



PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE

Introduction to Deep Learning

Marco Rorro

m.rorro@cineca.it

CINECA – SCAI SuperComputing Applications and Innovation Department



Table of Contents

Introduction

Machine Learning Background

Deep Learning: a review

Convolutional Networks



Introduction

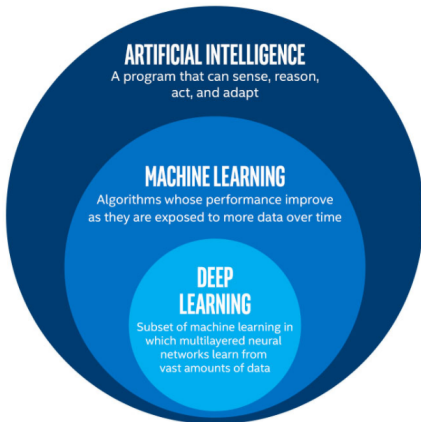
Machine Learning Background

Deep Learning: a review

Convolutional Networks

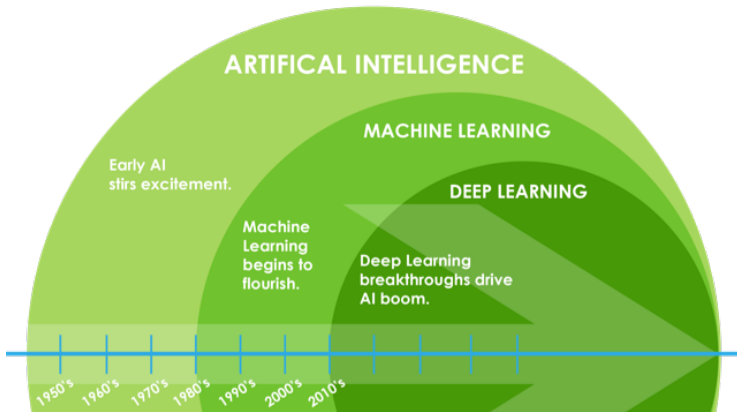


Artificial Intelligence, Machine Learning and Deep Learning



Source: Cousins of AI, <https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55>

A bit of history



Source: <https://buzzrobot.com/difference-between-artificial-intelligence-machine-learning-and-deep-learning-ccfd779eca7b>



Introduction

Machine Learning Background

Deep Learning: a review

Convolutional Networks



Learning Algorithm

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if **its performance** at tasks in T , as measured by P , **improves with experience** E .



Learning Algorithm

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if **its performance** at tasks in T , as measured by P , **improves with experience** E .

So we need to identify:

- ▶ the class of tasks T



Learning Algorithm

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if **its performance** at tasks in T , as measured by P , **improves with experience** E .

So we need to identify:

- ▶ the class of tasks T
- ▶ the measure of performance P



Learning Algorithm

Definition of Learning Algorithm [Mitchell 1997]¹

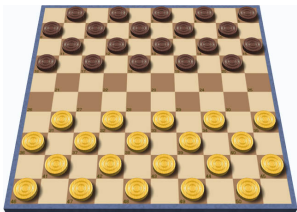
A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if **its performance** at tasks in T , as measured by P , **improves with experience** E .

So we need to identify:

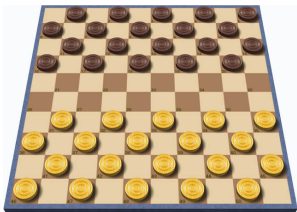
- ▶ the class of tasks T
- ▶ the measure of performance P
- ▶ the source of experience E



Example: checkers game

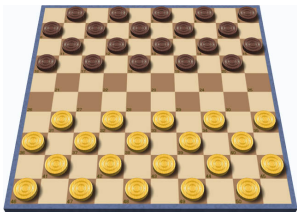


Example: checkers game



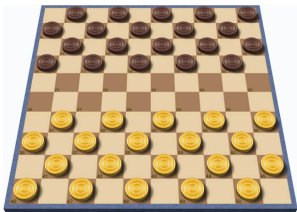
- ▶ **task class T:** playing checkers

Example: checkers game



- ▶ **task class T:** playing checkers
- ▶ **performance measure P:** fraction of games won against opponents

Example: checkers game

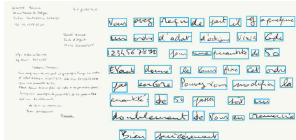


- ▶ **task class T:** playing checkers
- ▶ **performance measure P:** fraction of games won against opponents
- ▶ **training experience E:** playing practice games against itself



