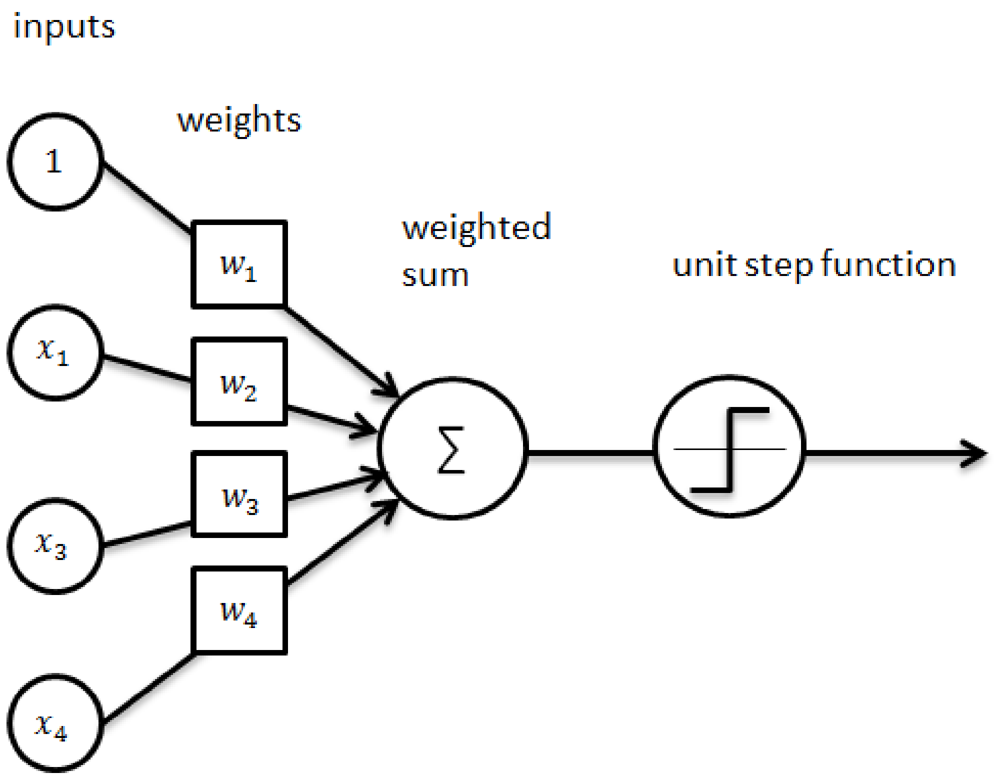
Introduction to Computer Vision

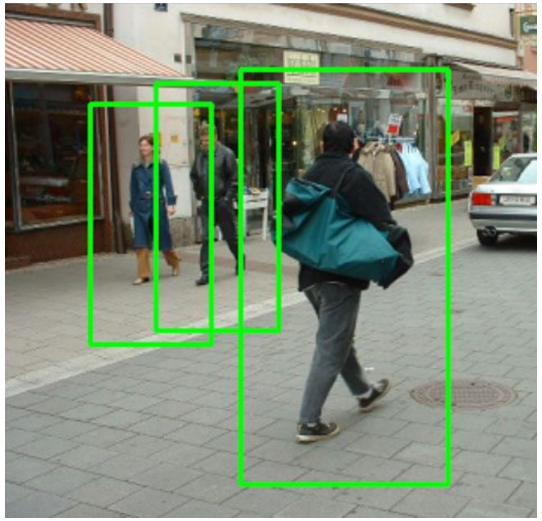
Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 18

Introduction to Neural Networks



Hand-Engineered Methods



- Image formation (+database+labels) — Captured+manual.
- Filtering (gradients/transforms) — Hand designed.
- Feature points (saliency+description) — Hand designed.
- Dictionary building (compression/quantization) — Hand designed.
- Classifier (decision making) — Learned.

Recognition: Classification | Object Detection | Segmentation

Learning-Based Methods

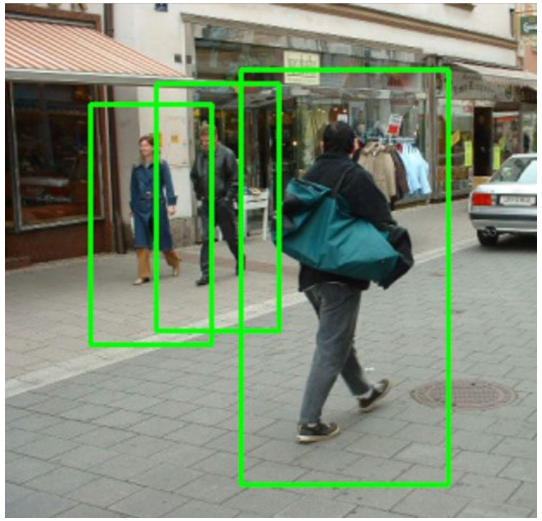


Image formation (+database+labels) — Captured+manual.

Filtering (gradients/transforms) — Learned.

Feature points (saliency+description) — Learned.

Dictionary building (compression/quantization) — Learned.

Classifier (decision making) — Learned.

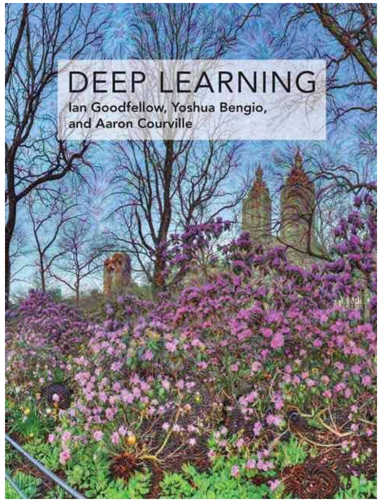
End to end learning

Recognition: Classification | Object Detection | Segmentation

Deep learning

- Modeling the visual world is incredibly complicated. We need high-capacity models.
- In the past, we didn't have enough data to fit these models. But now we do!
- We want a class of **high-capacity models** that are **easy to optimize:**

Deep neural networks!



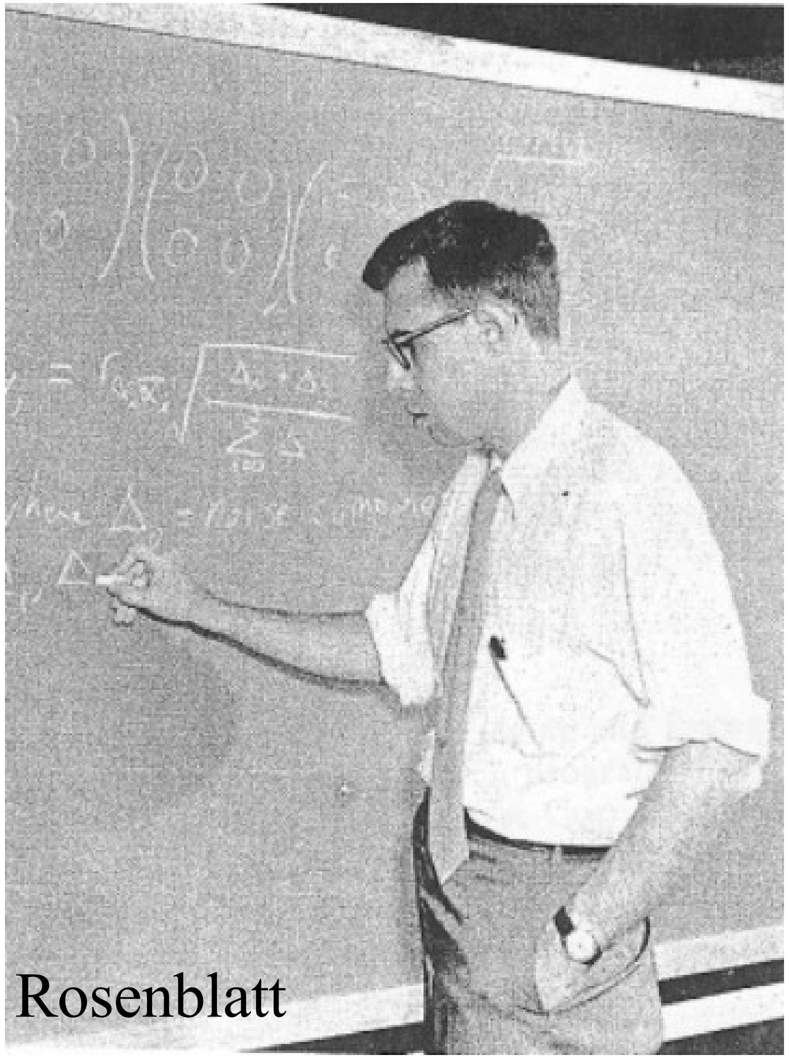
http://www.deeplearningbook.org/

By Ian Goodfellow, Yoshua Bengio and Aaron Courville

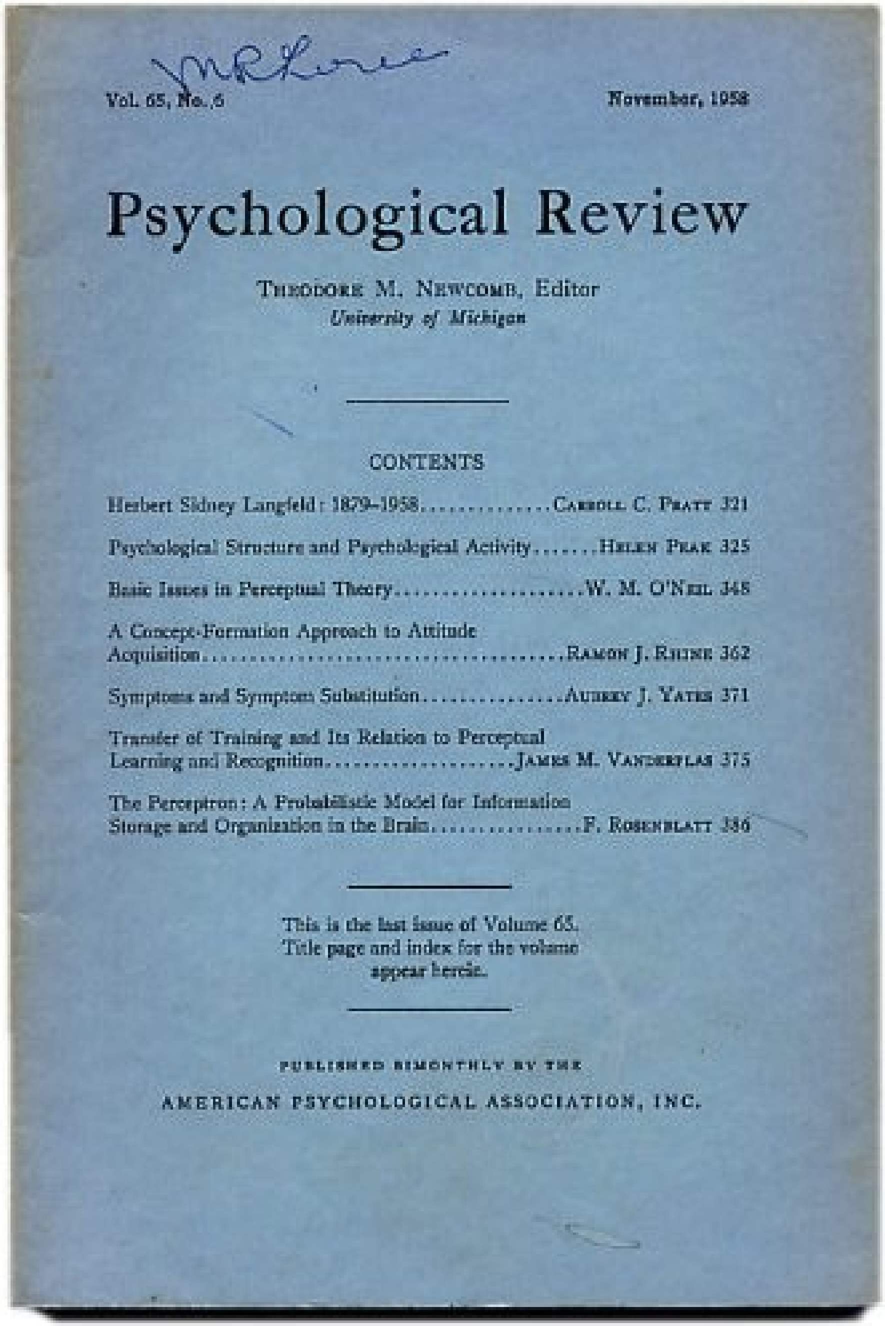
November 2016

A brief history of Neural Networks

Perceptrons, 1958



[http://www.ecse.rpi.edu/homepages/nagy/PDF_chrono/2011_Nagy_Pace_FR.pdf.](http://www.ecse.rpi.edu/homepages/nagy/PDF_chrono/2011_Nagy_Pace_FR.pdf) Photo by George Nagy



Vol. 65, No. 6 November, 1958

Psychological Review

THEODORE M. NEWCOMB, Editor
University of Michigan

CONTENTS

Herbert Sidney Langfeld: 1879–1958.............CARROLL C. PRATT 321

Psychological Structure and Psychological Activity.......HELEN PEAK 325

Basic Issues in Perceptual Theory..................W. M. O'NEIL 348

A Concept-Formation Approach to Attitude Acquisition....................................RAMON J. RHINE 362

Symptoms and Symptom Substitution..............AUBREY J. YATES 371

Transfer of Training and Its Relation to Perceptual Learning and Recognition..................JAMES M. VANDERPLAS 375

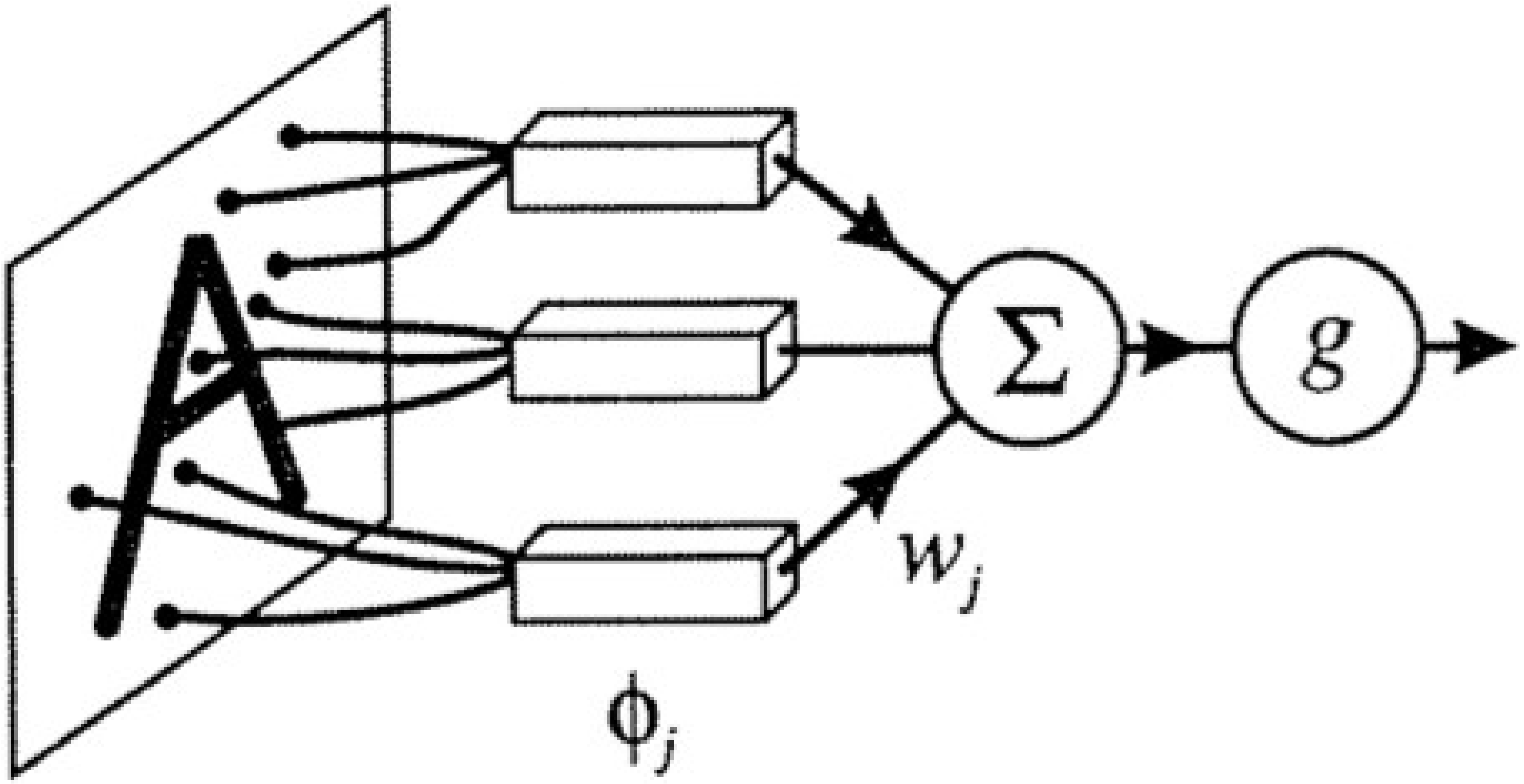
The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain...............F. ROSENBLATT 386

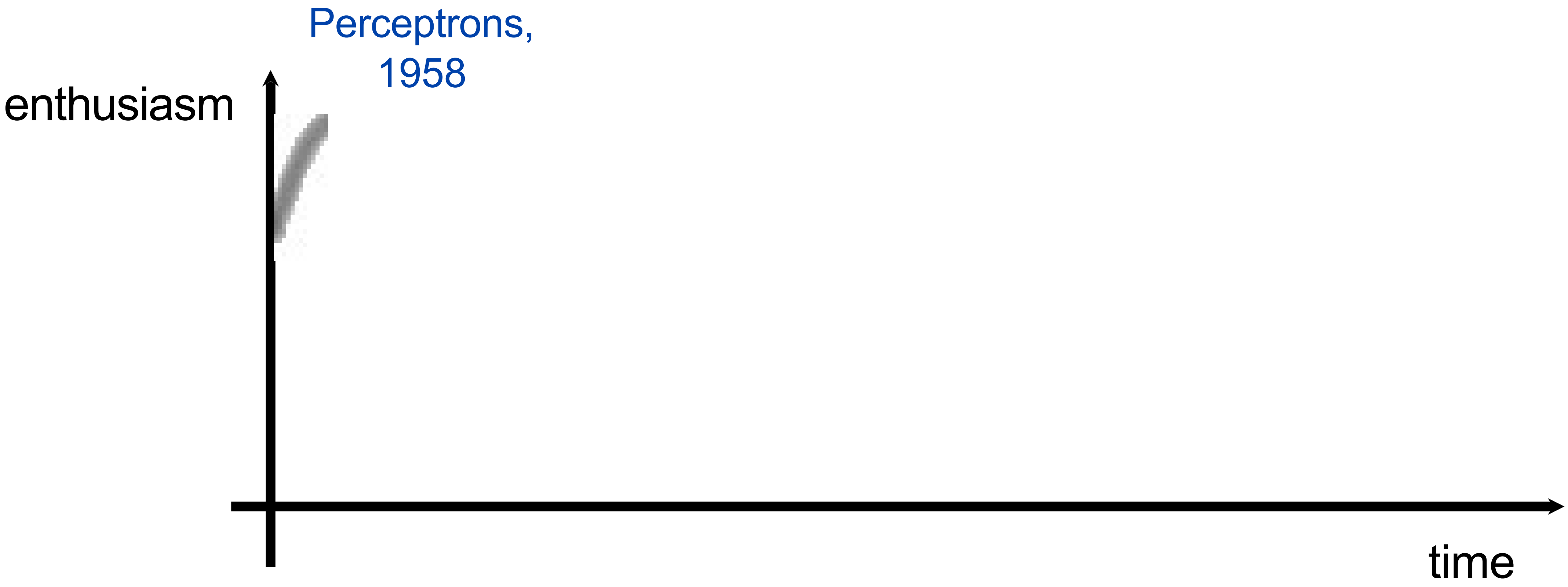
This is the last issue of Volume 65.
Title page and index for the volume
appear herein.

PUBLISHED BIMONTHLY BY THE
AMERICAN PSYCHOLOGICAL ASSOCIATION, INC.

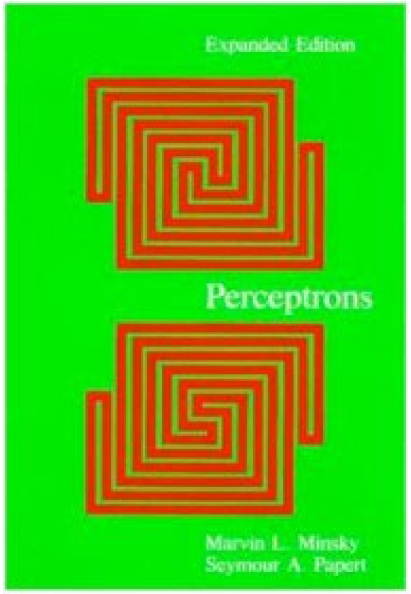
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf

Perceptrons, 1958




Perceptrons,
1958
enthusiasm
time

Minsky and Papert, Perceptrons, 1972



Perceptrons, expanded edition

An Introduction to Computational Geometry

By Marvin Minsky and Seymour A. Papert

Overview

Perceptrons - the first systematic study of parallelism in computation - has remained a classical work on threshold automata networks for nearly two decades. It marked a historical turn in artificial intelligence, and it is required reading for anyone who wants to understand the connectionist counterrevolution that is going on today.

