Introduction to Neural Networks: Which Architectures, for which Purposes

Guillaume Charpiat

TAU team, LRI, Paris-Sud / INRIA Saclay

Mathematical Coffees, Huawei

9^{th} of May, 2017

Overview

- What's a neural net?
- Architectures
- Natural gradient

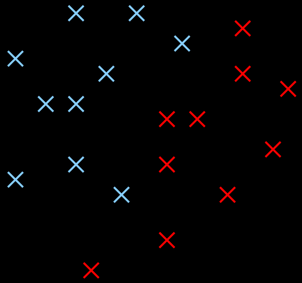
What's a neural net?

A classical machine learning tool

What's a neural net?

A classical machine learning tool

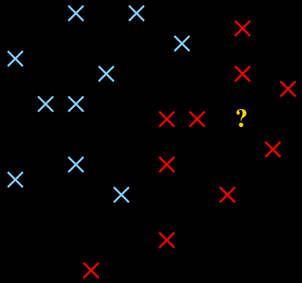
- Task to solve: explained with pairs (examples, expected answer)



What's a neural net?

A classical machine learning tool

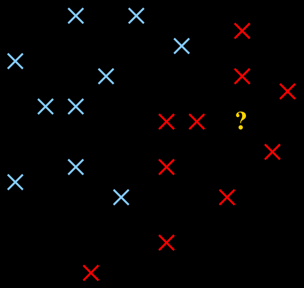
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer



What's a neural net?

A classical machine learning tool

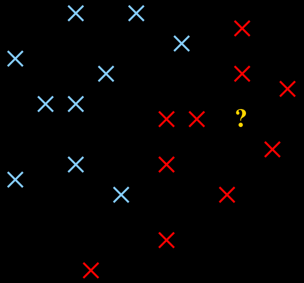
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...



What's a neural net?

A classical machine learning tool

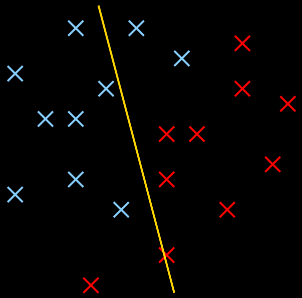
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

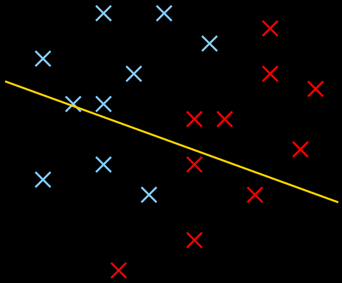
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

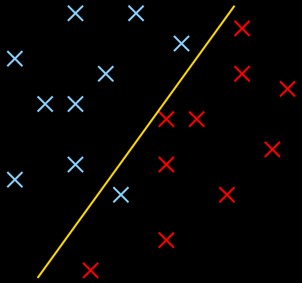
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

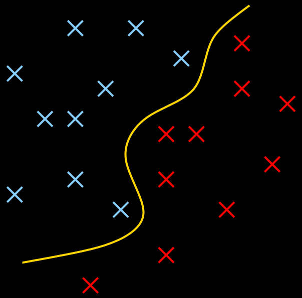
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

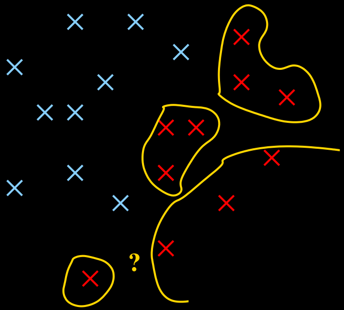
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

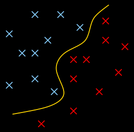
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

A classical machine learning tool

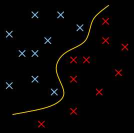
- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



What's a neural net?

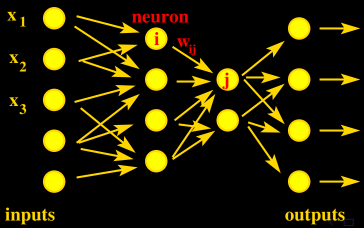
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

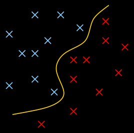
- very varied space of functions (the more layers, the more varied)



What's a neural net?