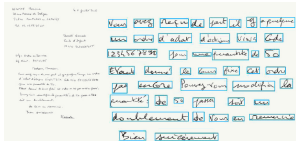
Example: handwritten characters recognition

8 2 9 4 4 6 4 9 7 0 9 2 7 5 1 5 9 1 0 3
2 3 5 9 1 7 6 2 8 2 2 5 0 7 4 9 7 8 3 2
1 1 8 3 6 1 0 3 1 0 0 1 1 2 7 3 0 4 6 5
2 6 4 7 1 8 9 9 3 0 7 1 0 2 0 3 5 4 6 5



Example: handwritten characters recognition

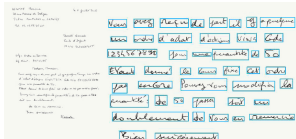
8 2 9 4 4 6 4 9 7 0 9 2 7 5 1 5 9 1 0 3
 1 3 5 9 1 7 6 2 8 2 2 5 0 7 4 9 7 8 3 2
 1 1 8 3 6 1 0 3 1 0 0 1 1 2 7 3 0 4 6 5
 2 6 4 7 1 8 9 9 3 0 7 1 0 2 0 3 5 4 6 5



- ▶ **task class T:** recognizing and classifying handwritten characters within images

Example: handwritten characters recognition

8 2 9 4 4 6 4 9 7 0 9 2 7 5 1 5 9 1 0 3
 1 3 5 9 1 7 6 2 8 2 2 5 0 7 4 9 7 8 3 2
 1 1 8 3 6 1 0 3 1 0 0 1 1 2 7 3 0 4 6 5
 2 6 4 7 1 8 9 9 3 0 7 1 0 2 0 3 5 4 6 5



- ▶ **task class T:** recognizing and classifying handwritten characters within images
- ▶ **performance measure P:** fraction of characters correctly classified

Example: handwritten characters recognition

8 2 9 4 4 6 4 9 7 0 9 2 7 5 1 5 9 1 0 3
2 3 5 9 1 7 6 2 8 2 2 5 0 7 4 9 7 8 3 2
1 1 8 3 6 1 0 3 1 0 0 1 1 2 7 3 0 4 6 5
2 6 4 7 1 8 9 9 3 0 7 1 0 2 0 3 5 4 6 5



- ▶ **task class T**: recognizing and classifying handwritten characters within images
- ▶ **performance measure P**: fraction of characters correctly classified
- ▶ **training experience E**: a database of handwritten characters with given classifications



Example: supervised learning

- ▶ **training experience E** : a number of training examples $E = \{z_1, z_2, z_3 \dots\}$
each example is a (input,target) pair: $Z_i = (X_i, Y_i)$



Example: supervised learning

- ▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3 \dots\}$
each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ **task class T:** a decision function f able to predict unknown Y from known X



Example: supervised learning

- ▶ **training experience E**: a number of training examples $E = \{z_1, z_2, z_3 \dots\}$
each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ **task class T**: a decision function f able to **predict** unknown Y from known X
- ▶ **performance measure P**: a **loss function** L to measure the (non-symmetric) distance
 $L(f, Z_i) = d(f(X_i), Y_i)$



Example: supervised learning

- ▶ **training experience E**: a number of training examples $E = \{z_1, z_2, z_3 \dots\}$
each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ **task class T**: a decision function f able to **predict** unknown Y from known X
- ▶ **performance measure P**: a **loss function** L to measure the (non-symmetric) distance
 $L(f, Z_i) = d(f(X_i), Y_i)$

Examples:

- ▶ **regression**
 - ▶ X is a real-valued scalar or vector
 - ▶ Y is a scalar real value
 - ▶ f is able to predict Y_i value from X_i
 - ▶ L is usually the euclidean norm



Example: supervised learning

- ▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3 \dots\}$
each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ **task class T:** a decision function f able to **predict** unknown Y from known X
- ▶ **performance measure P:** a **loss function** L to measure the (non-symmetric) distance
 $L(f, Z_i) = d(f(X_i), Y_i)$

Examples:

- ▶ **regression**
 - ▶ X is a real-valued scalar or vector
 - ▶ Y is a scalar real value
 - ▶ f is able to predict Y_i value from X_i
 - ▶ L is usually the euclidean norm
- ▶ **classification**
 - ▶ X is a real-valued scalar or vector (features)
 - ▶ Y is an integer (label) corresponding to a class index
 - ▶ f is able to provide the probability of X_i being in class Y_i
 - ▶ L is usually the negative log-likelihood