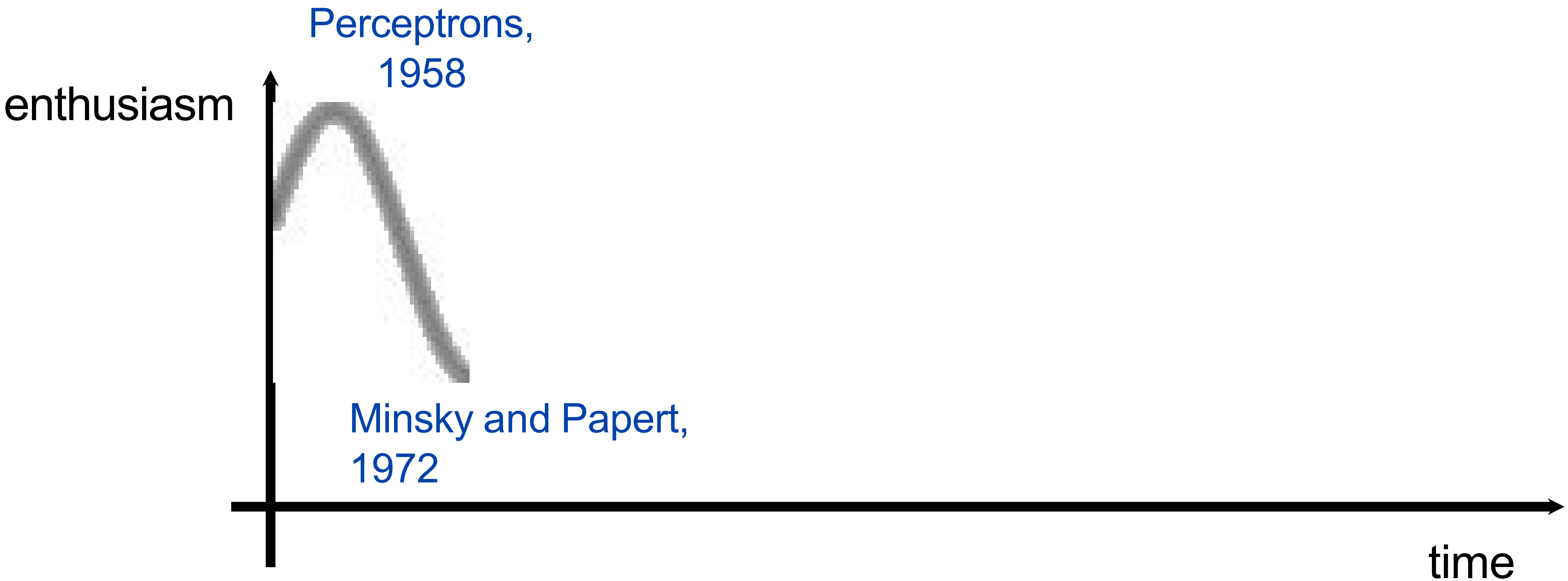
Artificial-intelligence research, which for a time concentrated on the programming of ton Neumann computers, is swinging back to the idea that intelligence might emerge from the activity of networks of neuronlike entities. Minsky and Papert's book was the first example of a mathematical analysis carried far enough to show the exact limitations of a class of computing machines that could seriously be considered as models of the brain. Now the new developments in mathematical tools, the recent interest of physicists in the theory of disordered matter, the new insights into and psychological models of how the brain works, and the evolution of fast computers that can simulate networks of automata have given *Perceptrons* new importance.

Witnessing the swing of the intellectual pendulum, Minsky and Papert have added a new chapter in which they discuss the current state of parallel computers, review developments since the appearance of the 1972 edition, and identify new research directions related to connectionism. They note a central theoretical challenge facing connectionism: the challenge to reach a deeper understanding of how "objects" or "agents" with individuality can emerge in a network. Progress in this area would link connectionism with what the authors have called "society theories of mind."

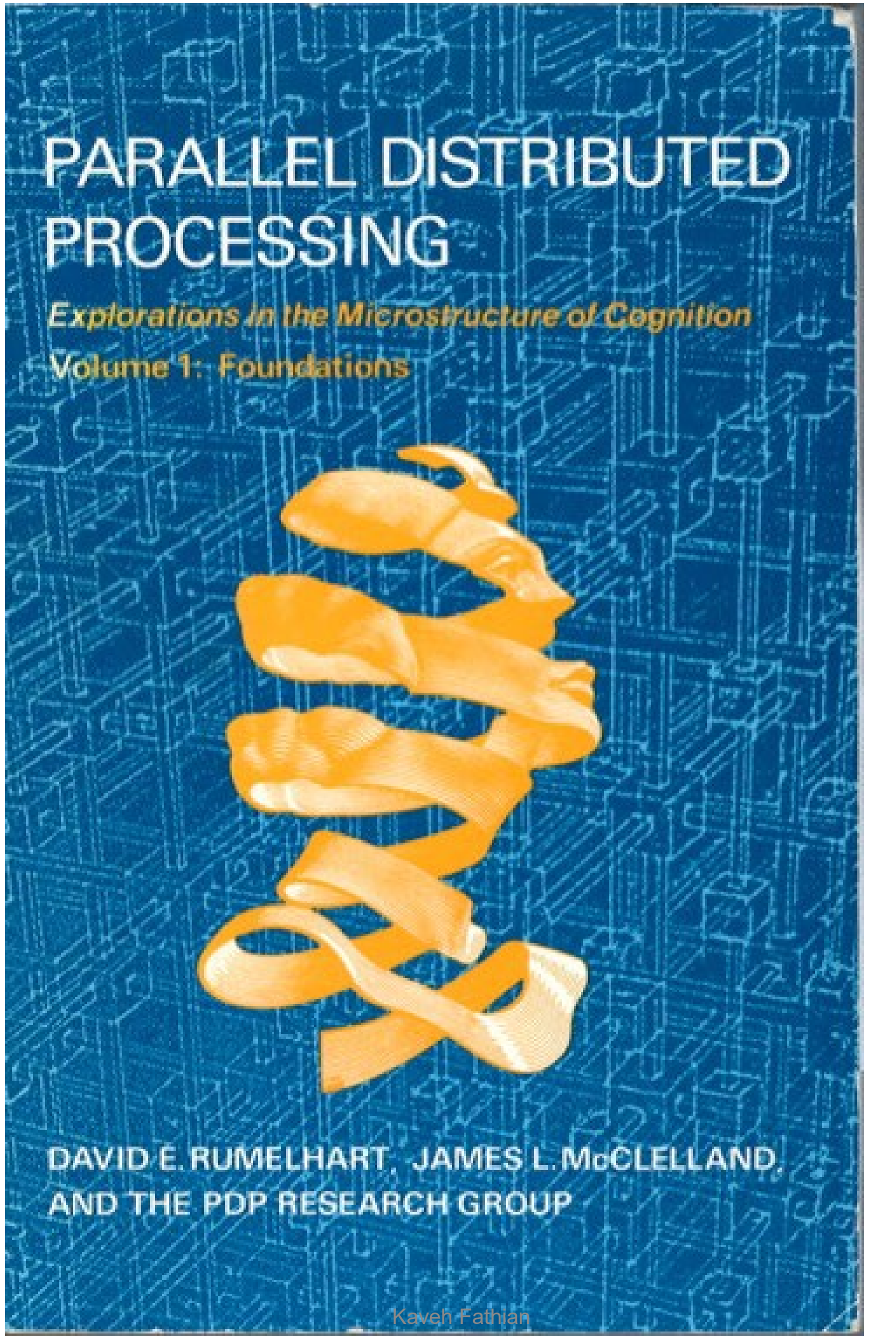
FOR BUYING OPTIONS, START HERE

Select Shipping Destination

Paperback | $35.00 Short | £24.95 |
ISBN: 9780262631112 | 308 pp. | 6 x 8.9 in | December 1987

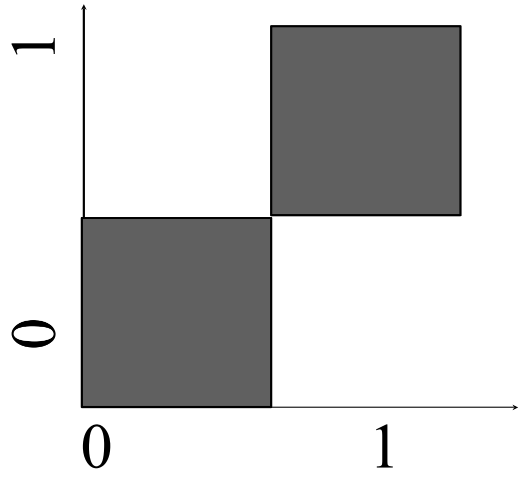

Perceptrons,
1958
enthusiasm
Minsky and Papert,
1972
time

Parallel Distributed Processing (PDP), 1986

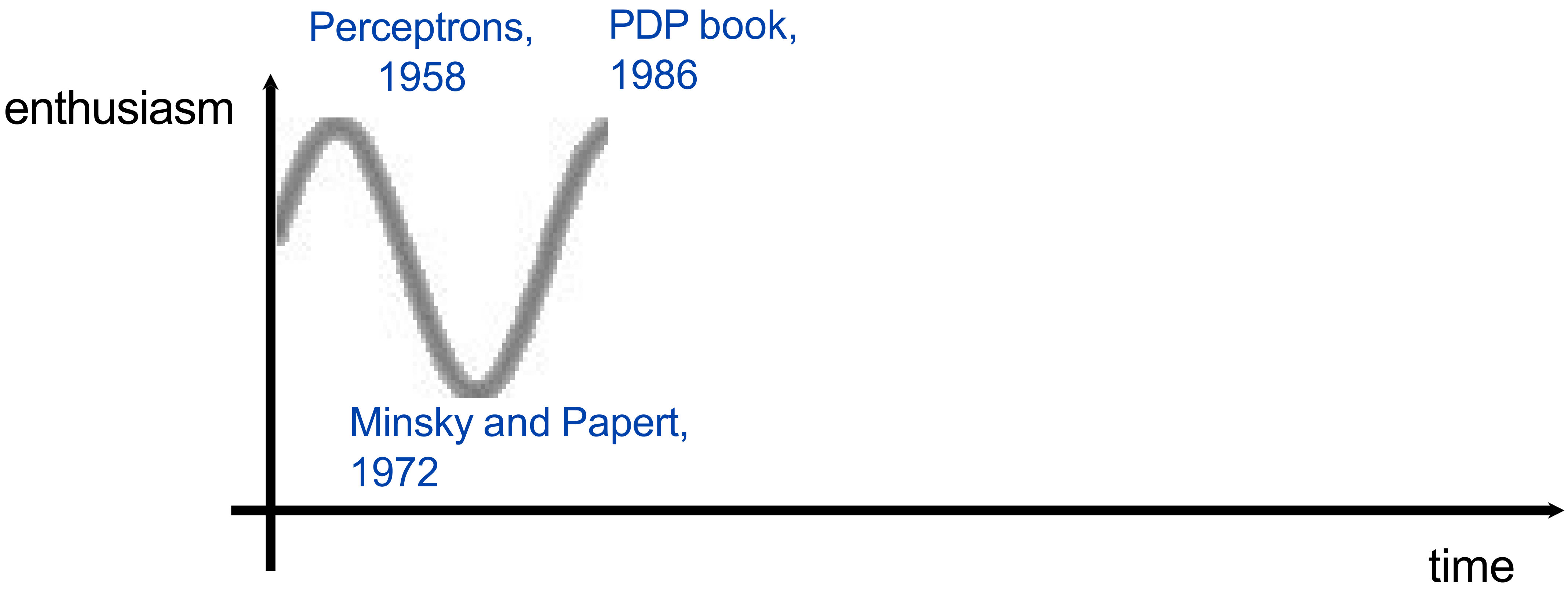


XOR problem

Inputs		Output
0	0	0
1	0	1
0	1	1
1	1	0



- PDP authors pointed to the backpropagation algorithm as a breakthrough.
- This allowed multi-layer neural networks to be trained.
- Functions that a multi-layer network can represent but a single-layer network cannot: XOR function.


Perceptrons,
1958
PDP book,
1986
enthusiasm
Minsky and Papert,
1972
time

LeCun conv nets, 1998

PROC. OF THE IEEE, NOVEMBER 1998 7

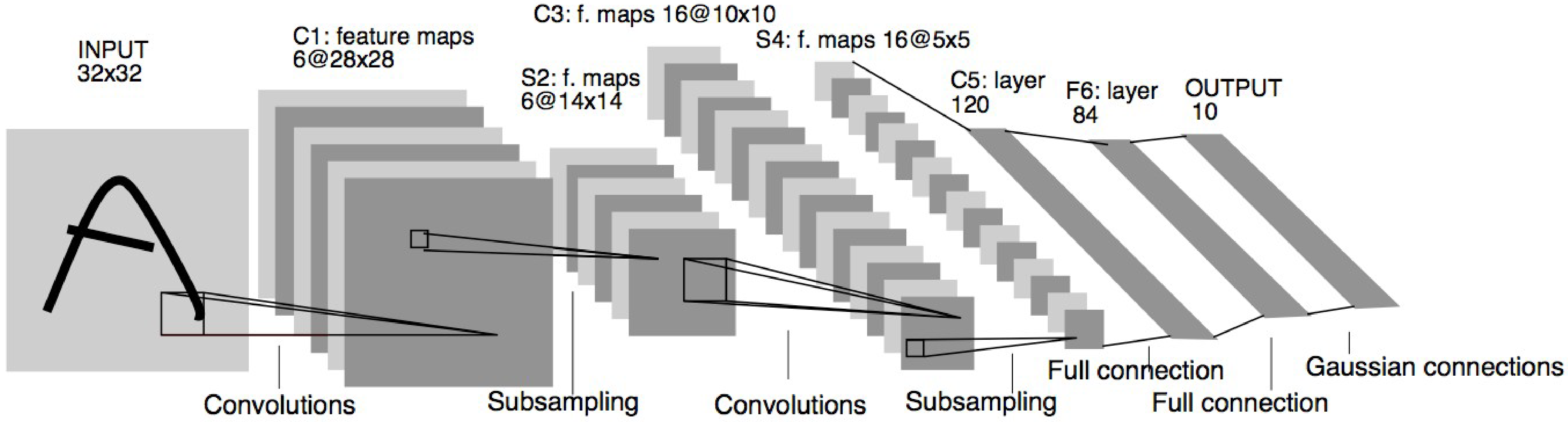


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Demos: http://yann.lecun.com/exdb/lenet/index.html

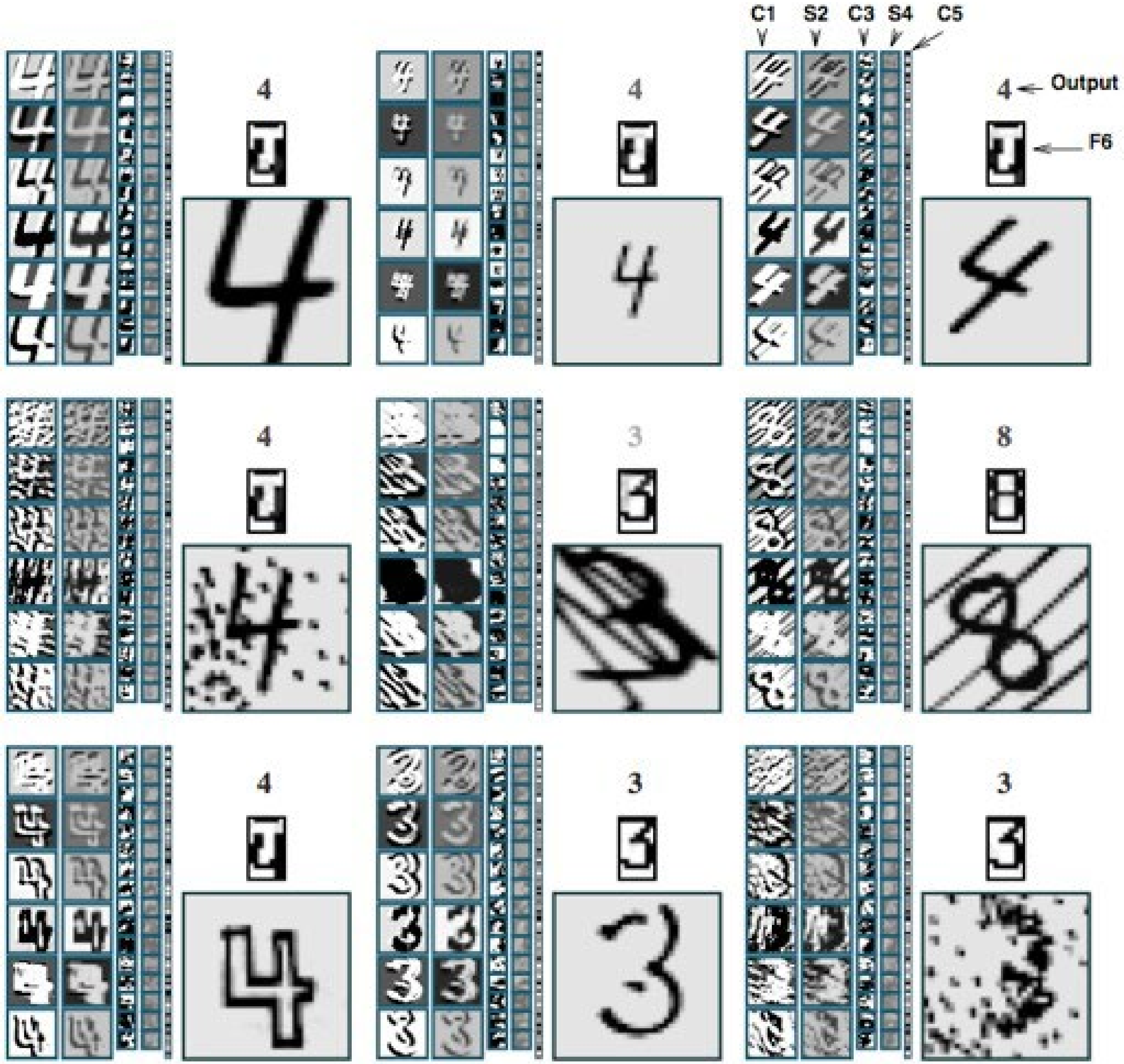
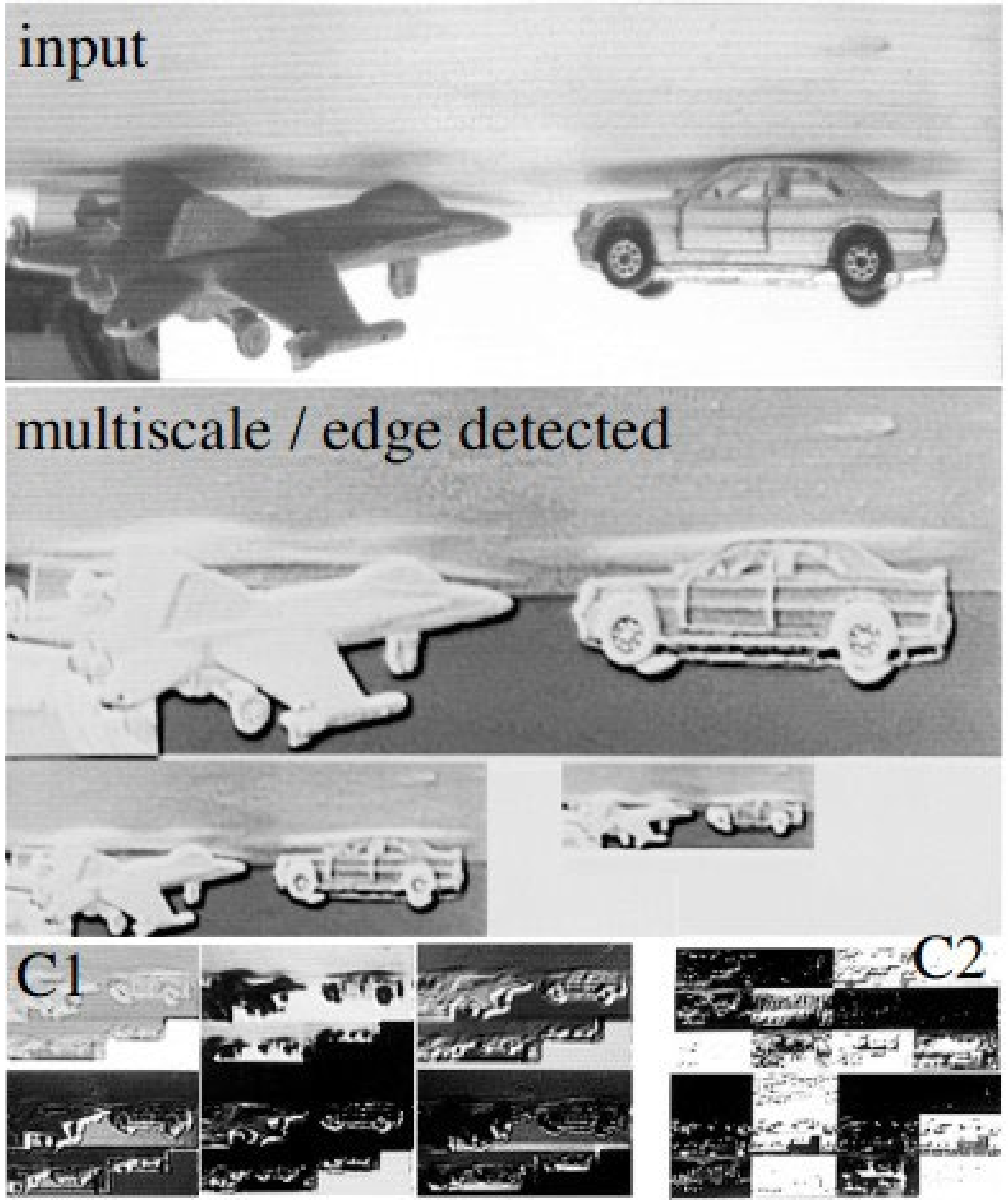


Fig. 13. Examples of unusual, distorted, and noisy characters correctly recognized by LeNet-5. The grey-level of the output label represents the penalty (lighter for higher penalties).



http://pub.clement.farabet.net/ecvw09.pdf

Neural networks to recognize handwritten digits?
yes

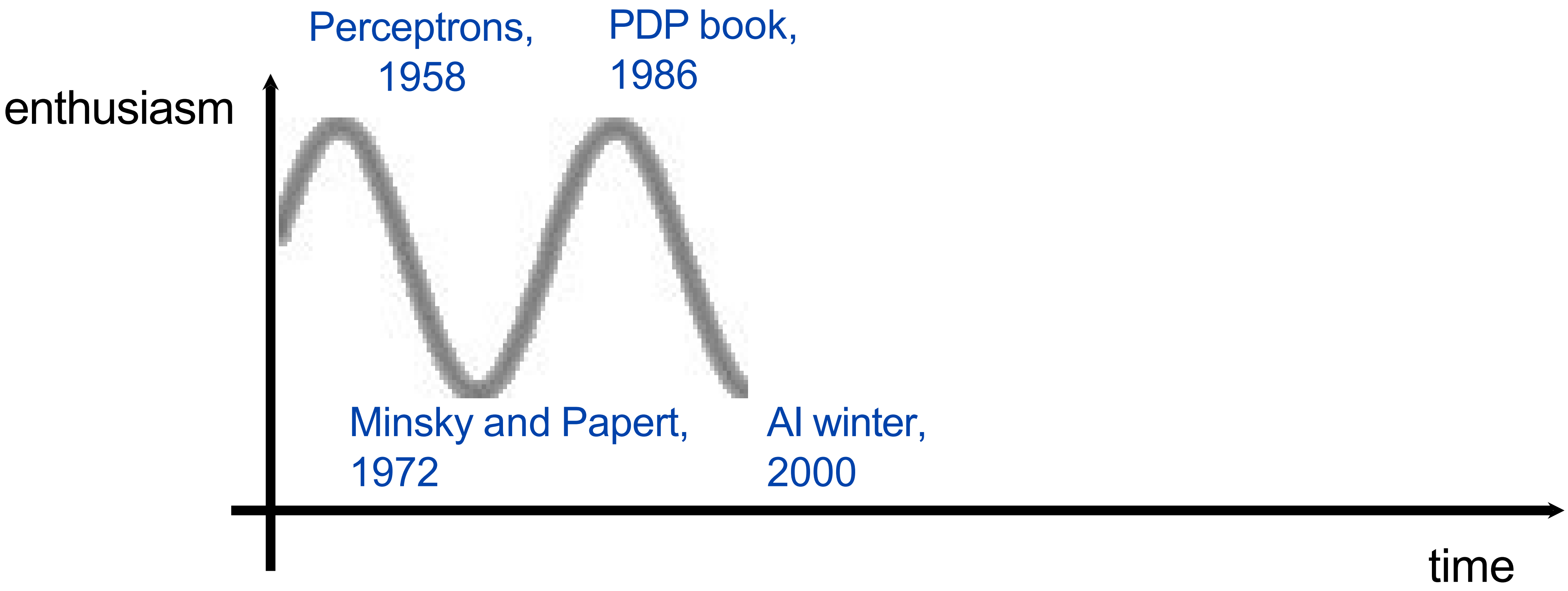
Neural networks for tougher problems?
not really