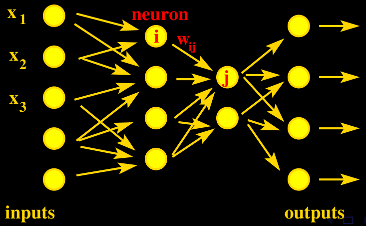
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

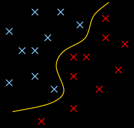
- very varied space of functions (the more layers, the more varied)
- parameters: connection weights w_{ij} between neurons



What's a neural net?

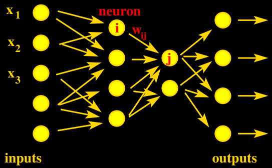
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

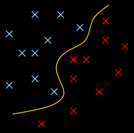
- very varied space of functions (the more layers, the more varied)
- parameters: connection weights w_{ij} between neurons
- parameters: $\theta = (w_{ij})_{i,j}$: many many many (millions)



What's a neural net?

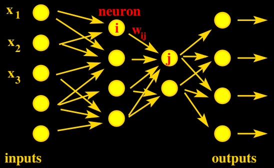
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

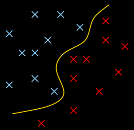
- very varied space of functions (the more layers, the more varied)
- parameters: connection weights w_{ij} between neurons
- parameters: $\theta = (w_{ij})_{i,j}$: many many many (millions)
- meta-parameters: architecture, type of neurons...



What's a neural net?

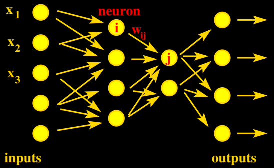
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

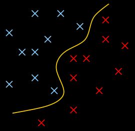
- very varied space of functions (the more layers, the more varied)
- parameters: connection weights w_{ij} between neurons
- parameters: $\theta = (w_{ij})_{i,j}$: many many many (millions)
- meta-parameters: architecture, type of neurons...
- very big space $(f_\theta)_\theta \Longrightarrow$ difficult optimization



What's a neural net?

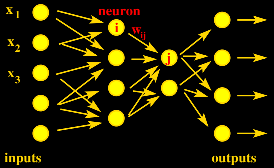
A classical machine learning tool

- Task to solve: explained with pairs (examples, expected answer)
- Search for best function: example $\mapsto$ answer
- Generalization power: regularizer, or restricted space of functions
- e.g.: linear functions, polynomials with degree $\leqslant 3$, mixture of Gaussians...
- $\Longrightarrow$ estimate the parameters of the best function in that space



Neural networks

- very varied space of functions (the more layers, the more varied)
- parameters: connection weights w_{ij} between neurons
- parameters: $\theta = (w_{ij})_{i,j}$: many many many (millions)
- meta-parameters: architecture, type of neurons...
- very big space $(f_\theta)_\theta \Longrightarrow$ difficult optimization
- gradient descent techniques: $\frac{d\theta}{dt} = -\nabla_\theta\, C(\theta)$



Architectures

Influence of the architecture

- We're searching for a function f_θ optimizing some criterion $C(f_\theta)$.
- Optimization in the space of parameters: $\theta \in \mathcal{P}_\mathcal{A}$
  $\Longrightarrow$ search space of functions $\mathcal{F}_\mathcal{A} = \{f_\theta, \text{ for } \theta \in \mathcal{P}_\mathcal{A}\}$: depends on the architecture $\mathcal{A}$
- more likely functions to be found when initializing with random coefficients
  $\Longrightarrow$ architecture $\mathcal{A}$ = prior on functions

Simplest architectures

Theorem: one layer can approximate any function if wide enough

In practice: many many parameters

$\Longrightarrow$ difficult to optimize, search space too big

Hierarchical networks: several layers

Aim: develop a series of features, from low-level (close to data, such as values) to high-level (detected objects).

Fully connected network

Issue: many neurons

$\Longrightarrow$ how to reduce the number of required parameters?

