Earth Data Science

course

Crash course of

deep learning

2022/2023 – December 7th, 2022

Grégory Sainton
PhD, Research engineer
sainton@ipgp.fr

Goals

Demystify Deep Learning

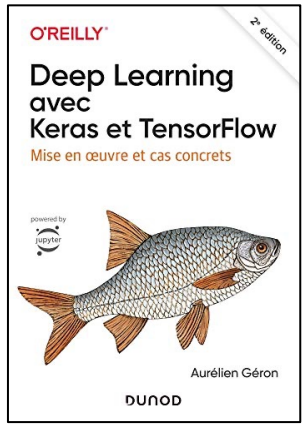
Understand ML *vs* DL

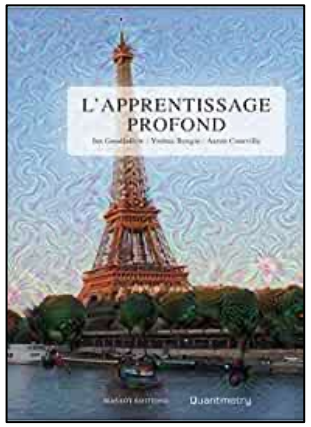
Discover Keras through few notebooks

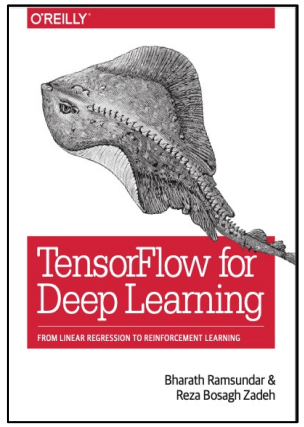
Inspired from FIDLE courses

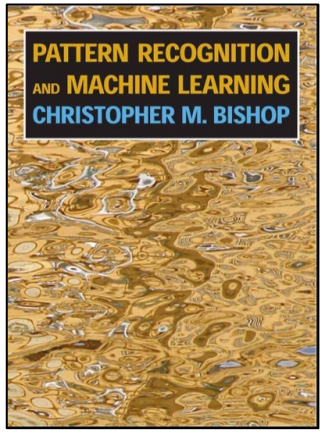
Few books/sites to prepare this course

A little selection









FIDLE Formation Introduction au Deep Learning

Online course by CNRS

Machine Learning by Andrew Ng

coursera

Online course by Andrew NG

Not exhaustive for sure…

How to learn deep learning

Online

- www.kaggle.com : many dataset, many notebooks and event contests needed
- www.coursera.com – Introduction to Deep Learning by Andrew Ng (Free)
- https://openclassrooms.com/fr - Initiez-vous au Deep Learning
- Have a look in Google Colab in training notebooks : https://colab.research.google.com/
- And many more…

Outlines

Generality	Machine learning, deep learning, IA… Supervised vs unsupervised Artificial network Gradient descent Artificial neural network
Deep Learning	Deep network Training concept Loss and accuracy Pathologies in training Intuition about regularization Set your hyperparameters
Specific networks	Convolution neural networks Recurrent neural networks

Outlines

Generality	Machine learning, deep learning, IA… Supervised vs unsupervised Artificial network Gradient descent Artificial neural network
Deep Learning	Deep network Training concept Loss and accuracy Pathologies in training Intuition about regularization Set your hyperparameters
Specific networks	Convolution neural networks Recurrent neural networks

Machine learning, deep learning, AI…

Artificial intelligence (AI)
Formal definition

John McCarthy

Marvin Lee Minsky

Reproduction of human intellectual activities

A program that can sense, reason, act and adapt.

John McCarthy: term of « Articificial Intelligence »

Marvin Lee Minsky: « The building of computer programs which perform tasks which are, for the moment, performed in a more satisfactory way by humans because they require high level mental processes such as: perception learning, memory organization and critical reasoning »

Oxford English Dictionnary: "Theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages."

Machine learning, deep learning, AI…

Artificial intelligence (AI)
Myths

Reproduction of human intellectual activities

A program that can sense, reason, act and adapt.


Ex-Machina (2014)


Avengers, age of Ultron (2015)


Maymax (2014)


Wall.E (2008)


Her (2013)

2001, A Space Odyssey (1968)

Very well represented in the fiction but not reality -> **Strong AI**

Machine learning, deep learning, AI…

Weak AI vs…

- Weak AI, also known as narrow AI, focuses on performing a specific task:
  - answering questions based on user input.
  - can perform one type of task.
- It relies on human interference to define the parameters of its learning algorithms.
- For example: self-driving cars, virtual assistants, translators, image recognition…

…Strong AI

- can perform a variety of functions.
- eventually teaching itself to solve for new problems.
- it develops a human-like consciousness instead of simulating it.
- … still a theoretical concept !

Machine learning, deep learning, IA…

Artificial intelligence (AI)

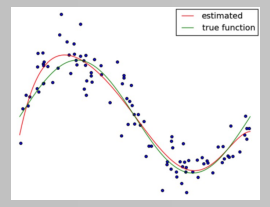
Reproduction of human intellectual activities

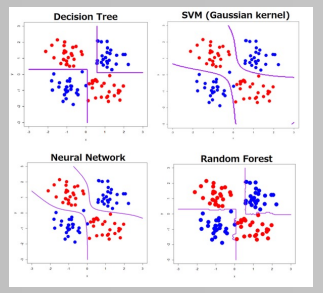
Programs that can sense, reason, act and adapt.

Machine learning (ML)

Learning of rules and patterns hidden in data

Algorithms whose performance improve as they are exposed to more data over time.





- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

Main scope of the EDS course

Machine learning, deep learning, IA…

Artificial intelligence (AI)

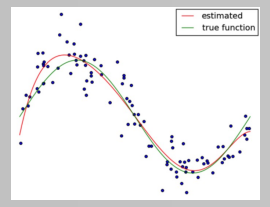
Reproduction of human intellectual activities

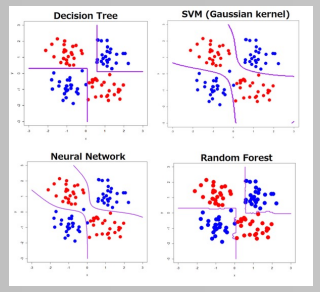
Programs that can sense, reason, act and adapt.

Machine learning (ML)

Learning of rules and patterns hidden in data

Algorithms whose performance improve as they are exposed to more data over time.

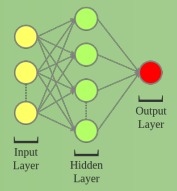




- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

Neural Networks (NN)

'Brain activities emulation'



Machine learning, deep learning, IA…

Artificial intelligence (AI)

Machine learning (ML)

Neural Networks (NN)

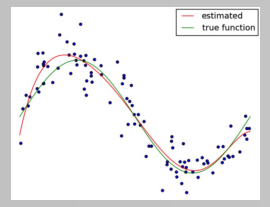
Deep learning (DL)

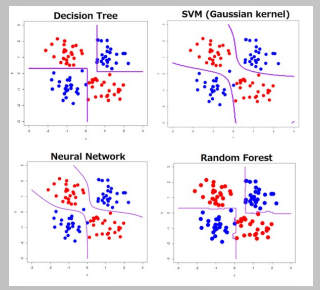
Reproduction of human intellectual activities

Programs that can sense, reason, act and adapt.

Learning of rules and patterns hidden in data

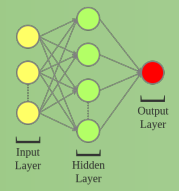
Algorithms whose performance improve as they are exposed to more data over time.





