Artificial Intelligence

Neural Networks

Lesson 4: Artificial Neural Networks

Vincenzo Piuri

Università degli Studi di Milano

Contents

Artificial Neural Networks (1)

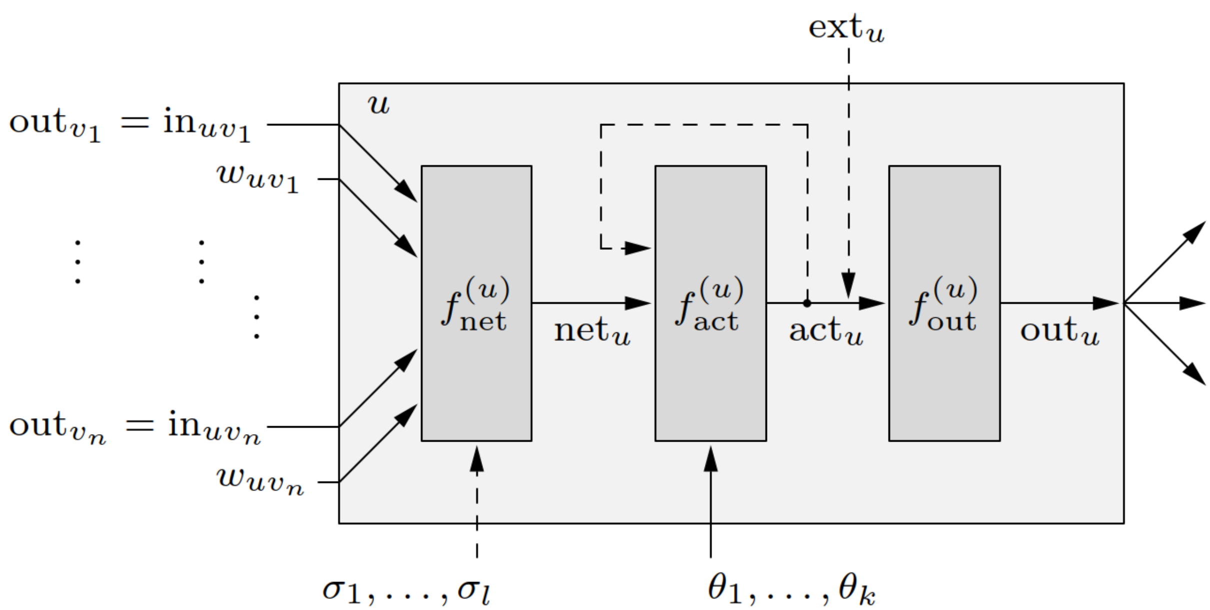
General definition of a neural network:

An (artificial) neural network is a (directed) graph $G = (U, C)$, whose vertices $u \in U$ are called neurons or units and whose edges $c \in C$ are called connections.

The set U of vertices is partitioned into

- the set U_{in} of input neurons,
- the set U_{out} of output neurons, and
- the set U_{hidden} of hidden neurons.

Artificial Neural Networks (2)

