Intelligenza Artificiale per Sistemi Industriali

Lezione 04 — 4 Marzo 2026

Alice Plebe

www.aliceplebe.com alice.plebe@unitn.it

1 Go to **wooclap.com**

2 Enter the event code in the top banner

Event code

MDDNSD

OBIETTIVI

- Il neurone artificiale e le prime ANN
- Unorganized machines, e perceptron
- Fallimenti e riprese, progetto PDP
- Backpropagation

Prime reti artificiali

TURING, ANCORA UNA VOLTA

- Alan Turing: *Intelligent Machinery,* 1948
- Macchine che si **modificano** e **apprendono**
- Connessione con il cervello umano

Intelligent Machinery

A. M. Turing
[1912—1954]

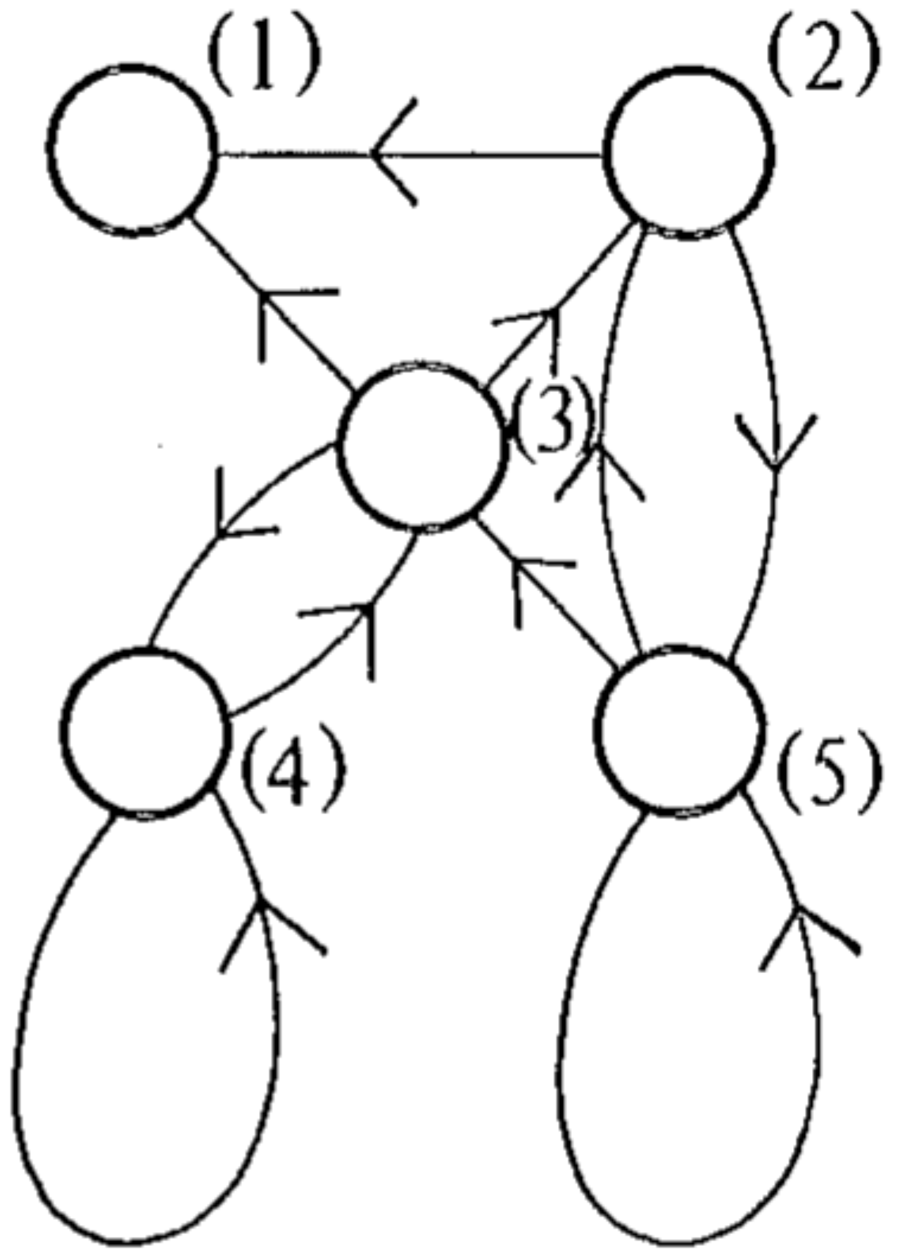
Abstract

The possible ways in which machinery might be made to show intelligent behaviour are discussed. The analogy with the human brain is used as a guiding principle. It is pointed out that the potentialities of the human intelligence can only be realized if suitable education is provided. The investigation mainly centres round an analogous teaching process applied to machines. The idea of an unorganized machine is defined, and it is suggested that the infant human cortex is of this nature. Simple examples of such machines are given, and their education by means of rewards and punishments is discussed. In one case the education process is carried through until the organization is similar to that of an ACE.