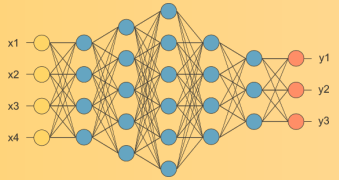
- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

'Brain activities emulation'



Multi-layer NN architecture

Subset of machine learning in which multilayered neural networks learn from vast amounts of data.



Machine learning, deep learning, IA…

Artificial intelligence (AI)

Machine learning (ML)

Neural Networks (NN)

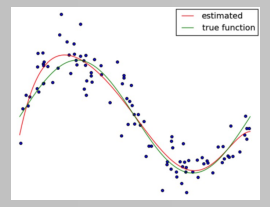
Deep learning (DL)

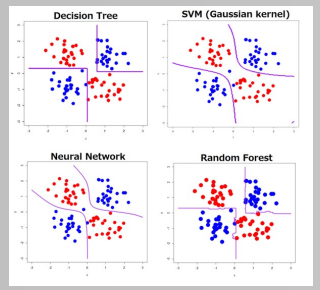
Reproduction of human intellectual activities

Programs that can sense, reason, act and adapt.

Learning of rules and patterns hidden in data

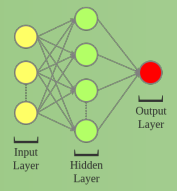
Algorithms whose performance improve as they are exposed to more data over time.





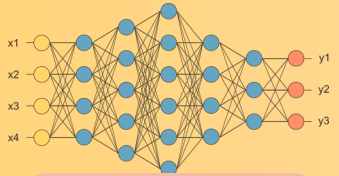
- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

'Brain activities emulation'

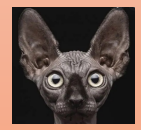


Multi-layer NN architecture

Subset of machine learning in which multilayered neural networks learn from vast amounts of data.



Convolutionnal Neural Netwoks



Computer vision, image processing

Recurrent Neural Netwoks

Natural language processing, time series

Machine learning, deep learning, IA…

Artificial intelligence (AI)

Machine learning (ML)

Neural Networks (NN)

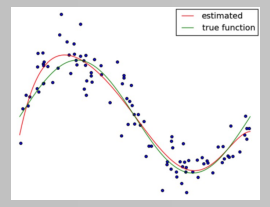
Deep learning (DL)

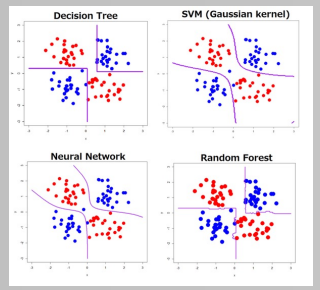
Reproduction of human intellectual activities

Programs that can sense, reason, act and adapt.

Learning of rules and patterns hidden in data

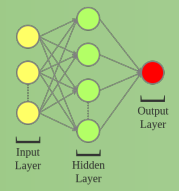
Algorithms whose performance improve as they are exposed to more data over time.





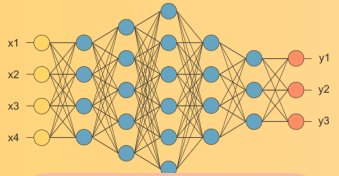
- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

Brain activities emulation

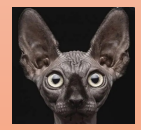


Multi-layer NN architecture

Subset of machine learning in which multilayered neural networks learn from vast amounts of data.



Convolutionnal Neural Netwoks



Computer vision, image processing

Recurrent Neural Netwoks

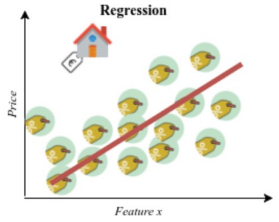
Natural language processing, time series

Supervised vs unsupervised

Supervised

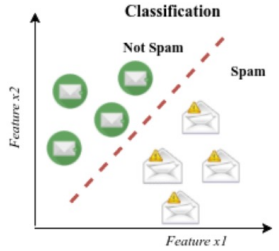
Regression

Predict quantitative info.



Classification

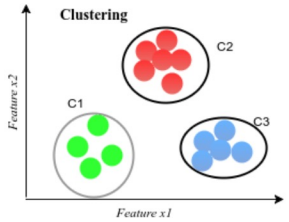
Predict qualitative info.



