Neural Networks and Deep Learning

Andrea Asperti

DISI - Department of Informatics: Science and Engineering
University of Bologna
Mura Anteo Zamboni 7, 40127, Bologna, ITALY
andrea.asperti@unibo.it

Deep Learning in a nutshell

Q: **What is Deep Learning?**

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Q: **To what kind of problems it can be applied?**

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Q: **To what kind of problems it can be applied?**

A: To all problems suitable for Machine Learning

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Q: **To what kind of problems it can be applied?**

A: To all problems suitable for Machine Learning

Q: **Are there problems where DL particularly excels?**

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Q: **To what kind of problems it can be applied?**

A: To all problems suitable for Machine Learning

Q: **Are there problems where DL particularly excels?**

A: All problems involving myriads of features, e.g. images/speech/text processing, recommendation systems, . . .

Deep Learning in a nutshell

Q: **What is Deep Learning?**

A: A branch of Machine Learning

Q: **To what kind of problems it can be applied?**

A: To all problems suitable for Machine Learning

Q: **Are there problems where DL particularly excels?**

A: All problems involving myriads of features, e.g. images/speech/text processing, recommendation systems, . . .

Deep Learning in a nutshell (2)

Q: **What is the basic technique for DL?**

Deep Learning in a nutshell (2)

Q: **What is the basic technique for DL?**

A: Neural Networks

Deep Learning in a nutshell (2)

Q: **What is the basic technique for DL?**

A: Neural Networks

Q: **Why “Deep”?**

Deep Learning in a nutshell (2)

Q: **What is the basic technique for DL?**

A: Neural Networks

Q: **Why “Deep”?**

A:
- because it exploits Deep Neural Networks, composed by many layers of neurons

Deep Learning in a nutshell (2)

Q: **What is the basic technique for DL?**

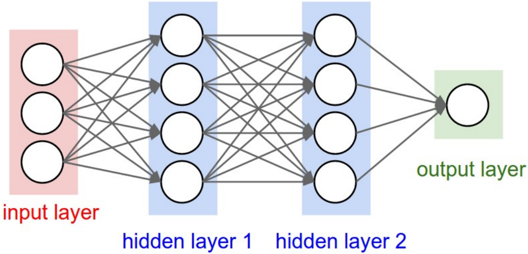
A: Neural Networks

Q: **Why "Deep"?**

A:
- because it exploits Deep Neural Networks, composed by many layers of neurons
- because it exploits deep features of data, that is features extracted from other features

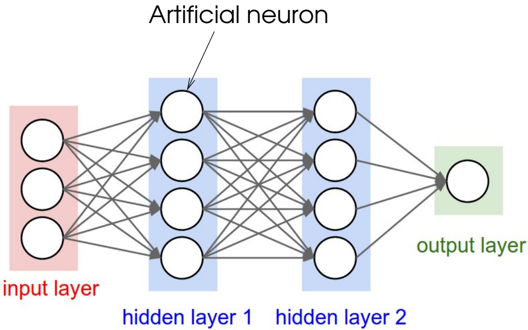
Next argument

Neural Networks



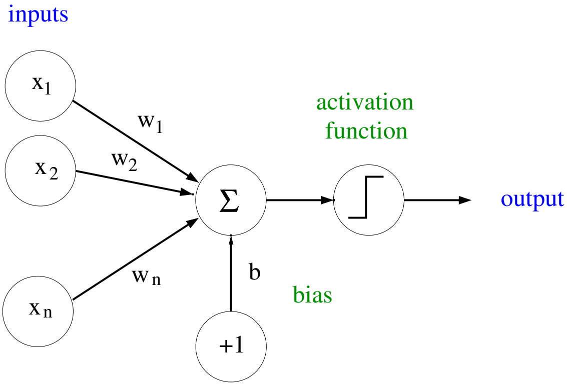
Neural Network

A network of (artificial) neurons



Each neuron takes multiple inputs and produces a single output (that can be passed as input to many other neurons).