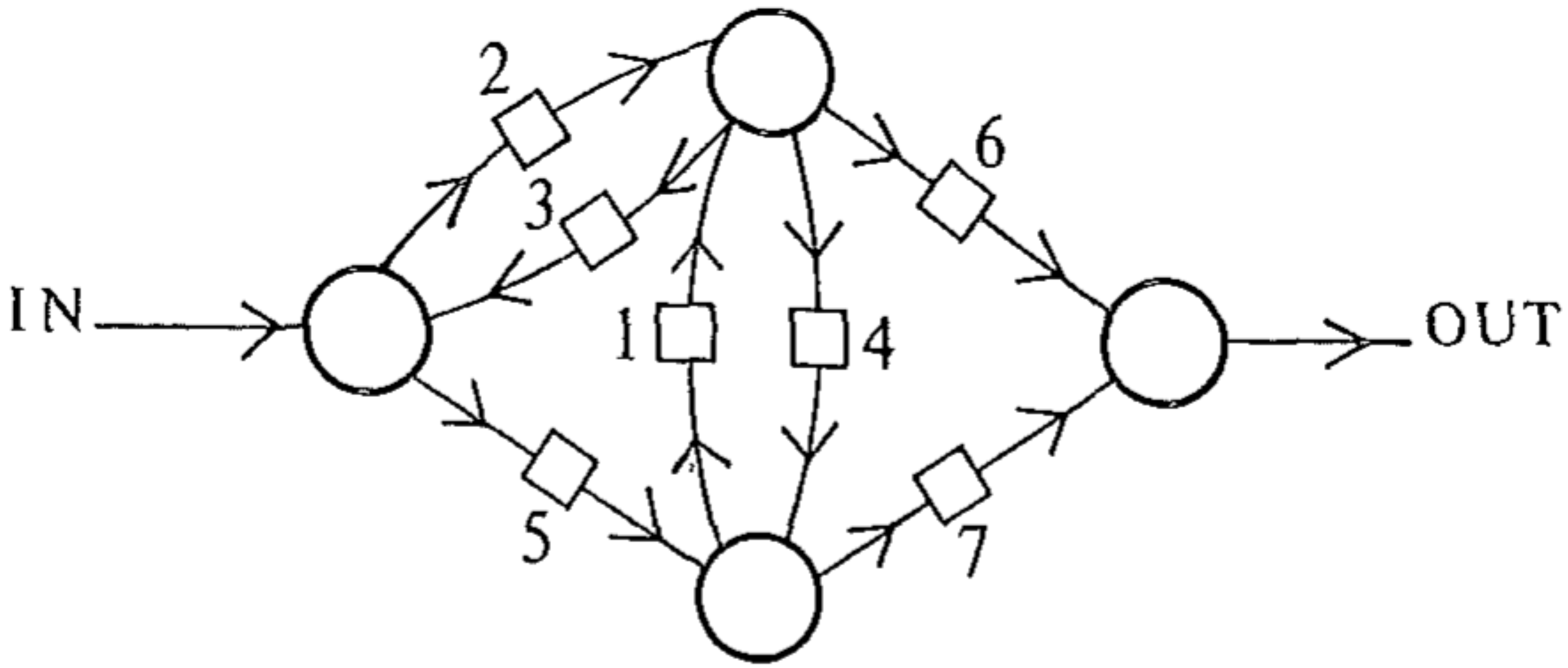
UNORGANIZED MACHINES

- Reti di elementi semplici senza particolare organizzazione iniziale
- L'organizzazione emerge attraverso l'apprendimento.

A-type



B-type



LA VISIONE DI TURING

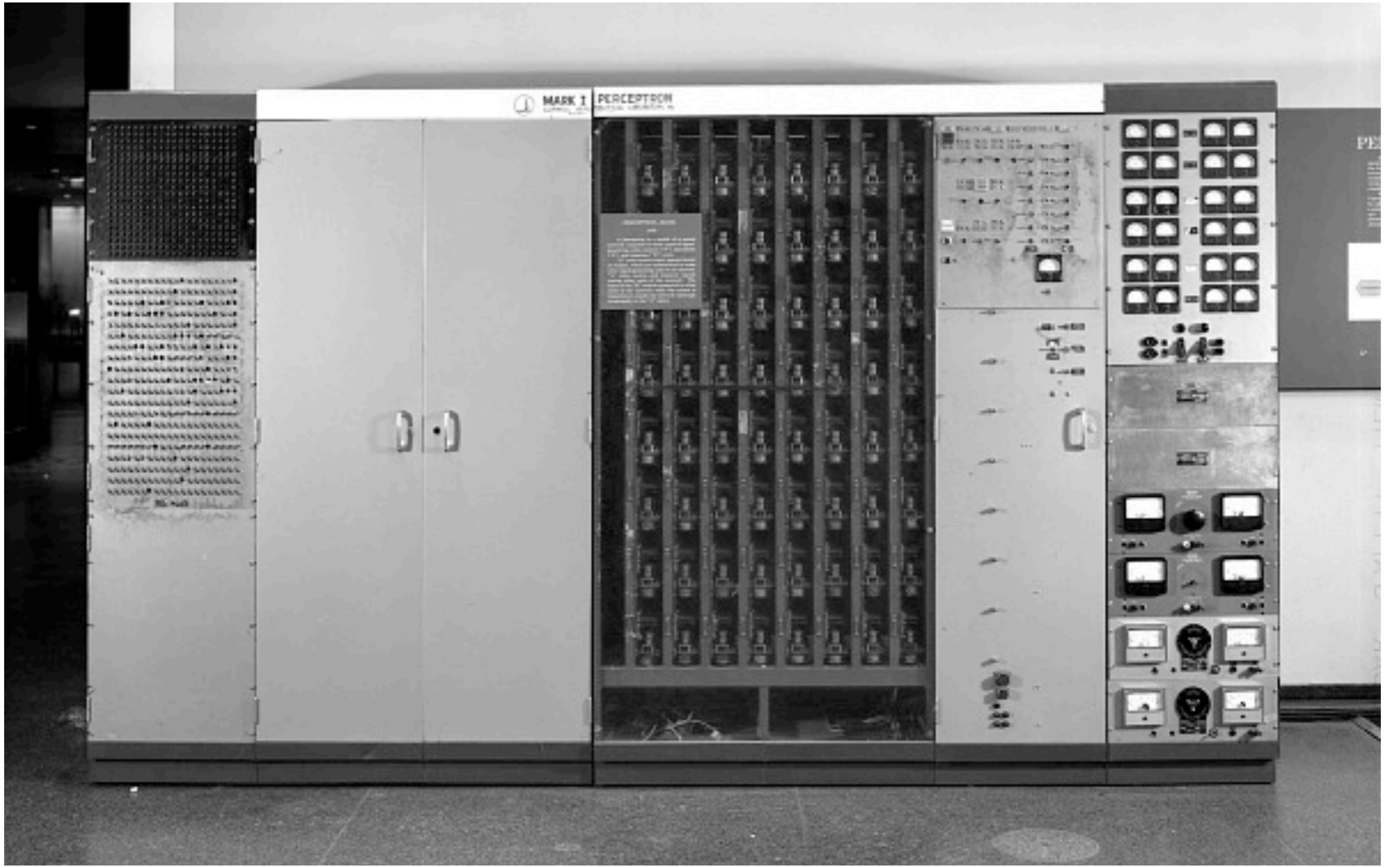
- Intelligenza distribuita, rete di unità semplici
- Apprendere è modificare connessioni
- Distinzione tra architettura e training
- Reward / punishment come controllo dell’apprendimento
- Cervello non come imitazione fedele, ma ispirazione ingegneristica