Unsupervised

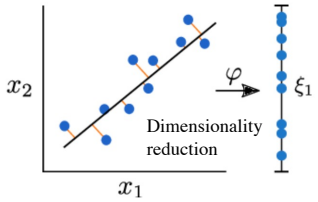
Clustering

Finding common features



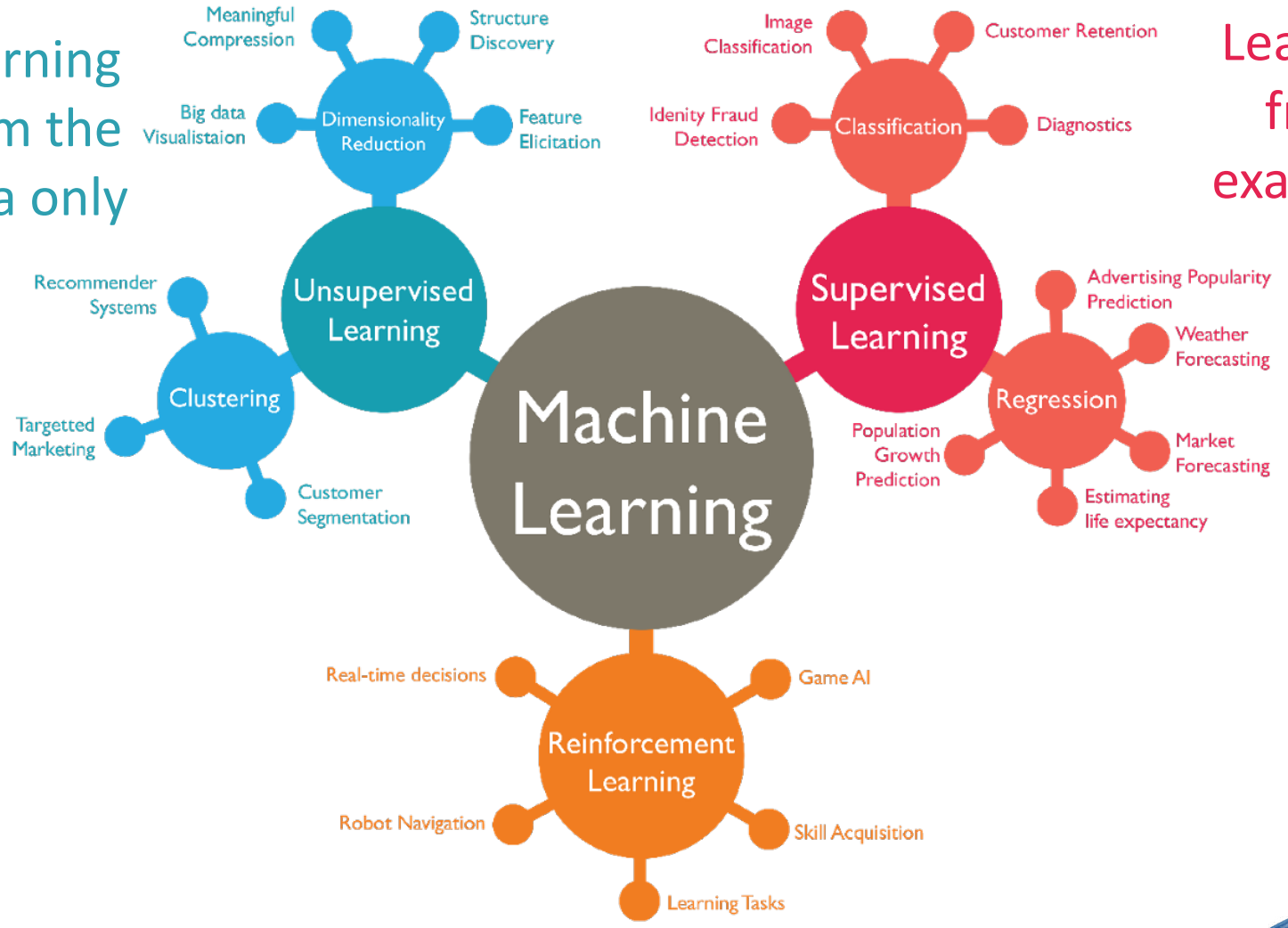
Dimensionality reduction

Reduce the number of reduction



Learning from the data only

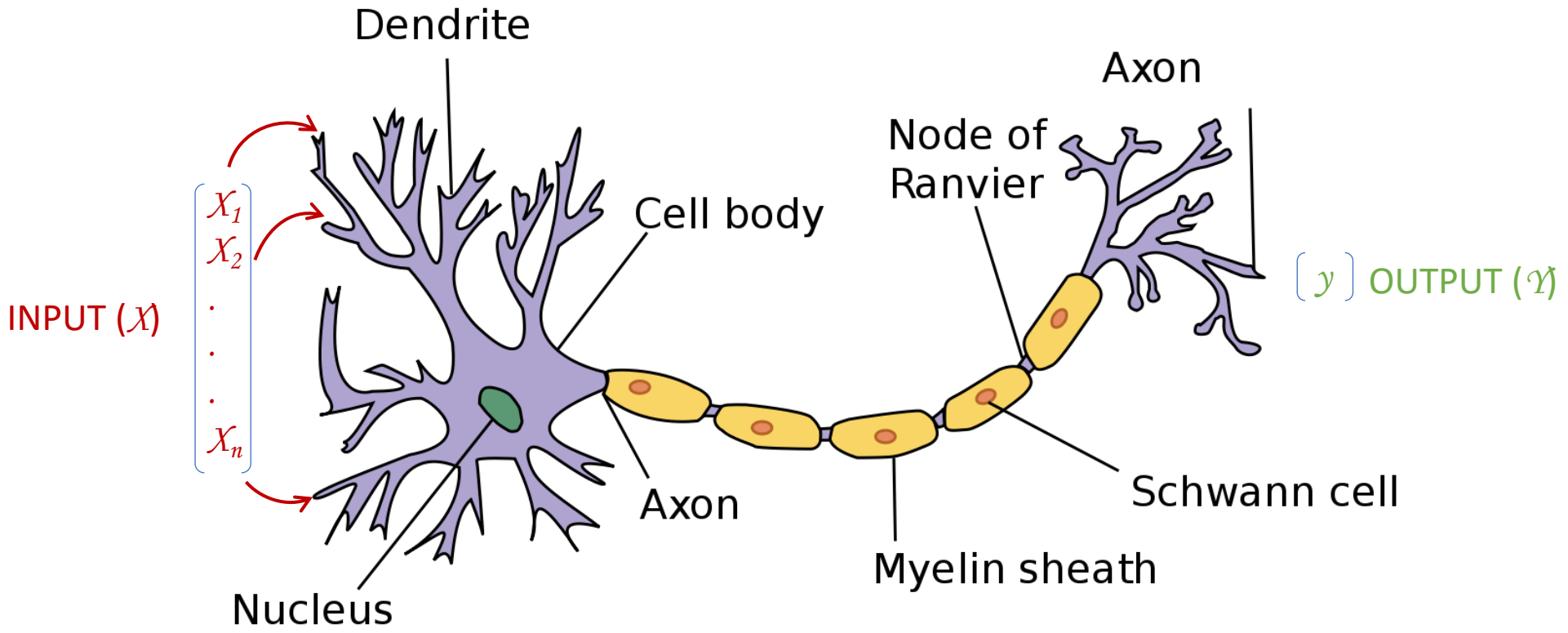
Learning from examples



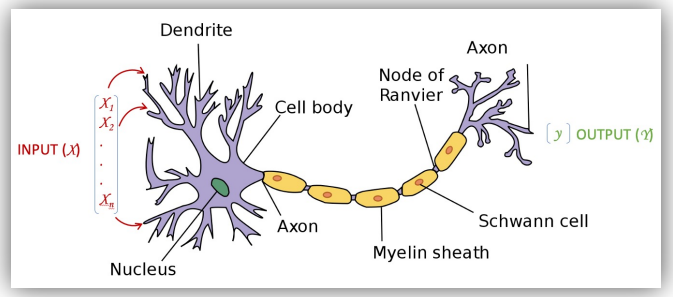
Biological neuron

Biological neurons receive short electrical impulses from other neurons (X).

If a neuron receives a sufficient number of signals, it fires its own signals (Y) .



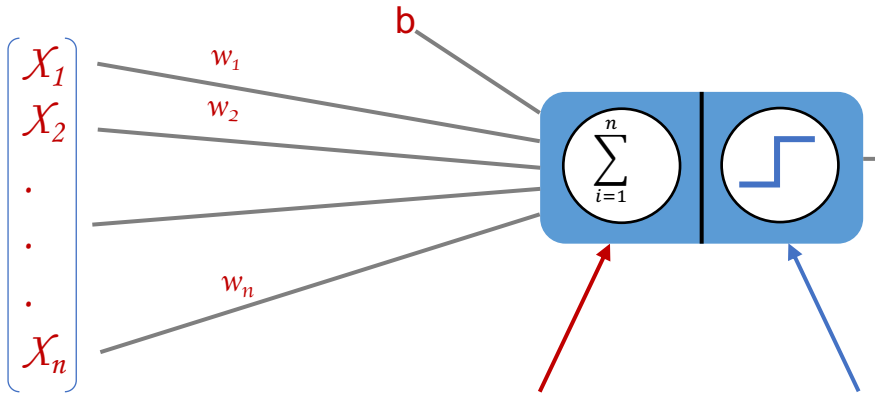
Artificial neuron



Biological neurons receive short electrical impulses from other neurons (X).

If a neuron receives a sufficient number of signals, it fires its own signals (Y) .

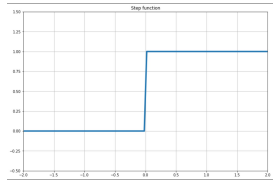
INPUT (X)



$$\hat{y} = \sigma(X^T W + b)$$

$$z = \sum_{1}^{n} x_i w_i + b = X^T W + b$$

$$\sigma(z) = \begin{cases} 0 \; if \; z < 0 \\ 1 \; if \; z \geq 0 \end{cases}$$ Activation function



Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408.

Perceptron is a basic neural network unit -> used for binary classification

Edible ?
YES / NO

But wait ! How to get the optimized w and b ?...