The artificial neuron

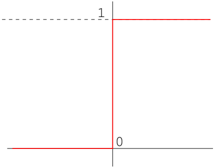


Each neuron (!) implements a logistic regressor

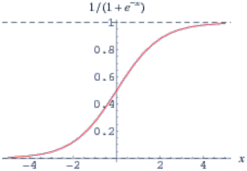
$$\sigma(wx + b)$$

Different activation functions

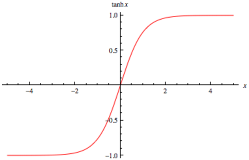
The activation function is responsible for threshold triggering.



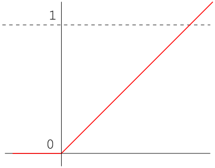
threshold: if $x > 0$ then 1 else 0



logistic function: $\frac{1}{1+e^{-x}}$

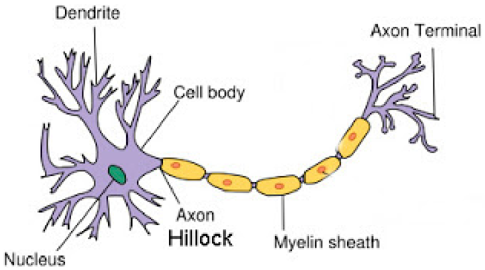


hyperbolic tangent: $\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$



rectified linear (RELU): if $x > 0$ then x else 0

The cortical neuron



- the dendritic tree of the cell collects inputs from other neurons, that get summed together
- when a triggering threshold is exceeded, the Axon Hillock generate an impulse that get transmitted through the axon to other neurons.

Some figures for human brains

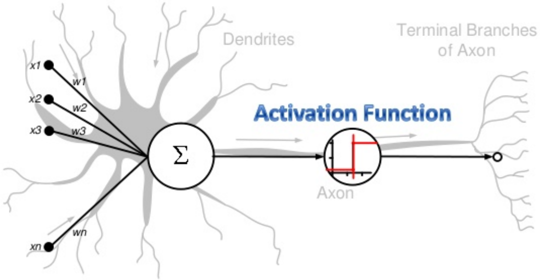
- number of neurons: $\sim 2 \cdot 10^{10}$
- switching time for neuron: $\sim .001$ s. (slow!)
- synapses (connections) per neuron: $\sim 10^{4-5}$
- time to recognize an image: $\sim .1$ s.

not too deep (< 100)

very high parallelism

A comparison with the cortical neuron

Artificial Neural Networks (ANN)



Slide credit : Andrew L. Nelson

Motivations behind neural computation

- to understand, via simulation, how the brain works
- to investigate a different paradigm of computation

  very far from traditional programming languages
- to solve practical problems difficult to address with algorithmic techniques

  useful even if the brain works in a different way

Next argument

Network topologies

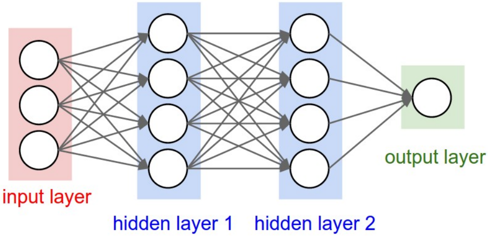
Feed-forward and recurrent networks

If the network is acyclic, it is called a feed-forward network.

If it has cycles it is called recurrent.
(no time to talk about them)

Layers

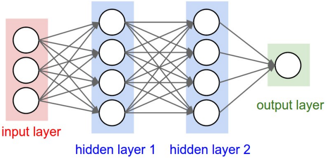
In a feed-forward network, neurons are usually organized in layers.



If there is more than one hidden layer the network is deep, otherwise it is called a shallow network.

Main layers in feed-forward networks: dense layer

Dense layer: each neuron at layer $k-1$ is connected to **each each** neuron at layer k.