Standard architectures

Exploit desired invariances

- E.g.: in computer vision, to process images, or in text analysis, to process text: precise location within the data is not relevant
- all data (all locations) should be processed "the same way", and the immediate spatial neighborhood is more important
- $\Longrightarrow$ translational invariance
  $\Longrightarrow$ convolutional networks
- few parameters, easy to optimize, closer to what one would do intuitively
- Note: several features (filters) for each location: 3D tensors of neurons

Examples

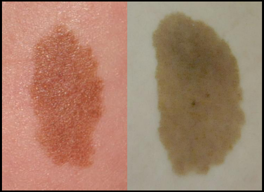
Classification of images

- dataset of skin pictures, from a hospital
- classes: operate / don't operate

Examples

Classification of images

- dataset of skin pictures, from a hospital
- classes: operate / don't operate
- difficulties: small part of the image, detection, white balance...
- work being done by Etienne Desbois (internship)

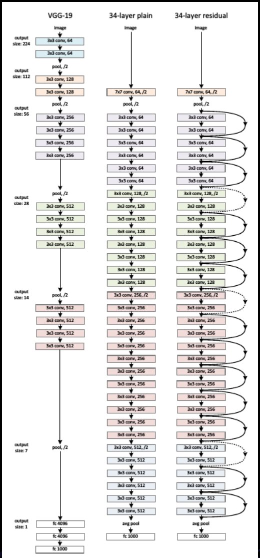


Impressive results in computer vision: Deeper architectures

Quantity of results in the last 4 years

Image classification

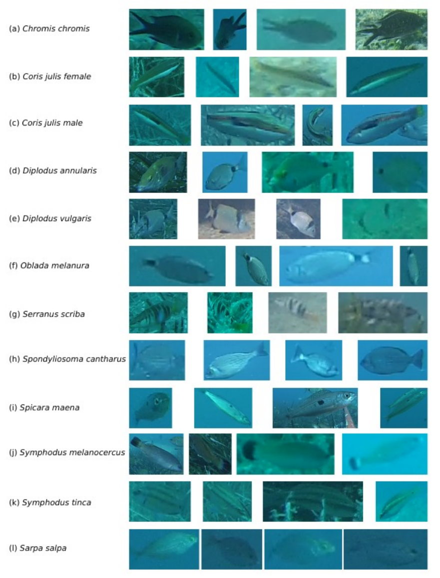
- ImageNet dataset: 1000 classes
- classification accuracy > 0.6 while many similar classes
- with (very) deep networks (15, 20... or 100 layers!)
- here: VGG and Resnet
- Deep Residual Learning for Image Recognition
  Kaiming He, Xiangyu Zhang, Shaoqing Ren Jian Sun
  Microsoft Research



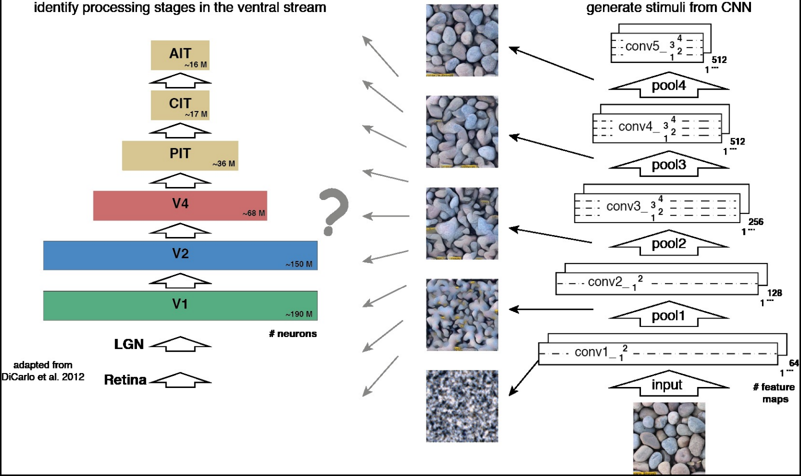
Impressive results in computer vision

Croatian Fish Dataset: Fine-grained classification of fish species in their natural habitat

Jonas Jaeger, Marcel Simon, Joachim Denzler, Viviane Wolff, Klaus Fricke-Neuderth, Claudia Kruschel



Texture generation

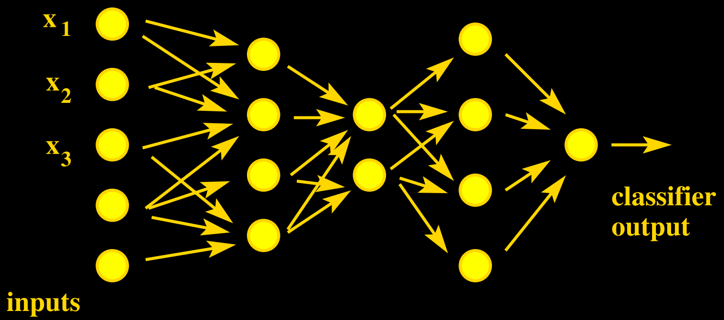


Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks
Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