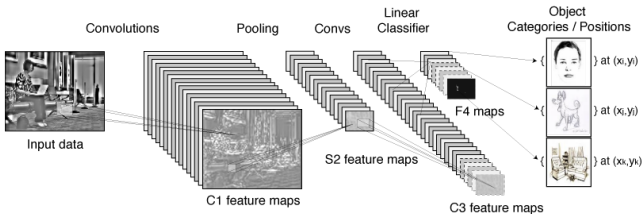
Introduction

Machine Learning Background

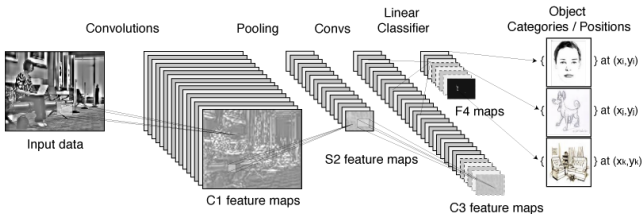
Deep Learning: a review

Convolutional Networks

Deep Learning

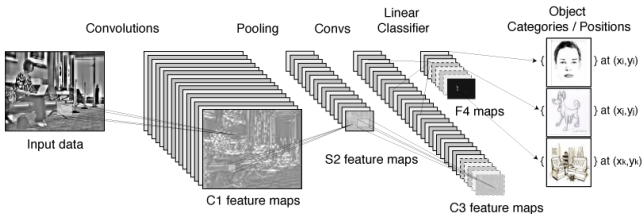


Deep Learning



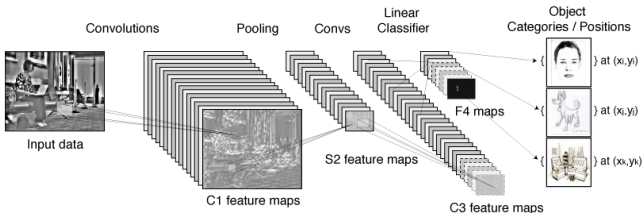
- ▶ Application of an **Artificial Neural Network** to a data set to find a **pattern**

Deep Learning



- ▶ Application of an **Artificial Neural Network** to a data set to find a **pattern**
- ▶ **Multiple** hidden layers (to mimic human brain processes associated to vision/hearing)

Deep Learning



- ▶ Application of an **Artificial Neural Network** to a data set to find a **pattern**
- ▶ **Multiple** hidden layers (to mimic human brain processes associated to vision/hearing)
- ▶ **Big data** sets and relevant number of **variables**



Deep Learning

A recently published review¹ 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).

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



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**.

"Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representation needed for detection or classification"



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:

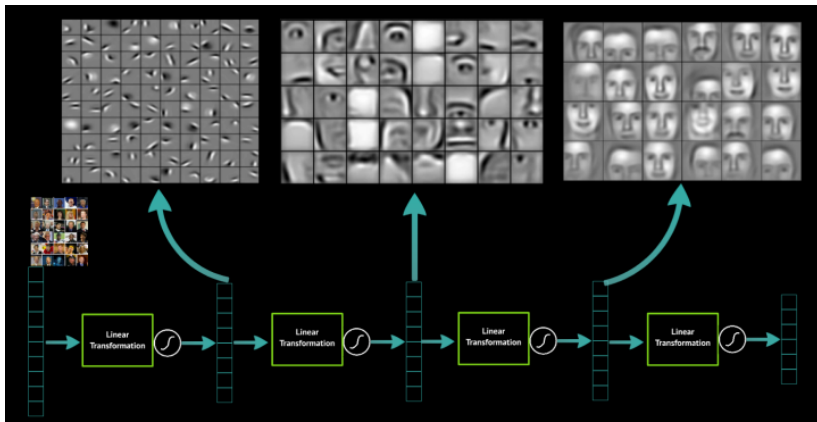
- ▶ no a-priori design of features;
- ▶ they are learned from data using a general purpose procedure.

Image Example

edges

motifs

familiar objects





Deep learning main results (I)

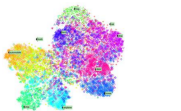
- ▶ Good at discovering **intricate structures** in high-dimensional data.
- ▶ Exhibits **superior** performances (compared to other ML techniques):
 - ▶ **image and speech** recognition;
 - ▶ prediction of the activity of potential **drug molecules**;
 - ▶ **analysing** particle accelerator data;
 - ▶ **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.



