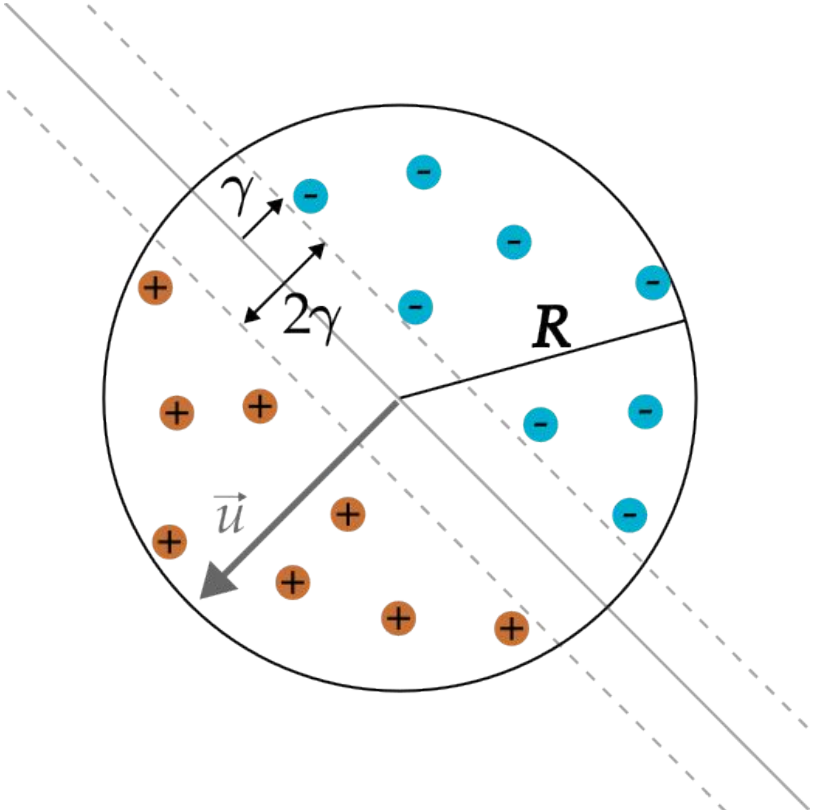


Linear separability and the margin, γ, of a dataset, D

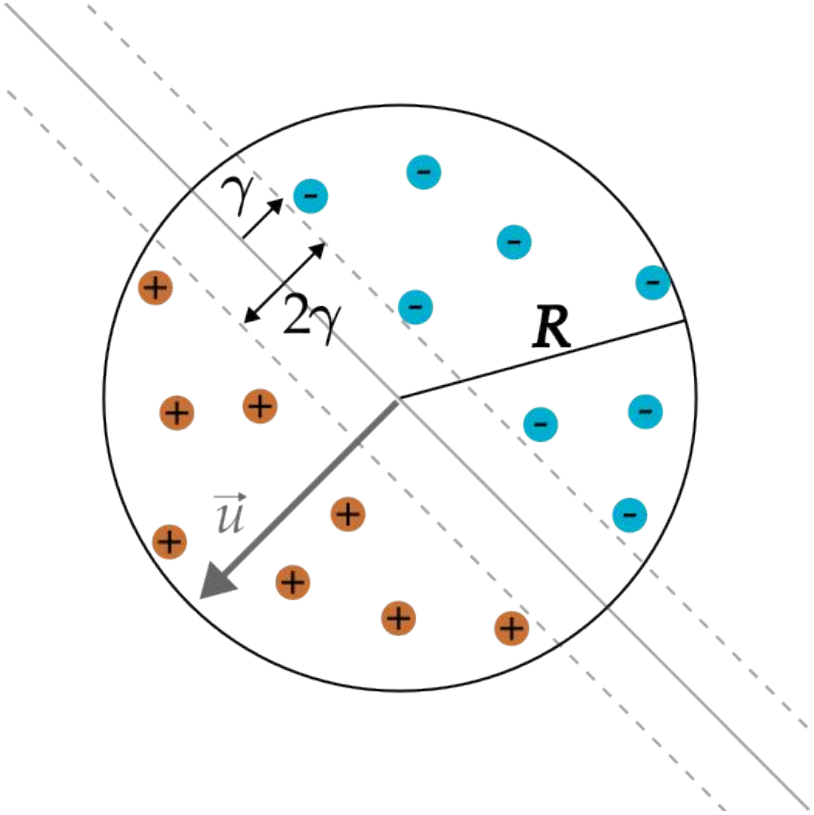


D is linearly separable if $\exists \vec{u}, \|\vec{u}\| = 1$,
and a margin γ, $\gamma > 0$,
s.t. $\forall (\vec{x}, y) \in D, y(\vec{u}^T x) > 0$

Finite number of errors on linearly separable data



Finite number of errors on linearly separable data



