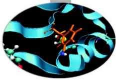
A Deep Learning primer

Riccardo Zanella – `r.zanella@cineca.it`

SuperComputing Applications and Innovation Department

Table of Contents



Deep Learning

A recently published review[1] can help on summarizing main aspects of deep learning.

1. Models are composed of multiple processing layers:
   - multiple layers of abstraction to learn data representations.
2. Improved state-of-the-art in:
   - speech recognition, object recognition, object detection;
   - drug discovery, genomics.
3. Discovers complex patterns in large datasets:
   - backpropagation to change layer parameters;
   - representation in each layer is based on previous layer results;
4. Specialized networks for different data;
   - deep convolutional networks: image, video, speech;
   - recurrent networks: sequential data (text, speech).

[1] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, **Deep Learning**, Nature 2015

SuperComputing Applications and Innovation

Limits of conventional ML techniques

LeCun Bengio and Hinton stress that:

Limits of conventional ML techniques

LeCun Bengio and Hinton stress that:

- conventional machine-learning techniques were limited in their ability to process natural data in their raw form;

Limits of conventional ML techniques

LeCun Bengio and Hinton stress that:

- conventional machine-learning techniques were limited in their ability to process natural data in their raw form;
- feature extraction is a necessary step for transforming raw data into an internal representation;

Limits of conventional ML techniques

LeCun Bengio and Hinton stress that:

- conventional machine-learning techniques were limited in their ability to process natural data in their raw form;
- feature extraction is a necessary step for transforming raw data into an internal representation;
- considerable domain expertise is needed to pick a representation suitable to the task.

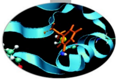
Limits of conventional ML techniques

LeCun Bengio and Hinton stress that:

- conventional machine-learning techniques were limited in their ability to process natural data in their raw form;
- feature extraction is a necessary step for transforming raw data into an internal representation;
- considerable domain expertise is needed to pick a representation suitable to the task.

On the other side, they consider deep learning methods as representation-learning methods.

Neural networks as representation learning methods



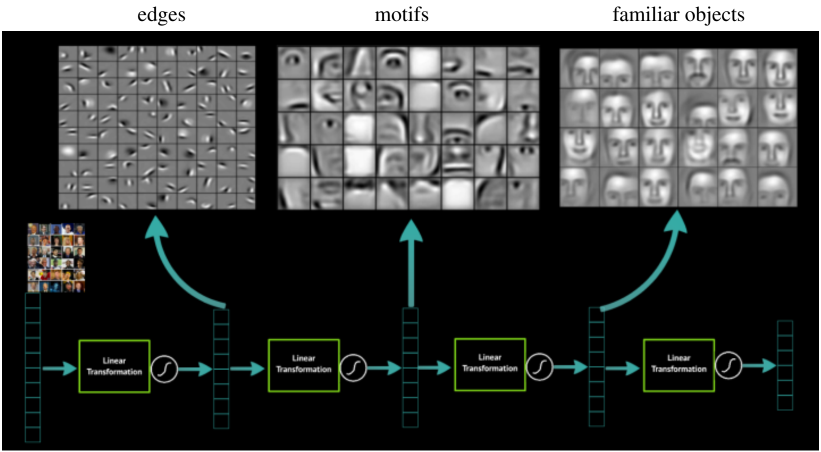
Data flow:

- input: raw data;
- output: detection/classification distribution probabilities;
- in the process: a layer is fed with data representation learned from previous layer.

Key aspects:

- no a-priori design of features;
- they are learned from data using a general purpose procedure.

Image Example



Deep learning main results (I)

- Good at discovering intricate structures in high-dimensional data.
- Exhibits superior performances (compared to other ML techniques):
  - image and speech recognition;
  - prediction of the activity of potential drug molecules;
  - analysing particle accelerator data;
  - reconstructing brain circuits;
  - predicting the effects of mutations in non-coding DNA on gene expression and disease.
- Shows promising results in natural language processing (NLP):
  - topic classification, sentiment analysis, question answering and language translation.