Neural net as feature factory

A feature factory

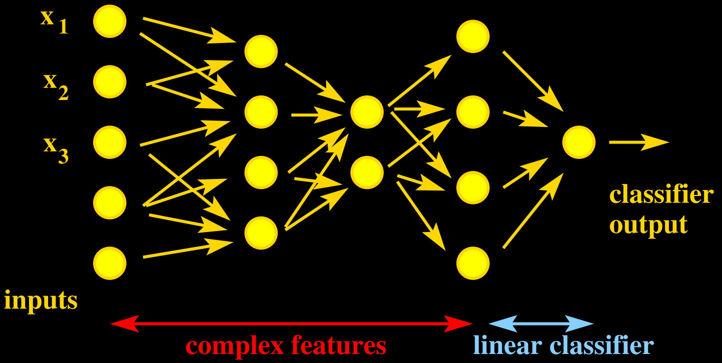
- set of hierarchical features



Neural net as feature factory

A feature factory

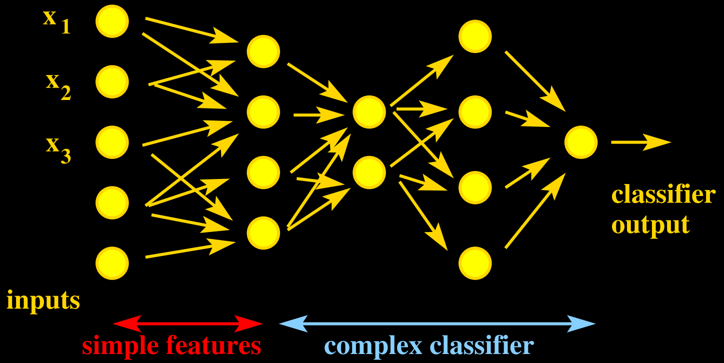
- set of hierarchical features



Neural net as feature factory

A feature factory

- set of hierarchical features



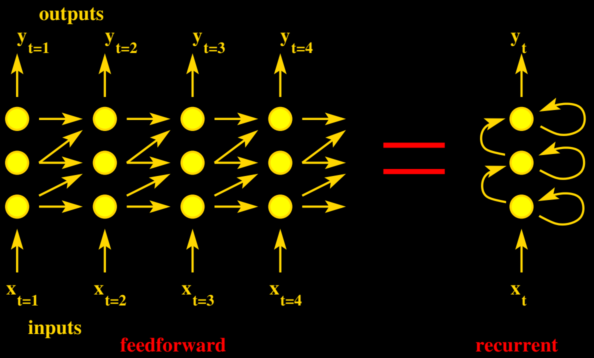
Style transfer

A Neural Algorithm of Artistic Style
Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

Recurrent networks (RNN)

Recurrent networks as dynamical systems

- recurrent networks (e.g.: LSTM, GRU)
- compute step by step with new inputs at each time t
- can be seen as a feedforward net with identical weights through time



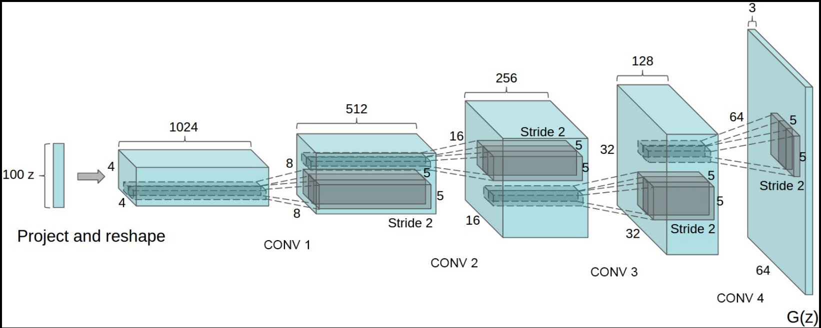
- several time scale dependencies? connect neurons accross various durations