COSA MANCA

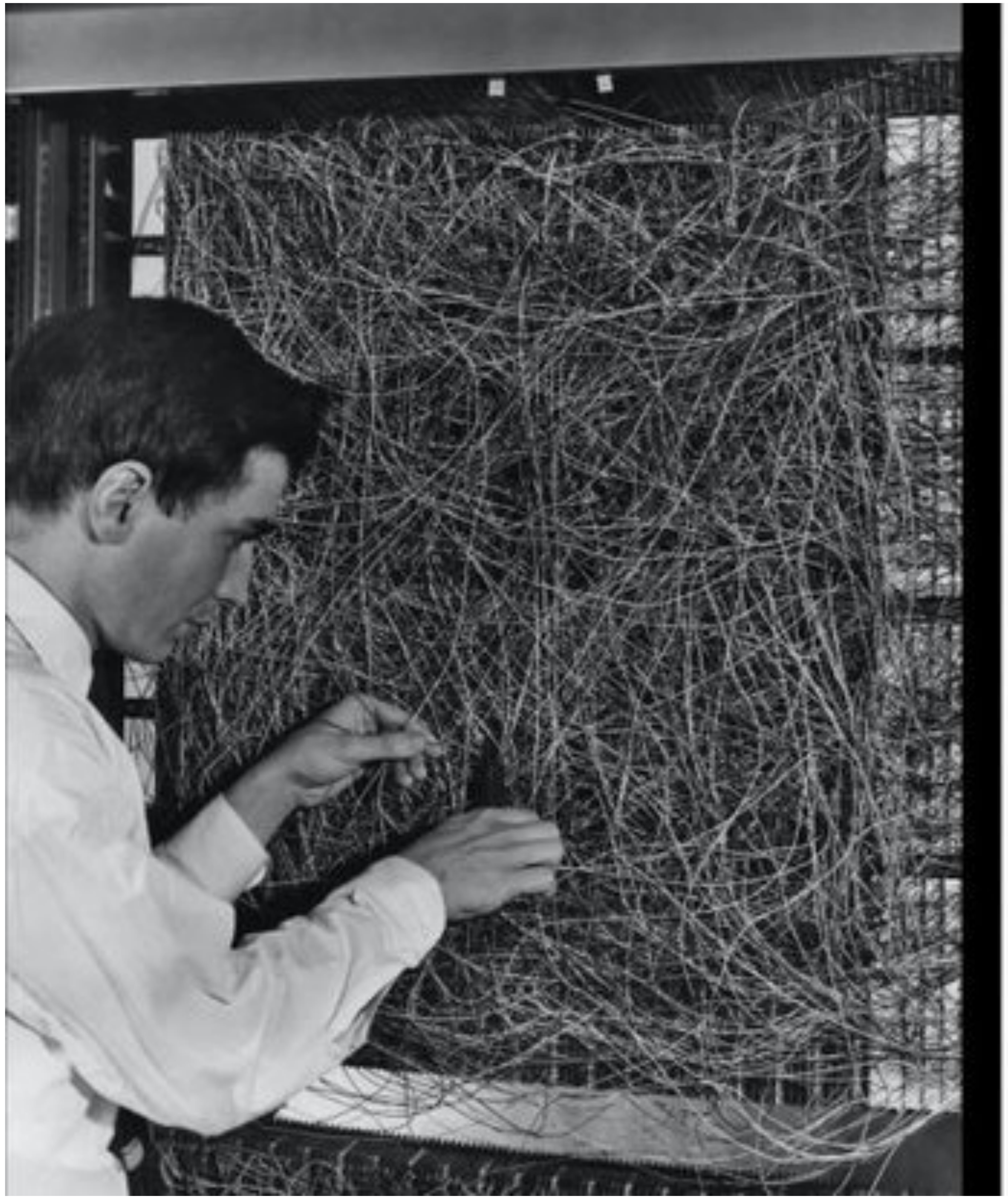
- Modello matematico di neurone artificiale
- Regola formale di apprendimento
- Dimostrazione che il sistema può imparare cose sofisticate

PERCEPTRON

- Frank Rosenblatt, 1957–1958
- Primo modello di neurone addestrabile in hardware



MARK I PERCEPTRON



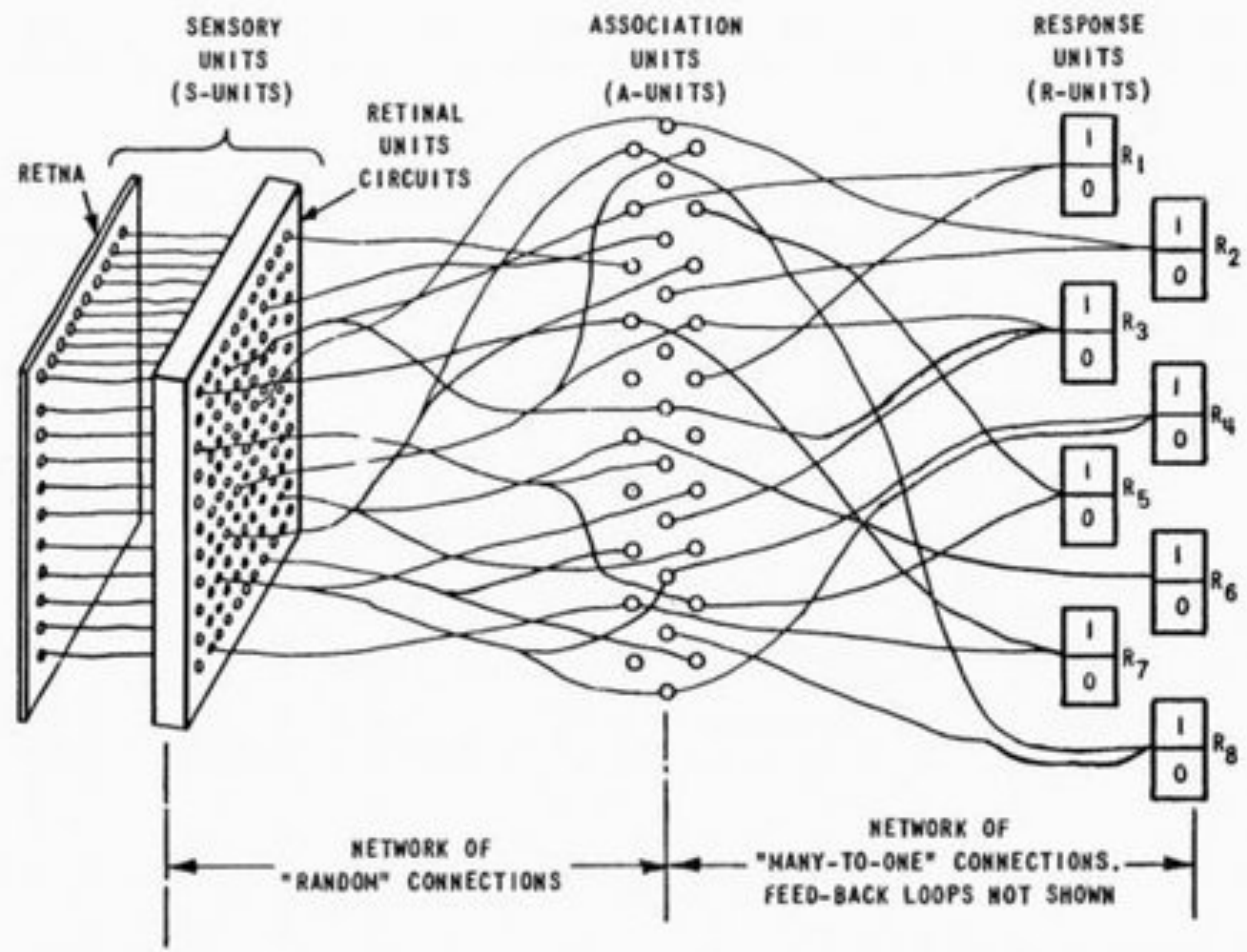
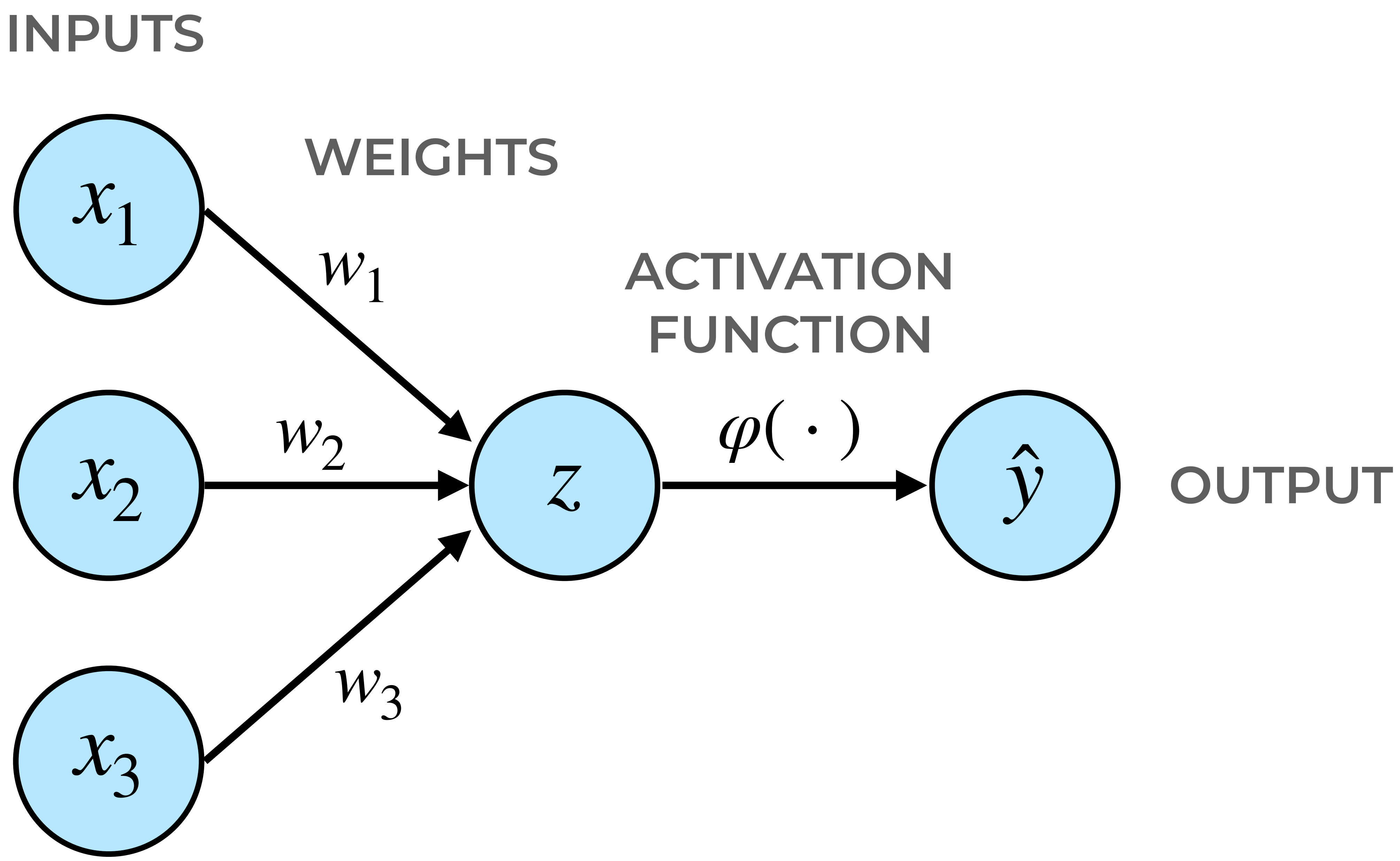