Gradient descent

Calculation vs. reality

How far the measure $\hat{y}$ is from the reality y

Cost function
To be minimized

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

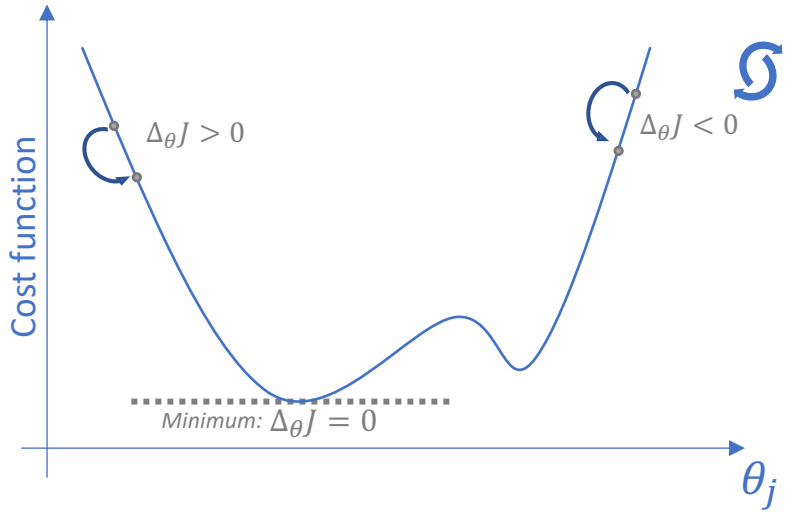
Mean Square Error (MSE) cost function is pretty common.

Gradient descent

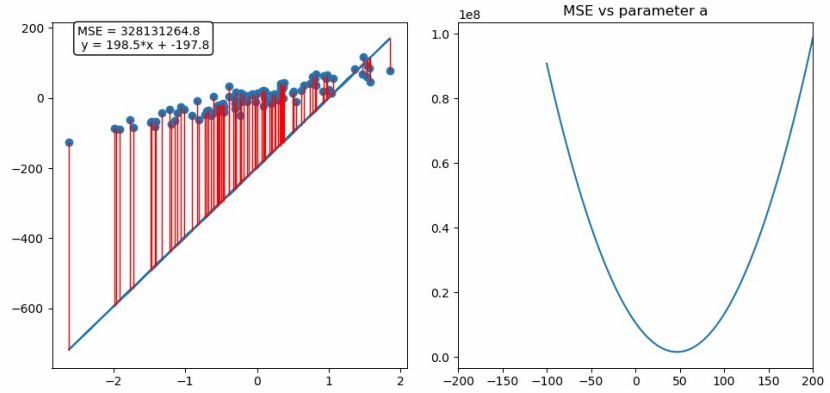
$$\theta_j^{new} = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

With α the learning rate

Minimization



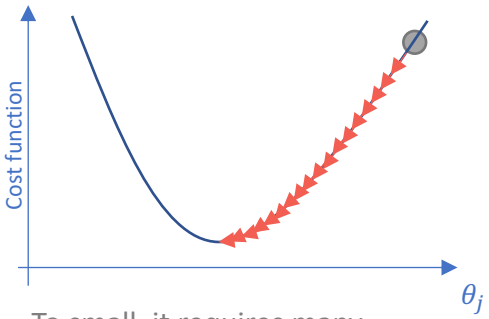
To be repeated until convergence



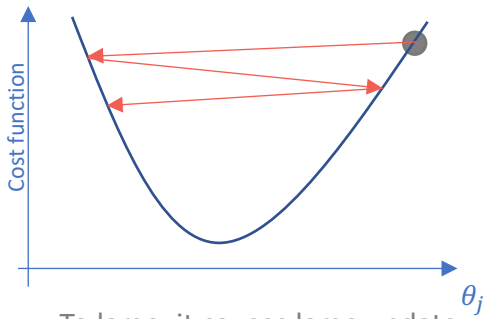
How to choose α ?

Gradient descent - 2

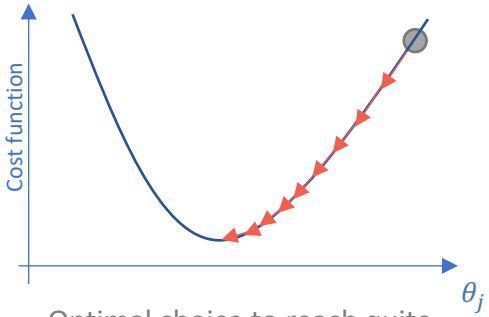
Learning rate α



To small, it requires many iteration, it takes very long time



To large, it causes large update, can lead to divergent behaviours

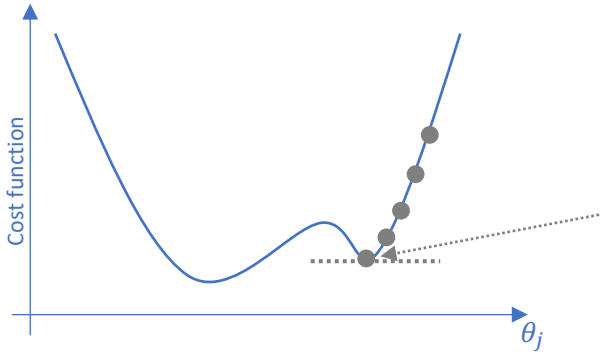


Optimal choice to reach quite rapidly the minimum

Advices to choose α

1. Plot $J(\theta)$ in function of the number of iteration: it must decrease
2. One can implement an automatic convergence test
3. Can be useful to test with $\alpha = 0.001; 0.01; 0.1; 1; 10;$ … and check the point 1.

The case of local minimum



Cost function stuck in a local minimum

-> Use more robust GD algorithms like

- **mini-batch gradient descent,**
- **stochastic gradient descent (SGD)**.

Gradient descent - 3

Batch GB
Stochastic GD
Mini-batch GD

- Whole training dataset at the time to compute GD -> **Batch GD**

- **Single training example** at the time to compute GD -> **Stochastic gradient descent (SGD)**

- **Multiple training examples** at the time (but less than the whole dataset) to compute GD -> **Mini-Batch Gradient Descent**

> « Most deep learning algorithms are based on an optimization algorithm called stochastic gradient descent. Deep Learning, Goodfellow et al.

Notebook – the perceptron

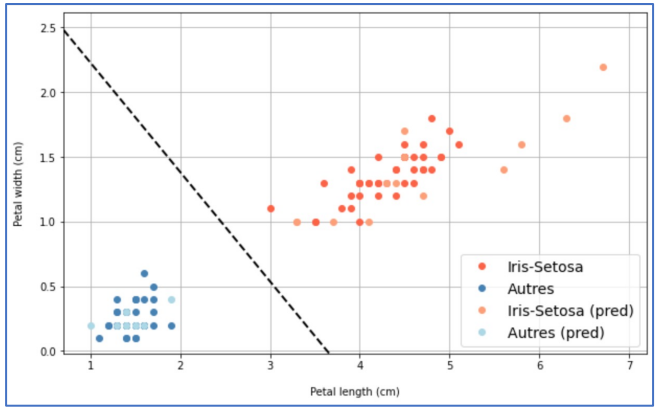
Goal

- First introduction with Scikit Learn
- Show few Pandas commands
- Play with the Perceptron as linear classifier
- Predict the size of the leaves of the IRIS dataset (Fisher, 1936)

Advices

This is just a quick look so…

- Don’t hesitate to play with the notebook.
- Look at the documentation which contain many explanations about the numerous parameters.
- Test the same classifier with different datasets
- Test the same dataset with different classifiers