Unsupervised approaches

Image generation

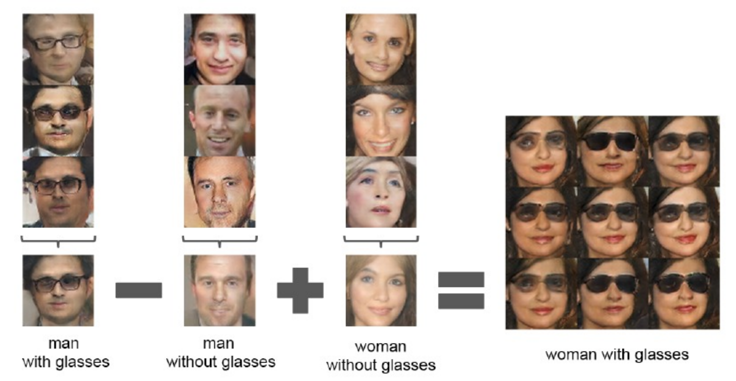
Unsupervised representation learning with deep convolutional generative adversarial networks
Alec Radford, Luke Metz, Soumith Chintala (Facebook AI Research)

Image generation



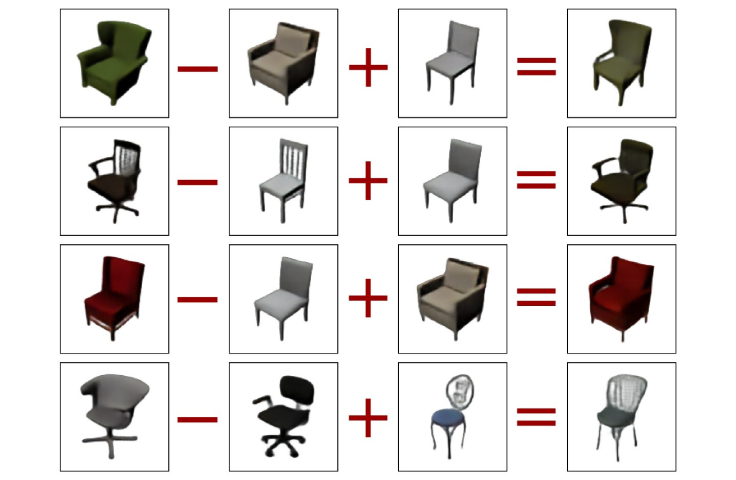
Unsupervised representation learning with deep convolutional generative adversarial networks
Alec Radford, Luke Metz, Soumith Chintala (Facebook AI Research)

Image generation : "face arithmetics"



Unsupervised representation learning with deep convolutional generative adversarial networks
Alec Radford, Luke Metz, Soumith Chintala (Facebook AI Research)

Image generation : chairs arithmetics...



Learning to Generate Chairs, Tables and Cars with Convolutional Networks
Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, Thomas Brox

Chair interpolation

Learning to Generate Chairs, Tables and Cars with Convolutional Networks

Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, Thomas Brox

Meanwhile, in Reinforcement Learning...

Several neural nets used (one to copy human experts), as parts of the main algorithm
Also: Atari games (no human knowledge included), etc.

And also

- Natural Language Processing
- Answering questions about text

Bilbo travelled to the cave. Gollum dropped the ring there. Bilbo took the ring.
Bilbo went back to the Shire. Bilbo left the ring there. Frodo got the ring.
Frodo journeyed to Mount-Doom. Frodo dropped the ring there. Sauron died.
Frodo went back to the Shire. Bilbo travelled to the Grey-havens. The End.
Where is the ring? A: Mount-Doom
Where is Bilbo now? A: Grey-havens
Where is Frodo now? A: Shire

Memory networks
Jason Weston, Sumit Chopra & Antoine Bordes (Facebook AI Research)

Problems and research tracks

Big data, scaling

- number of examples needed (huge)
- ML viewpoint: number of parameters $\Longrightarrow$ overfit
- input dimension: big (for images) $\Longrightarrow$ spurious correlations
- memory size (RAM), GPU/CPU consumption
- can't store history during learning

Problems and research tracks

Big data, scaling

- number of examples needed (huge)
- ML viewpoint: number of parameters $\Longrightarrow$ overfit
- input dimension: big (for images) $\Longrightarrow$ spurious correlations
- memory size (RAM), GPU/CPU consumption
- can't store history during learning

Optimization and meta-parameters

- initialization, optimization, sensitivity to adversarial noise
- for each new task, ask experts to build a new architecture
- and optimize over meta-parameters (precise architecure, type of neuron...)