If D is linearly separable with margin $\gamma > 0$, and $\forall(\vec{x}, y) \in D, \|\vec{x}\| \leq R$, the perceptron algorithm will have converged by update $k \leq \frac{R^2}{\gamma^2}$

A proof in two parts (Novikoff, 1962)

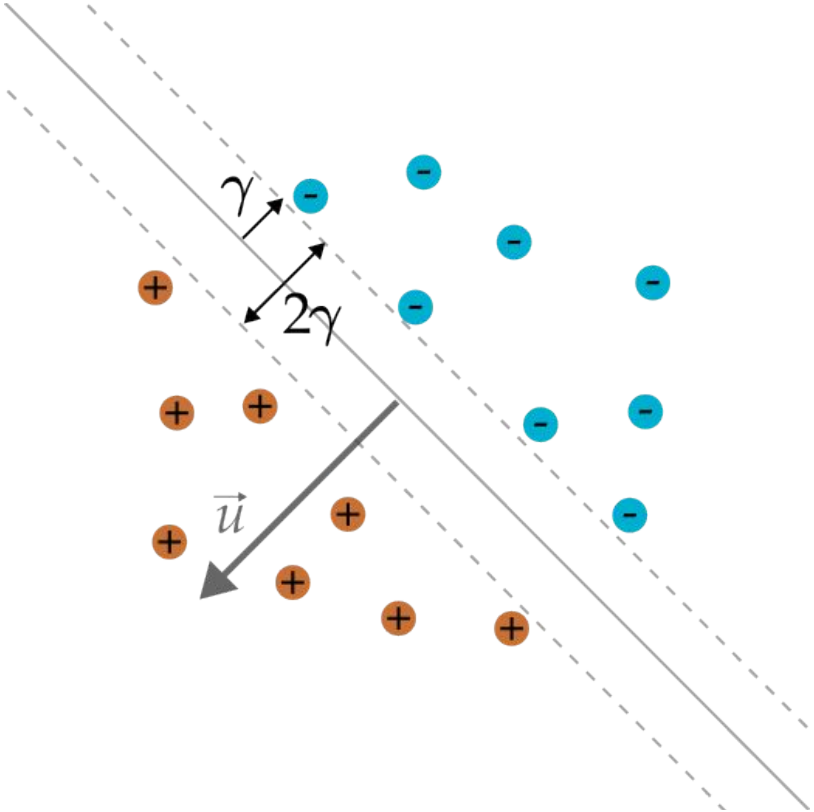
I. Do updates necessarily result in progress?
II. Will it ever stop?

A proof in two parts (Novikoff, 1962)