Figure I ORGANIZATION OF THE MARK I PERCEPTRON

PERCEPTRON

- Più input
- Pesi "sinaptici"
- Somma + bias
- Funzione di attivazione a soglia (action potential)
- Singolo output binario
- Classificatore lineare



$$z = \mathbf{w}^\top \mathbf{x} + b$$

$$\hat{y} = \varphi(z)$$

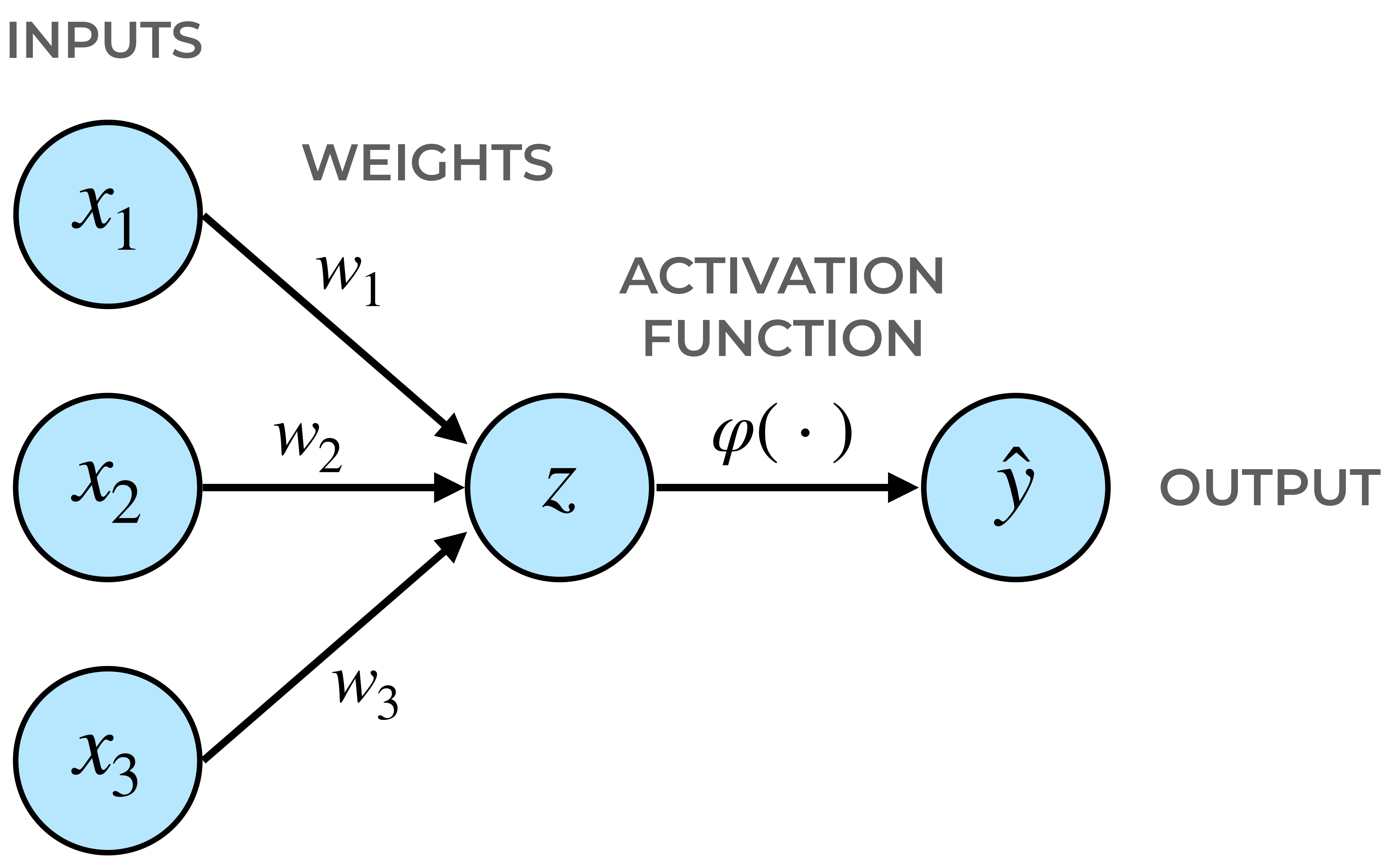
PERCEPTRON LEARNING RULE

- Aggiornamento dei pesi
  in base all'errore di classificazione

- Convergenza garantita
  nel caso lineare

$$w_i \leftarrow w_i + \eta \left(y - \hat{y}\right) x_i$$

learning rate   target   output



CRISI DEL PERCEPTRON

- Aspettative estremamente ottimistiche nel pubblico


Vol. VI, No. 2, Summer 1958

research trends

CORNELL AERONAUTICAL LABORATORY, INC., BUFFALO 21, NEW YORK

The Design of an Intelligent AUTOMATON

by FRANK ROSENBLATT

Introducing the perceptron — A machine which senses, recognizes, remembers, and responds like the human mind.

STORIES about the creation of machines having human qualities have long been a fascinating province in the realm of science fiction. Yet we are now about to witness the birth of such a machine — a machine capable of perceiving, recognizing, and identifying its surroundings without any human training or control.

Development of that machine has stemmed from a

First, in recent years our knowledge of the functioning of individual cells in the central nervous system has vastly increased.

Second, large numbers of engineers and mathematicians are, for the first time, undertaking serious study of the mathematical basis for thinking, perception, and the handling of information by the central nervous sys-