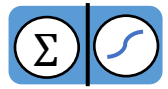


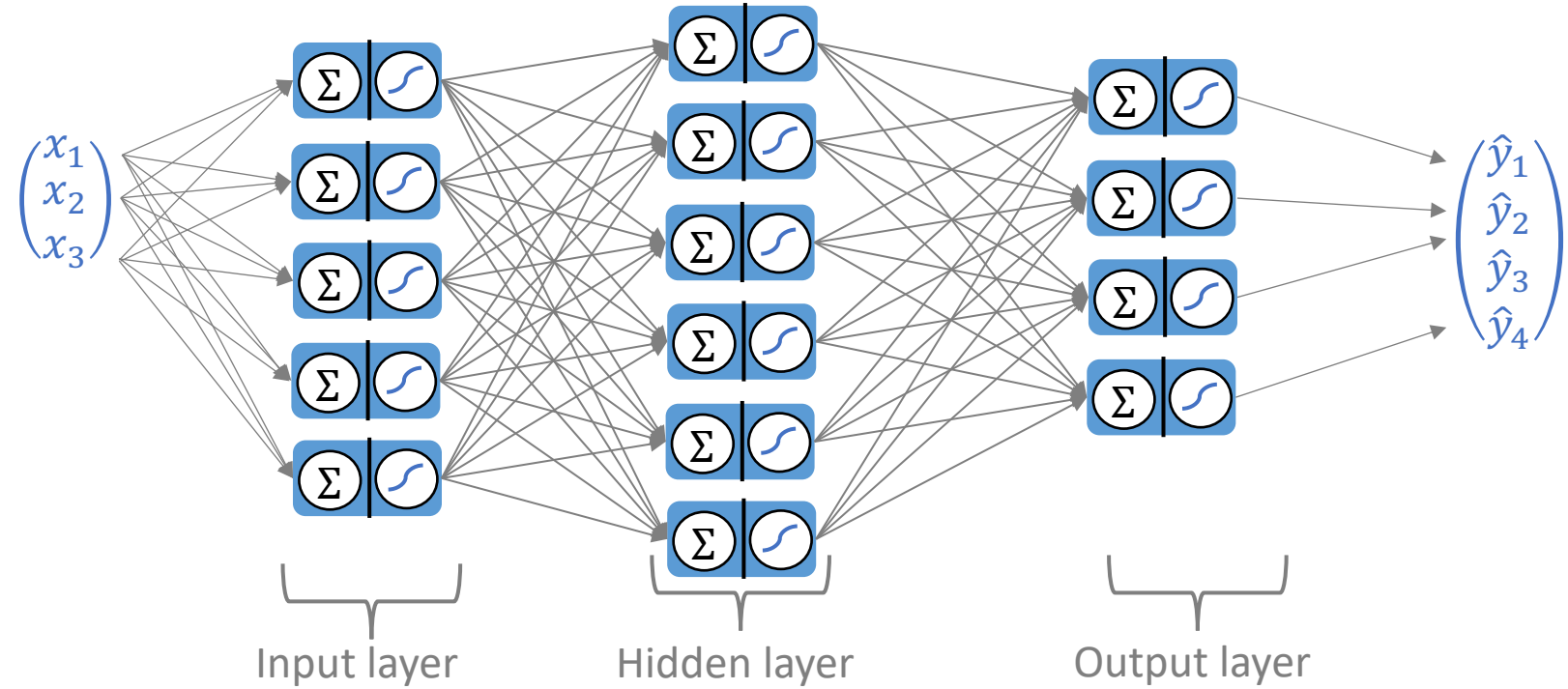
Let’s go…

Artificial neural network

From one neuron…

… to a network





This one is a fully connected network (FCN)

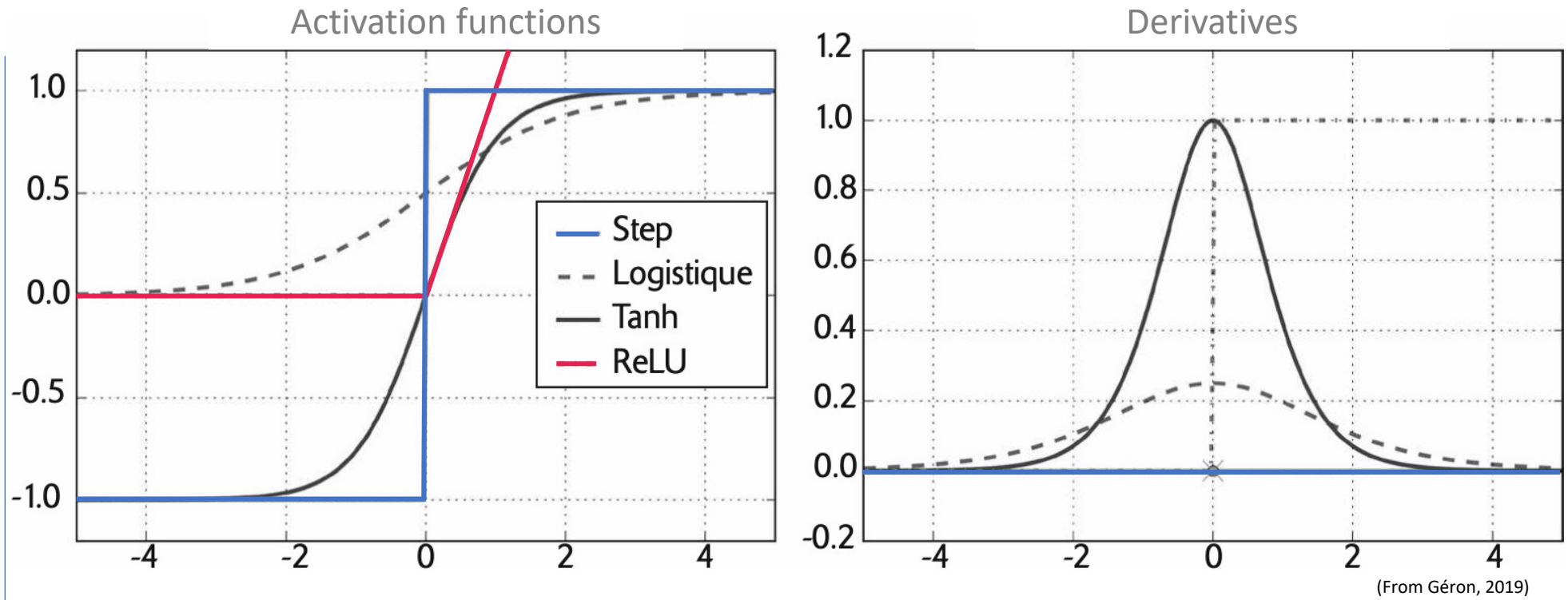
👉 One hidden layer
= shallow network

👉 More hidden layers
= deep network

From now, we enter in the realm of the deep learning

Activation functions

Plot of the activation functions



(From Géron, 2019)

Usage

- ReLU or Leaky ReLU in the hidden layers
- Sigmoid and SoftMax in the output layers

A Notebook is available to play with activation functions.

Artificial neural network (with a some maths)