I. Do updates necessarily result in progress?

II. Will it ever stop?

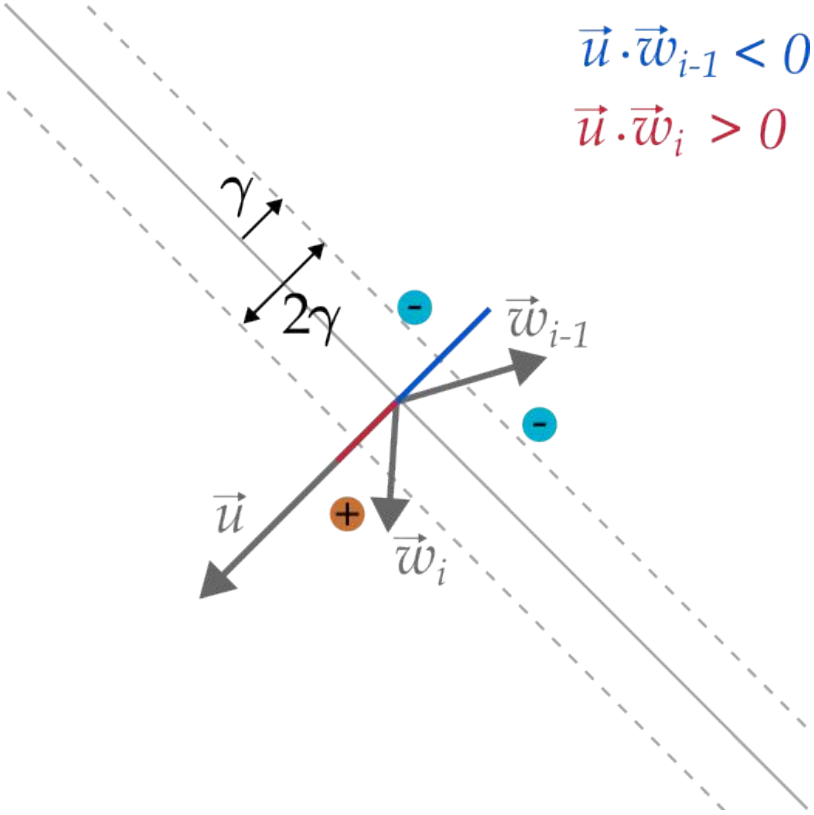
Do updates necessarily result in progress?



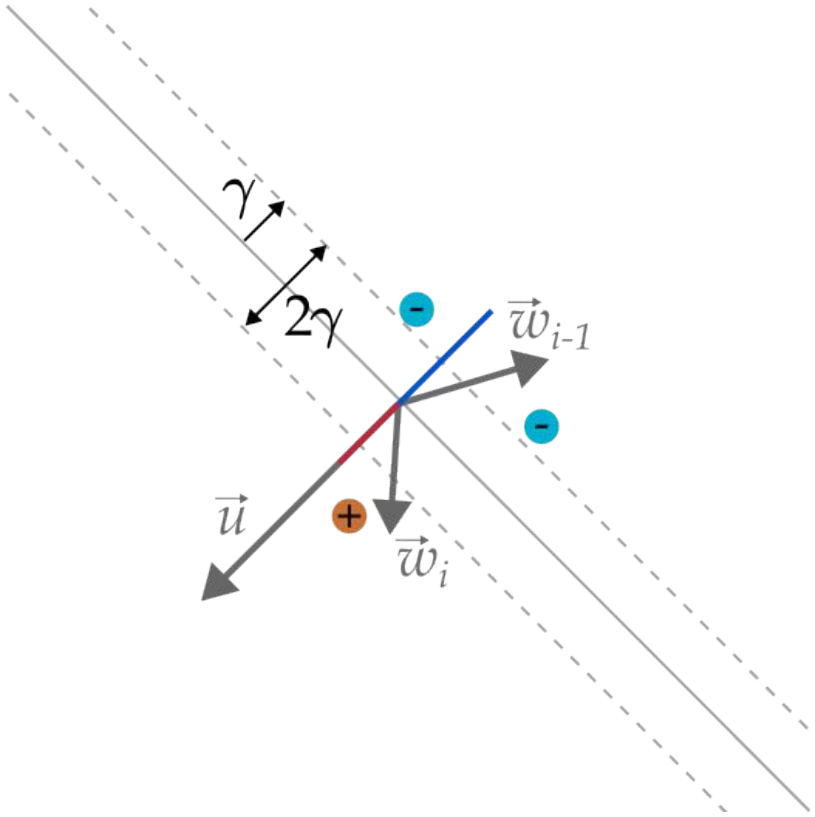
Do updates necessarily result in progress?