- Marvin Minsky & Seymour Papert: *Perceptrons*, 1969

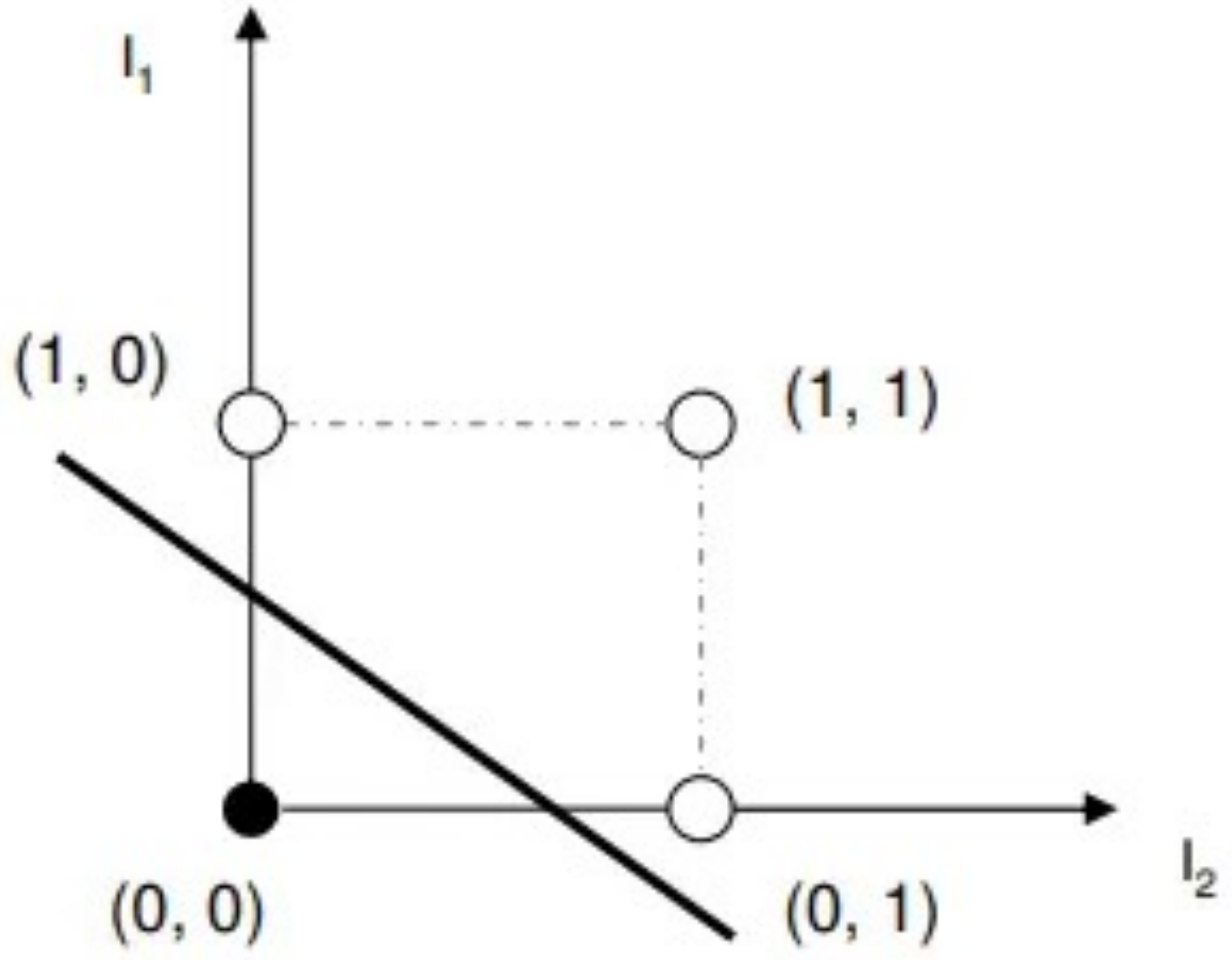
- Critica matematica rigorosa

- Scatena il primo AI winter

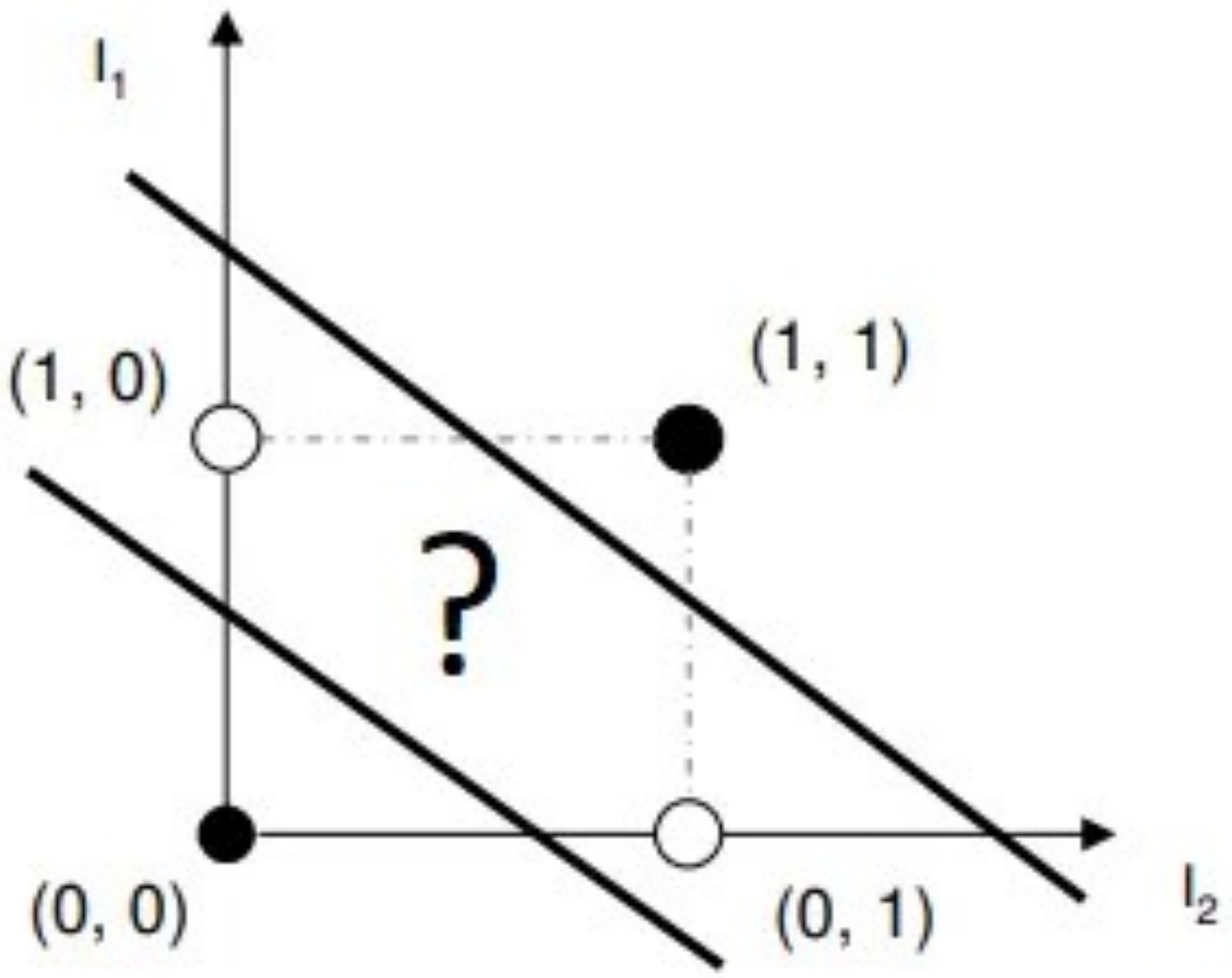
LIMITI DEL PERCEPTRON

- Solo problemi linearmente separabili
- Impossibilità di apprendere XOR
- Reti multistrato non addestrabili
- Training solo dell’ultimo strato

OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