Neural Information Processing Systems 2000

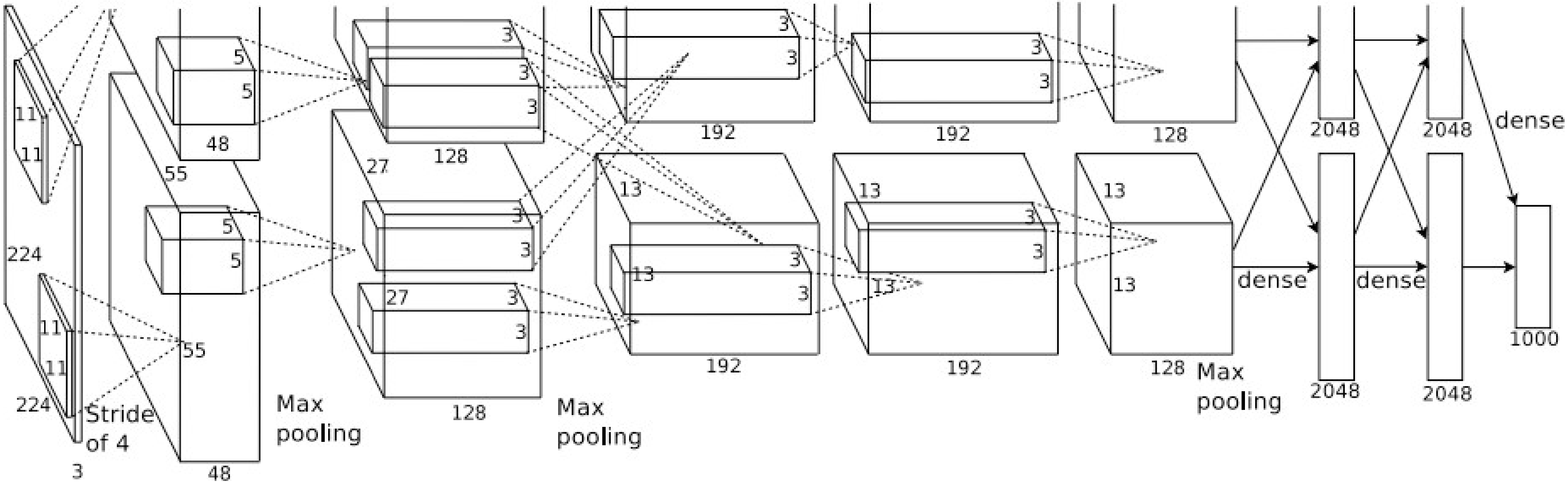
- Neural Information Processing Systems (NeurIPS), is the premier conference on machine learning.

- Evolved from an interdisciplinary conference to a machine learning conference.

- For the 2000 conference:
  - <u>Title words predictive of paper acceptance</u>: “Belief Propagation” and “Gaussian”.
  - <u>Title words predictive of paper rejection</u>: “Neural” and “Network”.


Perceptrons,
1958
PDP book,
1986
enthusiasm
Minsky and Papert,
1972
AI winter,
2000
time

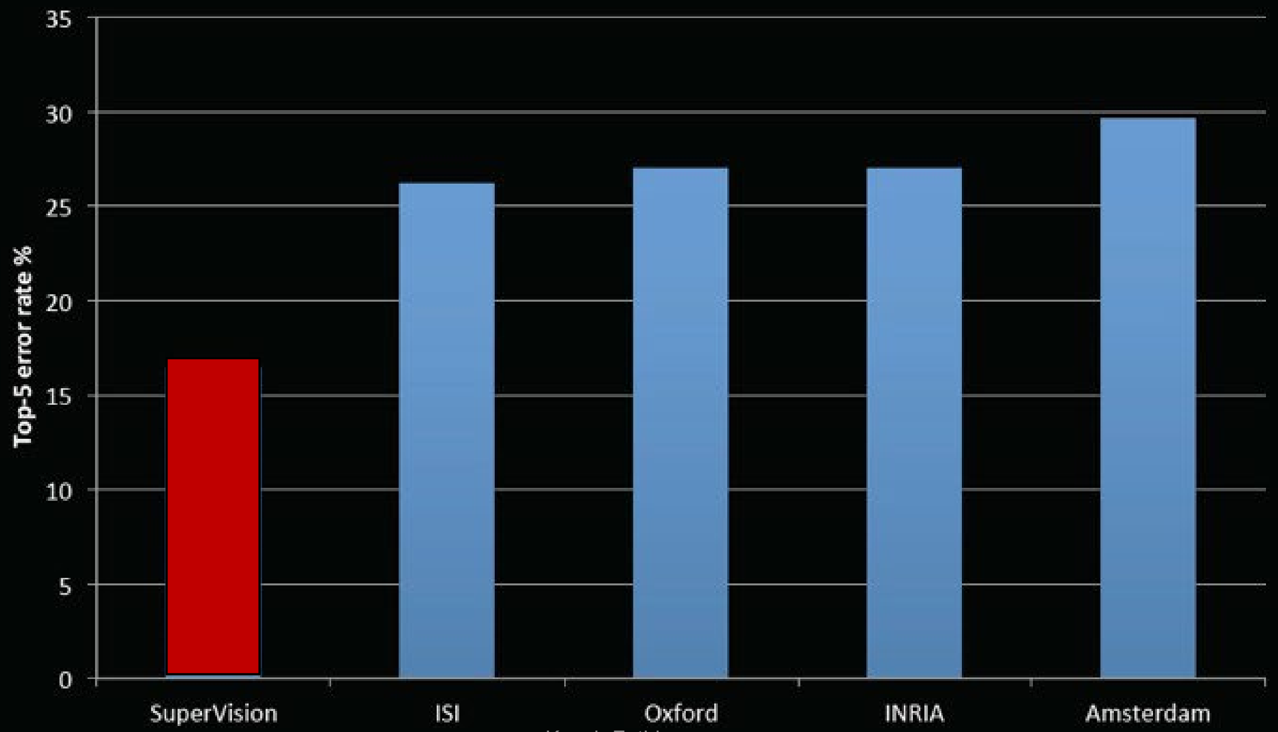
Krizhevsky, Sutskever, & Hinton, NeurIPS2012 “Alexnet”



Krizhevsky, Sutskever, & Hinton, NeurIPS 2012

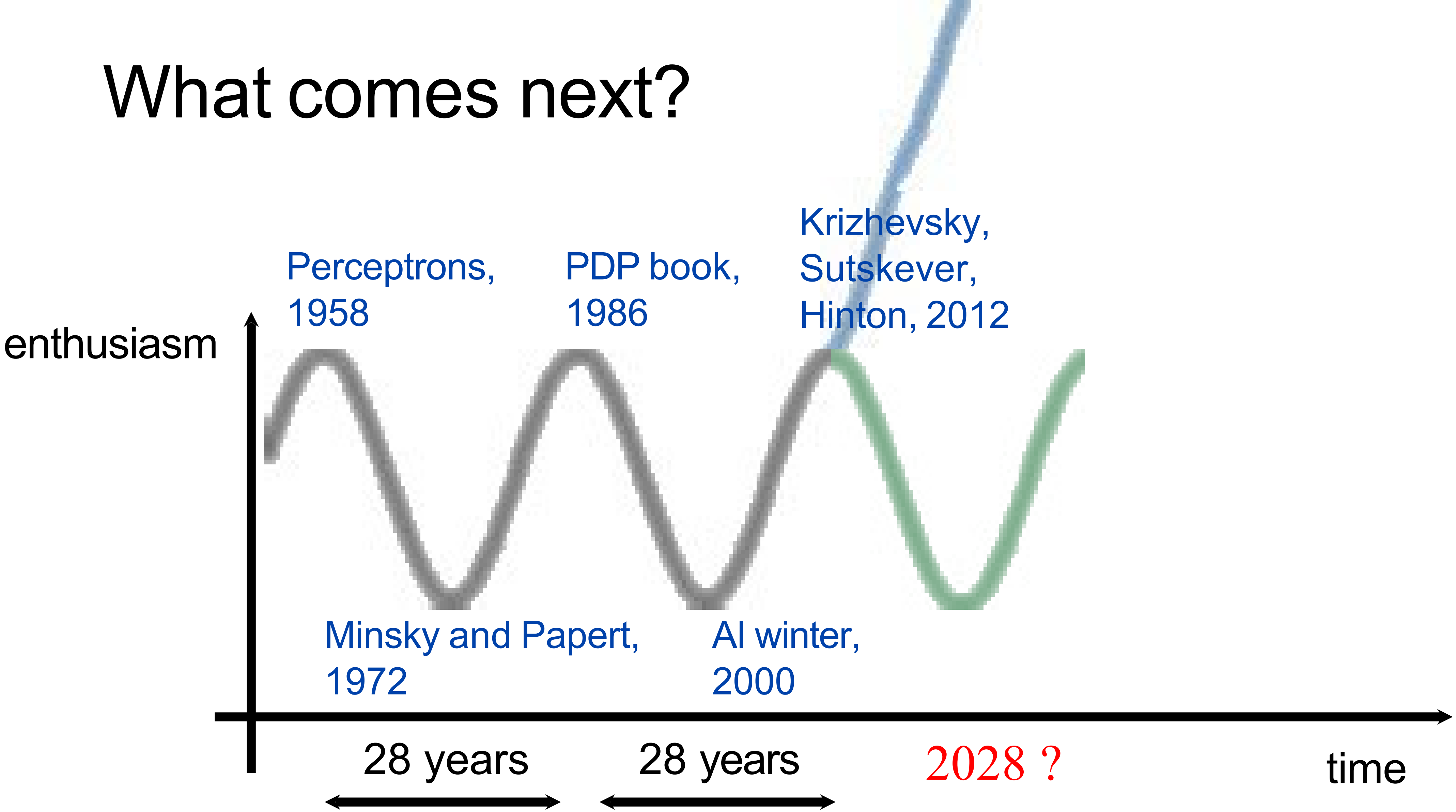
ImageNet Classification 2012

- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) – 26.2% error

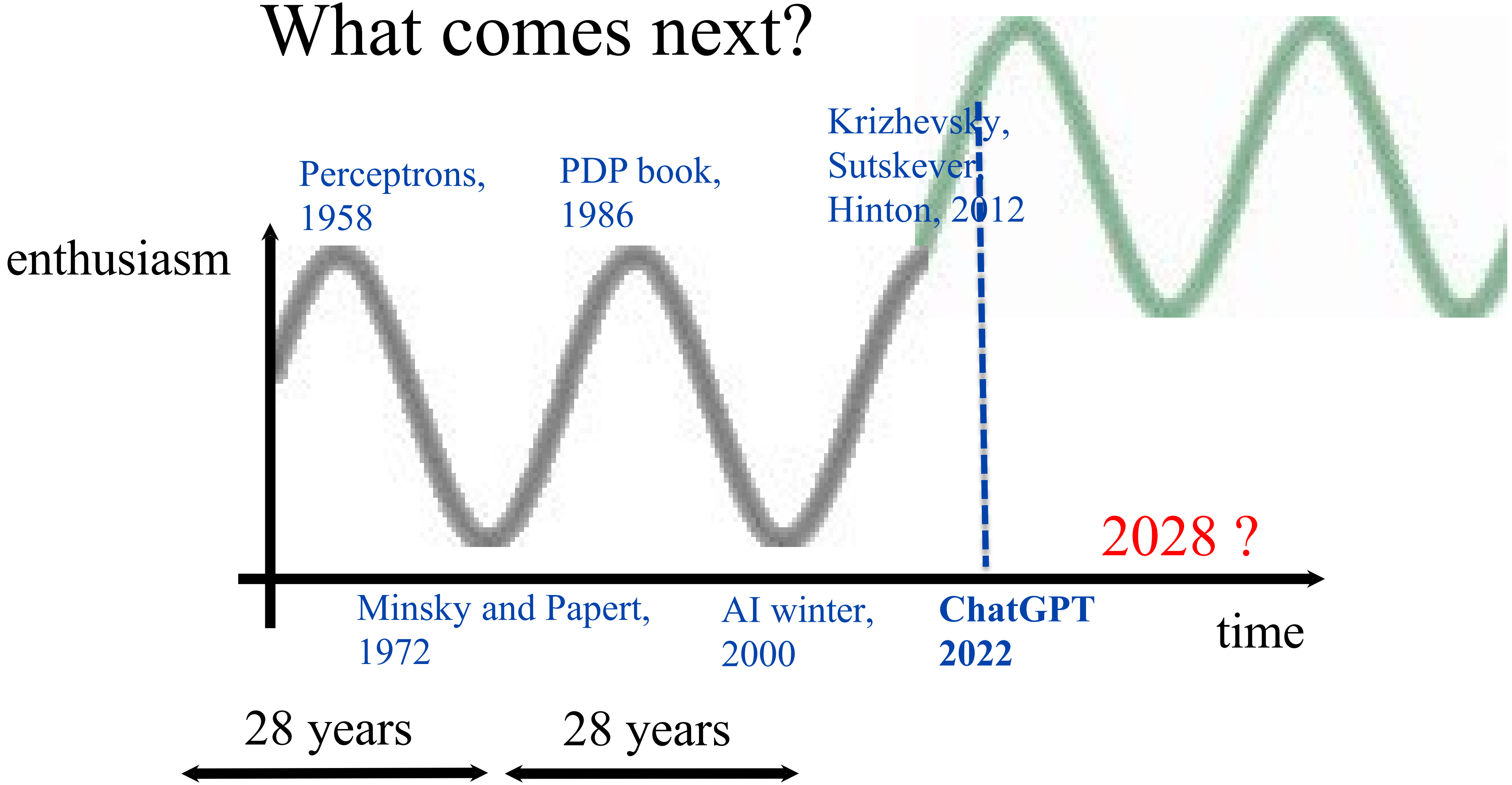



28 years
28 years
Perceptrons, 1958
PDP book, 1986
Krizhevsky, Sutskever, Hinton, 2012
enthusiasm
Minsky and Papert, 1972
AI winter, 2000
time

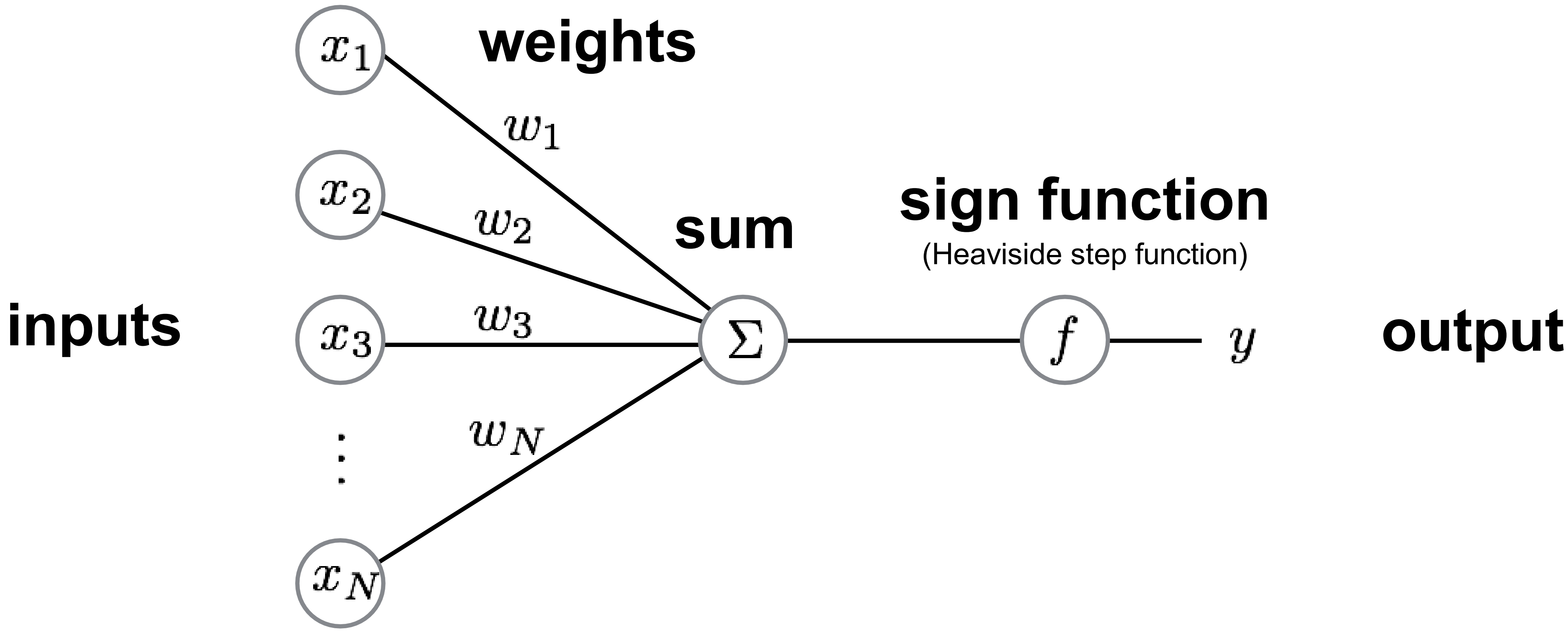
What comes next?



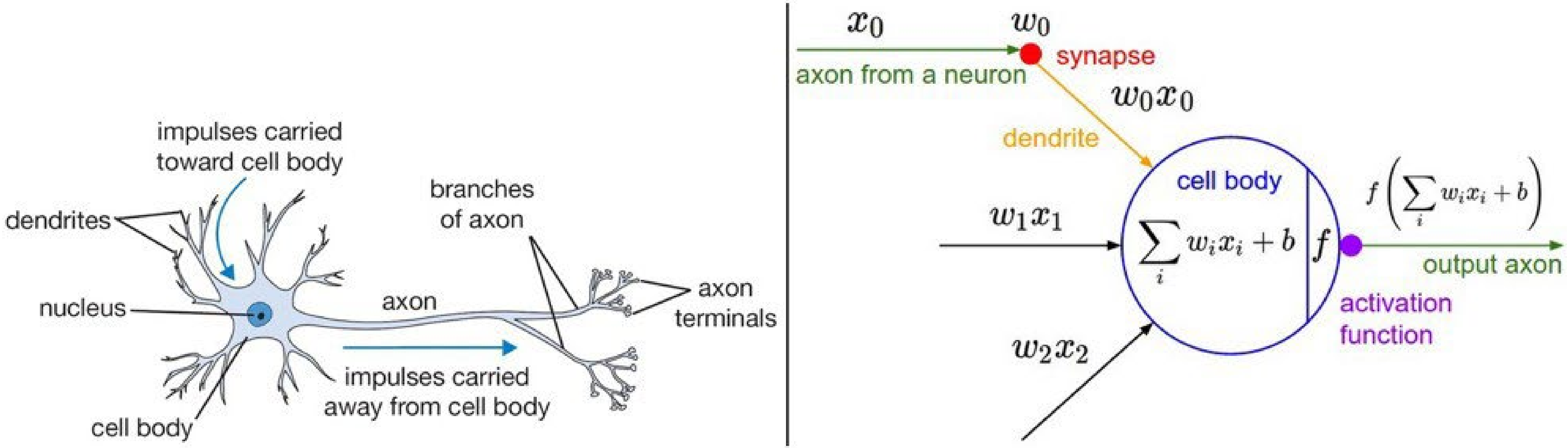
What comes next?