Deep learning main results (II)

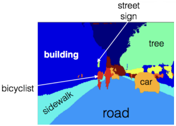
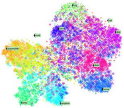
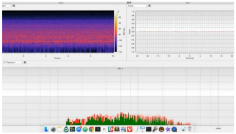


Image recognition

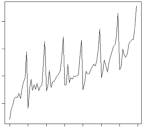


Natural language processing



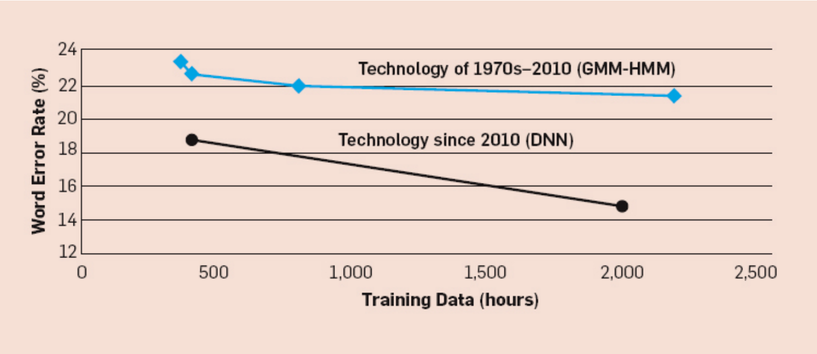
Speech recognition

Video activity detection



Tabular and time-series data applications

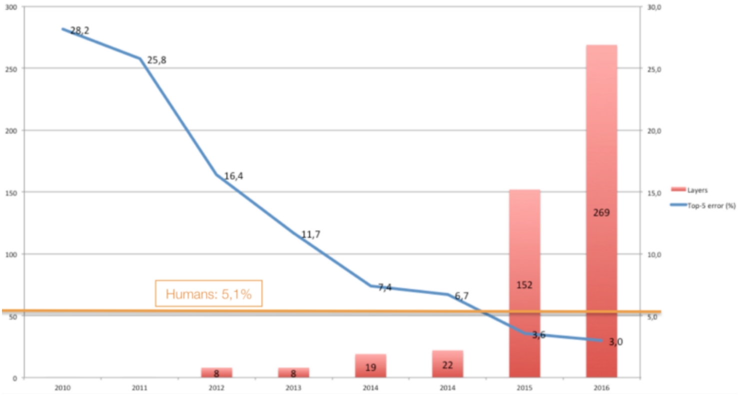
Accuracy on Speech Recognition



Source: Huang, Baker, Reddy, **A Historical Perspective of Speech Recognition**

GMM: Gaussian Mixture Models, **HMM:** Hidden Markov Models, **DNN:** Deep Neural Networks

How deep is deep learning?



Number of layers in ILSVRC winners, compared to accuracy.

How deep learning works?

In the following, we will see:

- the effect of adding a fully connected layer to an existing classifier;
- the effect of describing our data in a "wider" hyperspace.

How deep learning works?

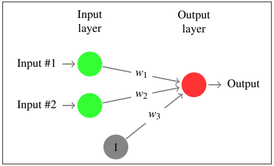
In the following, we will see:

- the effect of adding a fully connected layer to an existing classifier;
- the effect of describing our data in a “wider” hyperspace.

Idea from a blog post: Olah, **Neural Networks, Manifolds, and Topology**: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

2d example (I)

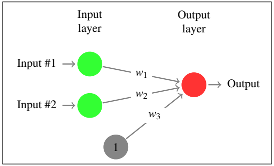
Define a simple network:



$$o_i = \langle [x_i \; y_i], [w_1 \; w_2] \rangle + w_3$$

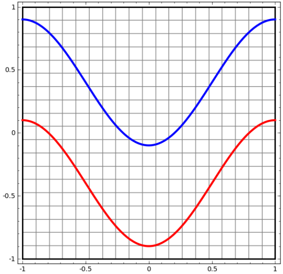
2d example (I)

Define a simple network:



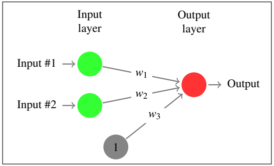
$$o_i = < [x_i \; y_i], [w_1 \; w_2] > + w_3$$