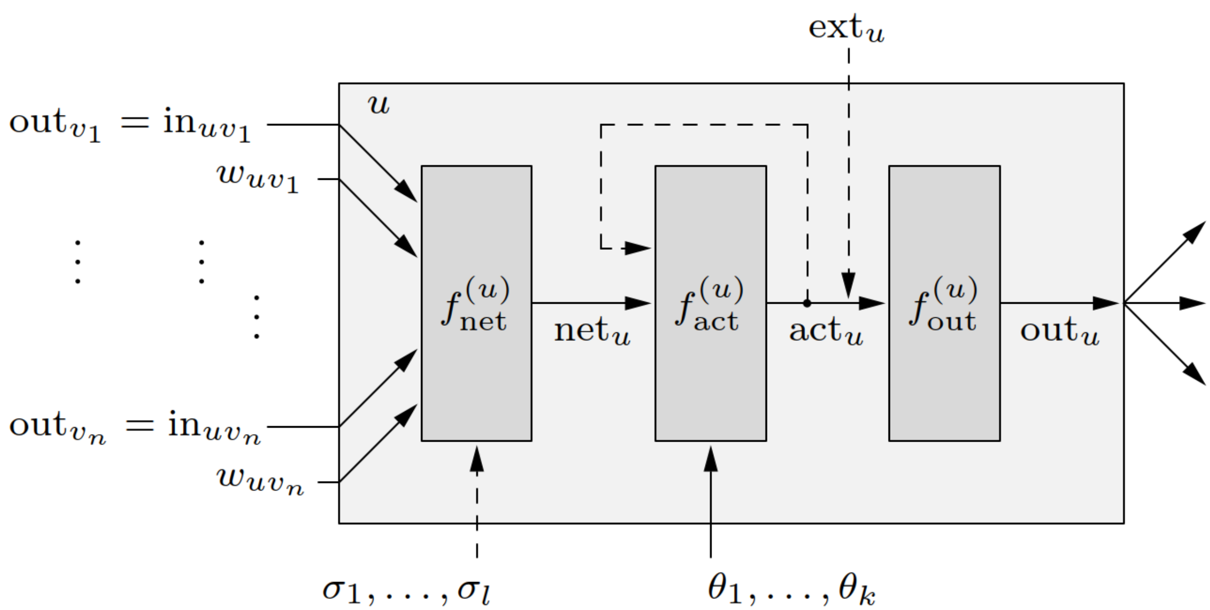


Each connection $(v, u) \in C$ possesses a weight w_{uv}

Each neuron $u \in U$ possesses three (real-valued) state variables:

- the network input net_u
- the activation act_u
- the output out_u

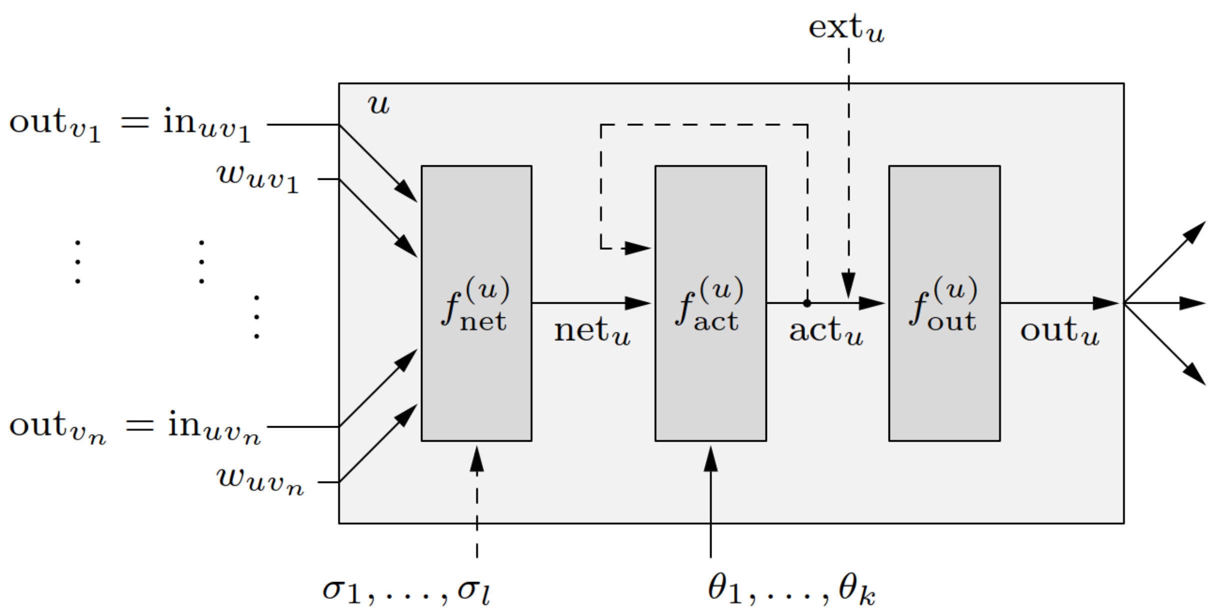
Artificial Neural Networks (3)



Each input neuron $u \in U_{in}$ also possesses a fourth (real-valued) state variable

- the external input ext_u

Artificial Neural Networks (4)



Each neuron $u \in U$ possesses three functions, used to compute the values of the state variables:

- the network input function $f_{net}^{(u)}$
- the activation function $f_{act}^{(u)}$
- the output function $f_{out}^{(u)}$

Artificial Neural Networks (5)

Types of (artificial) neural networks:

- if the graph of a neural network is acyclic, it is called a **feed-forward network**.
- if the graph of a neural network contains cycles (backward connections), it is called a **recurrent network**.

Artificial Neural Networks (6)

Operation of (artificial) neural networks:

- **Input phase**: external inputs are acquired by input neurons.
- **Work phase**: external inputs are switched off while new outputs are computed by each neuron.
  - **Recomputation** of a neuron output occurs if any of its inputs changes.
  - Temporal order of recomputation depends on the type of artificial neural network.
  - Work phase continues until the external outputs are steady, or a maximum number of recomputation iterations is reached.