The Perceptron



Aside: Inspiration from Biology



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

- Neural nets/perceptrons are **loosely** inspired by biology.
- But they certainly are **not** a model of how the brain works, or even how neurons work. ➔ cf. spiking neural networks

Perceptron Algorithm

1: **function** PERCEPTRON ALGORITHM

2: $\boldsymbol{w}^{(0)} \leftarrow \mathbf{0}$

3: **for** $t = 1, \ldots, T$ **do**

4: $\textsc{Receive}(\boldsymbol{x}^{(t)})$ $\boldsymbol{x} \in \{0,1\}^N$ N-d binary vector

5: $\hat{y}_A^{(t)} = \text{sign}\left(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\right)$ perceptron is just one line of code!

sign of zero is +1

6: $\textsc{Receive}(y^t)$ $y \in \{1, -1\}$

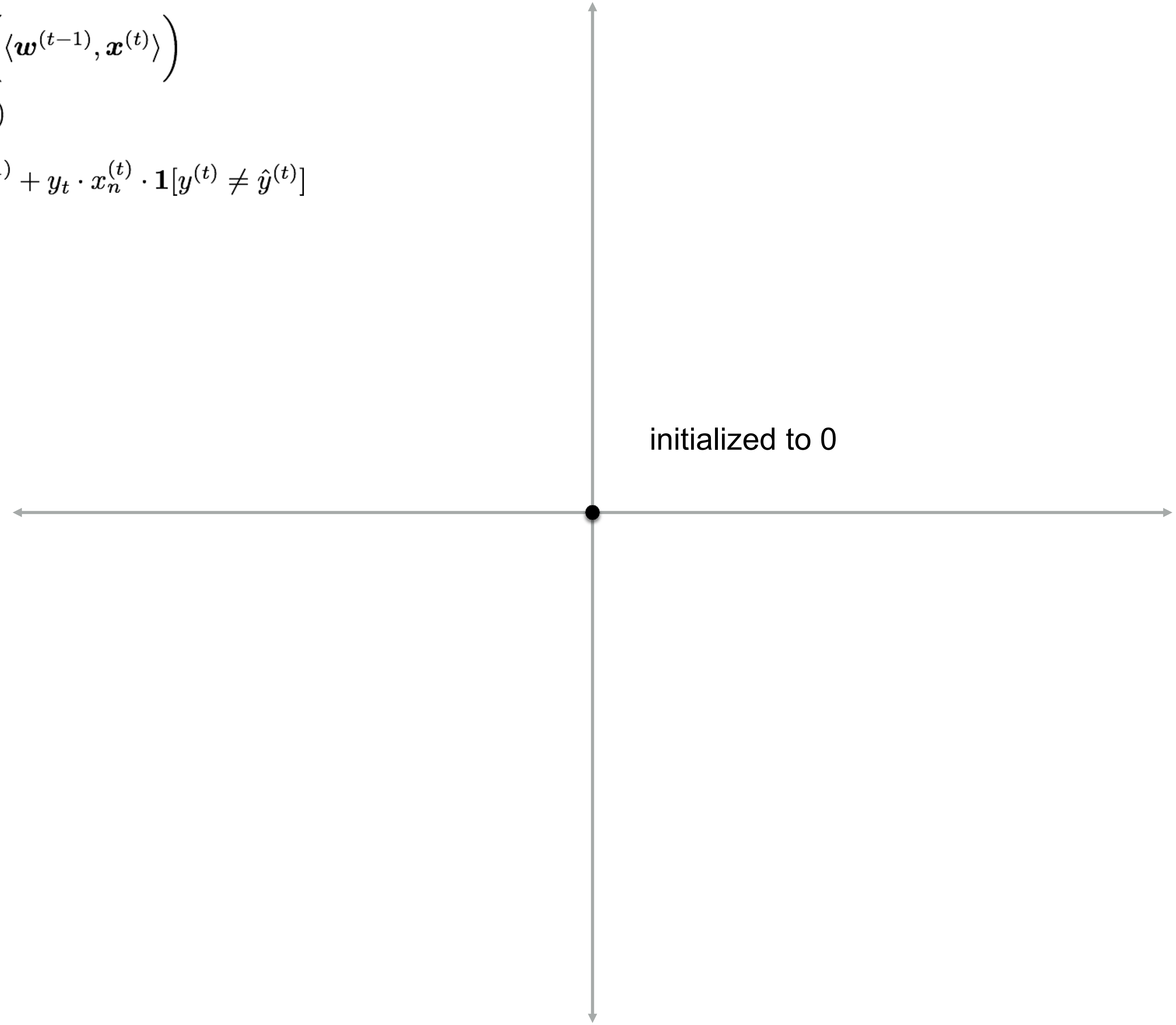
7: $w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$

$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

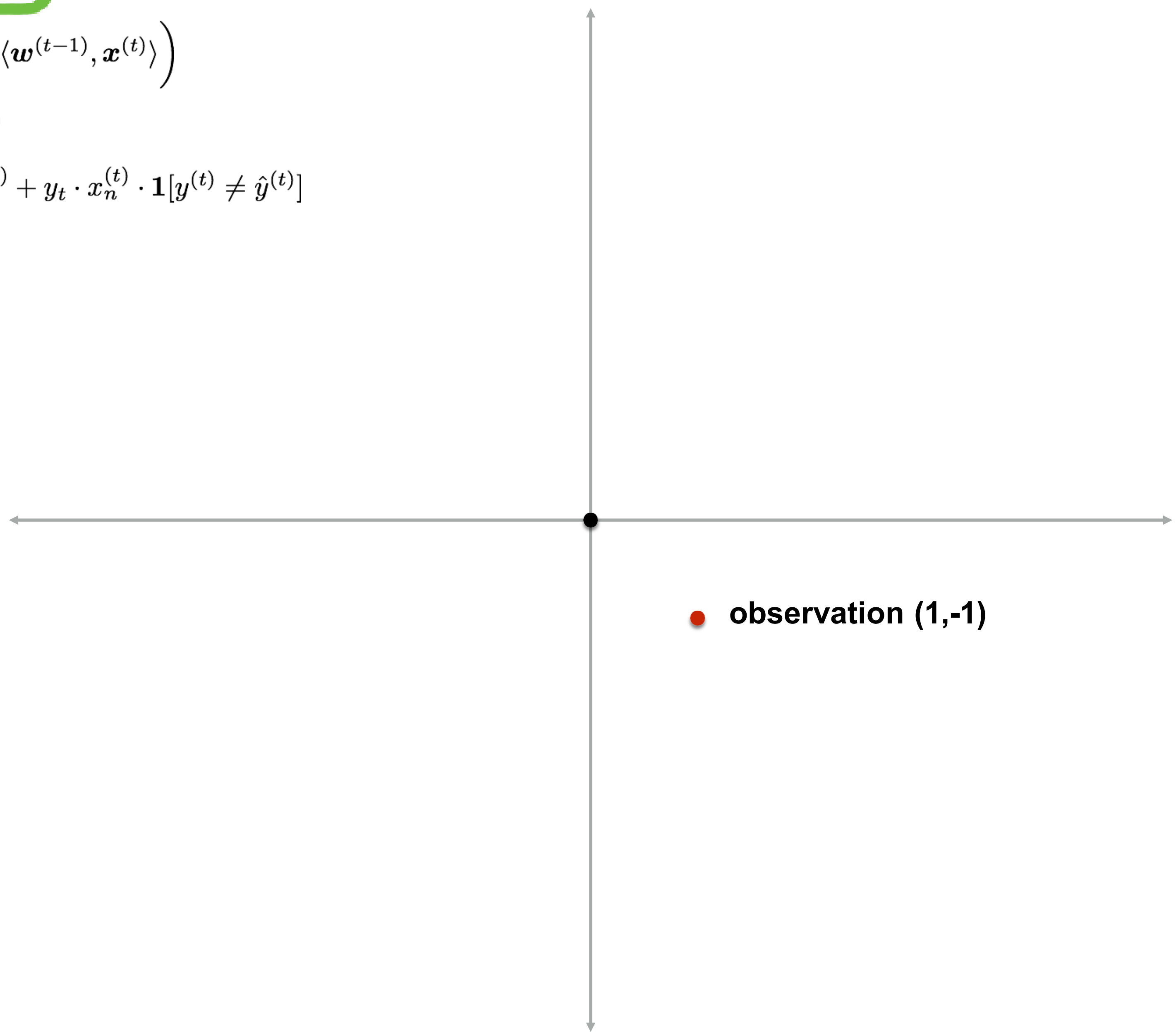


$$\boxed{\textsc{Receive}(\boldsymbol{x}^{(t)})}$$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$$\textsc{Receive}(y^t)$$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

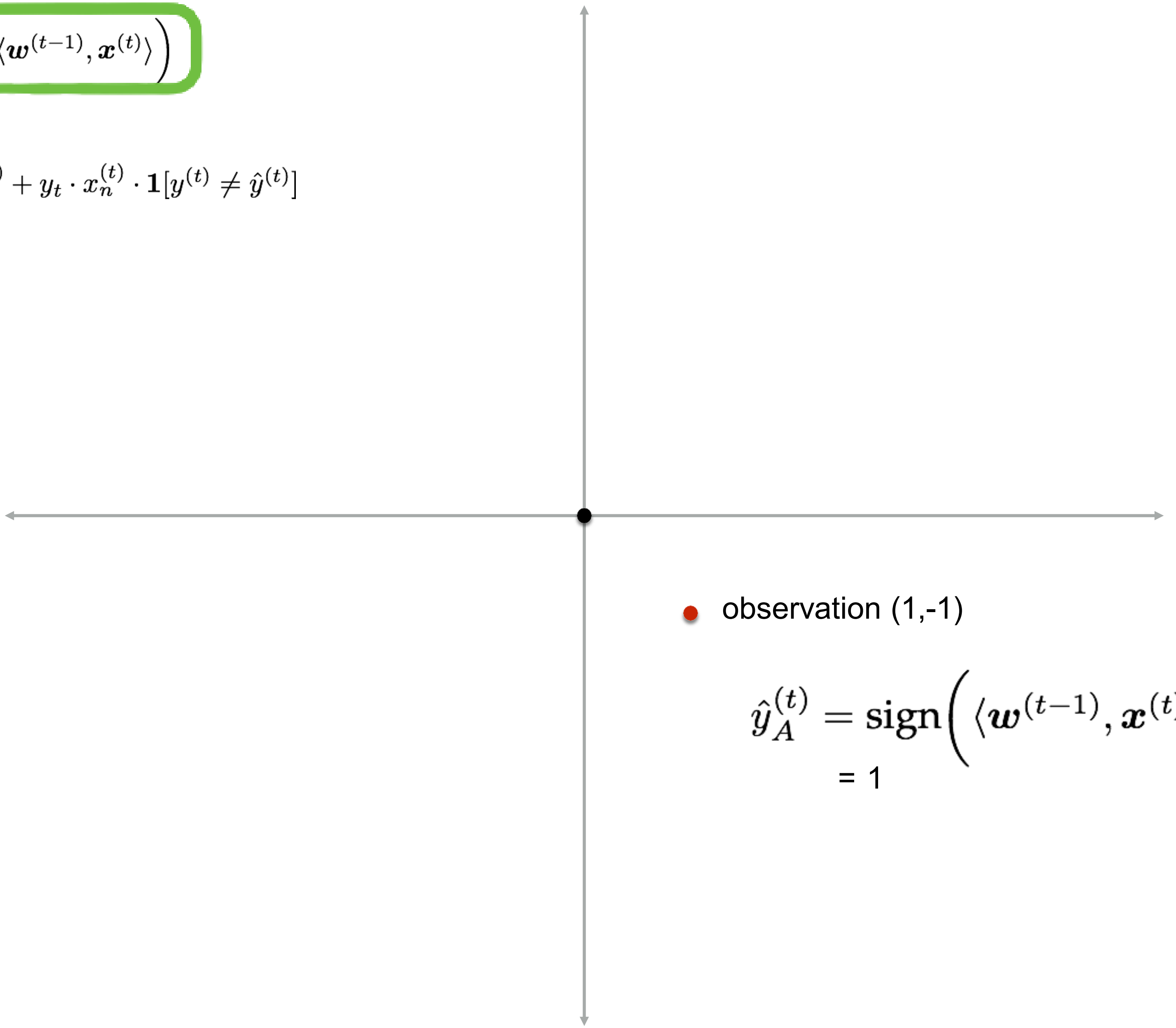


$\text{RECEIVE}(\boldsymbol{x}^{(t)})$

$$\boxed{\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)}$$

$\text{RECEIVE}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$



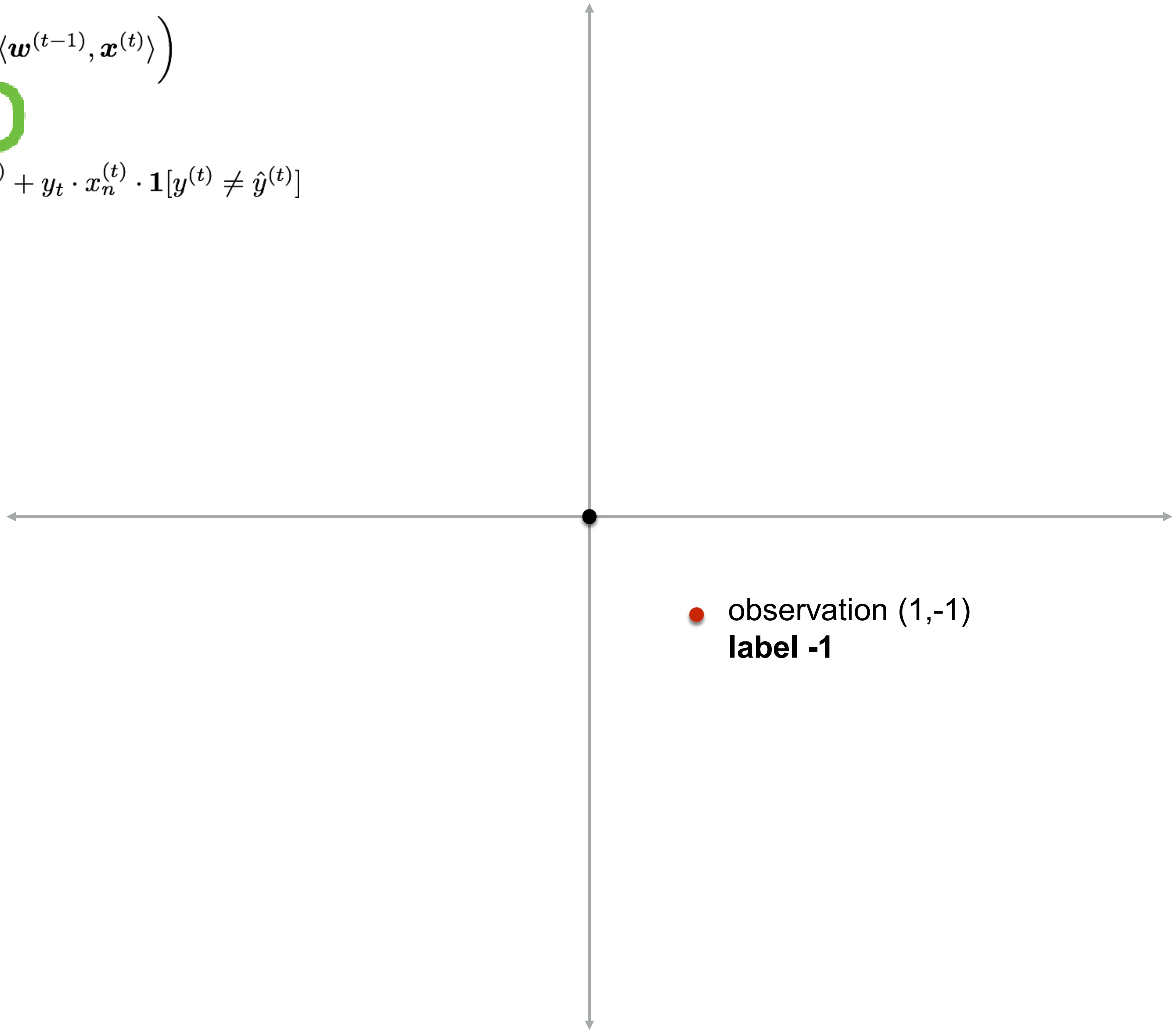
$$\underbrace{\hat{y}_A^{(t)}}_{=\,1} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle \Big)$$

$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$



$\text{RECEIVE}(\boldsymbol{x}^{(t)})$

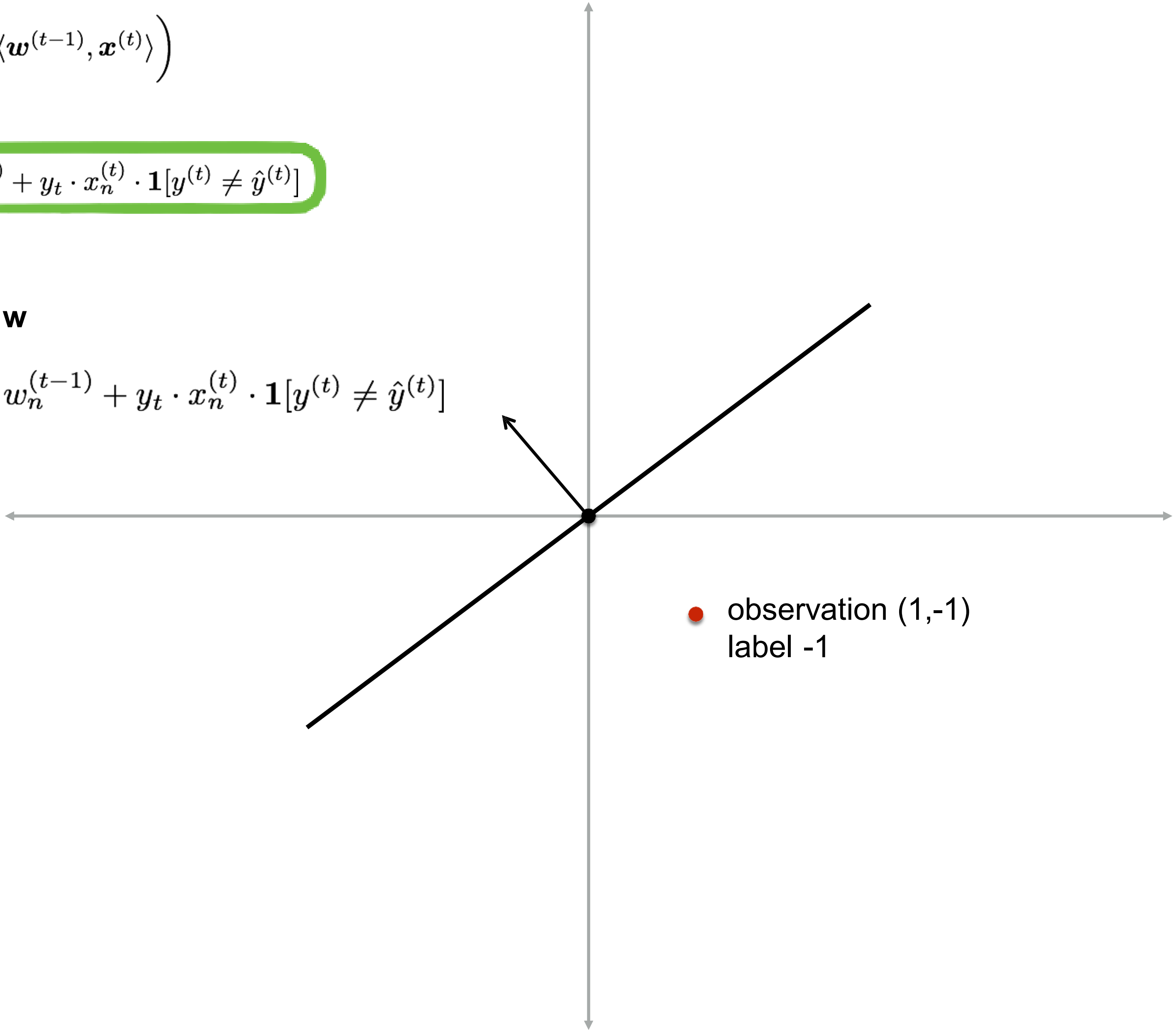
$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\text{RECEIVE}(y^t)$

$$\boxed{w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]}$$

update w

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$



$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \operatorname{sign}\left(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\right)$$

$\textsc{Receive}(y^t)$

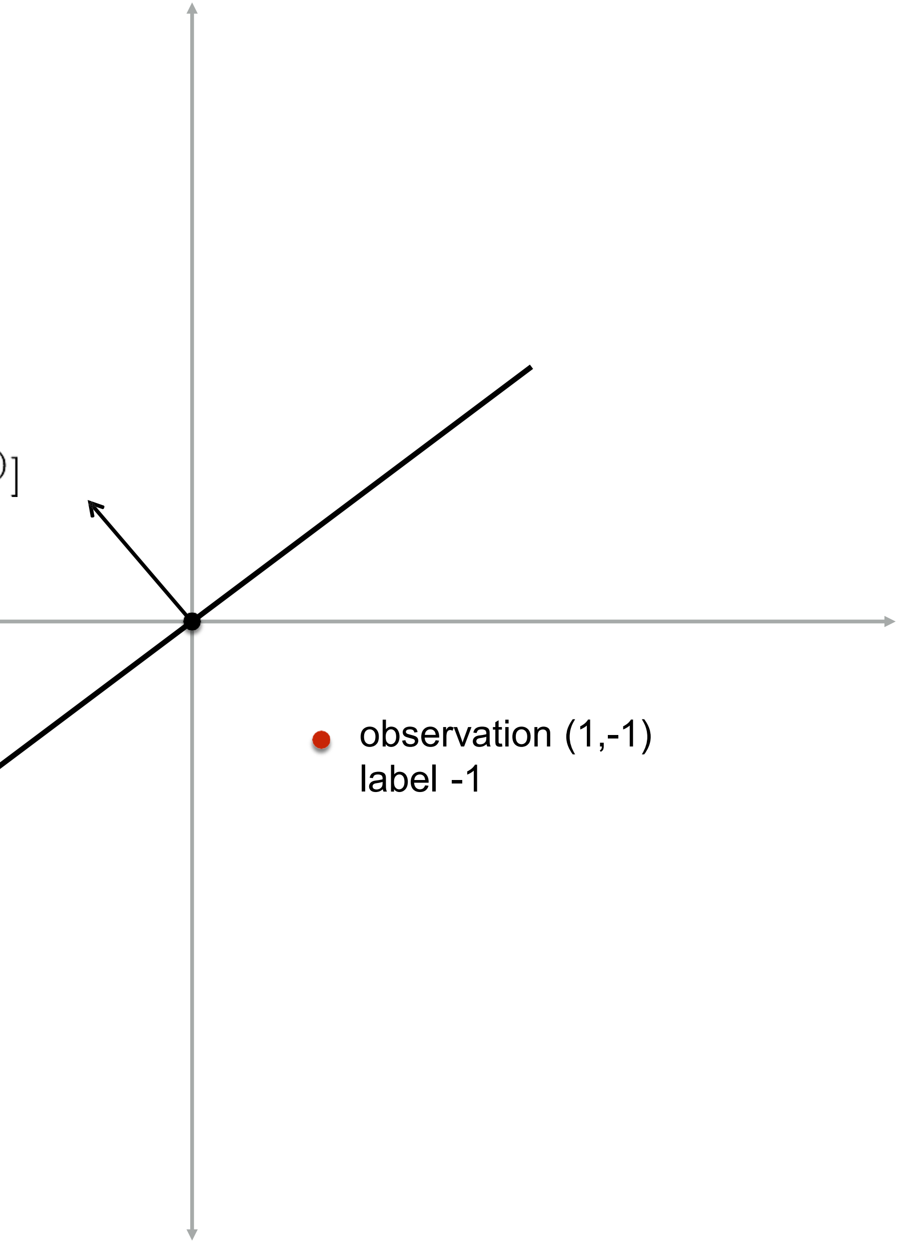
$$\boxed{w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]}$$

update w

no match!