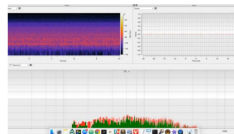
Deep learning main results (II)



Image recognition



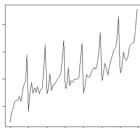
Natural language processing



Speech recognition

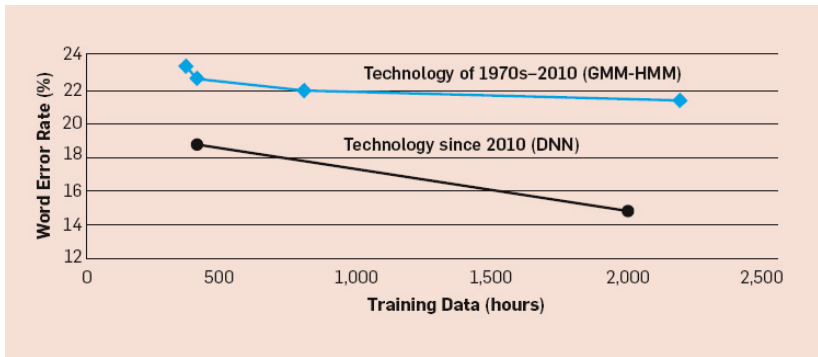


Video activity detection



Tabular and time-series data applications

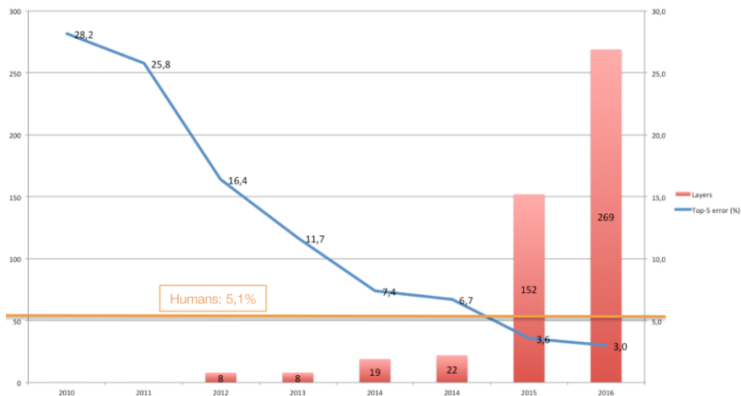
Accuracy on Speech Recognition



Source: Huang, Baker, Reddy, **A Historical Perspective of Speech Recognition**, Communications of the ACM, January 2014

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

How deep is deep learning?



Number of layers in ILSVRC (ImageNet Large Scale Visual Recognition Competition) 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:

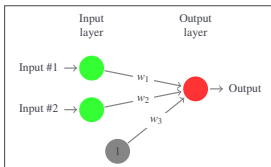
- ▶ the effect of **adding** a fully connected layer to an **existing classifier**;
- ▶ the effect of describing our data in a **“wider”** hyperspace.

Idea from a blog post: Olah, **Neural Networks, Manifolds, and Topology**:

<http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>

2d example (I)

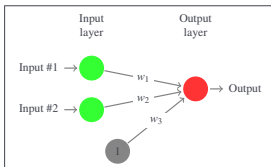
Define a simple network:



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

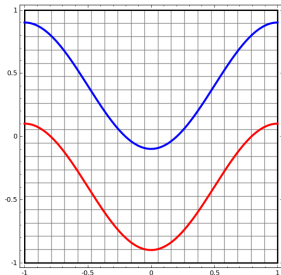
2d example (I)

Define a simple network:



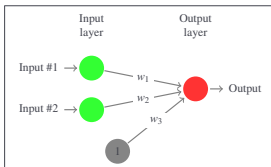
$$o_i = \langle [x_i \ y_i], [w_1 \ w_2] \rangle + w_3$$

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



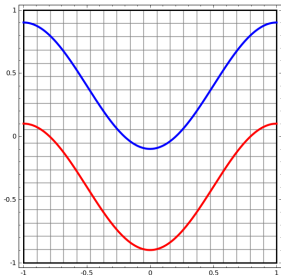
2d example (I)

Define a simple network:



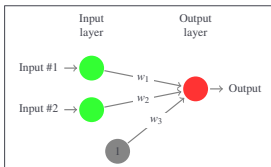
$$o_i = \langle [x_i \ y_i], [w_1 \ w_2] \rangle + w_3$$

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)

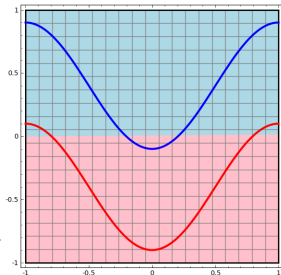
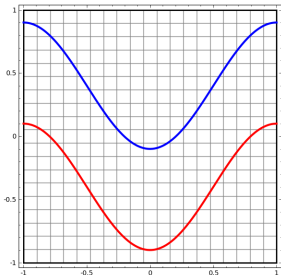
Define a simple network:



