Deep Learning

A (tentative) conceptual framework for deep learning from the perspective of computer science and neural networks

Fabio Massimo ZENNARO

University of Manchester
Scientific Methods II

March 7th, 2014

Aim of the presentation

This presentation aims at giving you a conceptual overview of the field of *deep learning* (DL) within the context of *machine learning*.

- ✓ We will deal with DL ideas, relationship between ideas and development of ideas.
- × We will avoid maths and CS details, applications and excessive oversimplification.

Reasons for being interested in deep learning

Wide applicability: DL may be applied to many different fields;

Cutting edge: DL provides state-of-art solutions to ML problems;

Trending: DL is the focus of interest of many companies:

- *Jun 2013*: Google acquired DNNResearch, a startup founded by G. Hinton [4]
- *Oct 2013*: Yahoo bought LookFlow to start a new DL group [6]
- *Dec 2013*: Facebook hired Y. LeCunn, head of the DL group in NYU [2]
- *Jan 2014*: Google acquired DeepMind, a DL London startup [3]
- *Feb 2014*: Netflix is deploying a GPU-based DL system on the cloud [5]

"We have never seen machine learning or artificial intelligence technologies so quickly make an impact in industry. It's very impressive."

(Kai Yu, director of Baidu's Institute of Deep Learning) [9]

Machine Learning

What is machine learning?

- Study of machines able to learn by themselves.
- Study of machines that can make sense of data.
- Study of machines that can learn to map inputs to outputs.

Machine learning studies algorithms that can learn meaningful functions from input to output.

Supervised and Unsupervised Learning

What is a meaningful function?

- A correct mapping between an input and an expected output relying on acquired knowledge (*supervised learning* or *classification*).
- A (statistically) valid mapping between an input and an unknown output relying on the properties of the input (*unsupervised learning* or *clustering*)

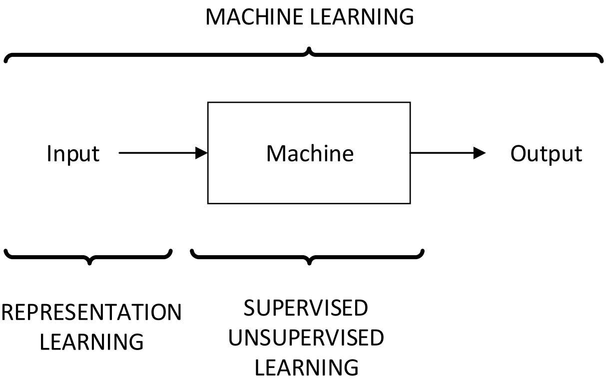
The results of learning are often dependent on having an informative input.

Representation Learning

How do we choose an informative input?

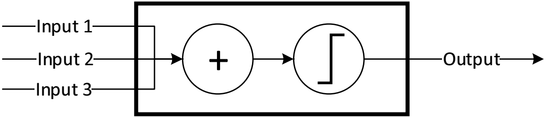
- We can extract, manually or automatically, a set of useful values from a large or noisy input (*feature extraction*).
- We can statistically transform the input in a way that highlights the information we are concerned about (*representation learning*)

Machine Learning



1943 - Artificial Neuron

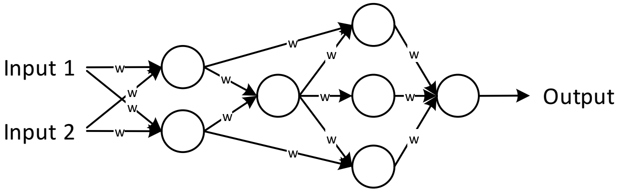
McCollough and Pitts introduce the model of an *artificial neuron* (AN), mimicking the activity of a spiking neuron [11].



An AN is a *vector-valued thresholding function*: it receives as an input a set of values, it sums the inputs and it applies the result to a non-linear thresholding function (e.g.: sigmoid)

1940s - Artificial Neural Networks

Combining ANs together gives rise to an *artificial neural network* (ANN), a model resembling a brain tissue.

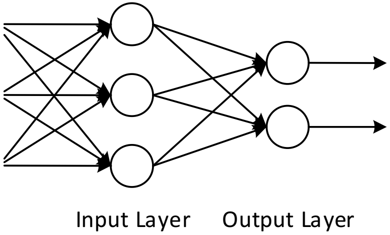


An ANN is a *weighted directed acyclic k-partite graph*; computation is performed by forwarding the inputs through the nodes of the networks and collecting its output.

Learning happens by tuning the weights.

1958 - Perceptrons

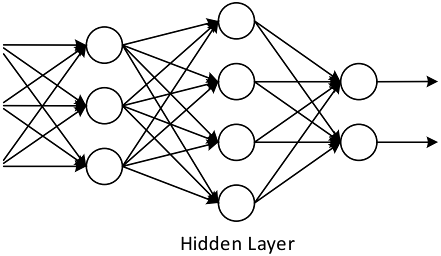
Rosenblatt defines the *perceptron*, an ANN model able to learn [12].