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

(-1,1) (0,0) -1 (1,-1) 1



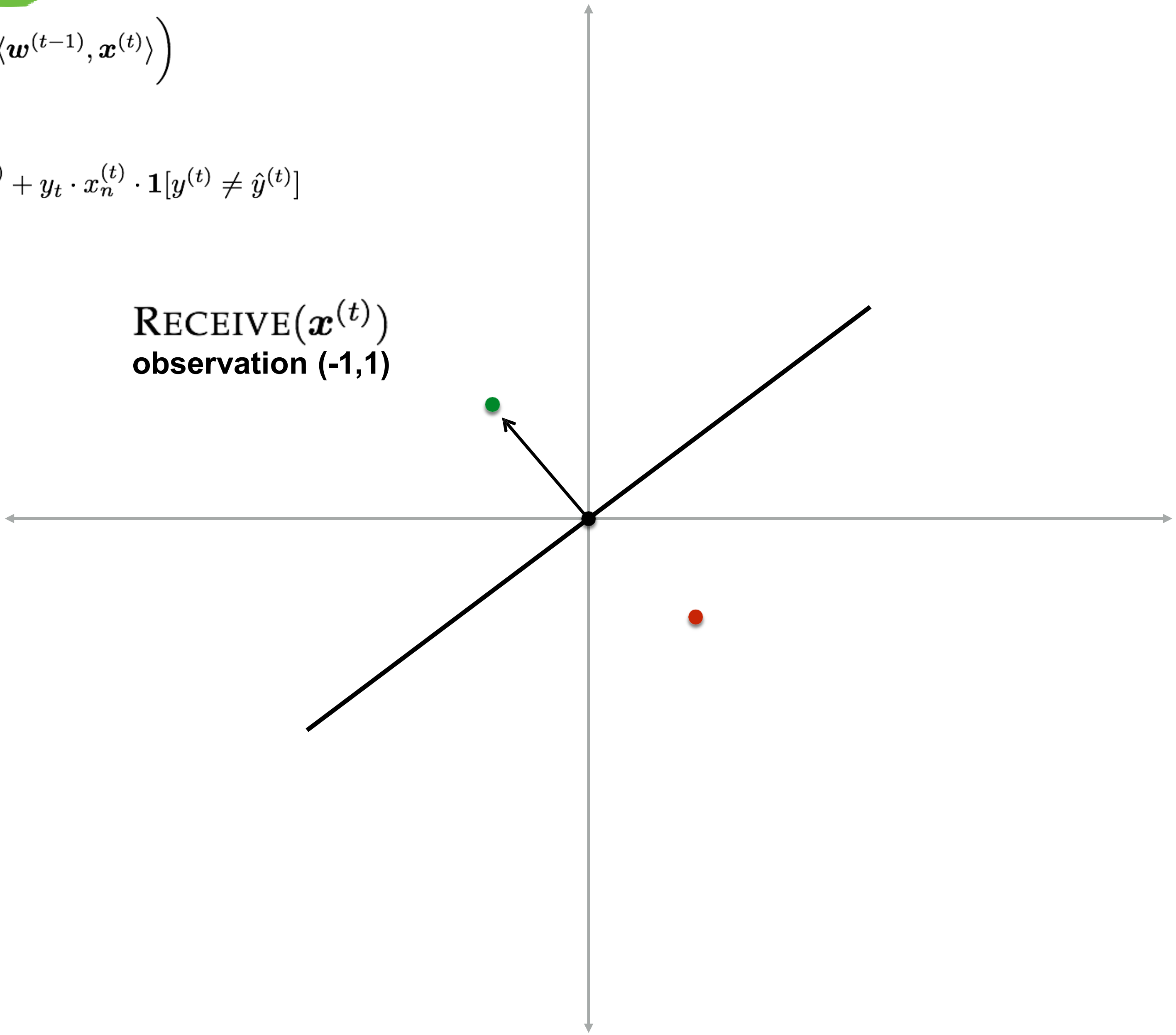
$\boxed{\textsc{Receive}(\boldsymbol{x}^{(t)})}$

$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$

$\textsc{Receive}(y^t)$

$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$

(-1,1)



$\textsc{Receive}(\boldsymbol{x}^{(t)})$

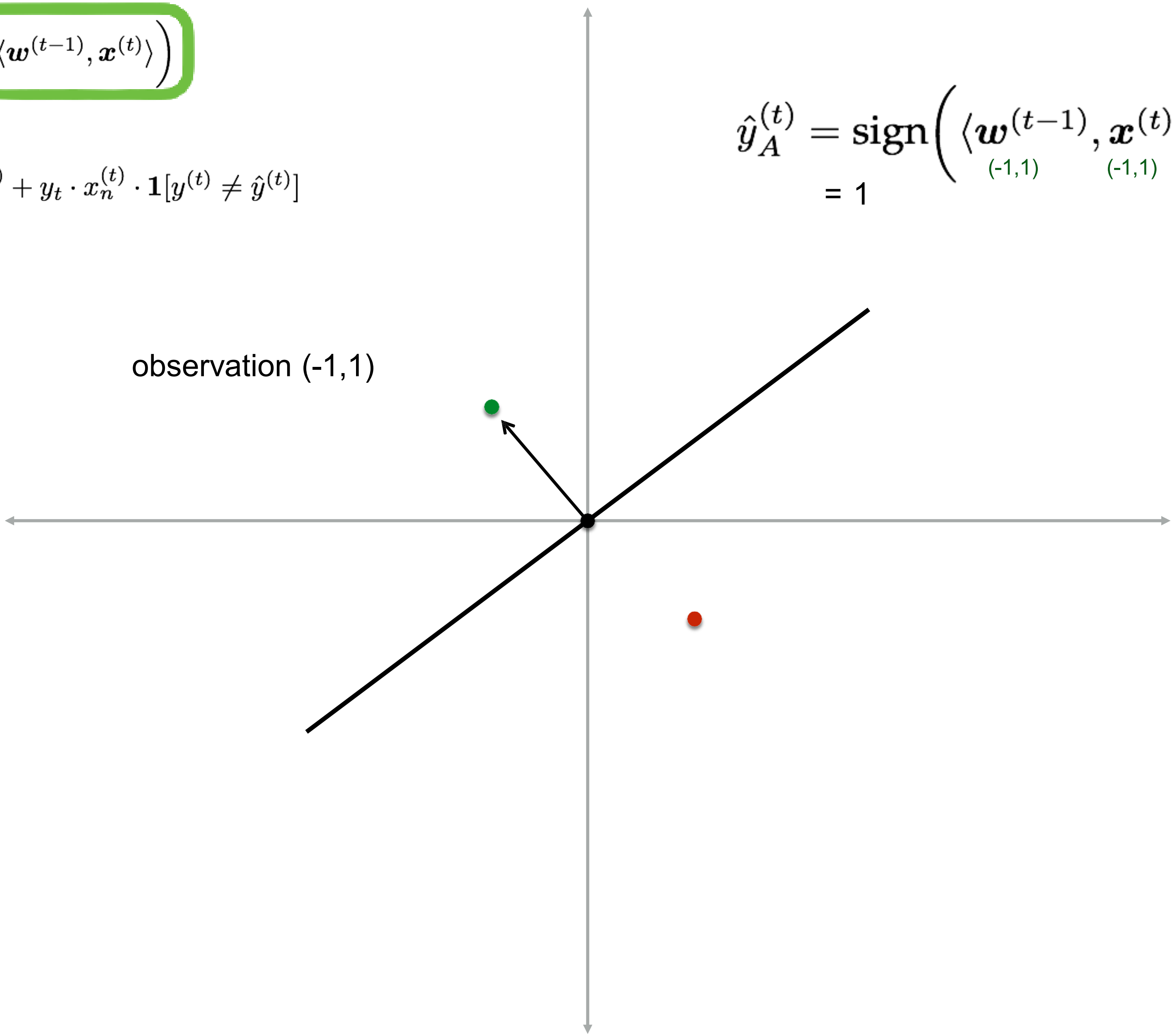
$$\boxed{\hat{y}_A^{(t)} = \operatorname{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)}$$

$\textsc{Receive}(y^t)$

$$\underbrace{w_n^{(t)}}_{(-1,1)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

$$\hat{y}_A^{(t)} = \operatorname{sign}\Big(\langle \underbrace{\boldsymbol{w}^{(t-1)}}_{(-1,1)}, \underbrace{\boldsymbol{x}^{(t)}}_{(-1,1)} \rangle\Big)$$

$= 1$



$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

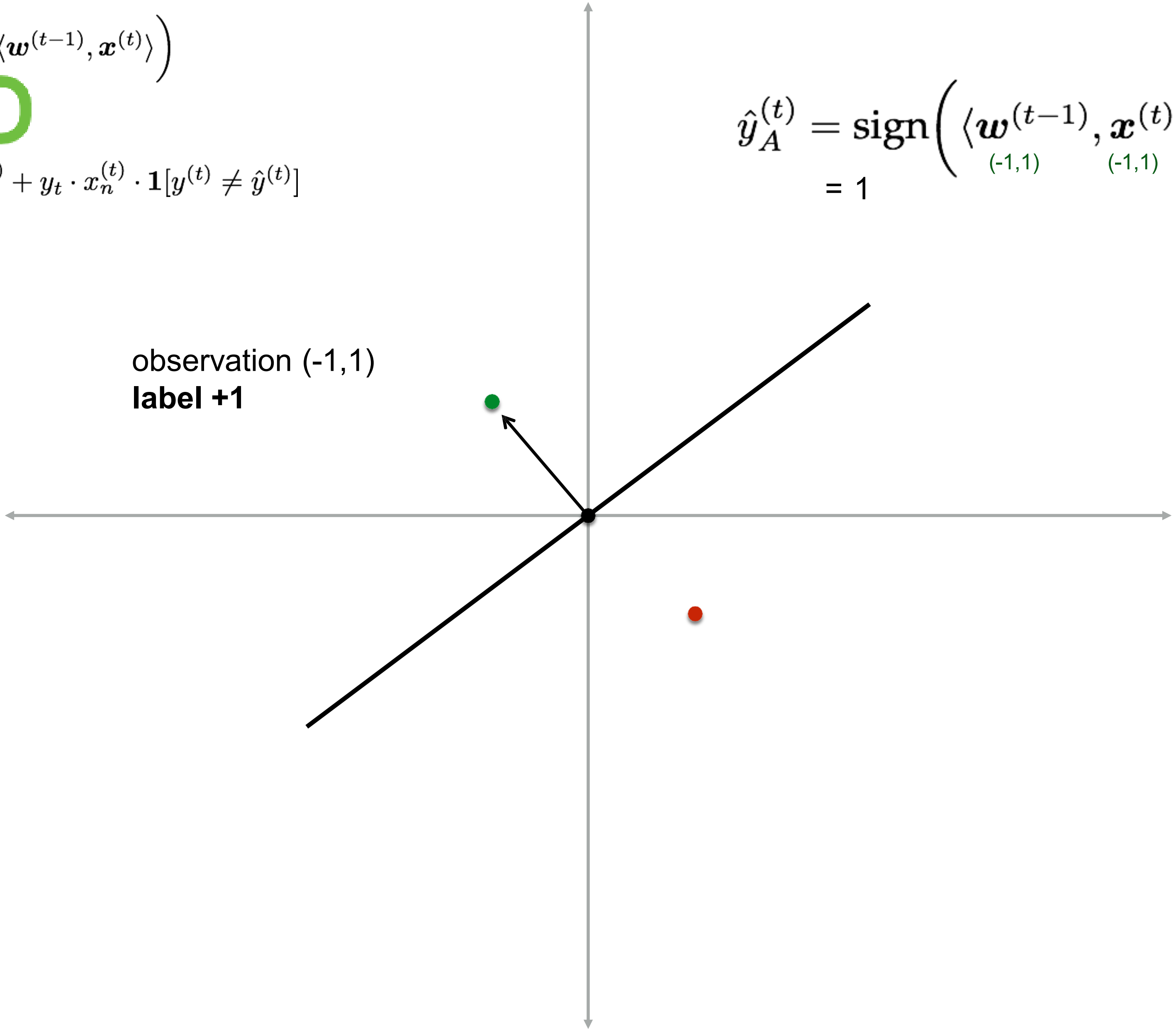
$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

(-1,1)

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \underset{(-1,1)}{\boldsymbol{w}^{(t-1)}}, \underset{(-1,1)}{\boldsymbol{x}^{(t)}} \rangle\Big)$$

$= 1$



$\text{RECEIVE}(\boldsymbol{x}^{(t)})$

$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$

$\text{RECEIVE}(y^t)$

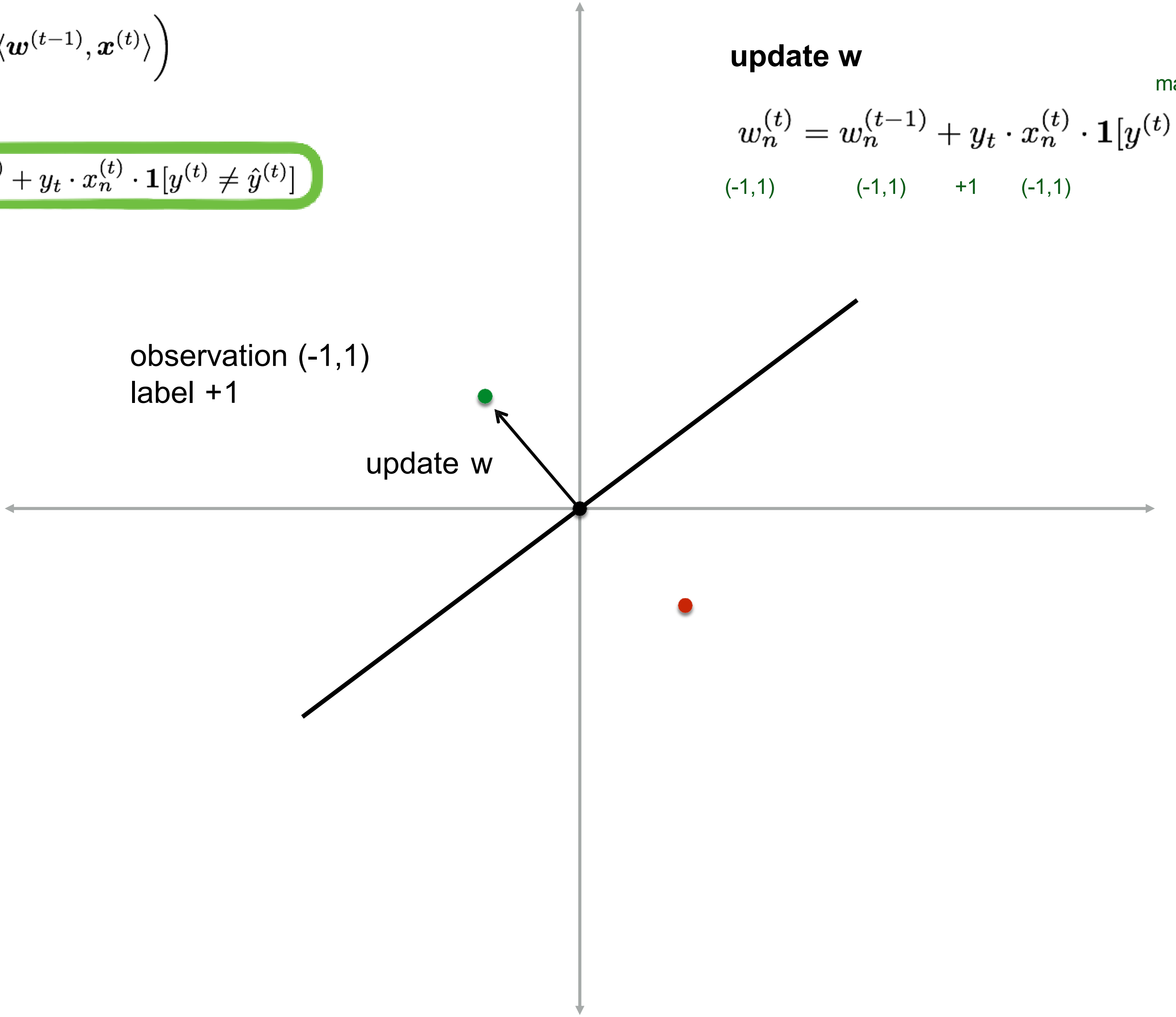
$\boxed{w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]}$

update w

match!