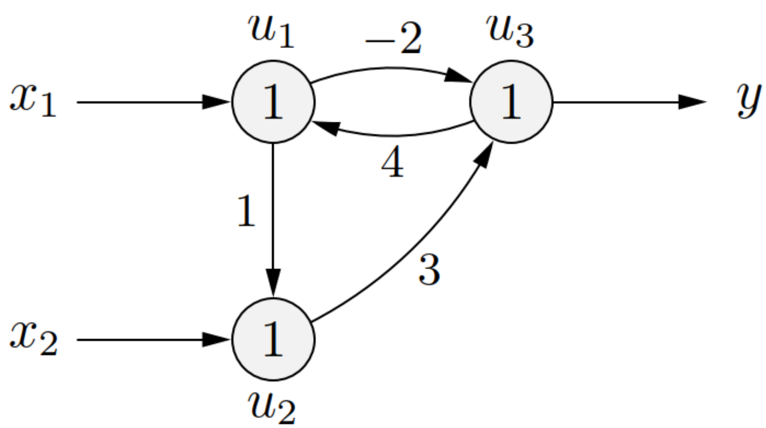
Examples of Neural Networks (1)

Feed-forward neural network

- Computation proceeds from input neurons progressively toward output neurons by following the topological order of the neuron in the network
- External inputs are frozen
- Input neuron compute their outputs which are maintained steady and forwarded to the connected neurons
- Neurons connected to preceeding neurons with steady outputs generate their respective outputs and propagated forward to the subsequent neurons, until the external outputs are generated.

Examples of Neural Networks (2)

A simple ***recurrent neural network***



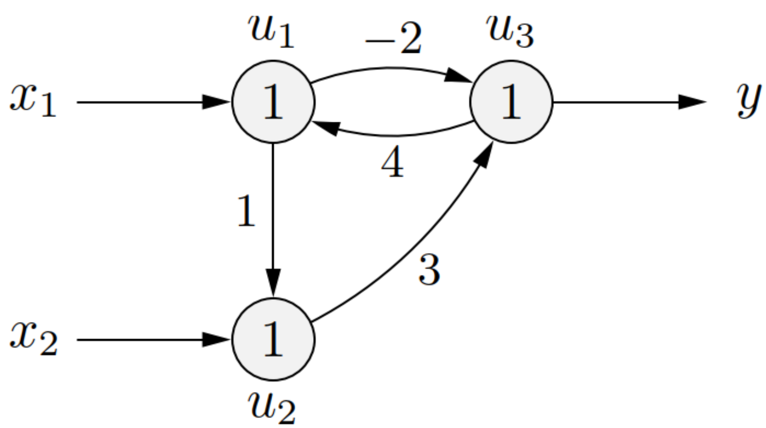
	u_1	u_2	u_3	
input phase	**1**	**0**	**0**	
work phase	1	0	**0**	$\text{net}_{u_3} = -2 \quad < 1$
	0	0	0	$\text{net}_{u_1} = 0 \quad < 1$
	0	**0**	0	$\text{net}_{u_2} = 0 \quad < 1$
	0	0	**0**	$\text{net}_{u_3} = 0 \quad < 1$
	0	0	0	$\text{net}_{u_1} = 0 \quad < 1$

- Order in which the neurons are updated:

$$u_3, u_1, u_2, u_3, u_1, u_2, u_3, \ldots$$

Examples of Neural Networks (3)

A simple recurrent neural network

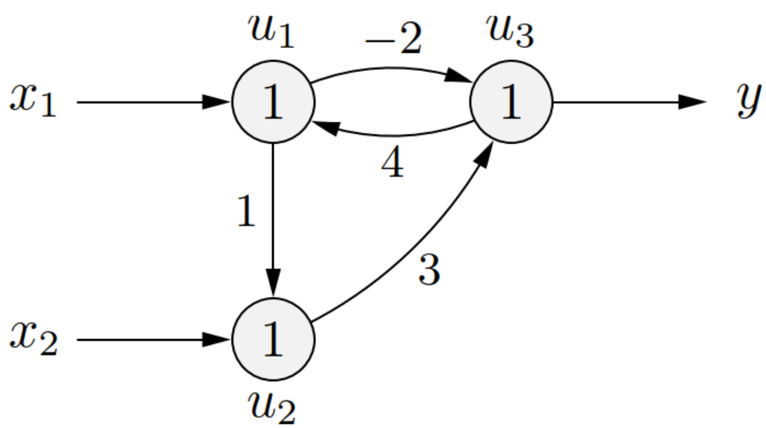


	u_1	u_2	u_3	
input phase	**1**	**0**	**0**	
work phase	1	0	**0**	$\text{net}_{u_3} = -2 \quad < 1$
	0	0	0	$\text{net}_{u_1} = 0 \quad < 1$
	0	**0**	0	$\text{net}_{u_2} = 0 \quad < 1$
	0	0	**0**	$\text{net}_{u_3} = 0 \quad < 1$
	0	0	0	$\text{net}_{u_1} = 0 \quad < 1$

- *Input phase*: initial activations/outputs
- *Work phase*: activations/outputs of the next neuron to update (bold) are computed from the outputs of the other neurons and the weights/threshold.
- *Stable state*: a unique output is reached.