A perceptron is a *complete weighted directed acyclic bipartite graph*; ANs are organized into two layers, an *input layer* and an *output layer*.

1950s - Multi-Layer Perceptrons

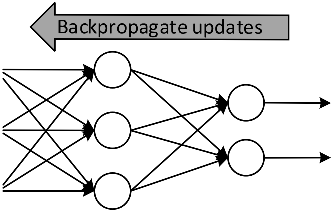
Adding layers to the original perceptron, more complex *multi-layer perceptron* (MLP) can be created.



A MLP extends a perceptron by adding a variable number of *hidden layers* between the input layer and the output layer.

1974 - Backpropagation Algorithm

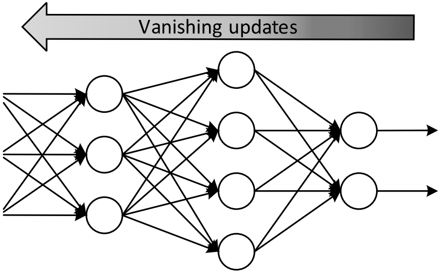
Werbos develops the *backpropagation algorithm*, an optimization technique allowing the efficient training of perceptrons [14].



The backpropagation algorithm is a variation of the *gradient descent algorithm*: it tunes a perceptron by making small changes to the weights of the network in order to make the output of a computation closer to the desired output.

1980s - Vanishing Gradient Problem

The training of a MLP with many hidden layers appears to suffer the problem of the *vanishing gradient* meaning that everytime information is backpropagated, part of the update is lost.



As a result, the optimization of MLPs end up stuck in *local minima* making MLPs performing poorly; MLPs were then abandoned in favour of other models, such as Support Vector Machines (SVM).

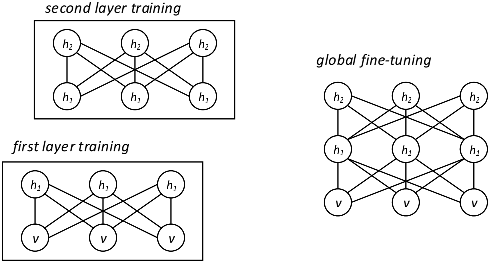
1989 - Universal Approximator Theorem

Cybenko proves the *universal approximator theorem* showing that a MLP can approximate any continuous function [7]

This result guarantees us that a MLP is powerful enough to approximate any continuous function (but it does not tell us how to find the best approximate function describing the relationship between a given input and a given output).

2006 - Deep Learning

Hinton and Salakhutdinov publish an article about *deep learning* (DL), an original way to train MLPs [10].



DL is based on the idea of a *two-stages training*:

1. An *unsupervised greedy layer-wise* training.
2. A *supervised global* fine-tuning.

Deep Neural Networks

Deep neural networks (DNN) are *neural networks* made up by more than one hidden layer.

MLP with two or more hidden layers are a prototypical instance of DNN.

Deep Learning

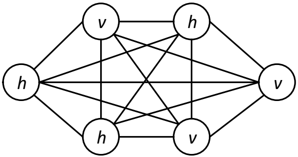
Deep learning (DL) is:

- A new area of machine learning [1].
- A *family of machine learning algorithms* relying on a two-stages training for tuning DNN.

Classical DL algorithms can be *Boltzmann machines*-based or *autoencoders*-based.

Boltzmann Machine

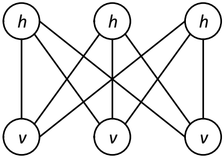
Boltzmann machines (BM) are a *network of symmetrically connected binary stochastic neuron-like units* [1]. BMs are not ANNs.



Units are partitioned among *visible* (v) and *hidden* (h) units. Upon receiving an input, units make stochastic decisions on their output until, in the long run, the network reaches an equilibrium given by a *Boltzmann distribution*.

Restricted Boltzmann Machine

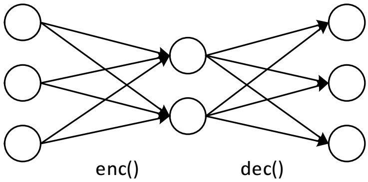
Restricted Boltzmann machines (RBM) are a subset of BMs with a constraint on their connections. [1].



RBMs are not allowed visible-visible or hidden-hidden connections. RBMs can be efficiently trained using the *Contrastive Divergence* algorithm.

Autoencoders

Autoencoders (AE) are *shallow neural networks* designed to encode an input.



An autoencoder is described by an *encoding* (enc) and a *decoding* (dec) function. Ideally, the output of an autoencoder is as similar to the input as possible.

Aside: A Semantic Note

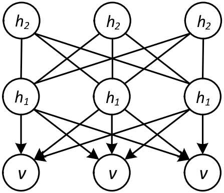
Deep denotes an architecture made up by several layers;

Stacking denotes the act of putting together several layers;

Stacked denotes an architecture built by stacking several layers.