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

(-1,1) (-1,1) +1 (-1,1) 0

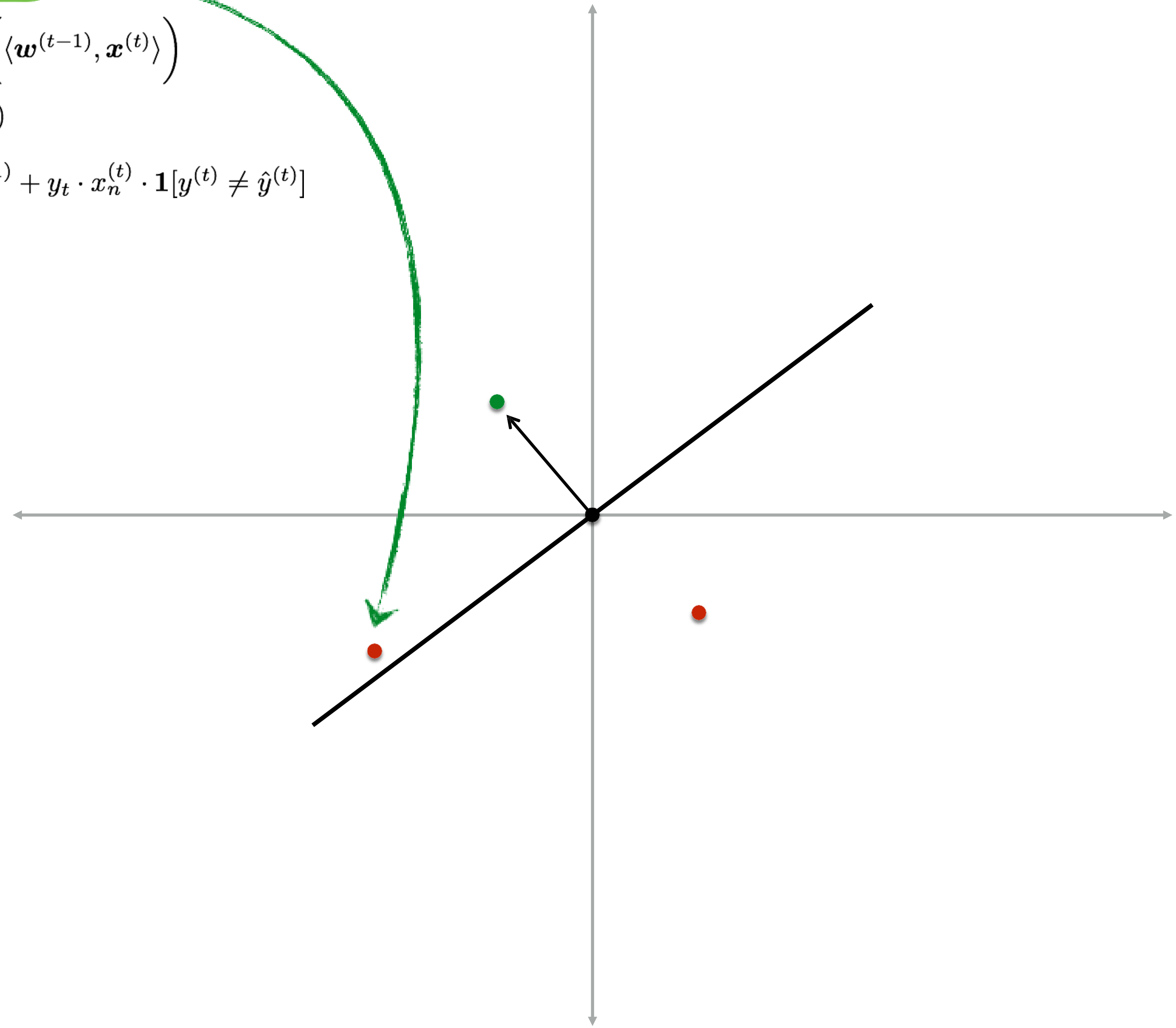


$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

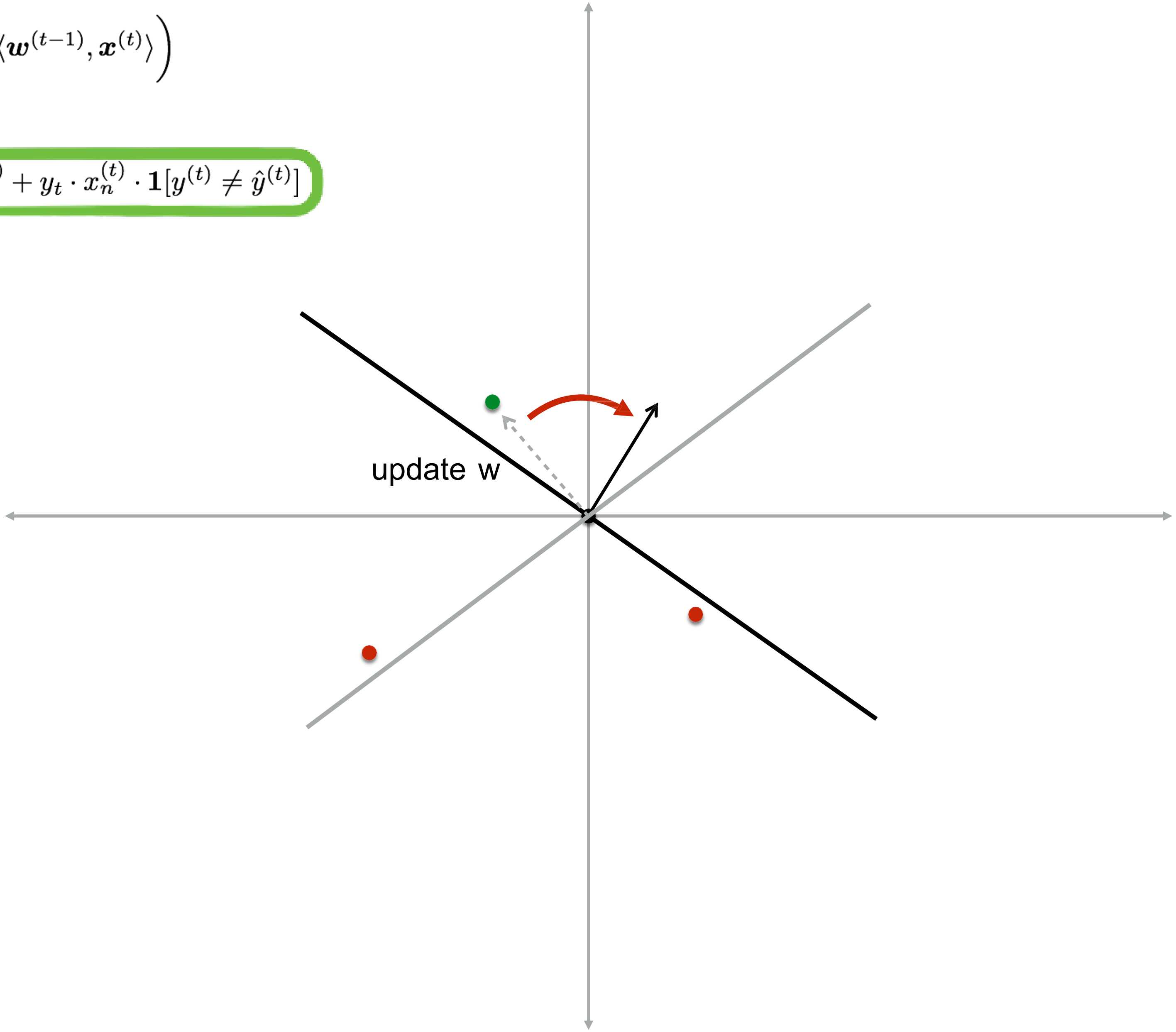


$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

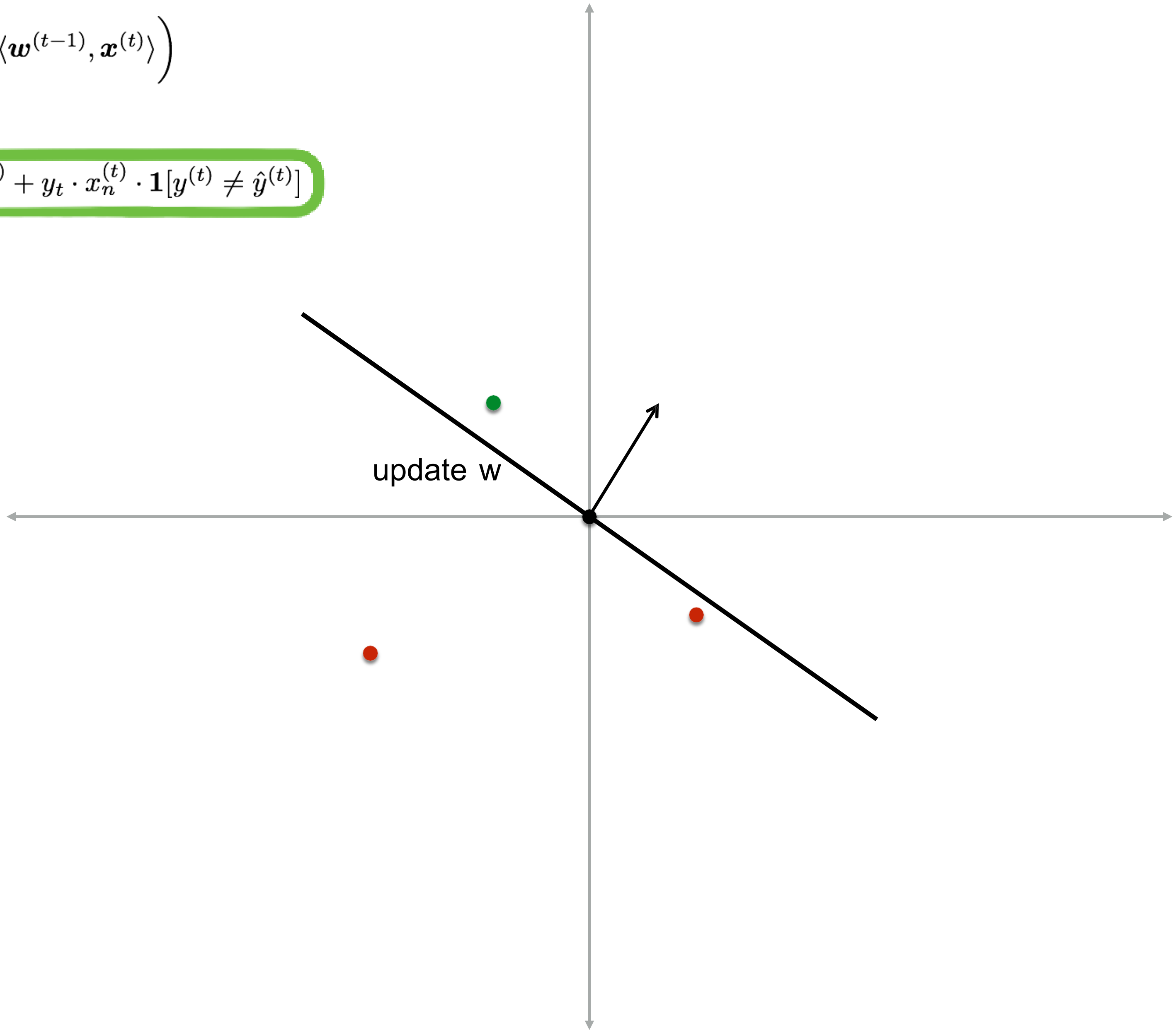


$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

$\textsc{Receive}(y^t)$

$$\boxed{w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]}$$

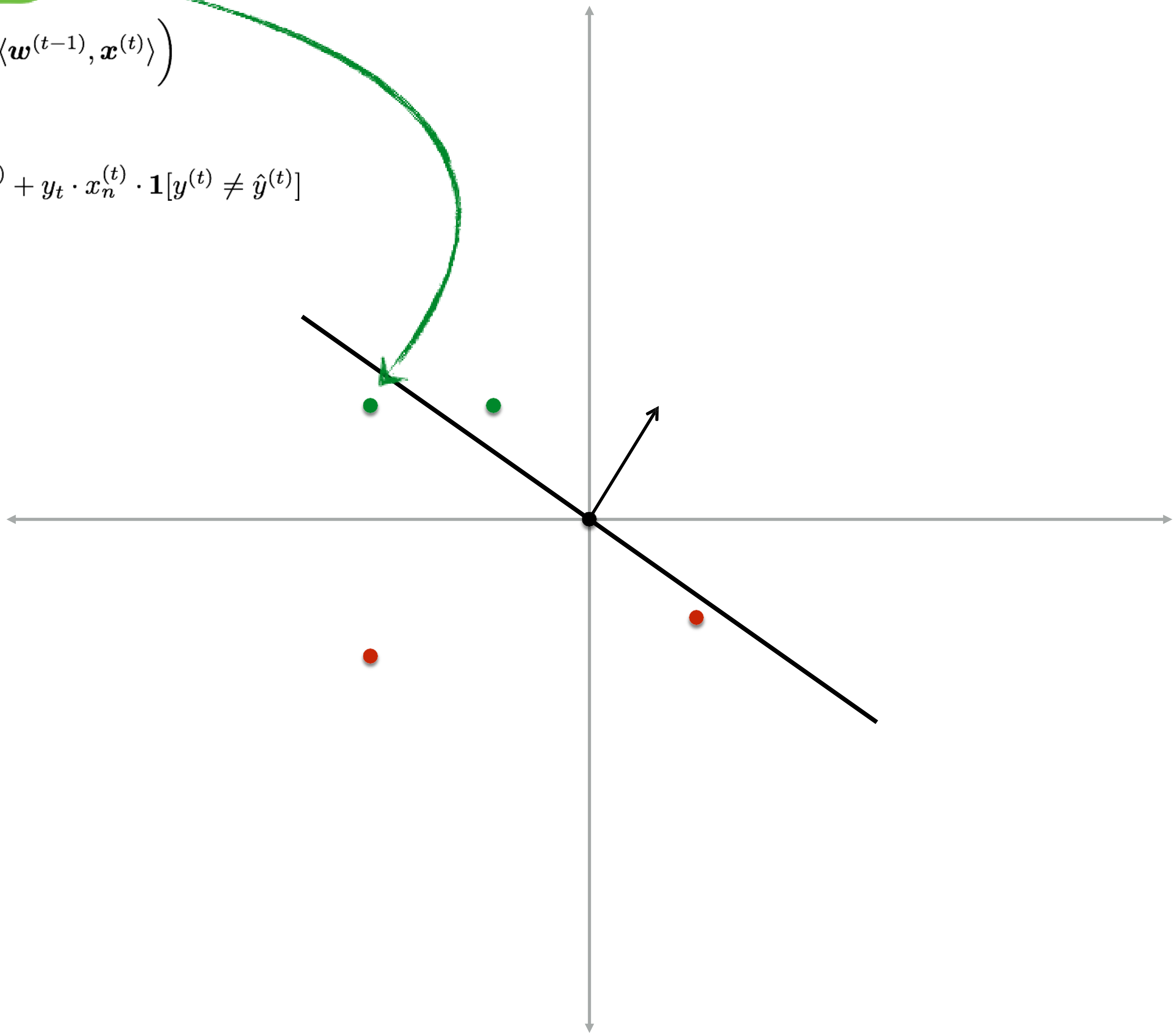


$\boxed{\textsc{Receive}(\boldsymbol{x}^{(t)})}$

$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$

$\textsc{Receive}(y^t)$

$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$

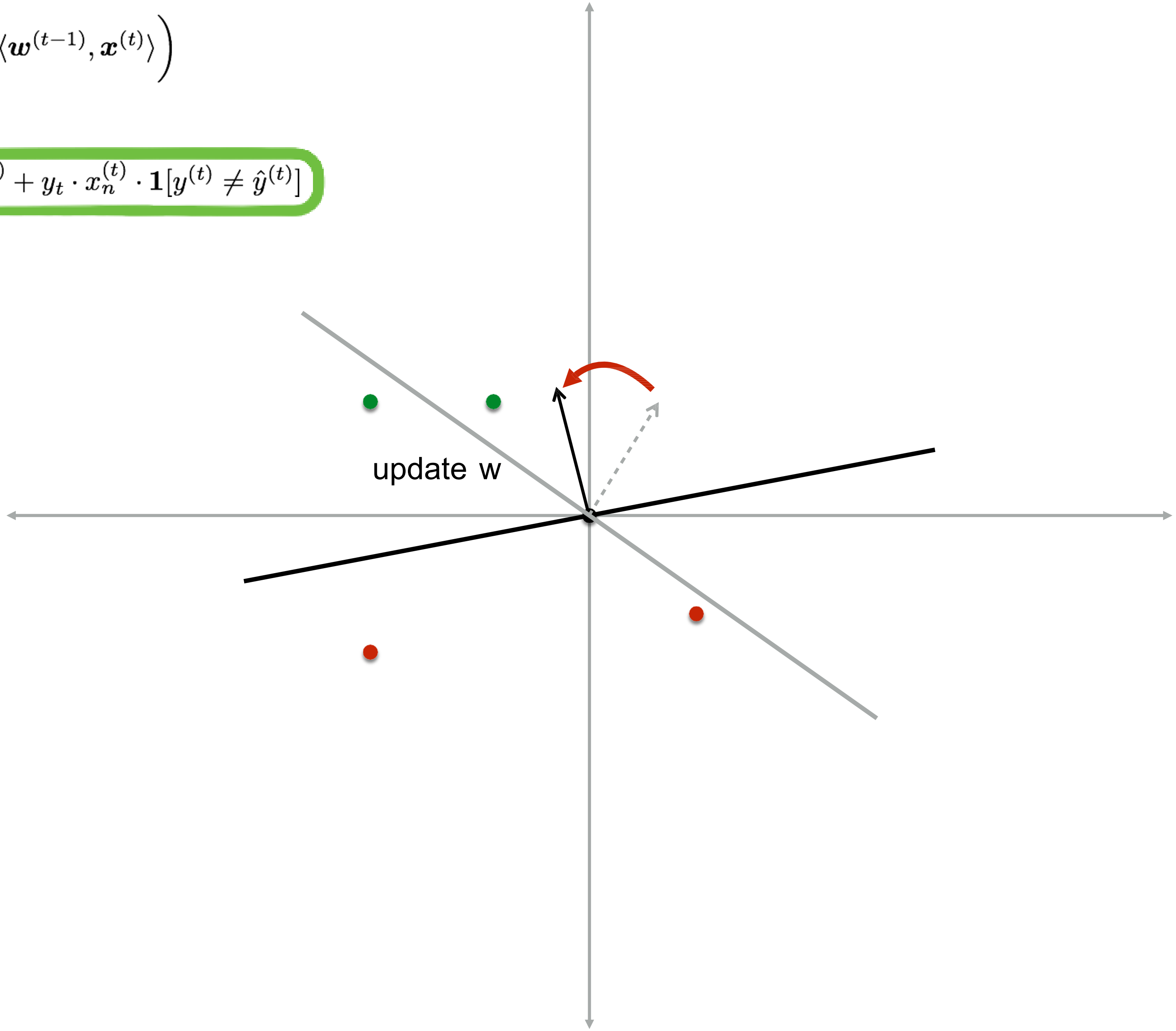


$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\left(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\right)$$

$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

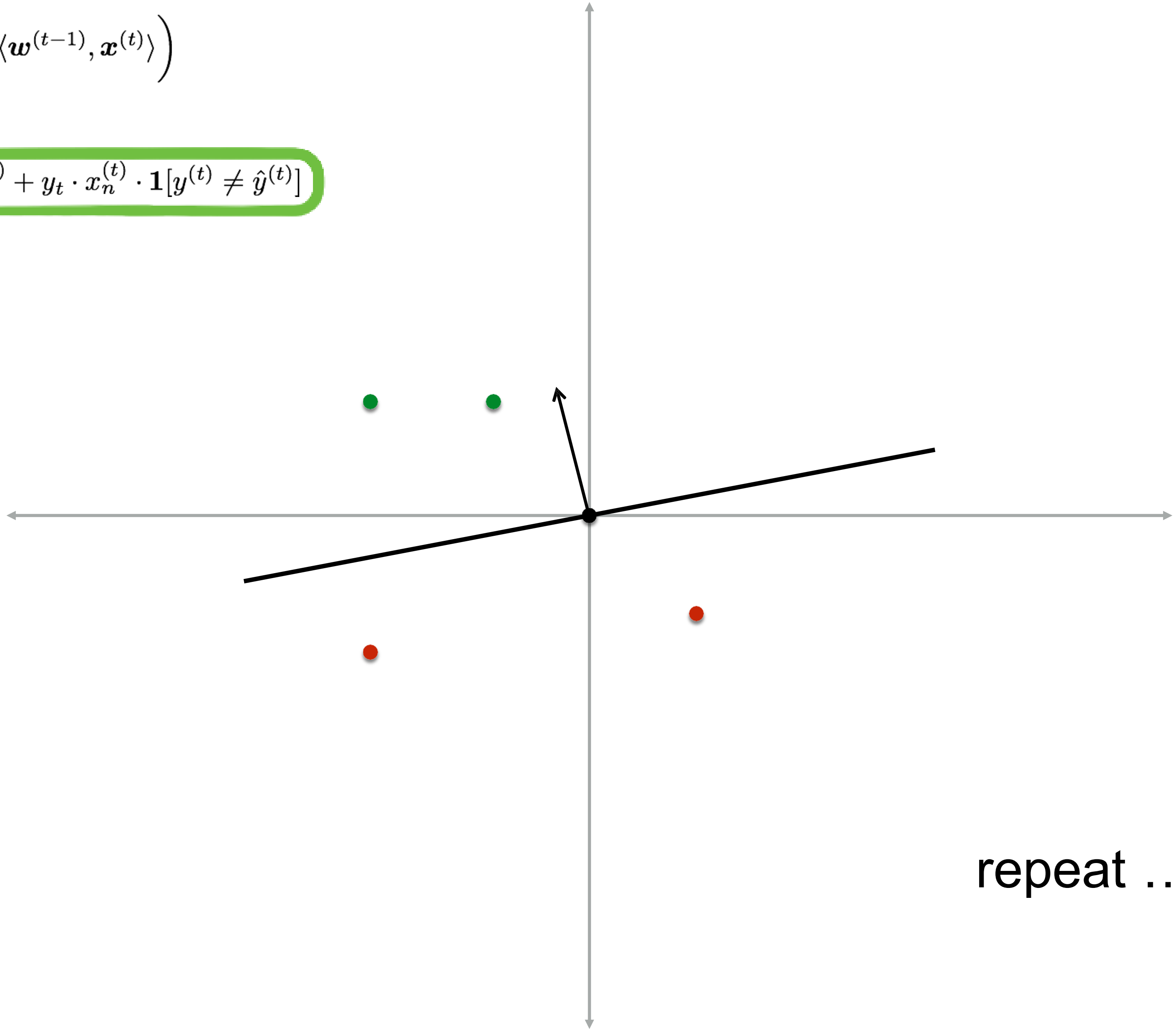


$\textsc{Receive}(\boldsymbol{x}^{(t)})$

$$\hat{y}_A^{(t)} = \text{sign}\Big(\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(t)} \rangle\Big)$$

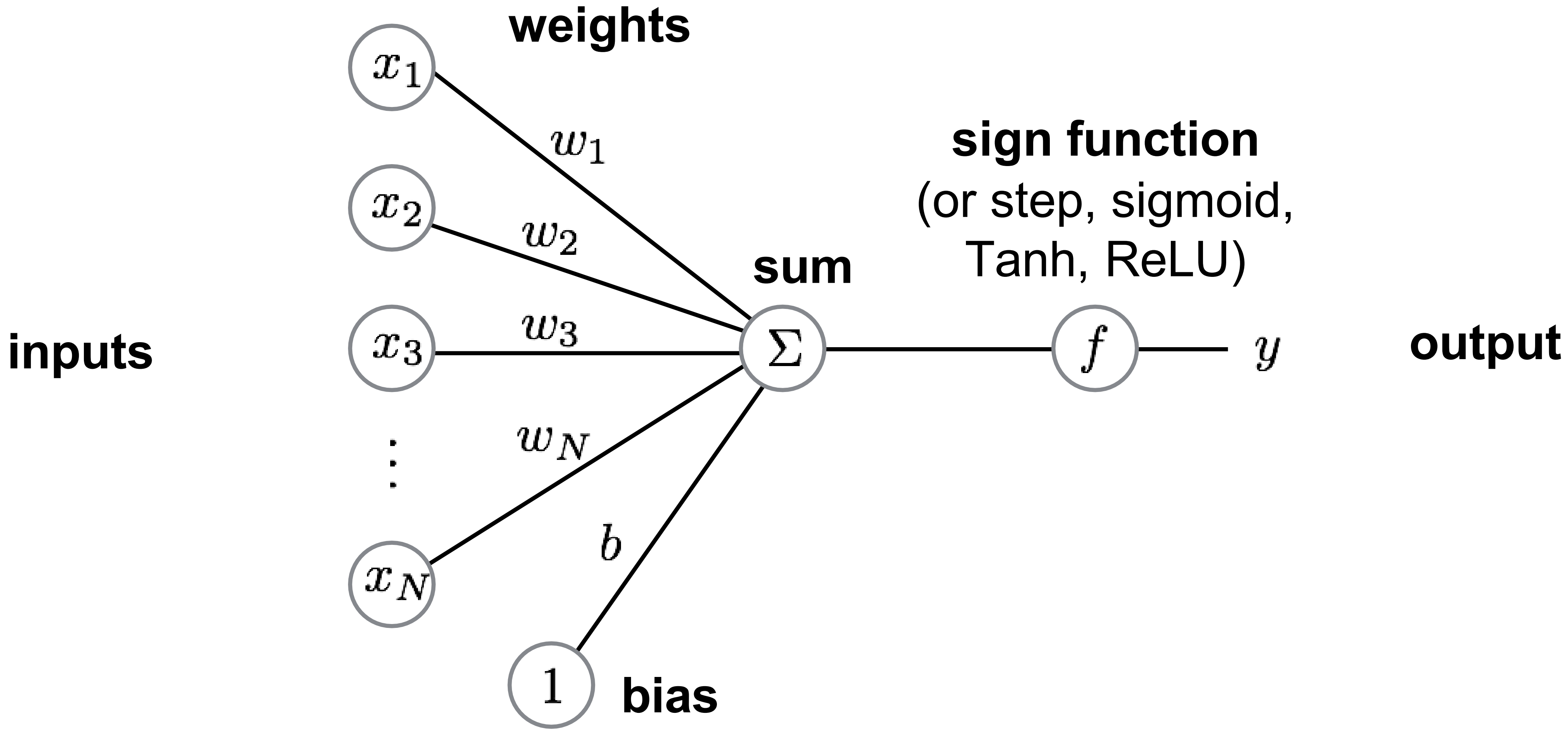
$\textsc{Receive}(y^t)$

$$w_n^{(t)} = w_n^{(t-1)} + y_t \cdot x_n^{(t)} \cdot \mathbf{1}[y^{(t)} \neq \hat{y}^{(t)}]$$

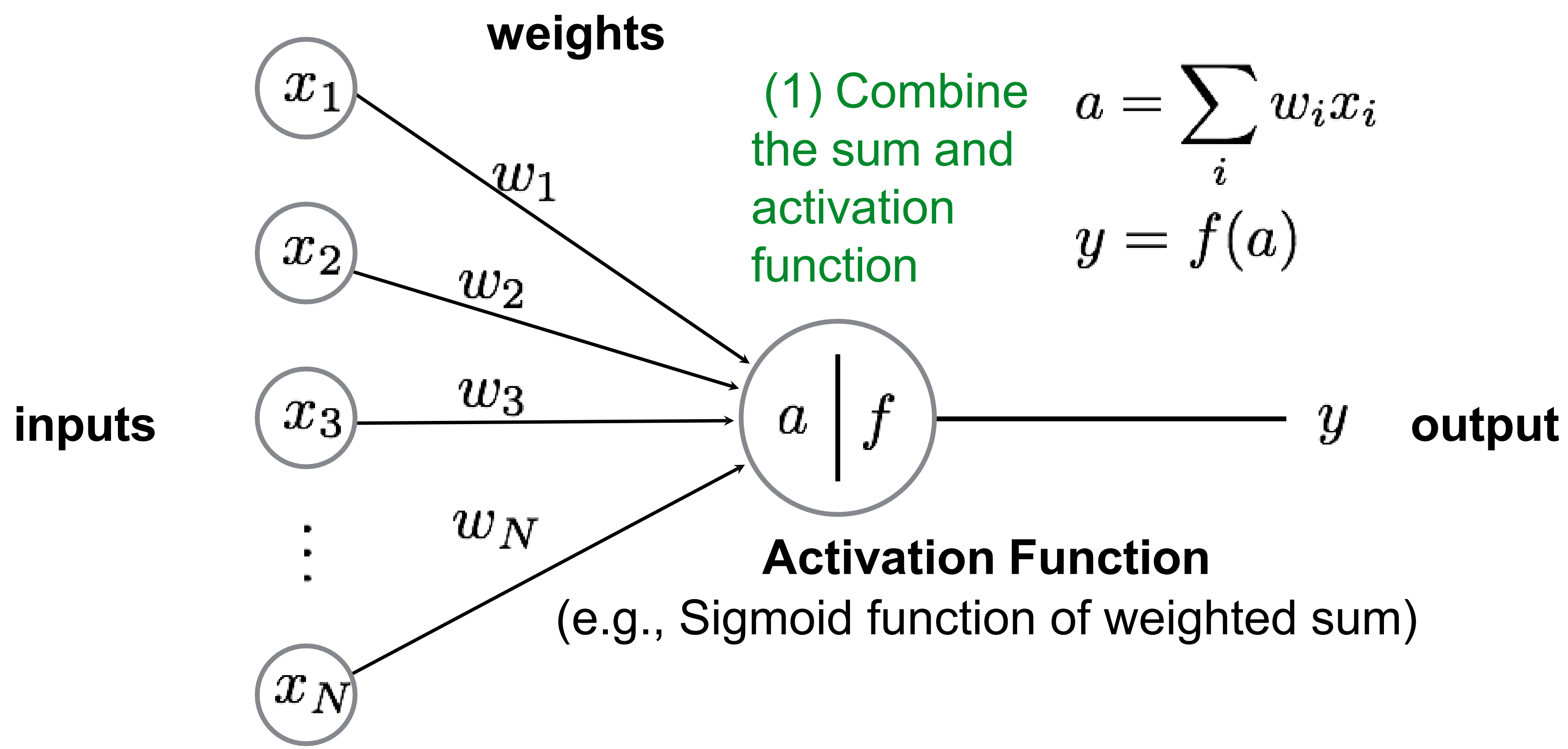


repeat …

The Perceptron



Another way to draw it...