AI WINTER PER LE RETI NEURALI

- Impatto devastante sul connessionismo
- Riduzione di finanziamenti e ricerca
- Abbandono quasi totale delle ANN
- Dominio dell’AI simbolica

(Prima) rinascita delle ANN

PDP PROJECT

- Emergono i limiti della AI simbolica

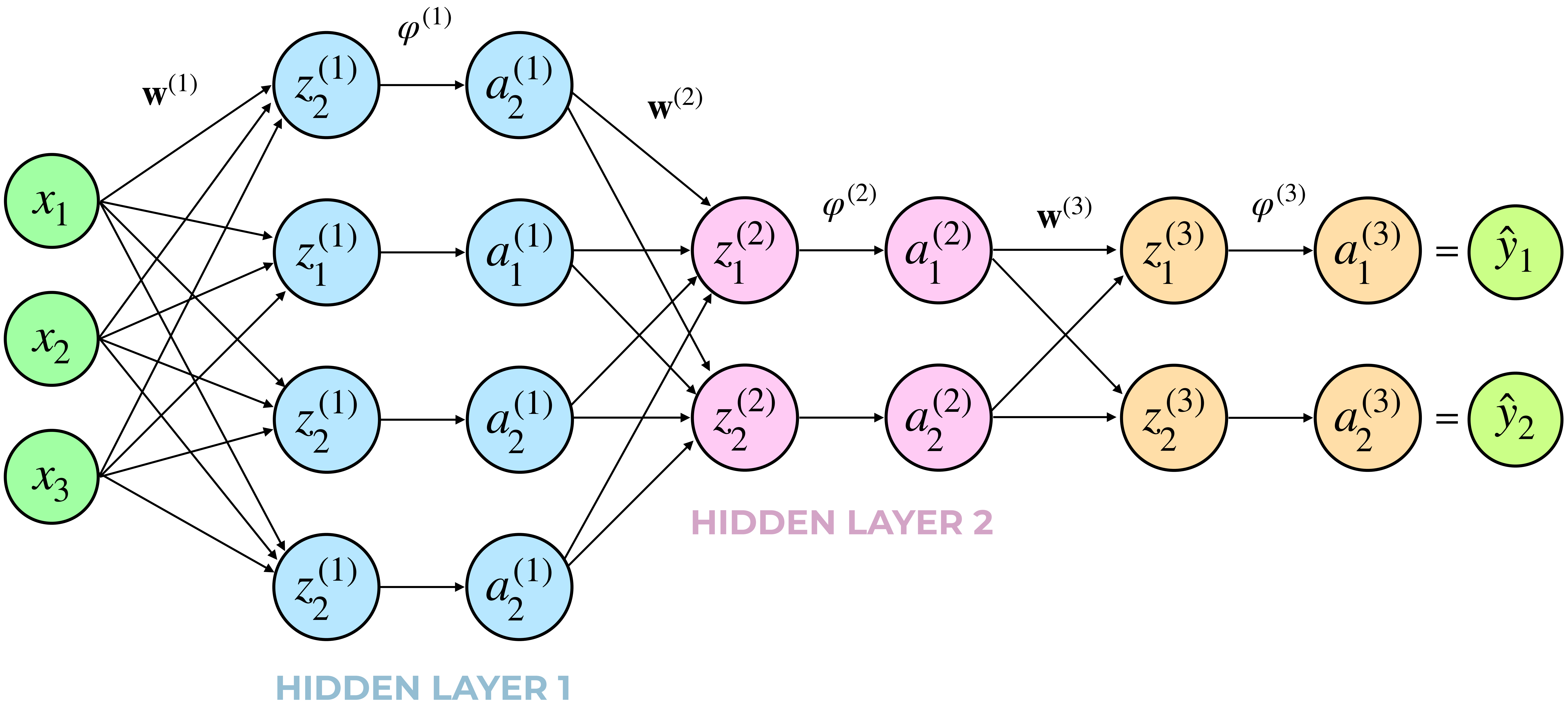
- Rumelhart & McClelland:
  Parallel Distributed Processing, metà '80