Labeled observations: $\forall i \; (x_i, y_i) \rightarrow l_i$



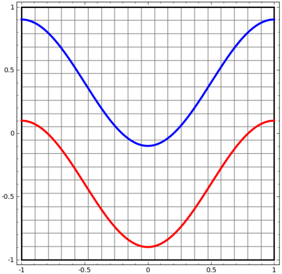
2d example (I)

Define a simple network:



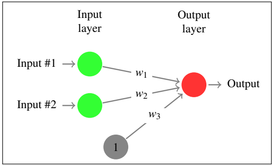
$$o_i = < [x_i \; y_i], [w_1 \; w_2] > + w_3$$

Labeled observations: $\forall i \; (x_i, y_i) \rightarrow l_i$ optimize: $w = \arg\min \sum_i (l_i - o_i)^2$



2d example (I)

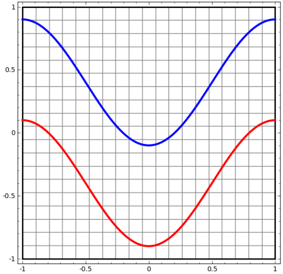
Define a simple network:

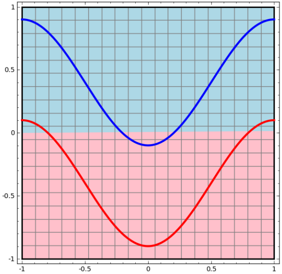


$$o_i = < [x_i \; y_i], [w_1 \; w_2] > + w_3$$

Labeled observations: $\forall i \; (x_i, y_i) \rightarrow l_i$

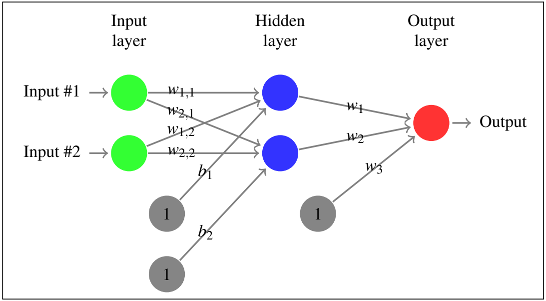
optimize: $w = \arg\min \sum_i (l_i - o_i)^2$





2d example (II)

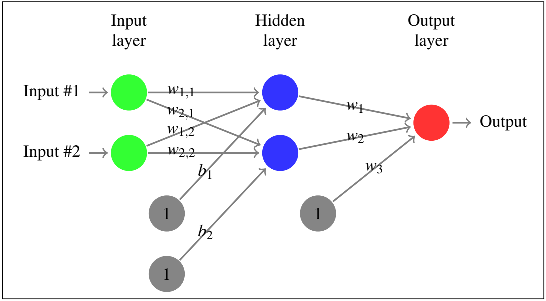
Add an hidden layer:



$$o_i = < f\left([x_i \; y_i] \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}^T \right), [w_1 \; w_2] > + w_3$$

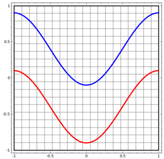
2d example (II)

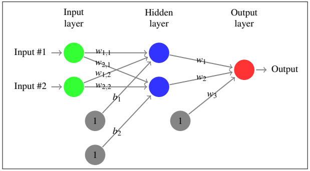
Add an hidden layer:

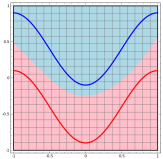


$$o_i = < \boxed{f\left([x_i\ y_i] \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}^T \right)}, [w_1\ w_2] > + w_3$$