(1) Combine the sum and activation function

$$a = \sum_i w_i x_i$$

$$y = f(a)$$

Activation Function

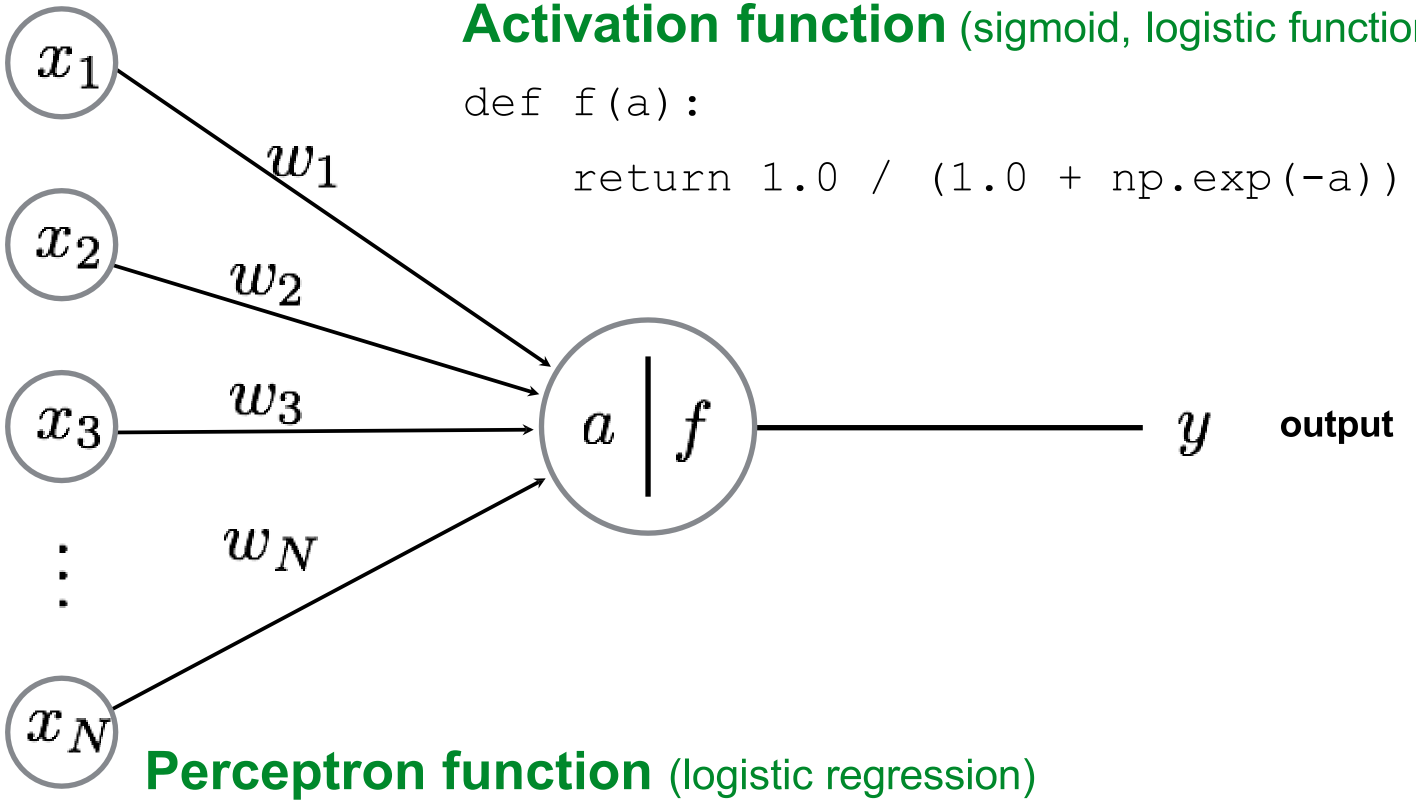
(e.g., Sigmoid function of weighted sum)

(2) suppress the bias term (less clutter)

$$x_N = 1$$

$$w_N = b$$

Programming the 'forward pass'



Activation function (sigmoid, logistic function)

```
def f(a):
    return 1.0 / (1.0 + np.exp(-a))
```

Perceptron function (logistic regression)

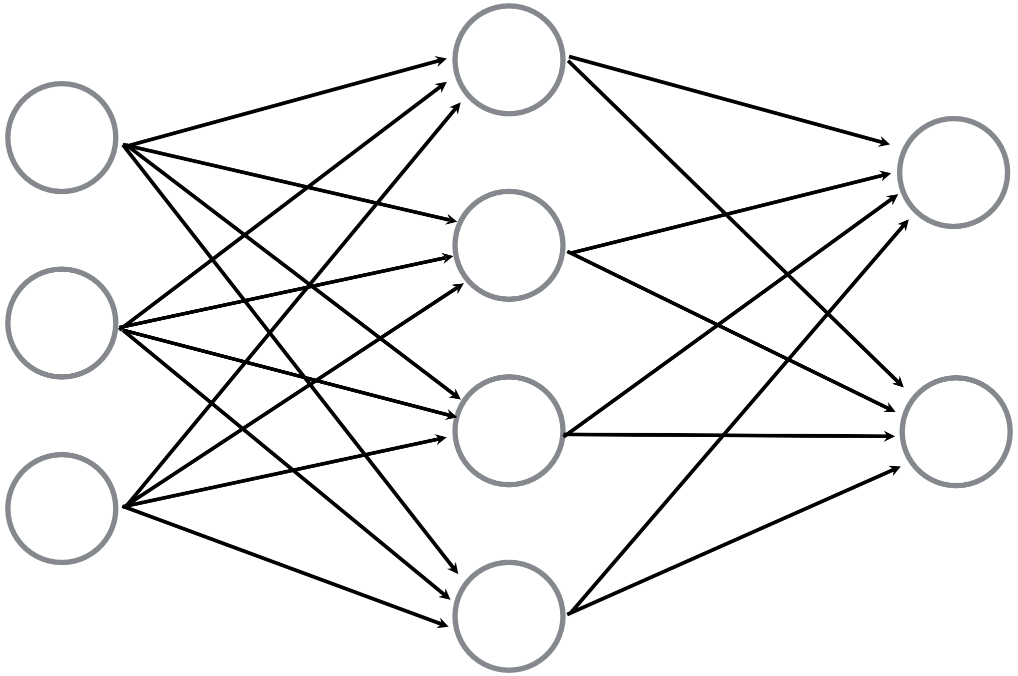
```
def perceptron(x, w):
    a = np.dot(x, w)
    return f(a)
```

Neural networks

Connect a bunch of perceptrons together …

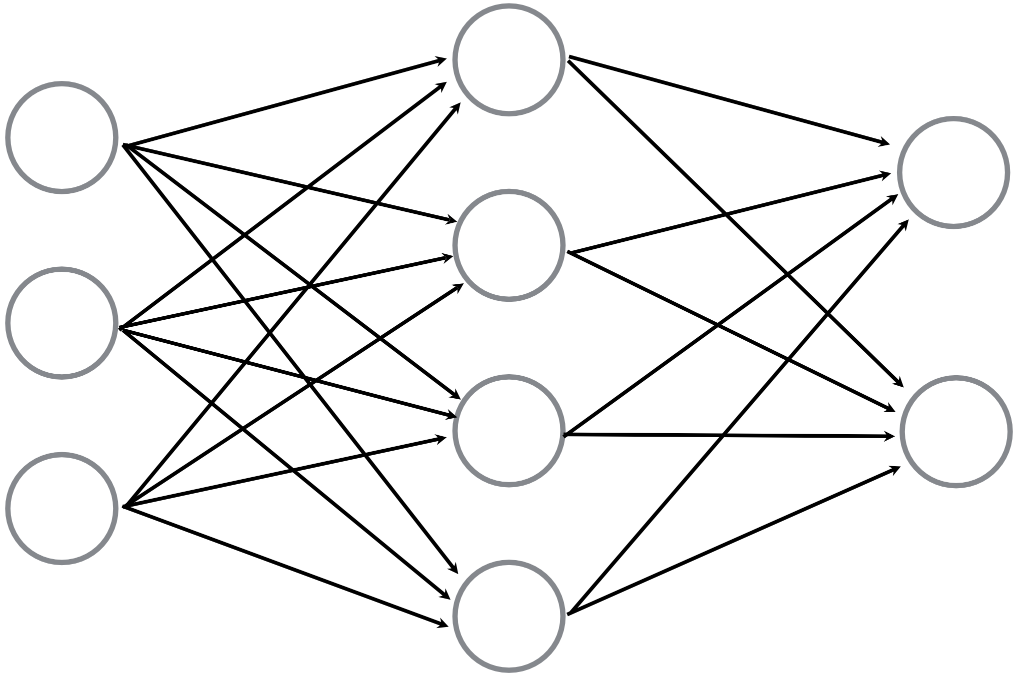
Neural Network

a collection of connected perceptrons



Neural Network

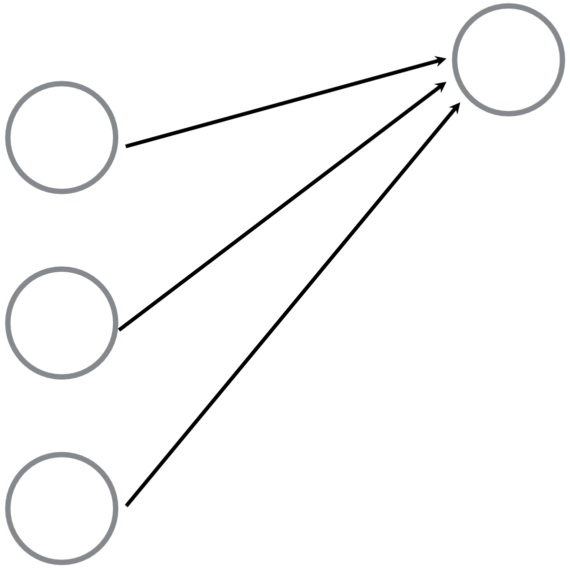
a collection of connected perceptrons



How many perceptrons in this neural network?

Neural Network

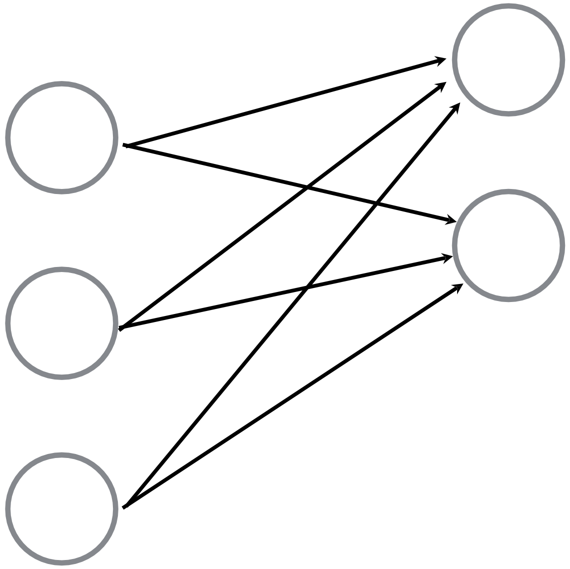
a collection of connected perceptrons



‘one perceptron’

Neural Network

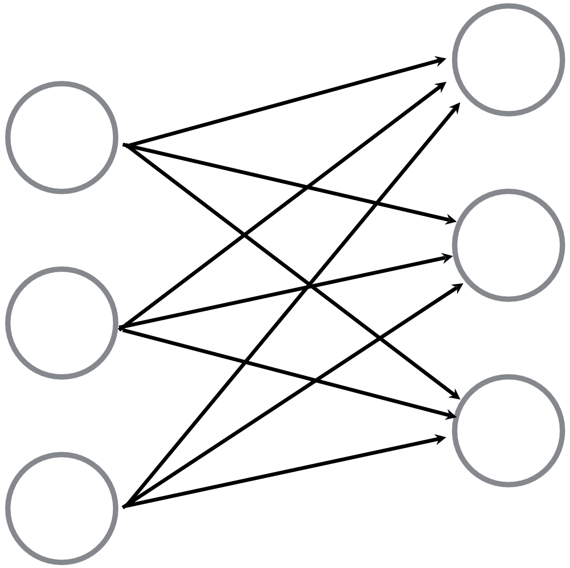
a collection of connected perceptrons



'two perceptrons'

Neural Network

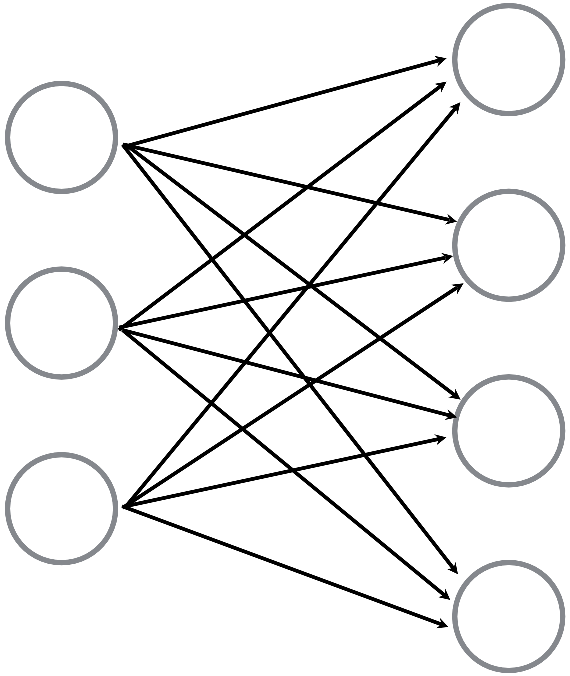
a collection of connected perceptrons



‘three perceptrons’

Neural Network

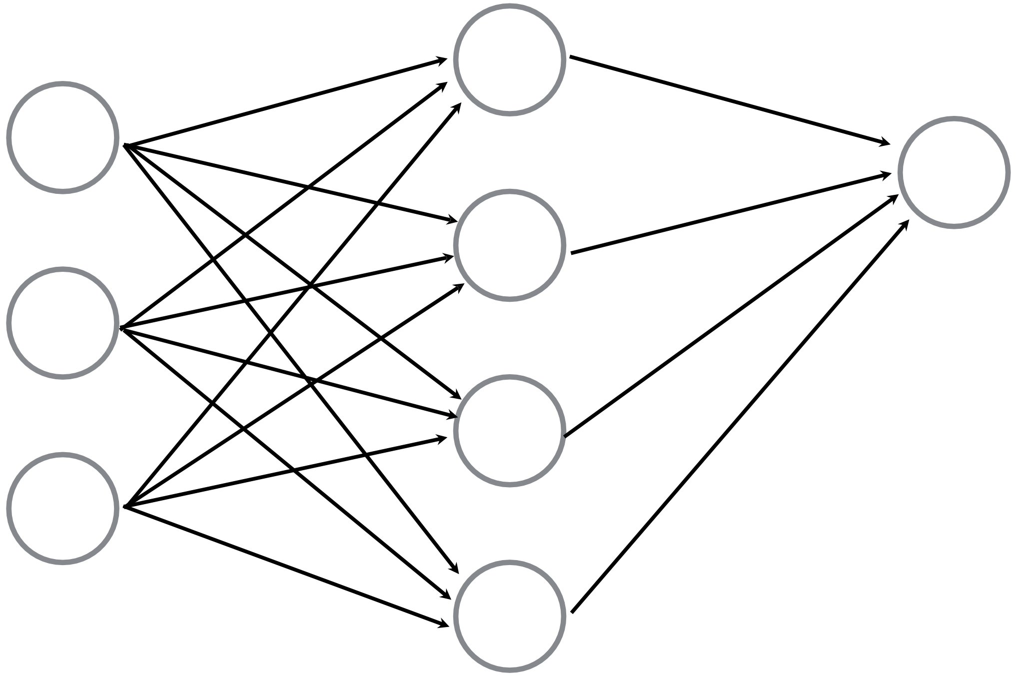
a collection of connected perceptrons



'four perceptrons'

Neural Network

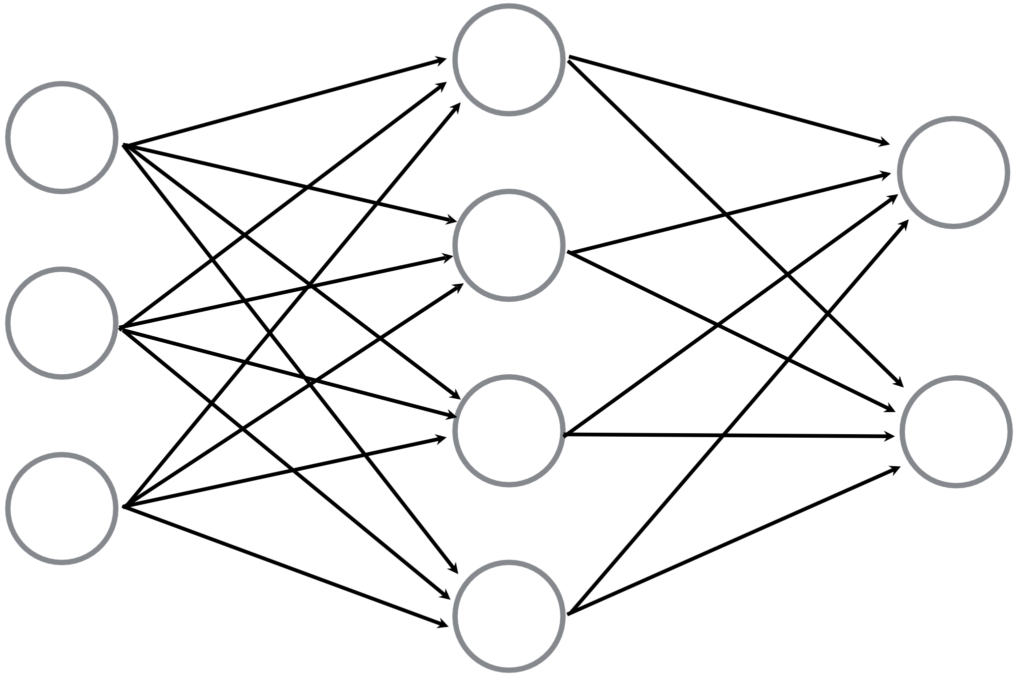
a collection of connected perceptrons



'five perceptrons'

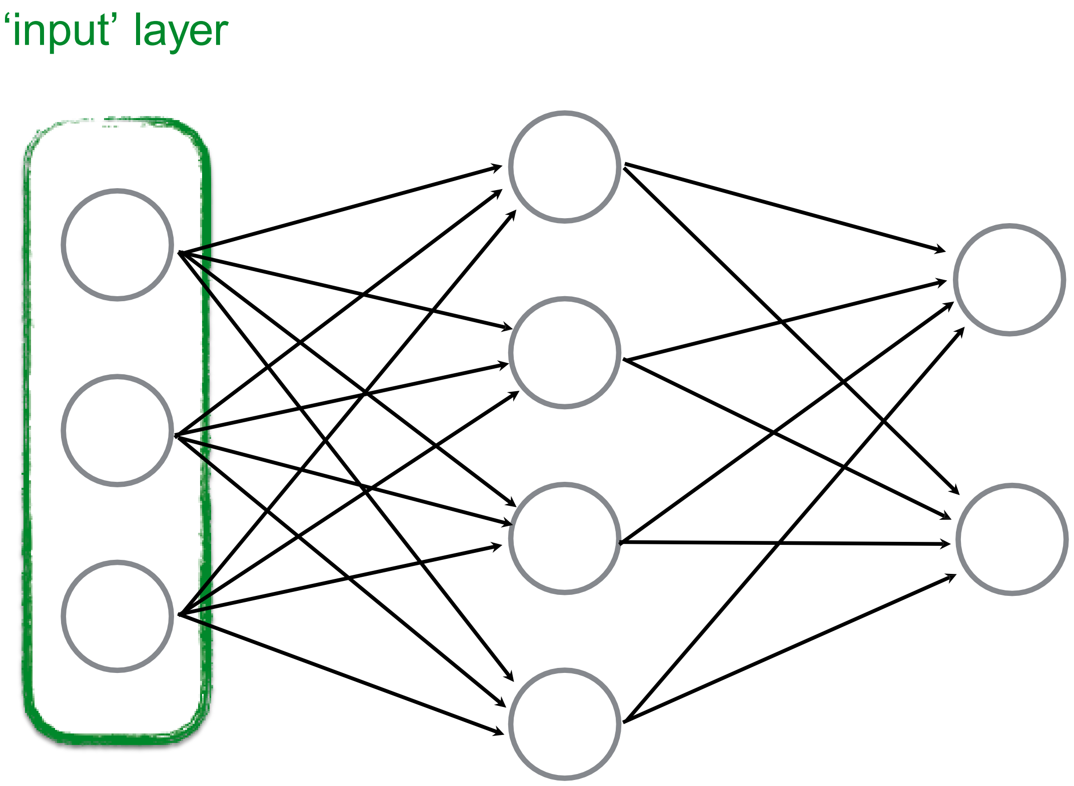
Neural Network

a collection of connected perceptrons



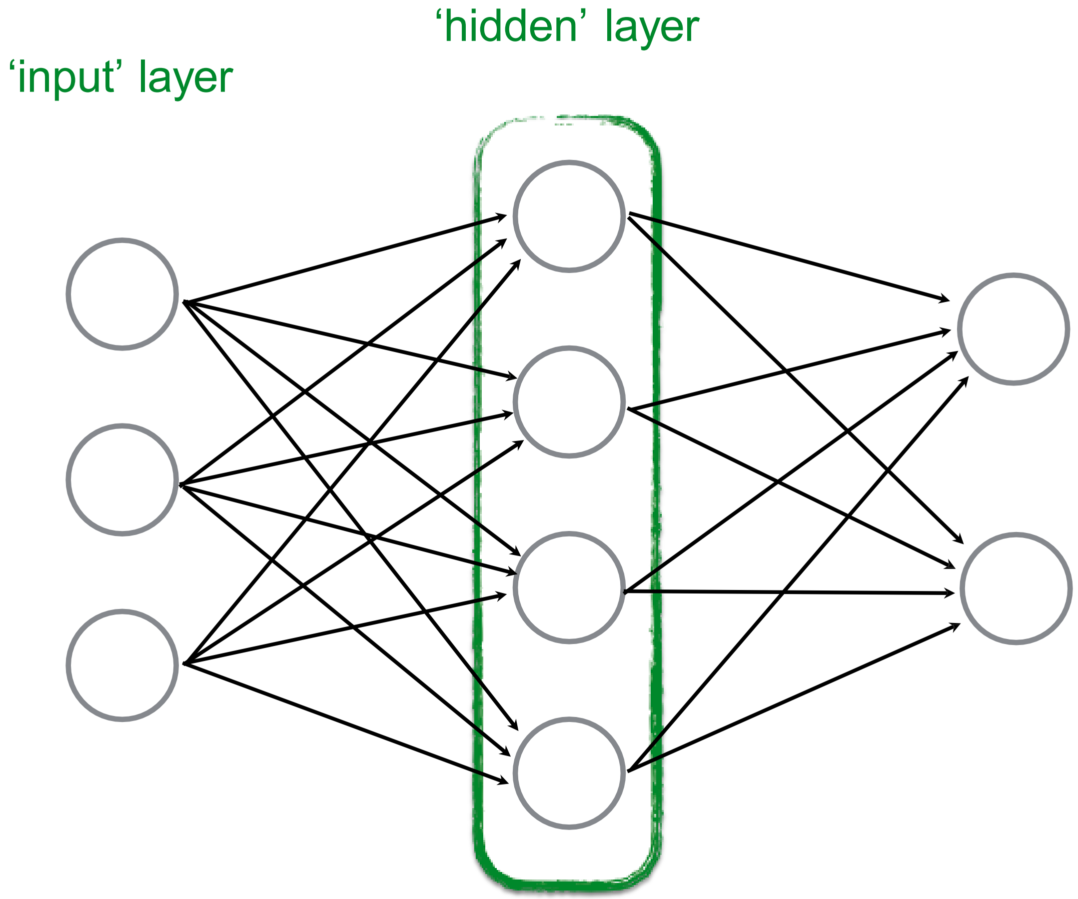
'six perceptrons'

Some terminology...