- Teoria della cognizione:

PROCESSING	→	*Interactions among*
DISTRIBUTED	→	*many simple, interconnected, neuron-like units*
PARALLEL	→	*that work simultaneously.*

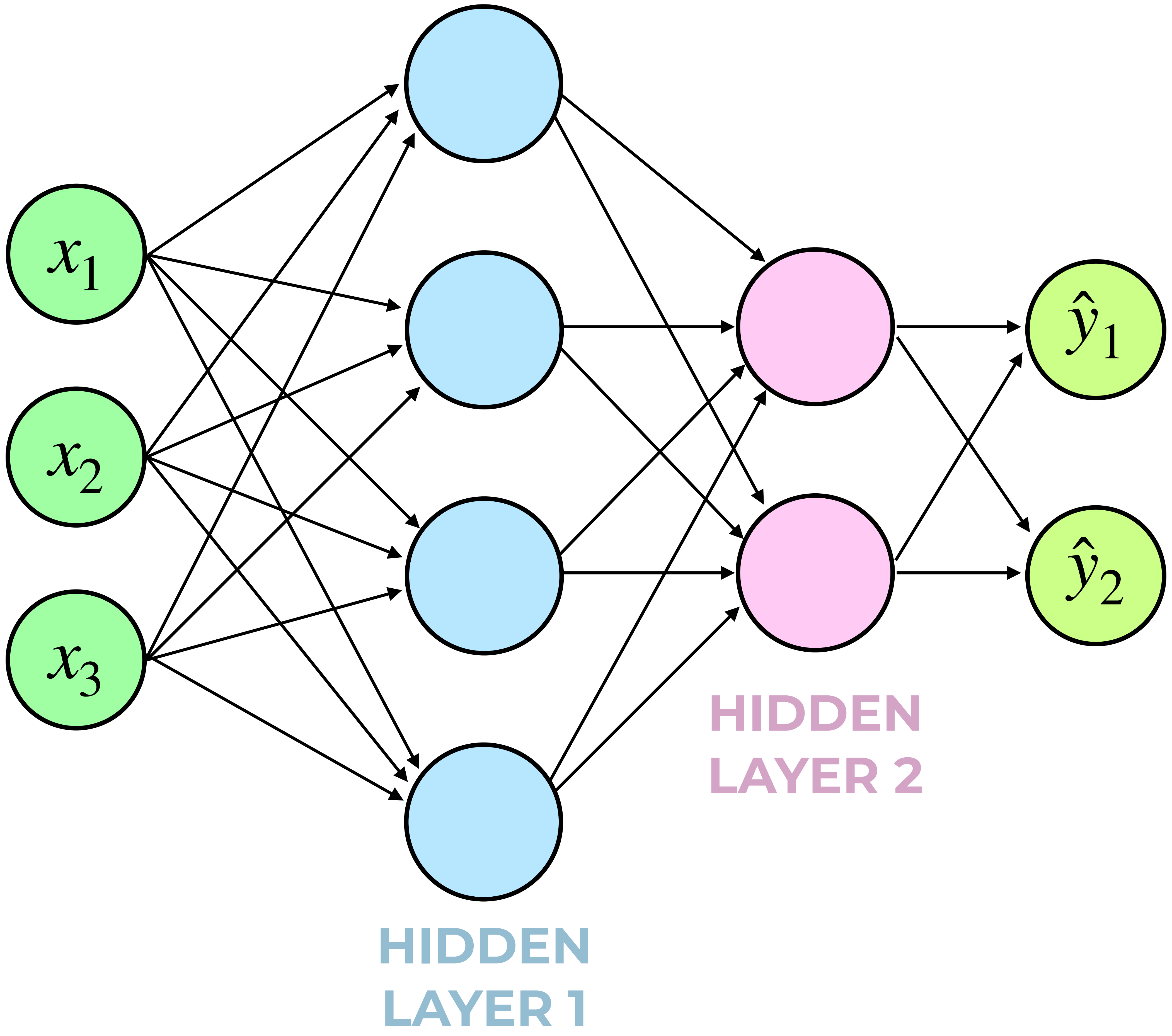
PDP PROJECT

- Obiettivo del progetto: trovare un meccanismo di apprendimento per **multi-layer perceptron (MLP)**



PDP PROJECT

- Obiettivo del progetto: trovare un meccanismo di apprendimento per **multi-layer perceptron (MLP)**



Learning representations by back-propagating errors

1986

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

BACKPROPAGATION (1/7)

- Problema: come modificare i pesi interni in funzione dell'errore finale?

- Obiettivo: minimizzare una funzione di errore $E\left(\hat{y}, y\right)$ rispetto ai pesi

- Approccio: discesa del gradiente $\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n}\right)$

- Regola di aggiornamento generale: $w \leftarrow w - \eta \frac{\partial E}{\partial w}$

- Nuovo problema: come calcolare $\frac{\partial E}{\partial w_i}$ per tutti i pesi di una rete multistrato?

BACKPROPAGATION (2/7)

- Soluzione: regola della catena

- Caso semplice: singolo neurone, singolo strato

$$z = wx + b$$

$$x \xrightarrow{w} z \xrightarrow{f(\cdot)} \hat{y}$$

$$\hat{y} = f(z)$$

$$E = \frac{1}{2}\left(\hat{y} - y\right)^2$$

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \hat{y}}\ \frac{\partial \hat{y}}{\partial z}\ \frac{\partial z}{\partial w} = \left(\hat{y} - y\right)\, f'(z)\, x$$

$$w \leftarrow w - \eta\left(\hat{y} - y\right)\, f'(z)\, x$$

BACKPROPAGATION (3/7)

- Definizione: quantità di errore locale δ

$$x \xrightarrow{\;w\;} z \xrightarrow{\;f(\,\cdot\,)\;} \hat{y}$$

$$\delta = \frac{\partial E}{\partial z} = \left(\hat{y} - y\right) f'(z)$$

$$\frac{\partial E}{\partial w} = \delta x$$

BACKPROPAGATION (4/7)

- Caso: rete a due strati, un neurone per strato

$$x \xrightarrow{w^{(1)}} z^{(1)} \xrightarrow{f^{(1)}} a^{(1)} \xrightarrow{w^{(2)}} z^{(2)} \xrightarrow{f^{(2)}} a^{(2)} = \hat{y}$$

$$z^{(1)} = w^{(1)}x + b^{(1)} \qquad z^{(2)} = w^{(2)}a^{(1)} + b^{(2)}$$

$$a^{(1)} = f^{(1)}\left(z^{(1)}\right) \qquad a^{(2)} = \hat{y} = f^{(2)}\left(z^{(2)}\right)$$