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

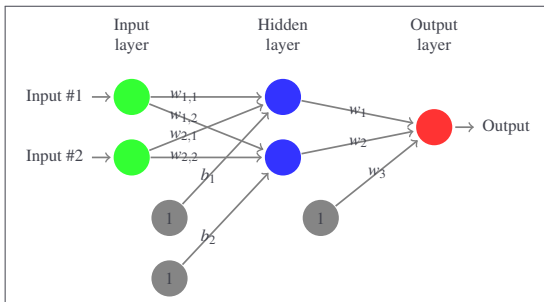
Labeled observations: $\forall i (x_i, y_i) \rightarrow l_i$

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



2d example (II)

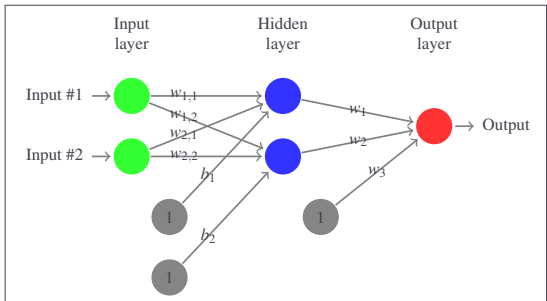
Add an hidden layer:



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

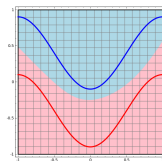
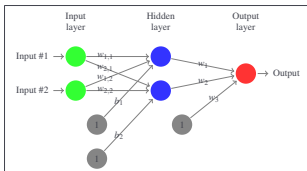
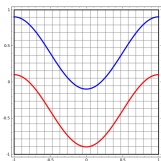
2d example (II)

Add an hidden layer:

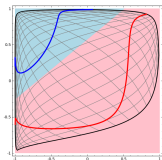
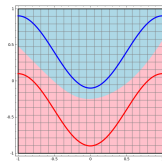
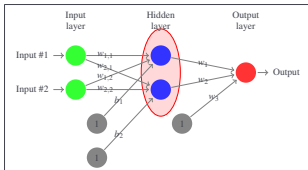
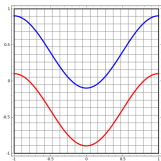


$$o_i = \left\langle f \left([x_i \ y_i] \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)^T, [w_1 \ w_2] \right\rangle + w_3$$

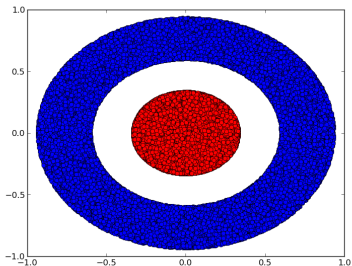
Hidden layer: evaluated features



Hidden layer: evaluated features

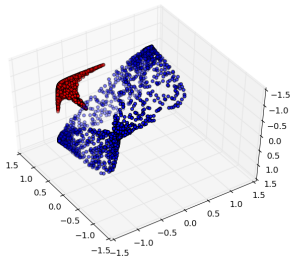
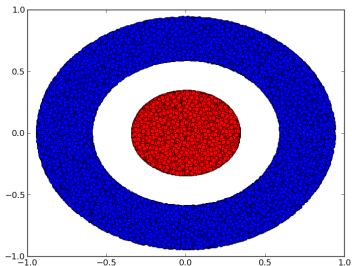


Increase the dimensionality



- ▶ It is impossible for a neural network to classify this dataset without having a layer that has 3 or more hidden units, regardless of depth
- ▶ Even if it can get an 80% of classification accuracy

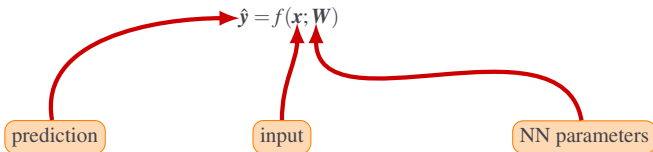
Increase the dimensionality



- ▶ It is impossible for a neural network to classify this dataset without having a layer that has 3 or more hidden units, regardless of depth
- ▶ Even if it can get an 80% of classification accuracy

Neural Network Internals: classification problem

- ▶ Given a single input, a trained neural network is able to predict a distribution of probability:





Neural Network Internals: classification problem

- ▶ Given a single input, a trained neural network is able to predict a distribution of probability:

$$\hat{y} = f(\mathbf{x}; \mathbf{W})$$

- ▶ NN parameters are chosen in order to minimize the average error on a given training set $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1, \dots, N}$:

$$J(\mathbf{W}) = \frac{1}{N} \sum_1^N L(f(\mathbf{x}^{(i)}; \mathbf{W}), \mathbf{y}^{(i)})$$