Examples of Neural Networks (4)

A simple recurrent neural network



	u_1	u_2	u_3	
input phase	**1**	**0**	**0**	
work phase	1	0	**0**	$\text{net}_{u_3} = -2 \quad < 1$
	1	**1**	0	$\text{net}_{u_2} = 1 \quad \geq 1$
	0	1	0	$\text{net}_{u_1} = 0 \quad < 1$
	0	1	**1**	$\text{net}_{u_3} = 3 \quad \geq 1$
	0	**0**	1	$\text{net}_{u_2} = 0 \quad < 1$
	1	0	1	$\text{net}_{u_1} = 4 \quad \geq 1$
	1	0	**0**	$\text{net}_{u_3} = -2 \quad < 1$

- Order in which the neurons are updated:

$$u_3, u_2, u_1, u_3, u_2, u_1, u_3, \ldots$$

- No stable state is reached (oscillation of output).

Training Neural Networks (1)

Learning updates the connection weights and possible other parameters (e.g., thresholds) to optimize an objective criterion.

- Fixed learning task
- Free learning task

Training Neural Networks (2)

A **fixed learning task** L_{fixed} for a neural network with

- n input neurons $U_{in} = \{u_1, \ldots u_n\}$
- m output neurons $U_{out} = \{v_1, \ldots v_m\}$

is a set of training patterns $l = (\overrightarrow{i^{(l)}}, \overrightarrow{o^{(l)}})$, each consisting of

- an input vector $\overrightarrow{i^{(l)}} = (ext_{u_1}^{(l)}, \ldots, ext_{u_n}^{(l)})$
- an output vector $\overrightarrow{o^{(l)}} = (o_{v_1}^{(l)}, \ldots, o_{v_m}^{(l)})$.

Supervised learning

Training Neural Networks (3)

A fixed learning task is solved,
if for all training patterns $l \in L_{fixed}$
the neural network computes,
from the external inputs contained
in the input vector $\overrightarrow{i^{(l)}}$ of a training pattern l,
the outputs contained in the corresponding
output vector $\overrightarrow{o^{(l)}}$.

Training Neural Networks (4)

Error of a fixed learning task:

- How well a neural network solves a given fixed learning task.
- Differences between desired and actual outputs.
  - Do not sum differences directly in order to avoid errors canceling each other.
  - Square has favorable properties for deriving the adaptation rules.

$$e = \sum_{l \in L_{\text{fixed}}} e^{(l)} = \sum_{v \in U_{\text{out}}} e_v = \sum_{l \in L_{\text{fixed}}} \sum_{v \in U_{\text{out}}} e_v^{(l)},$$

$$\text{where} \qquad e_v^{(l)} = \left(o_v^{(l)} - \text{out}_v^{(l)} \right)^2$$

Training Neural Networks (5)

A **free learning task** L_{free} for a neural network with

- n input neurons $U_{in} = \{u_1, \dots u_n\}$

is a set of training patterns $l = (\overrightarrow{i^{(l)}})$ each consisting of

- an input vector $\overrightarrow{i^{(l)}} = (ext_{u_1}^{(l)}, \dots, ext_{u_n}^{(l)})$.

Unsupervised learning

Training Neural Networks (6)

- There is no desired output for the training patterns.
- Outputs can be chosen freely by the training method.
- Solution: similar inputs should lead to similar outputs.
  - Learning should lead to clustering of similar input vectors so that the same output is produced for all vectors in the same cluster.

Preprocessing

Normalization of the input vectors

(e.g., **z-score normalization**)

- Expected value and (corrected) standard deviation for each input

$$\mu_k = \frac{1}{|L|} \sum_{l \in L} \mathrm{ext}_{u_k}^{(l)} \qquad \text{and} \qquad \sigma_k = \sqrt{\frac{1}{|L| - \mathbf{1}} \sum_{l \in L} \left(\mathrm{ext}_{u_k}^{(l)} - \mu_k \right)^2},$$

- Normalize the input vectors to expected value (arithmetic mean) equal to 0 and standard deviation equal to 1

$$\mathrm{ext}_{u_k}^{(l)(\mathrm{new})} = \frac{\mathrm{ext}_{u_k}^{(l)(\mathrm{old})} - \mu_k}{\sigma_k}$$

Data Representation

- Numeric data:
  - Real numbers
  - Integer numbers

- Non-numeric (symbolic) data:
  - Symbols
    - 1-in-N encoding