Problems and research tracks

Big data, scaling

- number of examples needed (huge)
- ML viewpoint: number of parameters $\Longrightarrow$ overfit
- input dimension: big (for images) $\Longrightarrow$ spurious correlations
- memory size (RAM), GPU/CPU consumption
- can't store history during learning

Optimization and meta-parameters

- initialization, optimization, sensitivity to adversarial noise
- for each new task, ask experts to build a new architecture
- and optimize over meta-parameters (precise architecure, type of neuron...)

Lack of theory

- No theoretical guarantee (that training a network will work)
- in practice: small networks (3 layers) are easy to learn and sufficient to provide descriptors for many tasks

Problems and research tracks

Big data, scaling

- number of examples needed (huge)
- ML viewpoint: number of parameters $\Longrightarrow$ overfit
- input dimension: big (for images) $\Longrightarrow$ spurious correlations
- memory size (RAM), GPU/CPU consumption
- can't store history during learning

Optimization and meta-parameters

- initialization, optimization, sensitivity to adversarial noise
- for each new task, ask experts to build a new architecture
- and optimize over meta-parameters (precise architecure, type of neuron...)

Lack of theory

- No theoretical guarantee (that training a network will work)
- in practice: small networks (3 layers) are easy to learn and sufficient to provide descriptors for many tasks

Learning a program

- no variable or memory in the network... how to learn a program?
- how to reuse a neural network as part of another task?

Problems and research tracks

Big data, scaling

- number of examples needed (huge)
- ML viewpoint: number of parameters $\Longrightarrow$ overfit
- input dimension: big (for images) $\Longrightarrow$ spurious correlations
- memory size (RAM), GPU/CPU consumption
- can't store history during learning $\Longrightarrow$ NoBackTrack

Optimization and meta-parameters

- initialization, optimization, sensitivity to adversarial noise
- for each new task, ask experts to build a new architecture $\Longrightarrow$ learn structure
- and optimize over meta-parameters (precise architecure, type of neuron...)

Lack of theory

- No theoretical guarantee (that training a network will work)
- in practice: small networks (3 layers) are easy to learn and sufficient to provide descriptors for many tasks

Learning a program $\Longrightarrow$ my long-term goal

- no variable or memory in the network... how to learn a program?
- how to reuse a neural network as part of another task?

Learning the structure of a network

Architecture design issues

- Impressive results in computer vision, but large networks hard to optimize
- many different architectures are tried
- many meta-parameters to tune (type of neurons, layer type and size, stride, ...)
- a lot of time lost

Learning the structure of a network

Architecture design issues

- Impressive results in computer vision, but large networks hard to optimize
- many different architectures are tried
- many meta-parameters to tune (type of neurons, layer type and size, stride, ...)
- a lot of time lost

Elements of design

- convolutional networks: suited for images (and text); exploit spatial information, reduce the number of parameters, invariance to translation
- recently, slightly more flexible architectures tried (skip layers when needed)
- more complex architectures too... (pseudo-recursive structures)
- stochastic architectures: e.g., drop-out (neurons deleted half of the time during training)
- stochastic weights (drawn according to Gaussian distribution, parameter = mean)
- neural networks are highly redundant/robust in the sense that compressing their weights by 90% might not affect them much

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy
- What is the equivalent of functions in neural networks?

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy
- What is the equivalent of functions in neural networks?
- $\Longrightarrow$ repeated blocks of neurons with similar weights

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy
- What is the equivalent of functions in neural networks?
- $\Longrightarrow$ repeated blocks of neurons with similar weights
- $\Longrightarrow$ self-similarity prior on neural networks
  $\Longrightarrow$ emerging structures during training

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy
- What is the equivalent of functions in neural networks?
- $\Longrightarrow$ repeated blocks of neurons with similar weights
- $\Longrightarrow$ self-similarity prior on neural networks
  $\Longrightarrow$ emerging structures during training
- simplest version: a block = one edge or one neuron $\Longrightarrow$ Bernouilli process
  $\Longrightarrow$ Pierre Wolinski PhD thesis

Learning the structure of a network (II)

(skippable)

An Information Theory viewpoint

- Kolmogorov complexity: length of the shortest program which can generate the data
- in practice: search for the simplest model suited to the data
- here, we see neural networks as programs and try to exploit any form of redundancy
- What is the equivalent of functions in neural networks?
- $\Longrightarrow$ repeated blocks of neurons with similar weights
- $\Longrightarrow$ self-similarity prior on neural networks
  $\Longrightarrow$ emerging structures during training
- simplest version: a block = one edge or one neuron $\Longrightarrow$ Bernouilli process
  $\Longrightarrow$ Pierre Wolinski PhD thesis
- noise during training $\Longleftrightarrow$ Jacobian regularization

Bonus

Architectures bonus: unsupervised learning

- generative models: auto-encoders
- adversarial approaches (DANN, GAN): to help improve the generated distribution / in order not to have to specify the task explicitely!