- Obiettivo: calcolare $\dfrac{\partial E}{\partial w^{(1)}}$ e $\dfrac{\partial E}{\partial w^{(2)}}$

BACKPROPAGATION (5/7)

- Caso: rete a due strati, un neurone per strato

$$x \xrightarrow{w^{(1)}} z^{(1)} \xrightarrow{f^{(1)}} a^{(1)} \xrightarrow{w^{(2)}} z^{(2)} \xrightarrow{f^{(2)}} a^{(2)} = \hat{y}$$

$$\frac{\partial E}{\partial w^{(2)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial w^{(2)}} = \left(\hat{y} - y\right) f'^{(2)}\left(z^{(2)}\right) \ a^{(1)}$$

$$\delta^{(2)} = \frac{\partial E}{\partial z^{(2)}} = \left(\hat{y} - y\right) f'^{(2)}\left(z^{(2)}\right)$$

$$\frac{\partial E}{\partial w^{(2)}} = \delta^{(2)} a^{(1)}$$

BACKPROPAGATION (6/7)

- Caso: rete a due strati, un neurone per strato

$$x \xrightarrow{w^{(1)}} z^{(1)} \xrightarrow{f^{(1)}} a^{(1)} \xrightarrow{w^{(2)}} z^{(2)} \xrightarrow{f^{(2)}} a^{(2)} = \hat{y}$$

$$\frac{\partial E}{\partial w^{(1)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w^{(1)}} = \left(\hat{y} - y\right) f'^{(2)}\left(z^{(2)}\right) \; w^{(2)} f'^{(1)}\left(z^{(1)}\right) \; x$$

$$\delta^{(1)} = \frac{\partial E}{\partial z^{(1)}} = \frac{\partial E}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} = \delta^{(2)} w^{(2)} f'^{(1)}\left(z^{(1)}\right)$$

$$\frac{\partial E}{\partial w^{(1)}} = \delta^{(1)} x$$

BACKPROPAGATION (7/7)

- Il gradiente rispetto a un peso è l'errore locale del neurone di arrivo moltiplicato per l'attivazione del neurone di partenza

$$\frac{\partial E}{\partial w_{i,j}^{(l)}} = \delta_j^{(l+1)}\, a_i^{(l)}$$

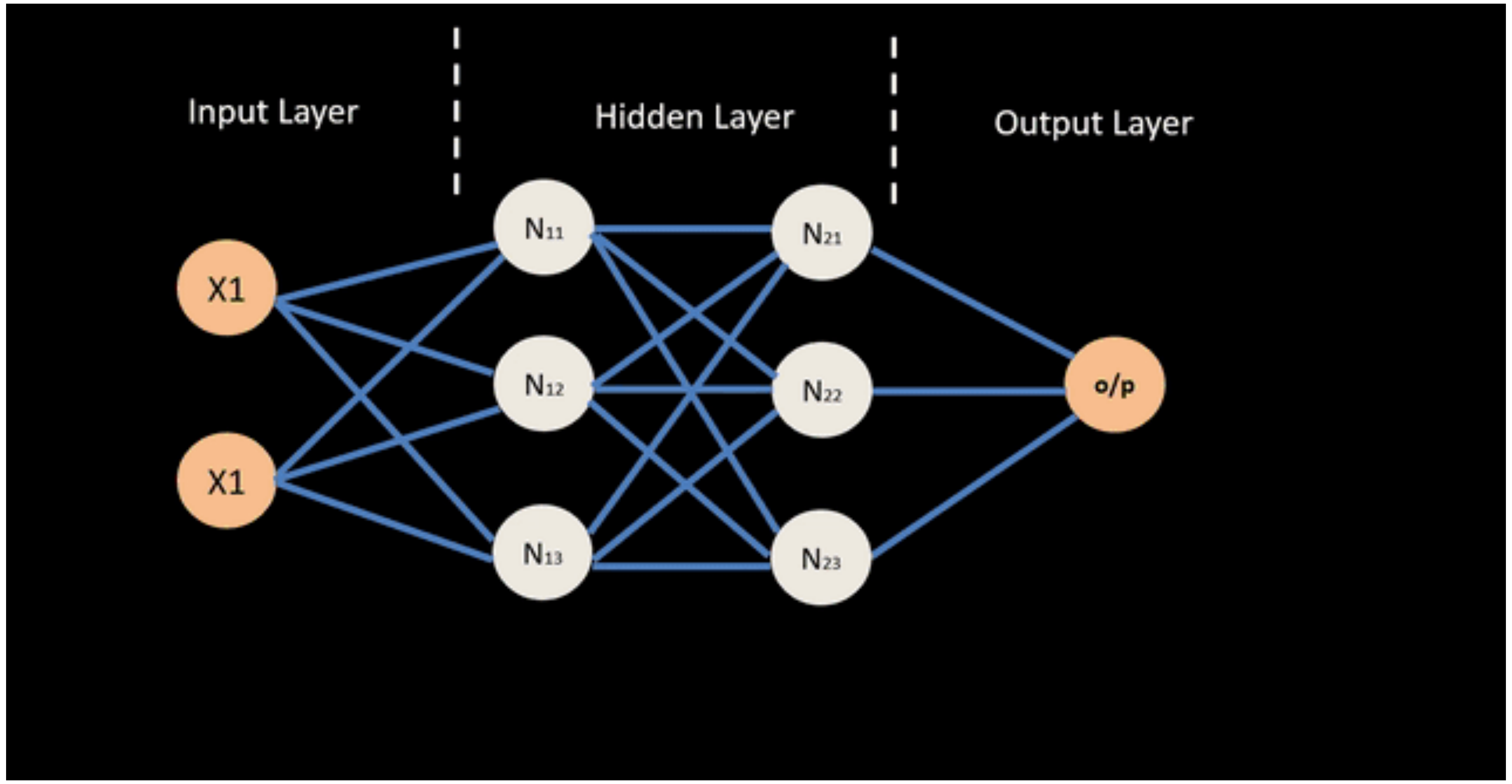
dove $w_{i,j}^{(l)}$ è il peso che collega il neurone i dello strato l al neurone j dello strato $l+1$.

- L'errore si propaga all'indietro come combinazione lineare pesata degli errori successivi, modulata dalla derivata locale

$$\delta_i^{(l)} = f'^{(l)}\left(z_i^{(l)}\right) \sum_j w_{i,j}^{(l)}\, \delta_j^{(l+1)}$$

BACK AND FORTH

Addestramento = forward pass + backward pass



RECAP

- Il limite del Perceptron era la linearità
- Una rete con almeno uno strato nascosto e funzioni di attivazione non lineari può rappresentare funzioni non linearmente separabili
- La backpropagation rende possibile addestrare questi strati nascosti
- La backpropagation non risolve tutto, però . . .