Neural Network Internals: classification problem

- ▶ Given a single input, a trained neural network is able to predict a distribution of probability:

$$\hat{y} = f(\mathbf{x}; \mathbf{W})$$

- ▶ NN parameters are chosen in order to minimize the average error on a given training set $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1, \dots, N}$:

$$J(\mathbf{W}) = \frac{1}{N} \sum_1^N L(f(\mathbf{x}^{(i)}; \mathbf{W}), \mathbf{y}^{(i)})$$

- ▶ a Stochastic Gradient Descent (SGD) algorithm step is:

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \varepsilon_k \hat{\mathbf{g}}^{(k)} \quad \text{where} \quad \hat{\mathbf{g}}^{(k)} = \frac{1}{n} \nabla_{\mathbf{W}} \sum_{l \in \text{batch}} L(f(\mathbf{x}^{(l)}; \mathbf{W}^{(k)}), \mathbf{y}^{(l)}), \quad n = \#\text{batch}$$

Backpropagation

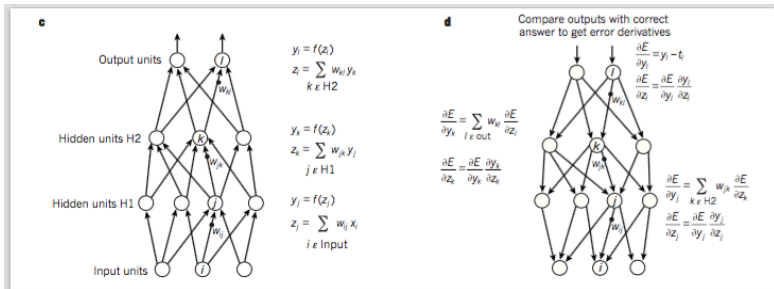


Image from Yann LeCun, Yoshua Bengio, Geoffrey Hinton, **Deep Learning**, Nature 2015.

See also:

<https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/>



Introduction

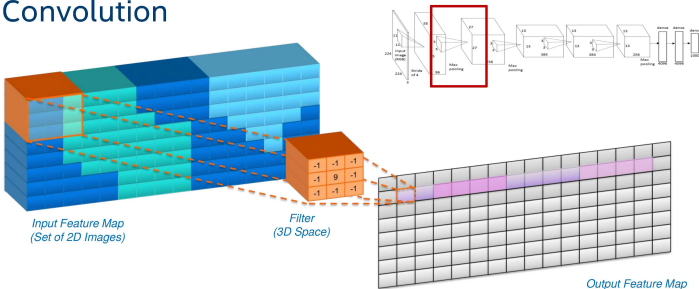
Machine Learning Background

Deep Learning: a review

Convolutional Networks

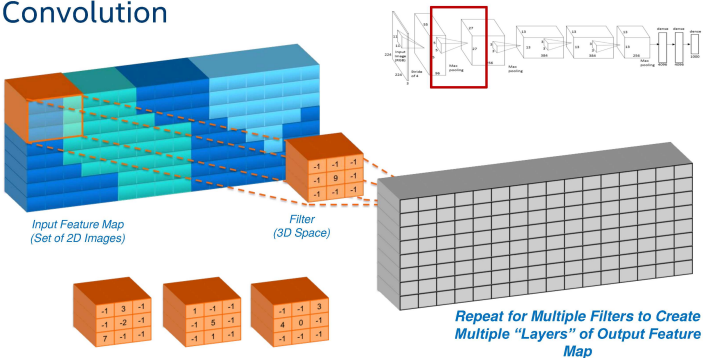
Convolutional layer (I)

Convolution

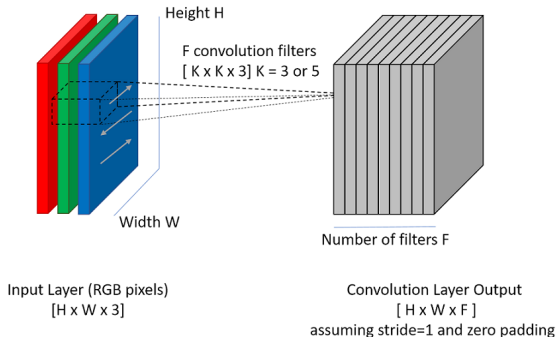


Convolutional layer (I)