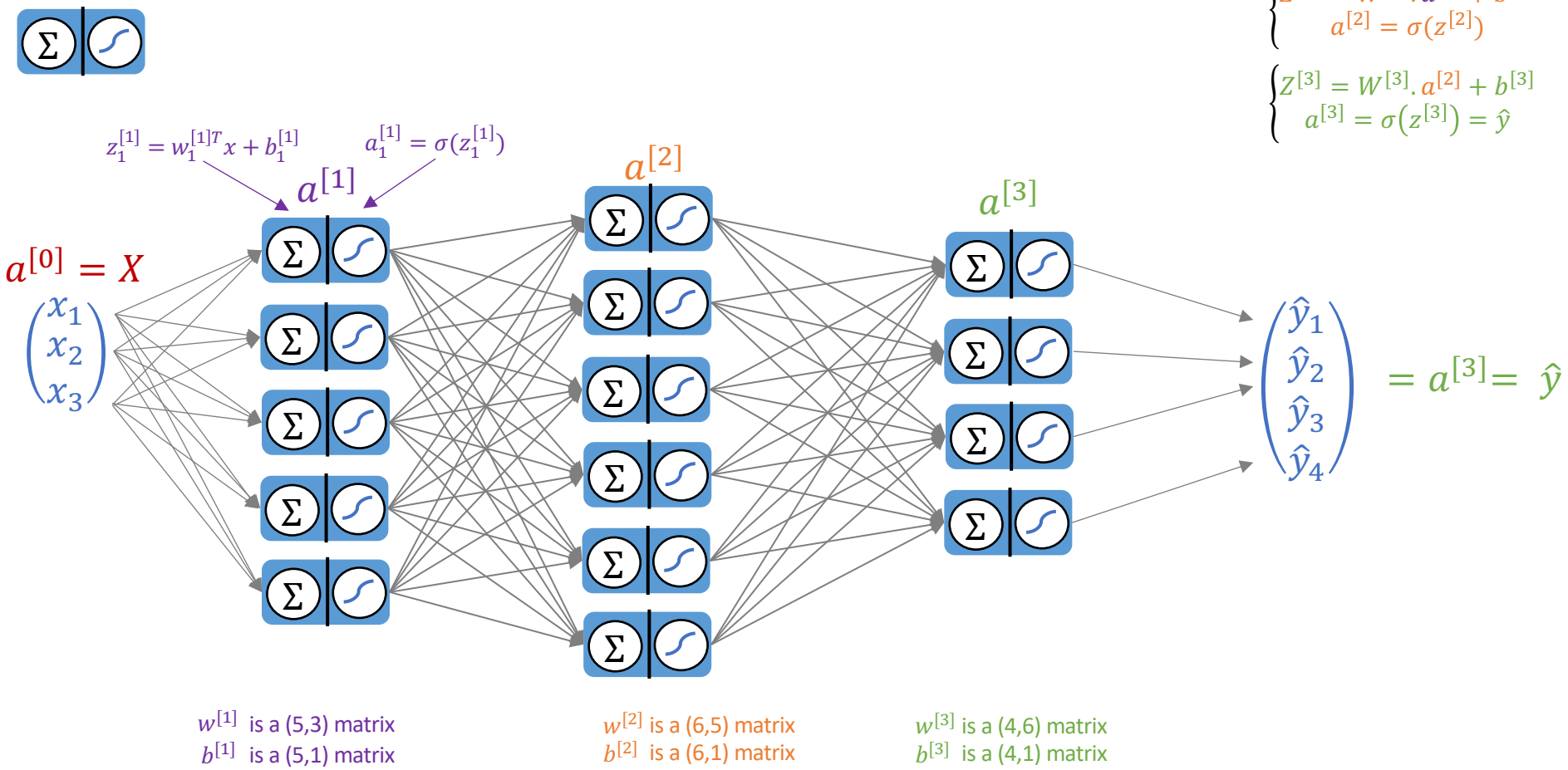
$$\begin{cases} Z^{[1]} = W^{[1]}.a^{[0]} + b^{[1]} \\ a^{[1]} = \sigma(z^{[1]}) \end{cases}$$

$$\begin{cases} Z^{[2]} = W^{[2]}.a^{[1]} + b^{[2]} \\ a^{[2]} = \sigma(z^{[2]}) \end{cases}$$

$$\begin{cases} Z^{[3]} = W^{[3]}.a^{[2]} + b^{[3]} \\ a^{[3]} = \sigma(z^{[3]}) = \hat{y} \end{cases}$$

From one neuron…

… to a network



$w^{[1]}$ is a (5,3) matrix
$b^{[1]}$ is a (5,1) matrix

$w^{[2]}$ is a (6,5) matrix
$b^{[2]}$ is a (6,1) matrix

$w^{[3]}$ is a (4,6) matrix
$b^{[3]}$ is a (4,1) matrix

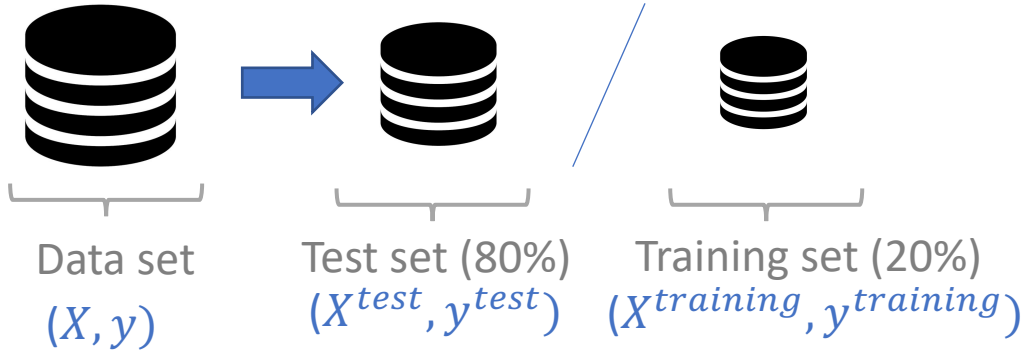
- One hidden layer = shallow network
- More hidden layers = deep network

Outlines

Generality	Machine learning, deep learning, IA… Supervised vs unsupervised Artificial network Gradient descent Artificial neural network
Deep Learning	Deep network Training concept Loss and accuracy Pathologies in training Intuition about regularization Set your hyperparameters
Specific networks	Convolution neural networks Recurrent neural networks

Deep network

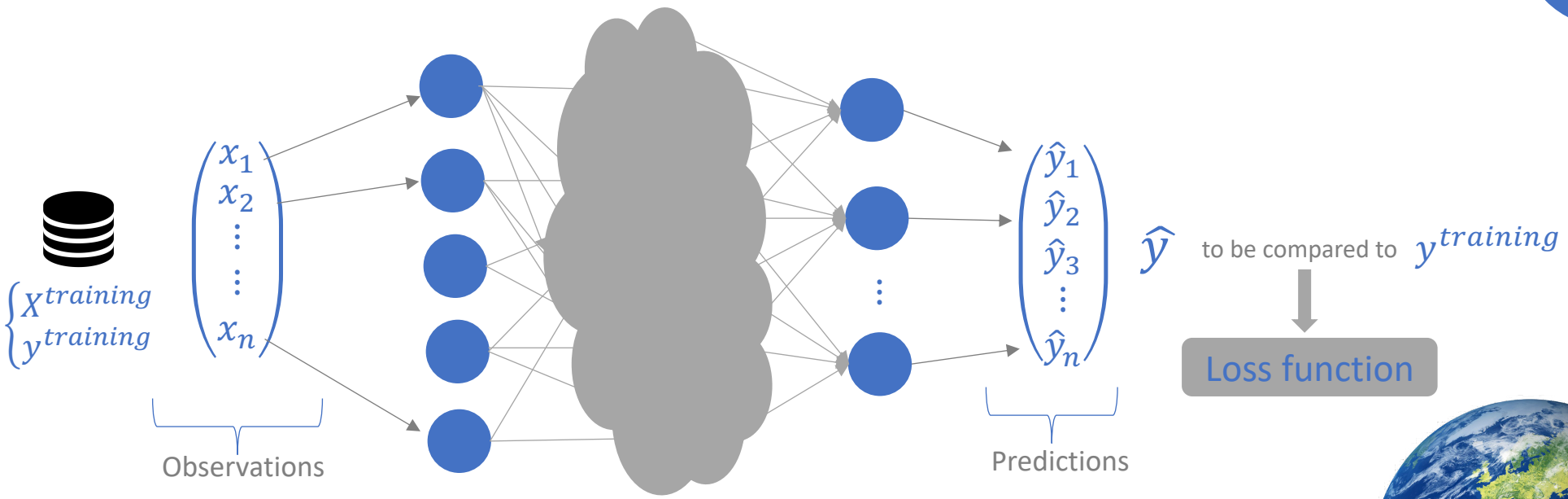
Data Set



$\begin{cases} X^{training} & \text{Observations} \\ y^{training} & \text{Outputs/Labels} \end{cases}$

Feed Forward

Apply gradient descent true all connexions of the network.