Deep Belief Networks

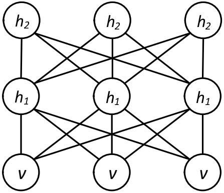
Deep belief networks (DBN) are directed-undirected probabilistic generative models made up by stacking RBMs (or AEs) [1]. DBNs are not ANNs.



First, each layer of a DBN is trained independently in a greedy way. Then, the DBN can undergo a global fine-tuning.

Deep Boltzmann Machines

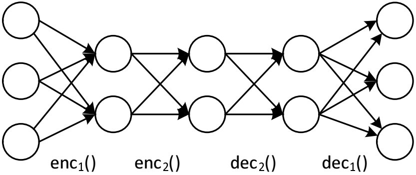
Deep Boltzmann machines (DBM) or *Stacked Restricted Boltzmann Machines* (SRBM) are undirected probabilistic generetive models made up by stacked RBMs [13]. DBMs are not ANNs.



Differently from DBN, a DBM can propagate information in both direction during training, leading to a better approximation.

Deep Autoencoders

Deep Autoencoders (DeepAE) or *Stacked Autoencoders* (SAE) are neural networks built by stacking several AEs [1].



First, each AE of a DeepAE is trained independently in a greedy way. Then, the DeepAE can undergo a global fine-tuning.

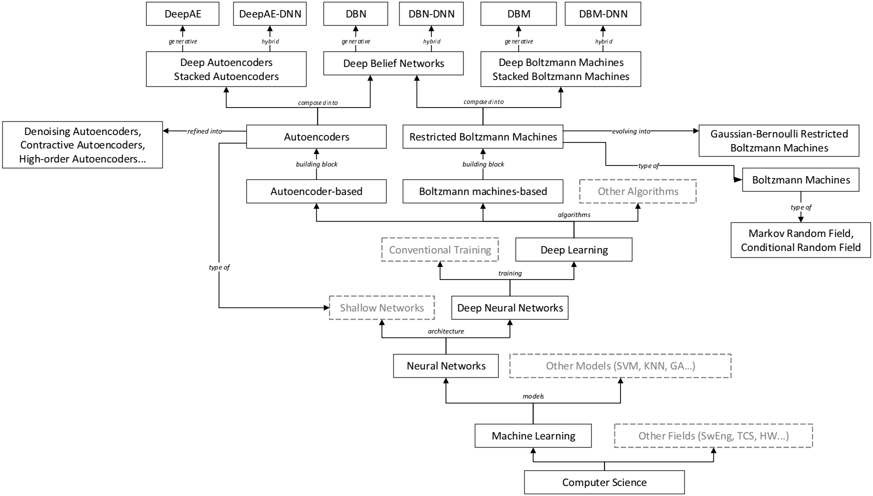
DL strategies

The networks we described (DBN, DBM, DeepAE) can be used in two different ways [8]:

Generative way: the networks can be used to generate representations;

Hybrid way: the networks are trained and their weights are used to initialize a DNN (DBN-DNN, DBM-DNN, DeepAE-DNN).

Final Conceptual Framework



Thanks!

Thank you very much for listening!

References I

[1] Deeplearning.net.
http://deeplearning.net/.

[2] Facebook hired LeCunn.
http://techcrunch.com/2013/12/09/facebook-artificial-intelligence-lab-lecun/.

[3] Google acquired DeepMind.
http://techcrunch.com/2014/01/26/google-deepmind/.

[4] Google acquired DNNResearch.
http://techcrunch.com/2013/03/12/google-scoops-up-neural-networks-startup-dnnresearch-to-boost-its-voice-and-image-search-tech/.

References II

[5] Netflix is developing GPU DL.
http://www.wired.com/wiredenterprise/2014/02/netflix-deep-learning/.

[6] Yahoo acquired LookFlow.
http://techcrunch.com/2013/10/23/yahoo-acquires-startup-lookflow-to-work-on-flickr-and-deep-learning/.

[7] George Cybenko.
Approximations by superpositions of sigmoidal functions.
Mathematics of Control, Signals, and Systems, 2:303–314, 1989.

References III

[8] Li Deng.
A tutorial survey of architectures, algorithms, and applications for deep learning.
In *APSIPA Transactions on Signal and Information Processing*, 2014.

[9] Daniela Hernandez.
Meet the man hired to make ai a reality.
Wired, 1, 2014.

[10] Geoffrey E. Hinton and Ruslan R. Salakhutdinov.
Reducing the dimensionality of data with neural networks.
Science, 28:504–507, 2006.

References IV

[11] Warren S. McCulloch and Walter Pitts.
A logical calculus of ideas immanent in nervous activity.
Bulletin of Mathematical Biophysics, 5:115–133, 1943.

[12] Franz Rosenblatt.
The perceptron: A probalistic model for information storage and organization in the brain.
Psychological Review, 65:386–408, 1958.

[13] Ruslan Salakhutdinov and Geoffrey Hinton.
Deep boltzmann machines.
In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

References V

[14] Paul J. Werbos.
Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.
PhD thesis, Harvard University, 1975.