Convolution

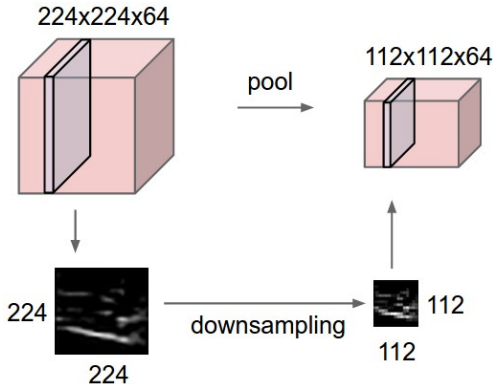


Convolutional layer (II)



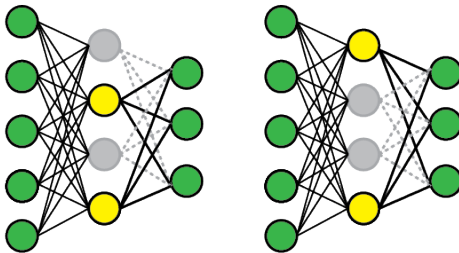
- ▶ Convolution Layer output: $H - \frac{K-1}{2}$, $W - \frac{K-1}{2}$ with *stride* = 1 and without padding
- ▶ For each filter we have $K \times K \times 3$ weights
- ▶ The filter convolves over all spatial locations, producing a scalar for each location
- ▶ An activation function is finally applied

Pooling layer



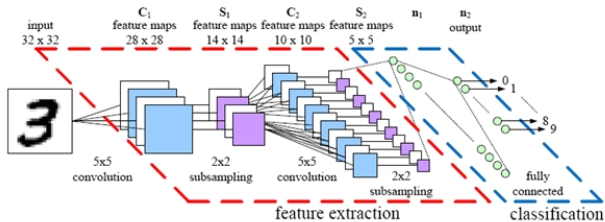
- ▶ makes the representations smaller and more manageable
- ▶ operates over each activation map independently
- ▶ Example: max pooling computes the maximum with $K \times K$ filter

Dropout layer



- ▶ In each forward pass, randomly set some neurons to zero.
- ▶ Probability of dropping is a hyperparameter; 0.5 is common

Convolutional network



- ▶ LeNet². The first successful convolutional neural network
- ▶ Designed to identify hand-written digits in the MNIST dataset
- ▶ LeNet-5 takes a single-channel 2D input
- ▶ Performs 6 convolution (5×5), then subsamples by max-pooling (2×2).
- ▶ The convolution-pooling layer sequence occurs again
- ▶ Finally 2 fully connected layer followed by a fully connected softmax layer is performed

²Yann Lecun and Léon Bottou and Yoshua Bengio and Patrick Haffner, **Gradient-based learning applied to document recognition**, 1998



Reference Convolutional Neural Networks

CNN used in image classification with reference results available:

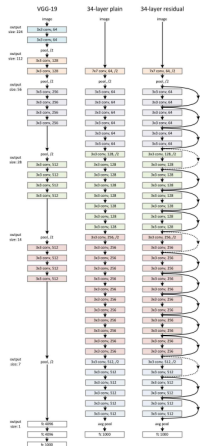
- ▶ VGG 16 (Simonyan, Zisserman, 2014)
 - ▶ **thirteen** convolutional layers (kernel size: 3×3 output channels: 64, 128, 256, 512)
 - ▶ five maxpool layers
 - ▶ three fully connected layers
 - ▶ Rectified Linear Unit **ReLU** as nonlinear activation function
- ▶ ResNet 50 and 152 (He, Zhang, Ren, Sun, 2015)
 - ▶ **fifty** and one hundred fifty-two convolutional layers
 - ▶ various kernel sizes ($7 \times 7, 3 \times 3, 1 \times 1$);
 - ▶ various output channels: 64, 128, 256, 512;
 - ▶ max-pool and fully-connected layers;
 - ▶ “residual” learning: output of convolutional layer is summed to input that generated it;
 - ▶ Rectified Linear Unit (**ReLU**) as nonlinear activation function
- ▶ Inception (Szegedy, Vanhoucke, Ioffe, Shlens, Wojna 2015)
 - ▶ six convolutional layers
 - ▶ ten **Inception** modules
 - ▶ Rectified Linear Unit **ReLU** as nonlinear activation function