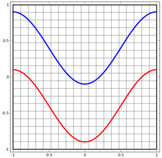
Hidden layer: evaluated features

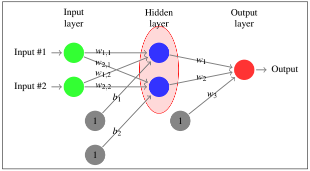


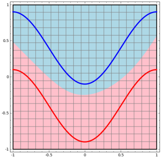


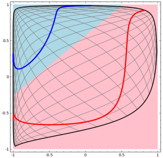


Hidden layer: evaluated features

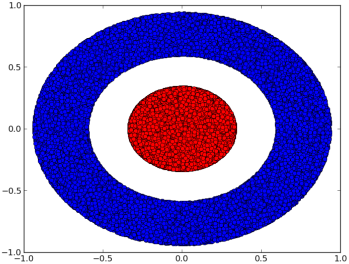




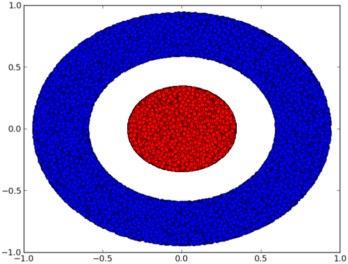


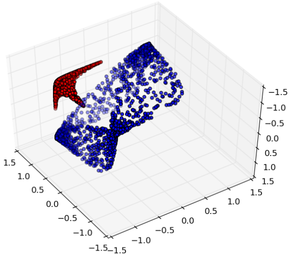


Increase the dimensionality

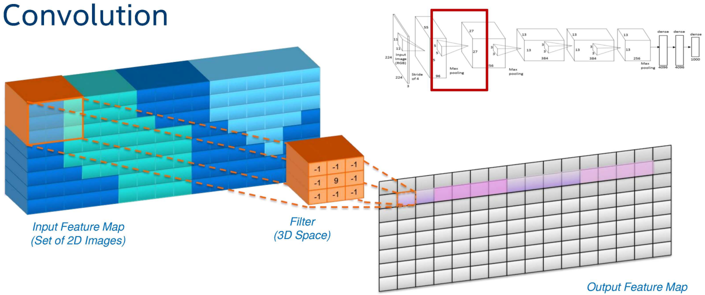


Increase the dimensionality



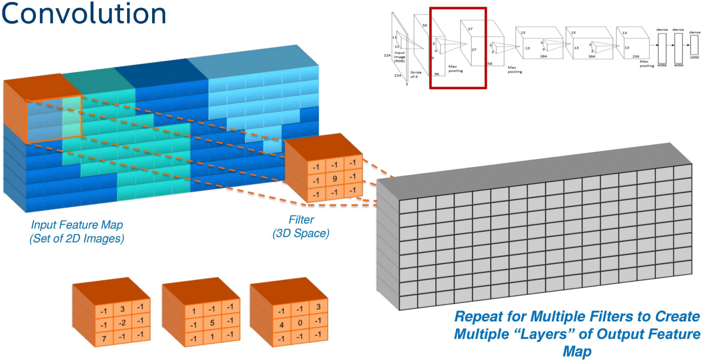


Convolutional layer (I)

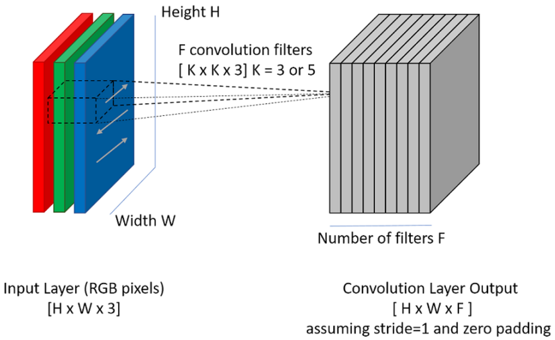


Convolutional layer (I)

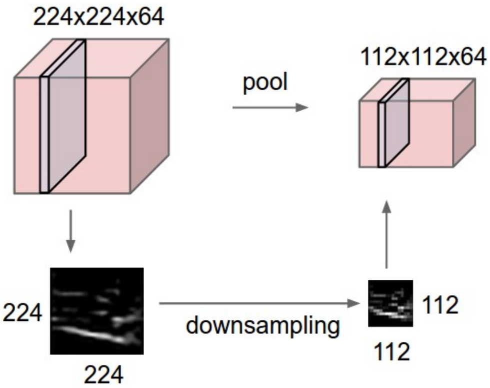
Convolution



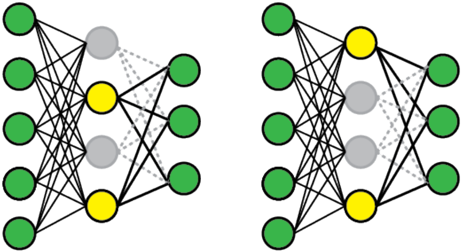
Convolutional layer (II)



Pooling layer



Dropout layer



Convolutional network