A single neuron:

$$I^n \cdot W^n + B^1 = O^1$$

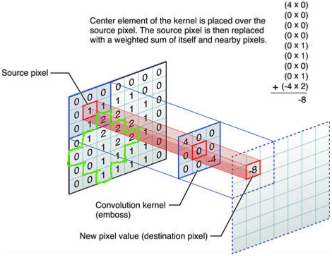
the operation can be vectorized to pruduce m outputs in parallel:

$$I^n \cdot W^{n\times m} + B^m = O^m$$

- dense layers usually work on flat (unstructured) inputs
- the order of elements in input is irrelevant

Main layers in feed-forward networks: convolutional layer

Convolutional layer: each neuron at layer $k-1$ is connected via a parametric **kernel** to a fixed subset of neurons at layer k. The kernel is convolved over the whole previous layer.



Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8

1. move the kernel K over a portion M of the input of equal size
2. compute the dot product $M \cdot K$ and possibly add a bias
3. shift the kernel and repeat

The dimension of the output only depends from the number of times the kernel is applied.

Input is structured, and the structure is reflected in the output.

Next arguments

Diving into DL

A first example

[demo]

Understanding DL

- understand the **different layers**, and their purpose
- understand how layers can be organized in **relevant architectures**
- understand the different possible **applications** of DL, and their specific solutions
- understand the main **issues, problems** and **costs**

Frameworks for DL

- TensorFlow/Keras, Google Brain
- PyTorch, Facebook
- MXNET, Apache

We shall mostly use Keras.

Historical remarks - Legacy

Legacy

1958	perceptron
1975	backpropagation
1980	convolutional layers
1992	Max-pooling
1997	LSTM
...	...

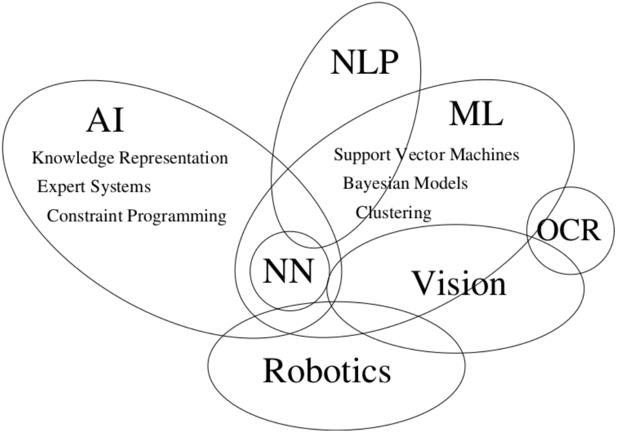
Extremely slow progress

The Deep Learning revolution

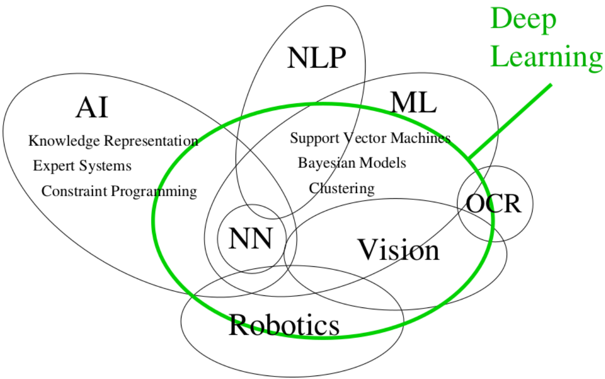
2011	**Google Brain foundation**	2017	**Pytorch release**
2012	ReLU and Dropout	2017	Mask-RCNN
2012	ImageNet Competition	2017	PPO
2013	DQN	2018	Transformers
2014	Inception v1	2018	BERT
2015	**Tensorflow release**	2018	GPT
2015	Keras release	2018	Soft Actor Critic
2015	Batchnormalization	2020	OpenAI Jukebox
2015	YOLO v1	2020	Vision Transformer
2015	**OpenAI foundation**	2021	**MXNet release**
2016	Residual connections	2022	DallE

Just to mention a few milestones ...

The situation at the beginning of the century



The deep learnig era



See my blog for a short historical perspective.