VGG

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

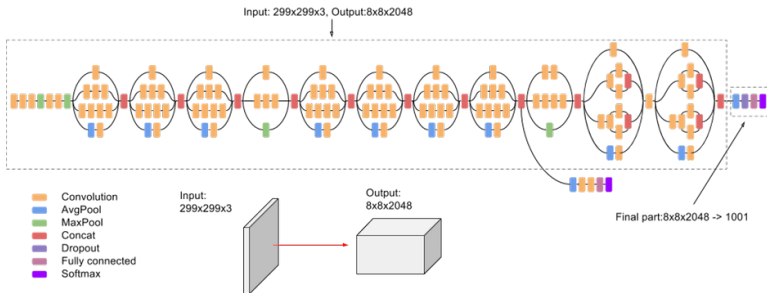
Source: Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014

ResNet



Source: He, Zhang, Ren & Sun, Deep Residual Learning for Image Recognition, 2015

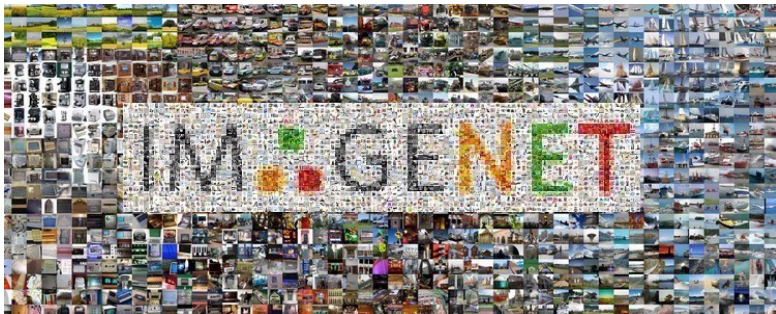
Inception



Source: Advanced Guide to Inception v3 on Cloud TPU

Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, Rethinking the Inception Architecture for Computer Vision, 2015

Imagenet dataset



- ▶ 22000 classes 11M labeled image examples
- ▶ Reduced to 1000 classes and 1.4M images
- ▶ The smaller dataset has both fine and coarse-grained classes
- ▶ Synthetic version keeps size intact (224x224)

image-net.org



Training Imagenet results

Table 1 : Training time and top-1 1-crop validation accuracy with ImageNet/ResNet-50

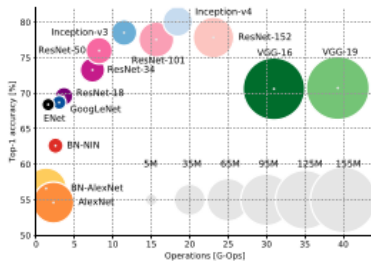
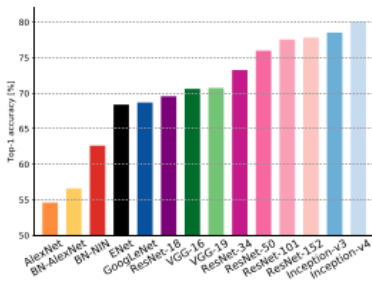
	Batch Size	Processor	DL Library	Time	Accuracy
He et al. [7]	256	Tesla P100 x8	Caffe	29 hours	75.3%
Goyal et al. [1]	8K	Tesla P100 x256	Caffe2	1 hour	76.3%
Smith et al. [4]	8K→16K	full TPU Pod	TensorFlow	30 mins	76.1%
Akiba et al. [5]	32K	Tesla P100 x1024	Chainer	15 mins	74.9%
Jia et al. [6]	64K	Tesla P40 x2048	TensorFlow	6.6 mins	75.8%
This work	34K→68K	Tesla V100 x2176	NNL	224 secs	75.03%

Table 2 : GPU scaling efficiency with ImageNet/ResNet-50 training

	Processor	Interconnect	GPU scaling efficiency
Goyal et al. [1]	Tesla P100 x256	50Gbit Ethernet	~90%
Akiba et al. [5]	Tesla P100 x1024	Infiniband FDR	80%
Jia et al. [6]	Tesla P40 x2048	100Gbit Ethernet	87.9%
This work	Tesla V100 x1088	Infiniband EDR x2	91.62%

Mikami, Suganuma, U-chupala, Tanaka & Kageyama, ImageNet/ResNet-50 Training in 224 Seconds

CNN comparison



- ▶ Inception-v4: Resnet + Inception
- ▶ VGG High memory and operations
- ▶ GoogLeNet very efficient
- ▶ Alexnet few operations but high memory and low accuracy
- ▶ Resnet moderate efficiency and high accuracy



Links and credits

- ▶ Fei-Fei Li, Justin Johnson, Serena Yeung - CS231n
- ▶ <https://developers.google.com/machine-learning/crash-course/>
- ▶ <https://eu.udacity.com/course/deep-learning--ud730>
- ▶ <https://www.kaggle.com/competitions>
- ▶ Credits to Riccardo Zanella for the first version of this course