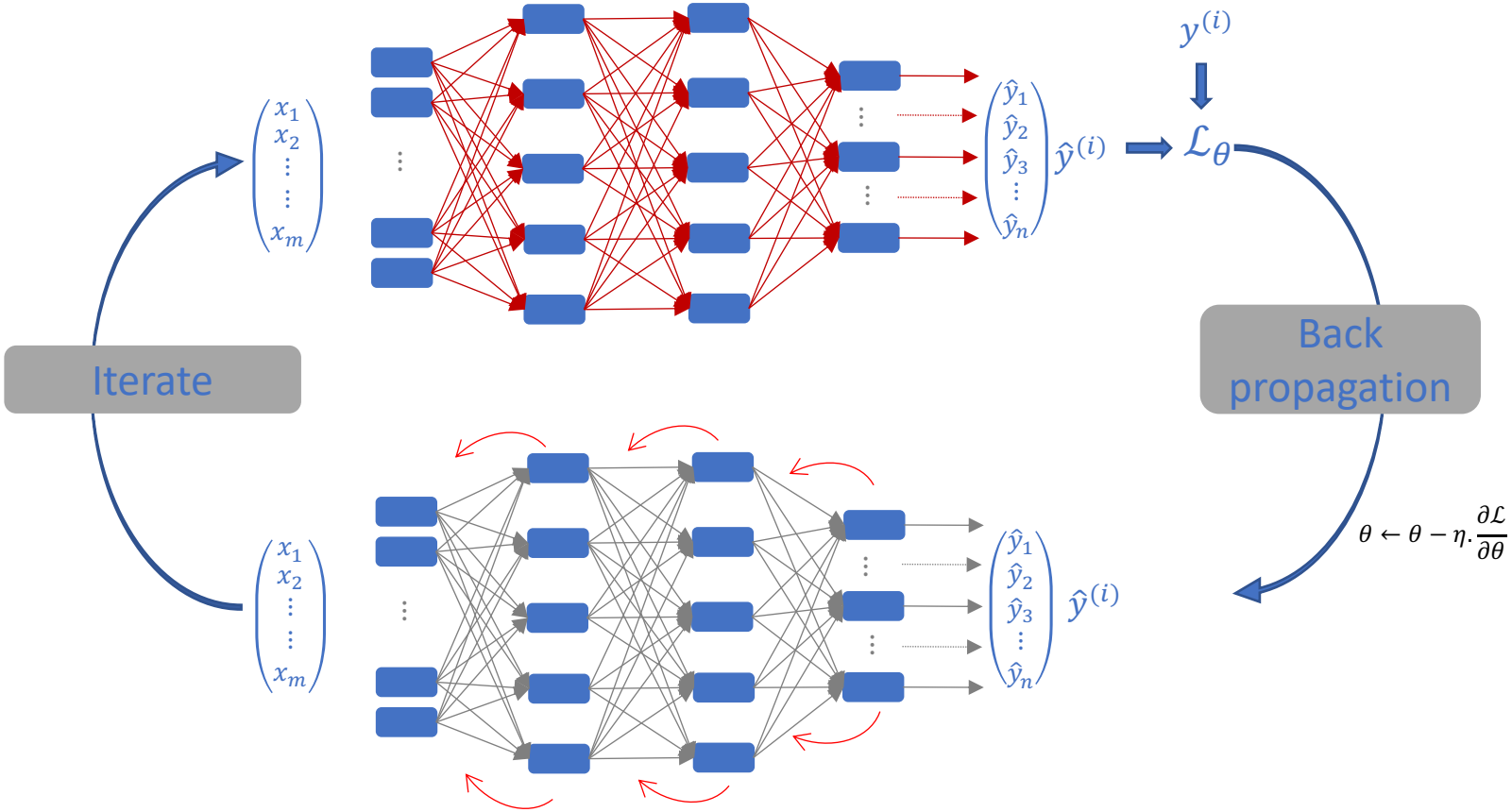


Deep network training

Loss function

$$\mathcal{L}_\theta = \frac{1}{2m}\sum_{i=1}^{m}\left\|y^{(i)} - \hat{y}^{(i)}\right\|^2$$

Feed Forward



Back propagation

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

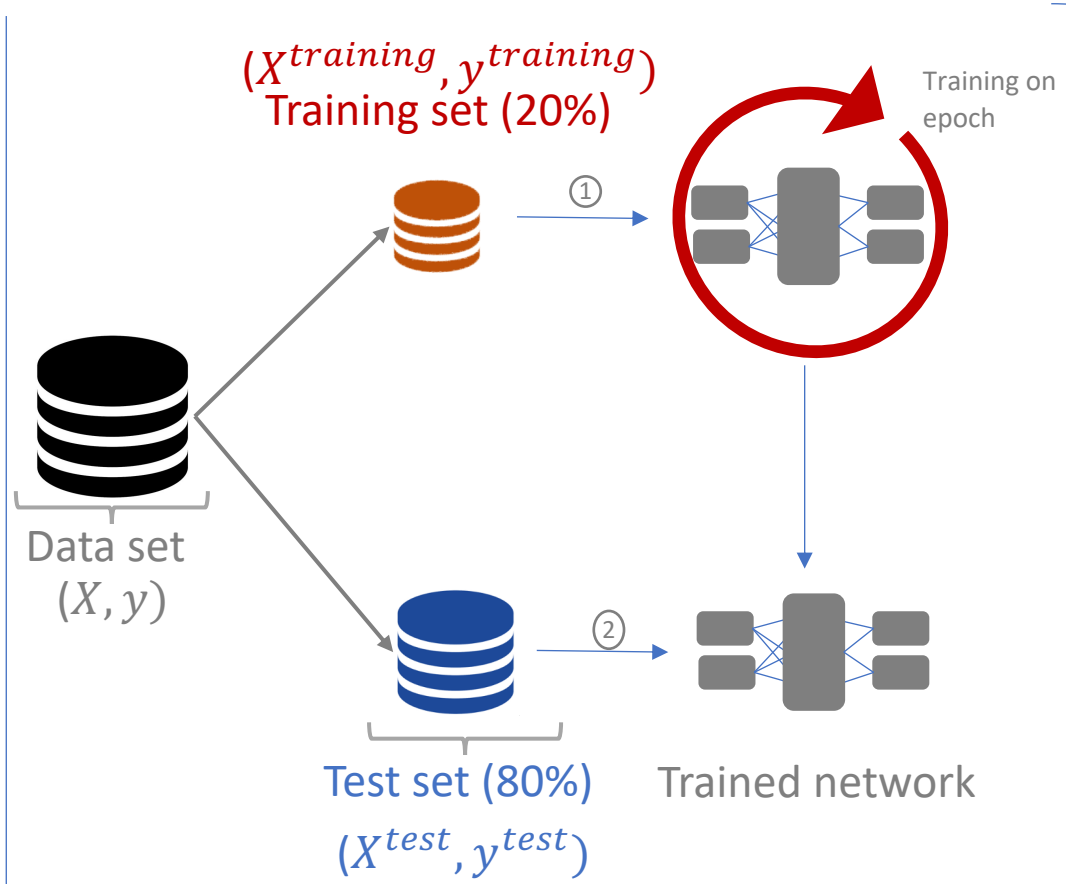
Algo. to adjust the weight of the connexions from the end to the beginning of the network.