$$\vec{u} \cdot \vec{w}_{i-1} < 0$$

$$\vec{u} \cdot \vec{w}_i > 0$$



Do updates necessarily result in progress?



$$\vec{w}_0 = \vec{0}$$
$$\vec{w}_i = \vec{w}_{i-1} + y \cdot \vec{x}$$

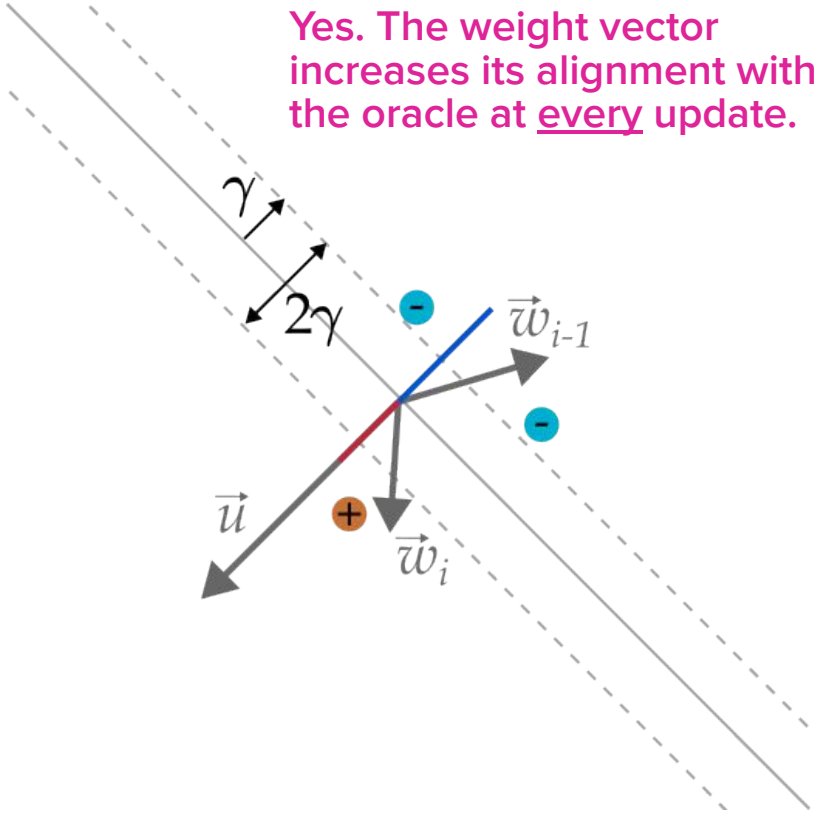
$$\begin{aligned}\vec{u} \cdot \vec{w}_i &= \vec{u} \cdot (\vec{w}_{i-1} + y \cdot \vec{x}) \\ &= \vec{u} \cdot \vec{w}_{i-1} + \vec{u} \cdot (y \cdot \vec{x}) \\ &= \vec{u} \cdot \vec{w}_{i-1} + y \cdot (\vec{u} \cdot \vec{x})\end{aligned}$$

$$y \cdot (\vec{u} \cdot \vec{x}) \geq \gamma$$

$$\vec{u} \cdot \vec{w}_i \geq \vec{u} \cdot \vec{w}_{i-1} + \gamma$$
$$\vec{u} \cdot w_k \geq k\gamma$$
$$\|\vec{w}_k\| \geq k\gamma$$

Do updates necessarily result in progress?

Yes. The weight vector increases its alignment with the oracle at <u>every</u> update.



$$\vec{w}_0 = \vec{0}$$
$$\vec{w}_i = \vec{w}_{i-1} + y \cdot \vec{x}$$

$$\begin{aligned}\vec{u} \cdot \vec{w}_i &= \vec{u} \cdot (\vec{w}_{i-1} + y \cdot \vec{x}) \\ &= \vec{u} \cdot \vec{w}_{i-1} + \vec{u} \cdot (y \cdot \vec{x}) \\ &= \vec{u} \cdot \vec{w}_{i-1} + y \cdot (\vec{u} \cdot \vec{x})\end{aligned}$$

$$y \cdot (\vec{u} \cdot \vec{x}) \geq \gamma$$

$$\vec{u} \cdot \vec{w}_i \geq \vec{u} \cdot \vec{w}_{i-1} + \gamma$$
$$\vec{u} \cdot w_k \geq k\gamma$$
$$\|\vec{w}_k\| \geq k\gamma$$

A proof in two parts (Novikoff, 1962)

I. Do updates necessarily result in progress?

Yes. The weight vector increases its alignment with the oracle at every update.