Examples or recurrent networks as PDEs

Semantic segmentation of images

- dataset of satellite images
- classes: road, building, grass, trees, lake, swimming pool...
- difficulties: very small objects, need for precise boundaries
- refine available segmentation with a Partial Differential Equation (PDE)
- learn it with a recurrent network

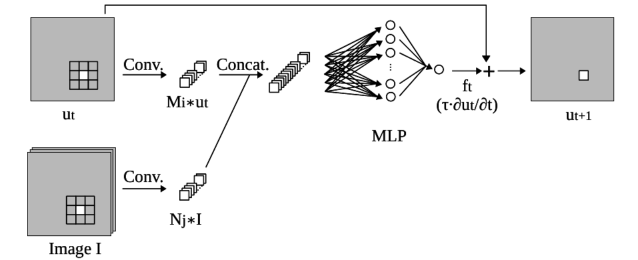
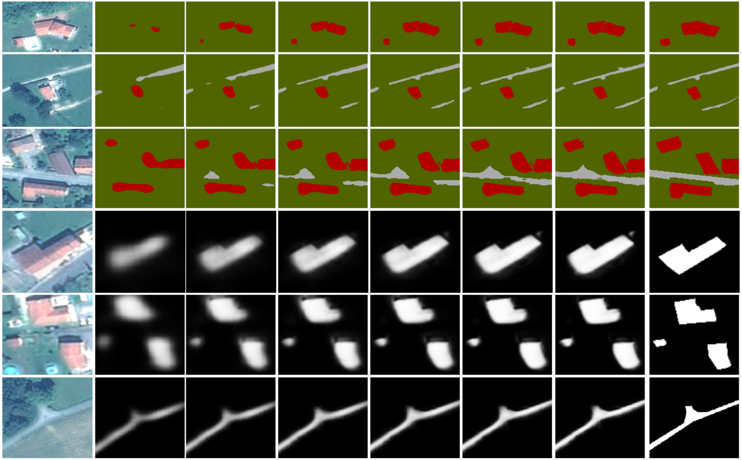


Figure 1: One PDE iteration represented as common neural network layers.

Examples or recurrent networks as PDEs



Color CNN map Intermediate RNN iterations RNN output Ground truth

Figure 3: Evolution of fragments of classification maps (top rows) and single-class fuzzy scores (bottom rows) through RNN iterations.

Examples or recurrent networks as PDEs

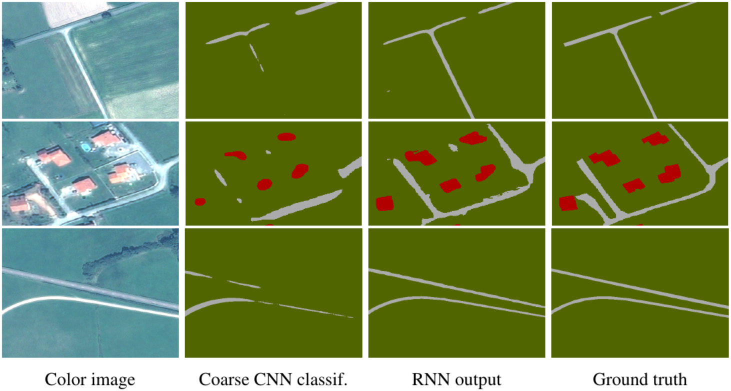
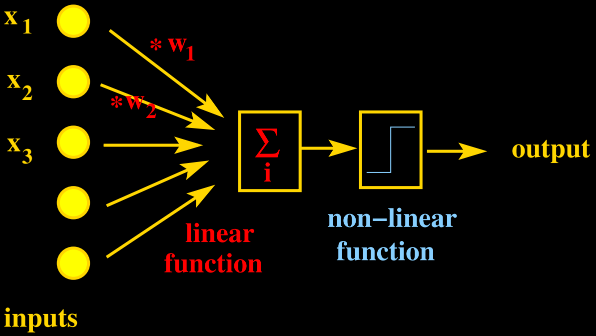


Figure 5: Initial coarse classifications and the enhanced maps by using RNNs.