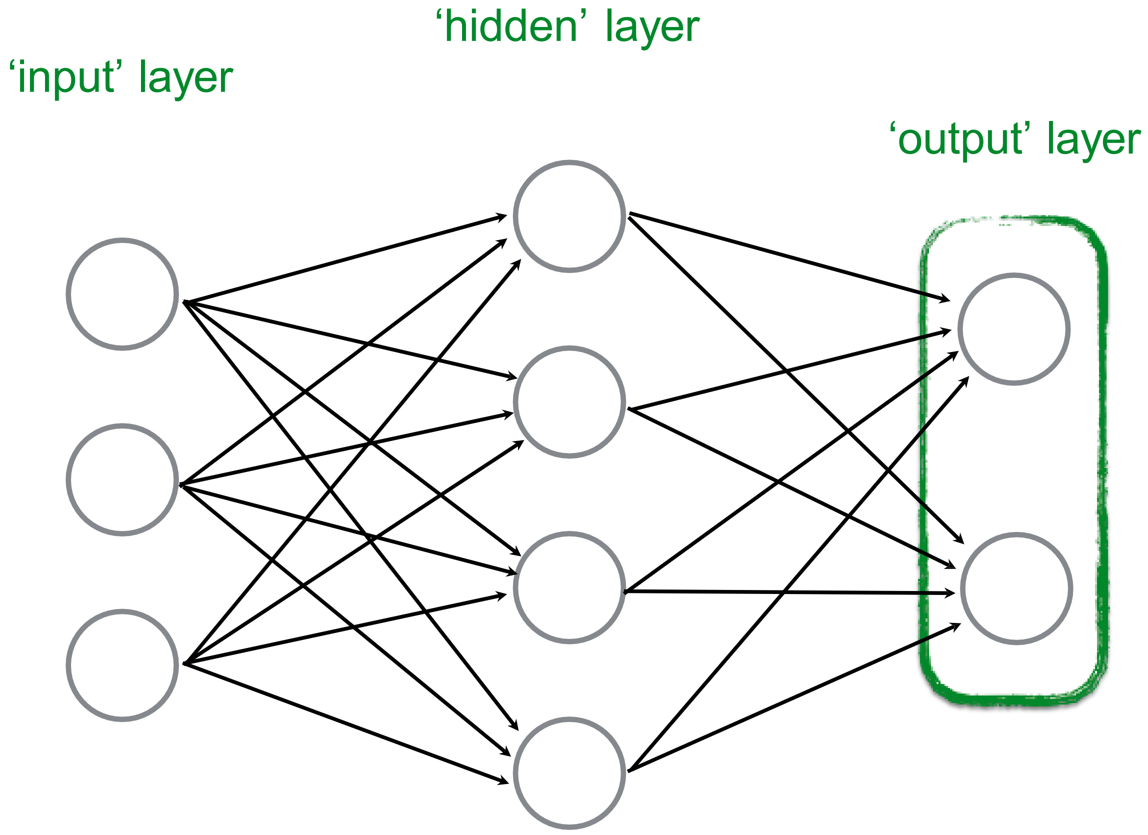
…also called a **Multi-layer Perceptron** (MLP)

Some terminology...



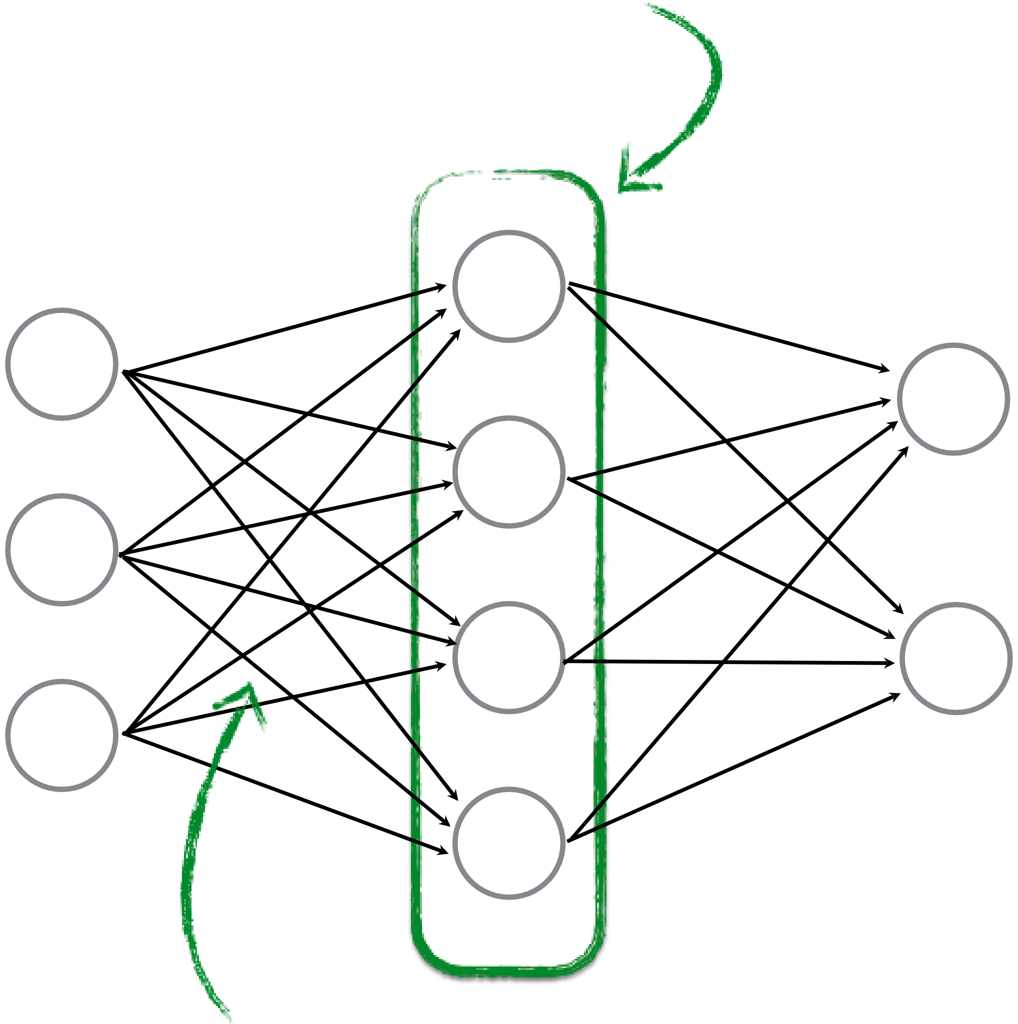
...also called a **Multi-layer Perceptron** (MLP)

Some terminology...



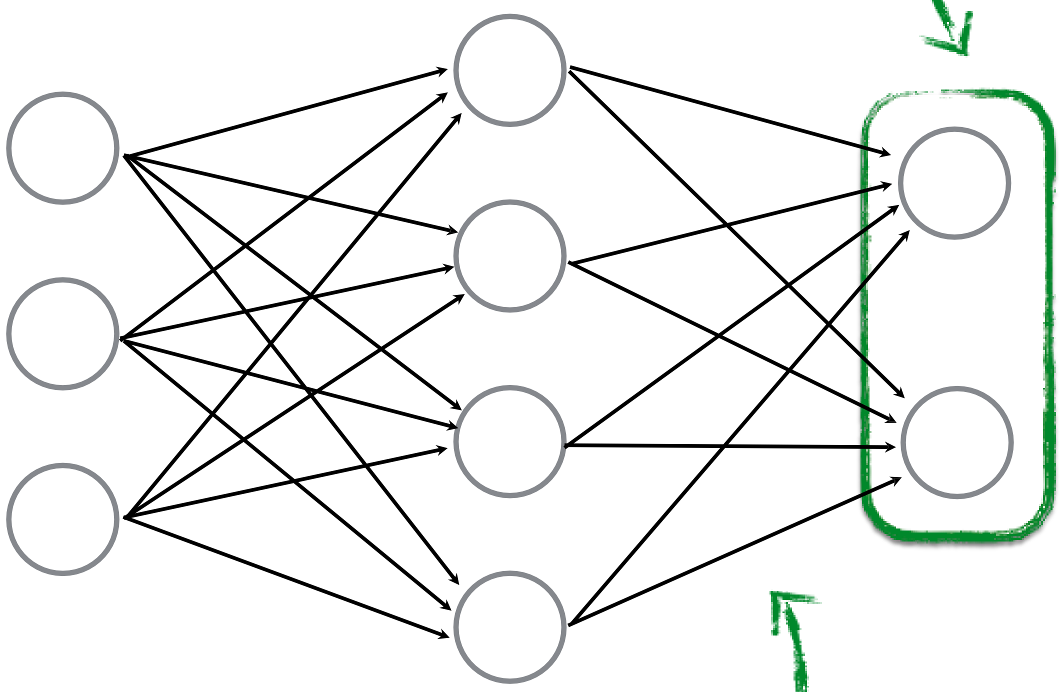
...also called a **Multi-layer Perceptron** (MLP)

this layer is a ‘fully connected layer’



all pairwise neurons between layers are connected

so is this

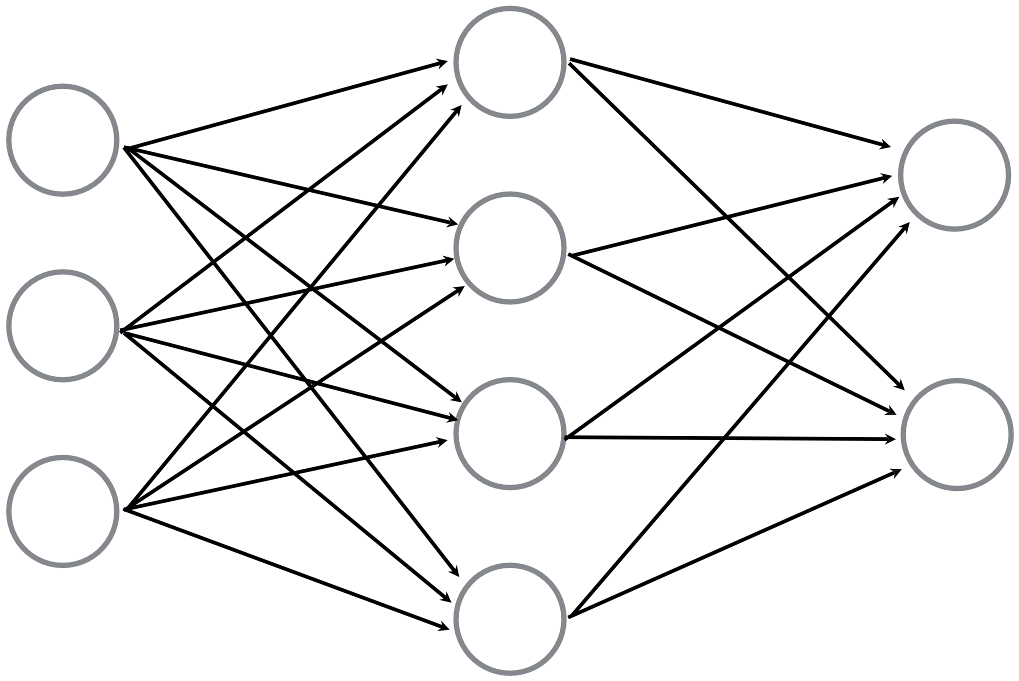


all pairwise neurons <u>between</u> layers are connected

Neural Network

How many neurons (perceptrons)?

How many weights (edges)?

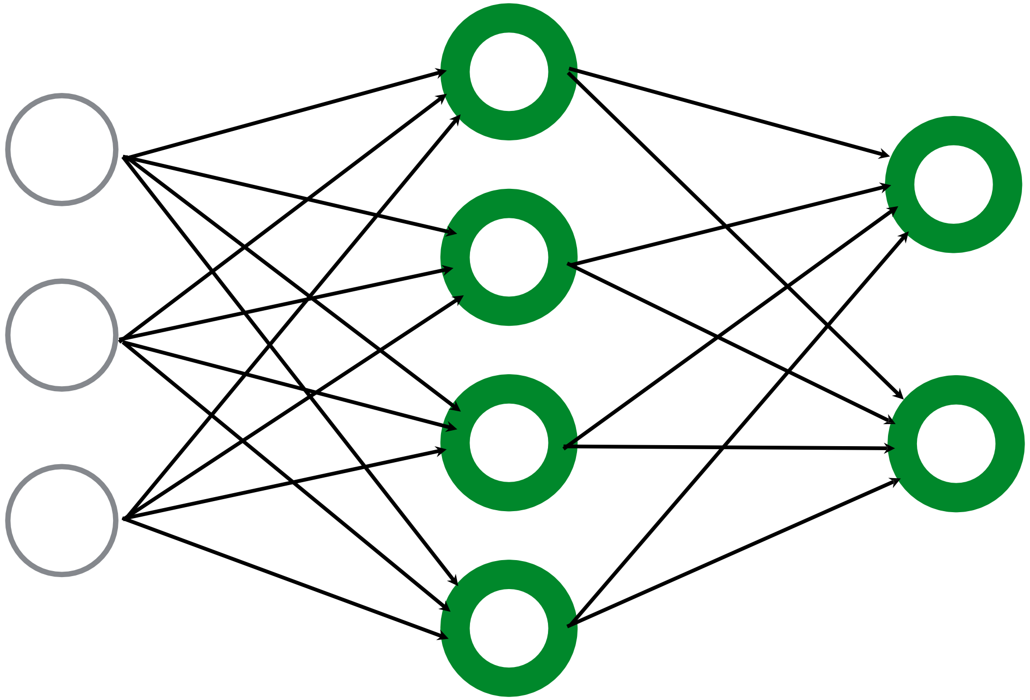


How many learnable parameters total?

Neural Network

How many neurons (perceptrons)? 4 + 2 = 6

How many weights (edges)?

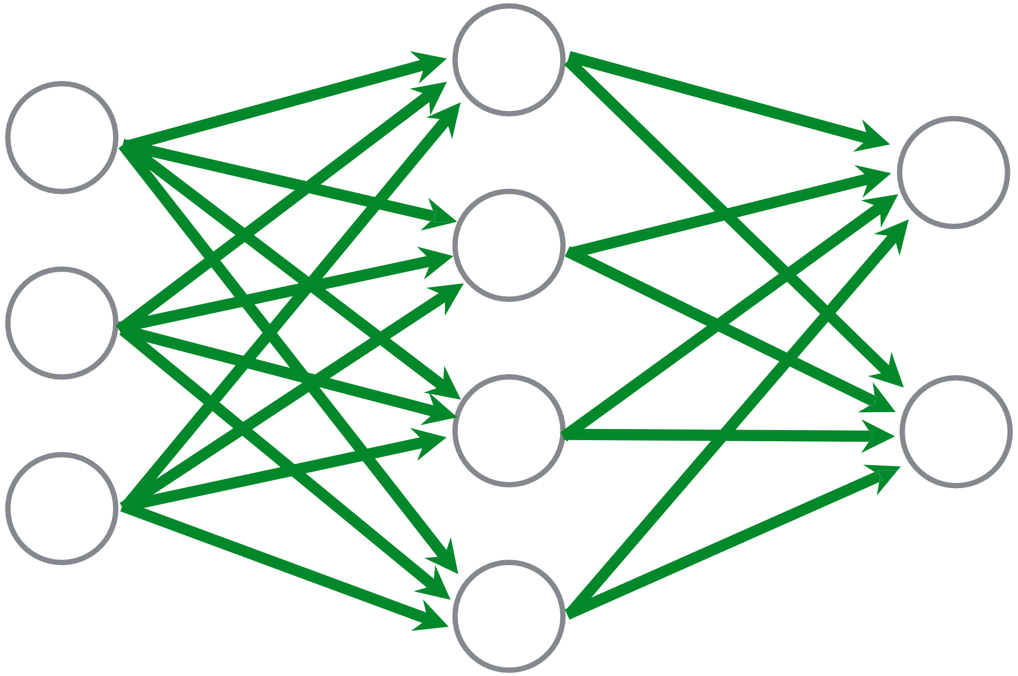


How many learnable parameters total?

Neural Network

How many neurons (perceptrons)? 4 + 2 = 6

How many weights (edges)? (3 x 4) + (4 x 2) = 20

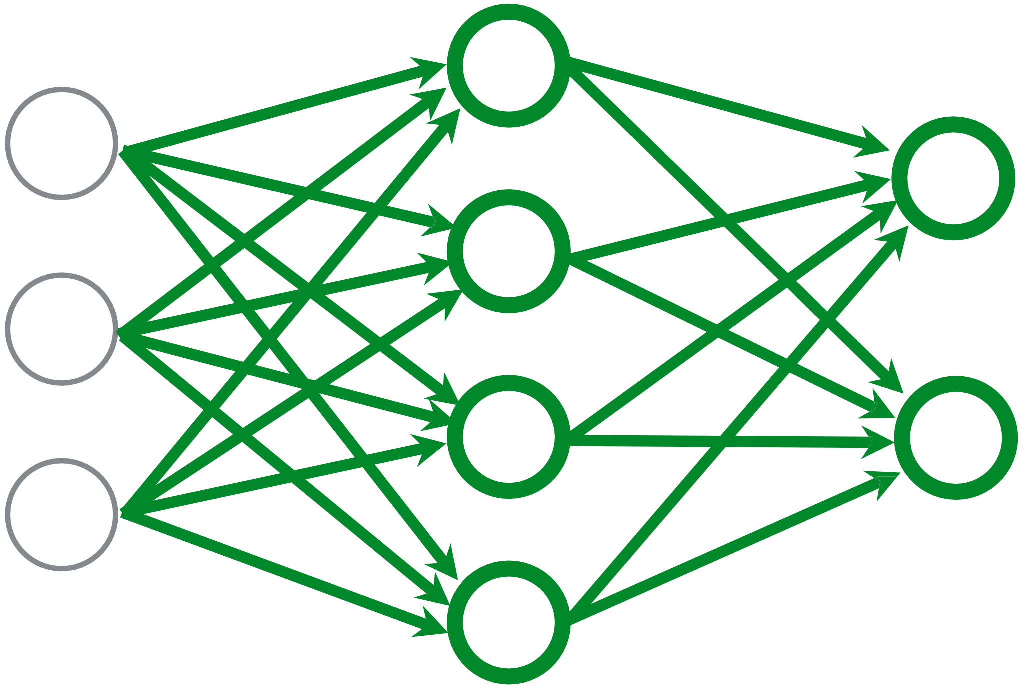


How many learnable parameters total?

Neural Network

How many neurons (perceptrons)? 4 + 2 = 6

How many weights (edges)? (3 x 4) + (4 x 2) = 20

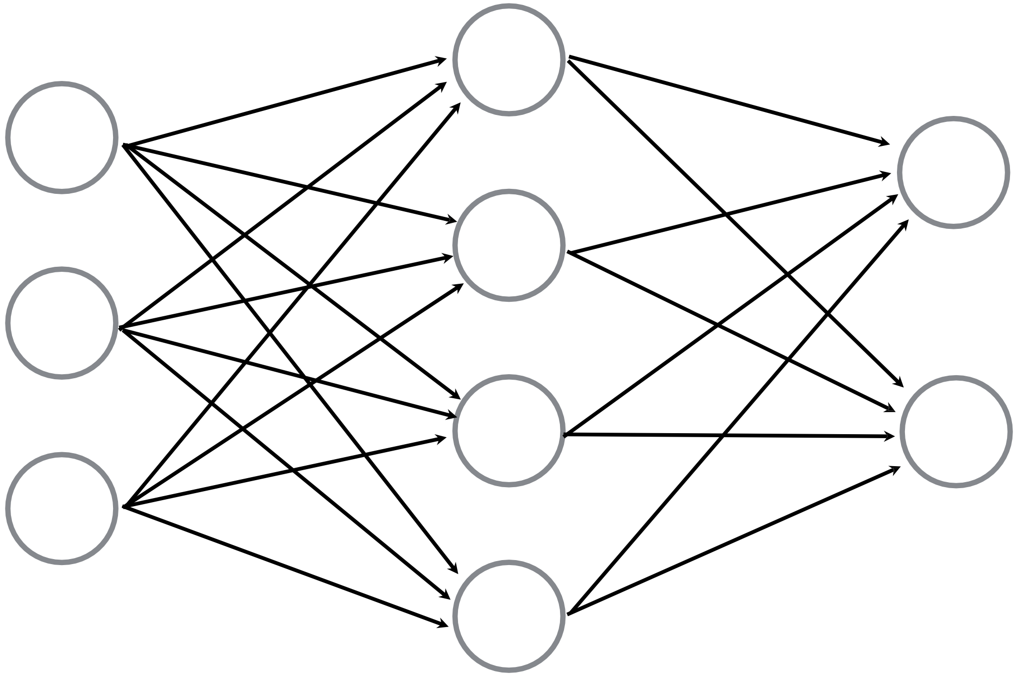


How many learnable parameters total? 20 + 4 + 2 = 26

bias terms

Neural Network

performance usually tops out at 2-3 layers, deeper networks don’t really improve performance...



...with the exception of **convolutional** networks for images