II. Will it ever stop?

A proof in two parts (Novikoff, 1962)

I. Do updates necessarily result in progress?

Yes. The weight vector increases its alignment with the oracle at every update.

II. Will it ever stop?

Will it ever stop?

Can now be interpreted as: “Is there an upper bound on the norm of the parameter vector?”

$$\vec{w}_i = \vec{w}_{i-1} + y \cdot \vec{x}$$

$$\|\vec{w}_i\| = \|\vec{w}_{i-1} + y \cdot \vec{x}\|$$

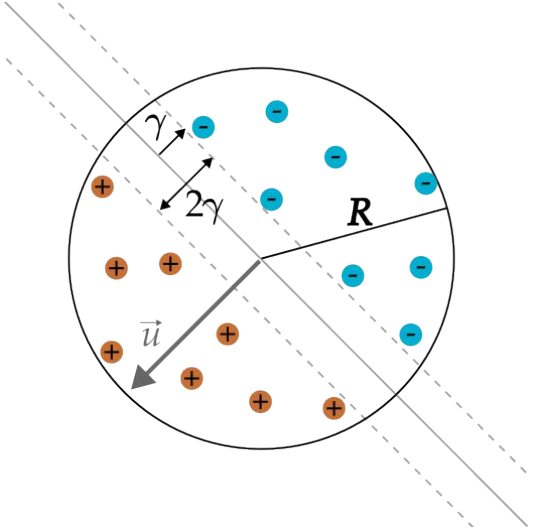
$$\|\vec{w}_i\|^2 = \|\vec{w}_{i-1} + y \cdot \vec{x}\|^2$$

$$= \|\vec{w}_{i-1}\|^2 + \boxed{\|y \cdot x\|^2 \leq R^2} + \boxed{2y(\vec{w}_{i-1} \cdot \vec{x}) \leq 0}$$

$$\|\vec{w}_i\|^2 \leq \|\vec{w}_{i-1}\|^2 + R^2$$

$$\|\vec{w}_k\|^2 \leq kR^2$$

$$\|\vec{a} + \vec{b}\|^2 = \|\vec{a}\|^2 + \|\vec{b}\|^2 + 2\vec{a} \cdot \vec{b}$$



Will it ever stop?