- joint work with Emmanuel Maggiori & Yuliya Tarabalka (INRIA Sophia-Antipolis)

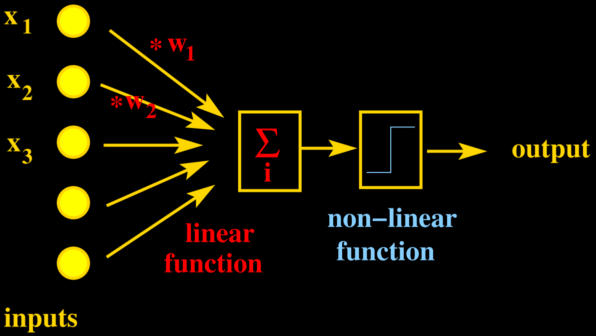
History

Perceptron [Rosenblatt, 1957]



History

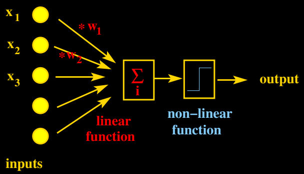
Perceptron [Rosenblatt, 1957]



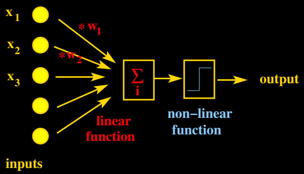
- only one layer
- = linear classifier
- inspired from brain and Hebb's work

History

Perceptron [Rosenblatt, 1957]



History

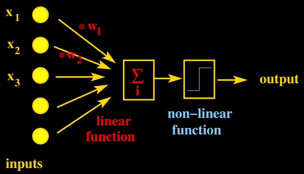


Perceptron [Rosenblatt, 1957]

Multi-layer perceptron (MLP)

- Backpropagation (derivation chain rule)

History



Perceptron [Rosenblatt, 1957]

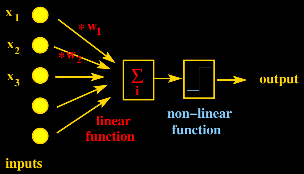
Multi-layer perceptron (MLP)

- Backpropagation (derivation chain rule)

Book Perceptrons [Minsky & Papert, 1969]

- Theorem: A perceptron (single layer) cannot learn XOR
- $\Longrightarrow$ break in research on neural networks

History



Perceptron [Rosenblatt, 1957]

Multi-layer perceptron (MLP)

- Backpropagation (derivation chain rule)

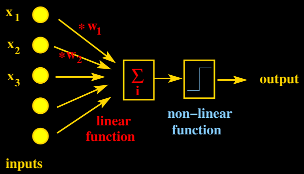
Book Perceptrons [Minsky & Papert, 1969]

- Theorem: A perceptron (single layer) cannot learn XOR
- $\Longrightarrow$ break in research on neural networks

End of 80's, 90's, 2000: resurgence

- in computer vision: LeCun, Hinton, Bengio, Schmidhuber...
- but no impact yet in the community (hand-made descriptors)

History



Perceptron [Rosenblatt, 1957]

Multi-layer perceptron (MLP)

- Backpropagation (derivation chain rule)

Book Perceptrons [Minsky & Papert, 1969]

- Theorem: A perceptron (single layer) cannot learn XOR
- $\Longrightarrow$ break in research on neural networks

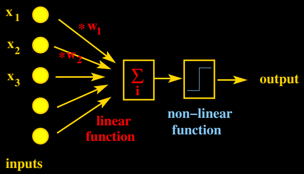
End of 80's, 90's, 2000: resurgence

- in computer vision: LeCun, Hinton, Bengio, Schmidhuber...
- but no impact yet in the community (hand-made descriptors)

2012: neural nets win great challenges in vision (image classification)

- "Deep Learning"

History



Perceptron [Rosenblatt, 1957]

Multi-layer perceptron (MLP)

- Backpropagation (derivation chain rule)

Book Perceptrons [Minsky & Papert, 1969]

- Theorem: A perceptron (single layer) cannot learn XOR
- $\Longrightarrow$ break in research on neural networks

End of 80's, 90's, 2000: resurgence

- in computer vision: LeCun, Hinton, Bengio, Schmidhuber...
- but no impact yet in the community (hand-made descriptors)

2012: neural nets win great challenges in vision (image classification)

- "Deep Learning"

Since then: explosion of results and popularity