Upper bound: $\|\vec{w}_k\|^2 \leq kR^2$

Lower bound: $\|\vec{w}_k\| \geq k\gamma$

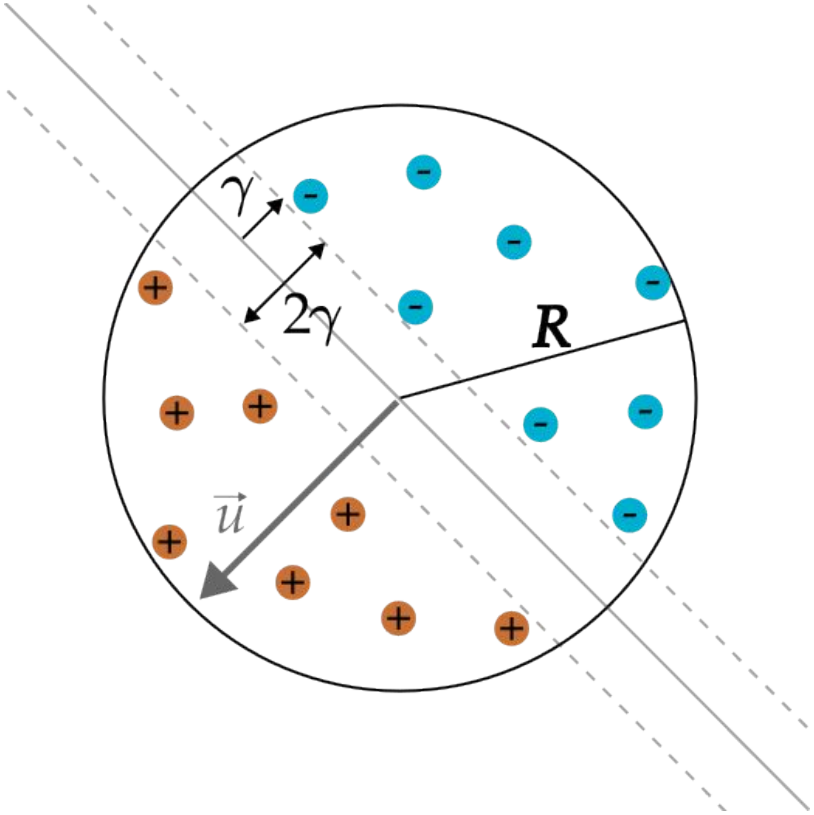
$$\|\vec{w}_k\|^2 \geq k^2\gamma^2$$

$$\implies k^2\gamma^2 \leq \|\vec{w}_k\|^2 \leq kR^2$$

$$\implies k^2\gamma^2 \leq kR^2$$

$$\implies k \leq \frac{R^2}{\gamma^2}$$

Finite number of errors on linearly separable data

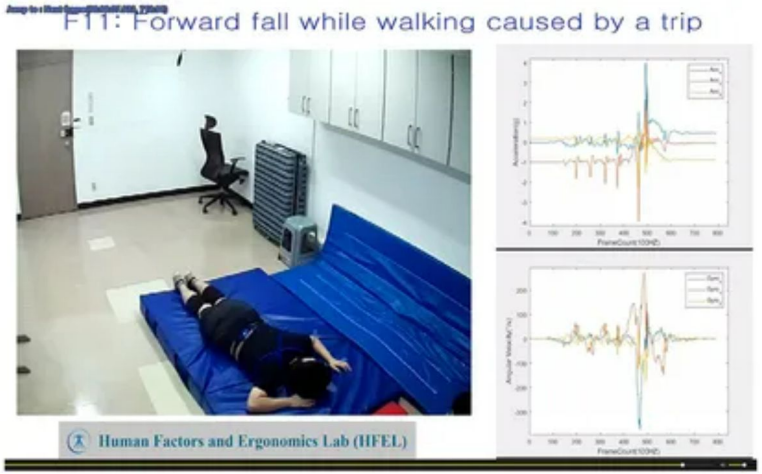


If D is linearly separable with margin $\gamma > 0$, and $\forall(\vec{x}, y) \in D, \|\vec{x}\| \leq R$, the perceptron algorithm will have converged by update $k \leq \frac{R^2}{\gamma^2}$

A few more properties

- The solution found by the Perceptron algorithm is not unique
  - There are infinitely many solutions
  - No guarantee to be the oracle associated with the margin of the dataset
- The Perceptron algorithm will not converge if data are not linearly separable
  - The algorithm will never halt, it will cycle
  - The algorithm is inappropriate for such problems
- Multiple valid termination conditions
  - Weights have stopped changing
  - Exhausted some update budget
  - Error on training or validation datasets has stopped
- There are different strategies for controlling the order of arrival of samples

Multiclass classification



Binary: Fall
Not Fall

Multiclass:

Activity
Stand for 30 s
Stand, slowly bendthe back with or without bending at knees, tie shoe lace, and get up
Pick up an object from the floor
Gently jump (try to reach an object)
Stand, sit to the ground, wait a moment, and get up with normal speed
Walk normally with turn (4 m)
Walk quickly with turn (4 m)
Jog normally with turn (4 m)
Jog quickly with turn (4 m)
Stumble while walking
Sit on a chair for 30 s
Sit on the sofa (back is inclined to the support) for 30 s
Sit down to a chair normally, and get up from a chair normally
Sit down to a chair quickly, and get up from a chair quickly
Sit a moment, trying to get up, and collapse into a chair
Stand, sit on the sofa (back is inclined to the support), and get up normally
Lie on the bed for 30 s
Sit a moment, lie down to the bed normally, and get up normally
Sit a moment, lie down to the bed quickly, and get up quickly
Walk upstairs and downstairs normally (five steps)
Walk upstairs and downstairs quickly (five steps)
Forward fall when trying to sit down
Backward fall when trying to sit down
Lateral fall when trying to sit down
Forward fall when trying to get up
Lateral fall when trying to get up
Forward fall while sitting, caused by fainting
Lateral fall while sitting, caused by fainting
Backward fall while sitting, caused by fainting
Vertical (forward) fall while walking caused by fainting
Fall while walking, use of hands to dampen fall, caused by fainting
Forward fall while walking caused by a trip
Forward fall while jogging caused by a trip
Forward fall while walking caused by a slip
Lateral fall while walking caused by a slip
Backward fall while walking caused by a slip

Yu, Xiaoqun, Jaehyuk Jang, and Shuping Xiong. *Frontiers in Aging Neuroscience* 13 (2021): 692865.

Learning a multiclass classifier with Perceptron

One vs all

- Train a classifier for each class
- Output: $\arg\max_i(w,x)$

One vs. one

- Train a classifier for each *pair* of classes
  - e.g. if 4 classes, 6 possible pairs
- Output: Majority vote

Lecture Outline

I. **What is needed in order to learn?**

The structure of observations and hypotheses

II. **How can we learn a hypothesis from data?**

The Perceptron Algorithm

III. **Why does this work?**

Convergence analysis and other properties

IV. **Summary + Housekeeping**

Aims

We should now be able to:

- ✓ Identify the components of a dataset required for supervised learning.
- ✓ Interpret the separating hyperplane hypothesis class geometrically.
- ✓ Implement the Perceptron algorithm and list its properties.
- ✓ Reproduce Novikoff's proof of the Perceptron convergence theorem.

Lecture	Date	Topics
0	05/09/2024	Introduction + Administrative Remarks
1	10/09/2024	Halfspaces the Perceptron Algorithm
2	12/09/2024	Linear Regression and Convexity
3	17/09/2024	Maximum Likelihood Estimation
4	19/09/2024	k-means Clustering
5	24/09/2024	k-NN Classification and Logistic Regression
6	26/09/2024	Hard-margin SVM
7	01/10/2024	Soft-margin SVM
8	03/10/2024	Kernel methods
9	08/10/2024	Decision Trees
10	10/10/2024	Bagging and Boosting
	15/10/2024	NO LECTURE - MIDTERM BREAK
	17/10/2024	NO LECTURE- MIDTERM BREAK
11	22/10/2024	Expectation Maximization Algorithm
12	24/10/2024	MLPs and Fully-Connected NNs
	29/10/2024	NO LECTURE - MIDTERM EXAM
13	31/10/2024	Convolutional Neural Networks
14	05/11/2024	Recurrent Neural Networks
15	07/11/2024	Attention and Transformers
16	12/11/2024	Graph Neural Networks (Time permitting)
17	14/11/2024	VAEs and GANs
18	19/11/2024	Flows
19	21/11/2024	Contrastive Learning (Time permitting)
20	26/11/2024	Robustness
21	28/11/2024	Privacy (Saber Malekmohammadi)
22	03/12/2024	Fairness

On the horizon

Table 2: Grading Scheme

	Assessment	Assessment Date	Weighting (CS480)	Weighting (CS680)
posted →	Assignment 1	September 27	7.5%	7.5%
	Assignment 2	October 14	7.5%	7.5%
	Assignment 3	November 8	7.5%	7.5%
	Assignment 4	November 22	7.5%	7.5%
	Exams			
	Midterm	October 29	30%	15%
	Final	TBD	40%	30%
examples are up →	Project (CS 680 only)			
	Pitch	September 19	N/A	2%
	Proposal	October 8	N/A	8%
	Report	December 3	N/A	15%
	Total		100%	100%