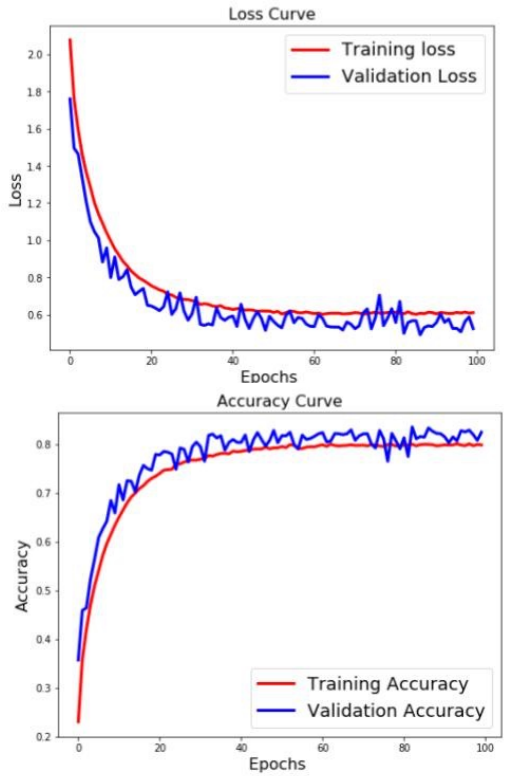
Feed forward + Back prop' = "Epoch"

Loss & accuracy

Training process

Never ever train on the test data !!!



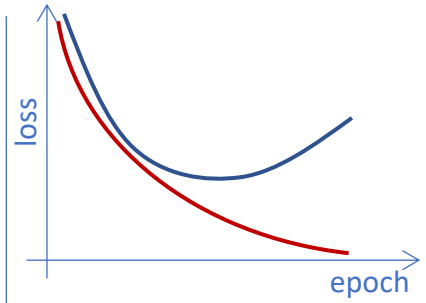


Accuracy is one metric for evaluating classification models.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Few training pathologies

Overfitting



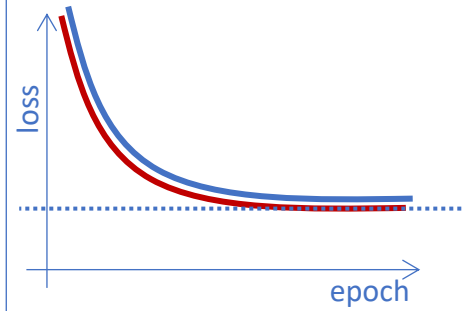
Symptoms:

- Validation loss decreases @ start then starts increase.
- Training loss continues to go down.

Try:

- More training data
- Add stronger regularization
- Data augmentation (flip, rotation, noise)
- Reduce complexity of the model

Underfitting



Symptoms:

- Training loss decreases at first but then stops.
- Training loss still high.

Try:

- Increase model capacity (more layers, increase layer size)
- Use more suitable network architecture
- Decrease regularization

Intuition about regularization 1/2

Logistic regression

$$\min_{w,b} J(w,b) \quad \text{where} \quad J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\widehat{y^{(i)}}, y^{(i)}\right) + \frac{\lambda}{2m}\|W\|_2^2$$

Add a penalty to the loss function of large weights. (Here, it is L^2 reg)
λ is a regularisation parameter (another hyperparameter)

Neural network

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\widehat{y^{(i)}}, y^{(i)}\right) + \frac{\lambda}{2m}\sum_{l=1}^{L}\|W^{[L]}\|^2$$

L is the number of layers

$$\text{with} \quad \|W^{[L]}\|^2 = \sum_{i=1}^{n^{[l+1]}}\sum_{j}^{n^{[l]}} (W_{ij}^{[l]})^2$$

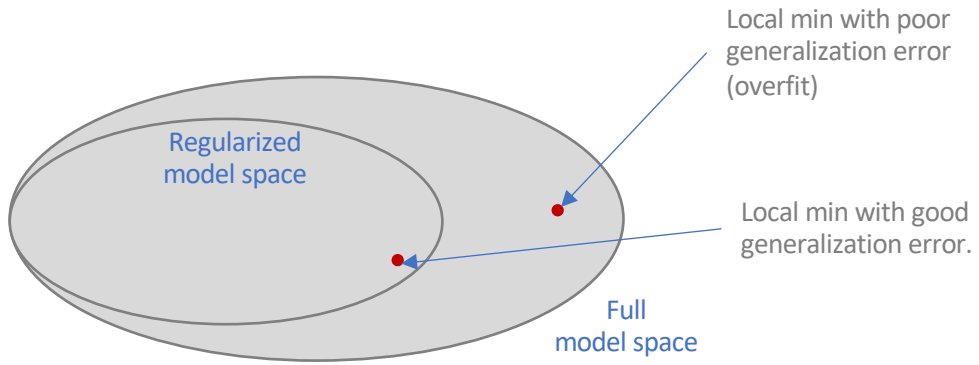
which is called the Frobenius norm

Back prop'

This « lambda » term is spread into the back propagation dans cause a Weight decay with help to lower overfitting

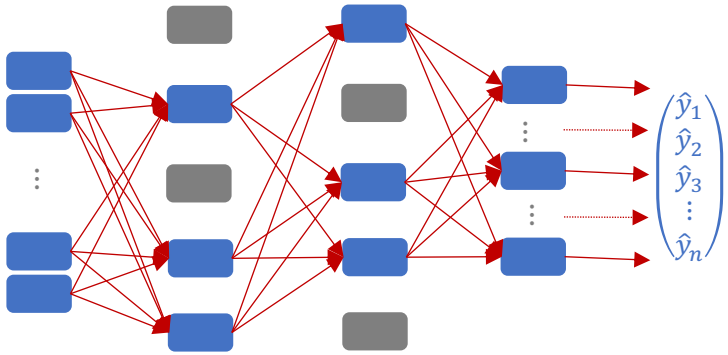
Intuition about regularization (2/2)

Early stopping



Early stopping limits the space of exploration but helps to prevent overfitting.

Drop out



Training: set weight of neurons to 0, randomly with probability p

- Prevents units from co-adapting to much.
- Force network to learn more robust features

Test: drop out is disabled

Set your hyperparameters

Check list

1. Learning rate
2. Number of hidden layers
3. Number of nodes in each hidden layers
4. Weight initialization methods
5. Which optimizer to choose ?
6. Activation functions per layers
7. Regularization
8. Number of epochs
9. …

Typical architecture for a Multi-layer Perceptron

Hyperparameters	Typical value
# of input neurons	One per feature (ex: 28x28=784 for MNIST)
# hidden layers (HL)	Depending on the problem: [1-5]
# of neurons per HL	Depending on the problem: [10-100]
# of output neurons	1 per predicted dimension
Activation function in HL	ReLU
Output activation function	Relu/softplus (for positive outputs) Logistic/tanh for (bounded outputs)
Loss function	MSE or MAE/Huber (in case of outliers)

(From Géron, 2019)

- It takes time to adjust them
- Trial and errors is the key.
- One can use *GridSearch* or *RandomSearch*

Notebook – Deep Neural Network

Goal

- First introduction with Keras framework
- Train a dense network with different hyperparameters
- Predict handwriting numbers after training with the famous MNIST Dataset
- Have a look on the statistical results

Advices

This is just a quick look so…

- Don’t hesitate to play with the notebook.
- Look the documentation which contain many explanations about the numerous parameters.
- Modify the structure of the network to improve the result
- If two heavy, use Google Collab for example

Let’s go…

Outlines

Generality	Machine learning, deep learning, IA… Supervised vs unsupervised Artificial network Gradient descent Artificial neural network
Deep Learning	Deep network Training concept Loss and accuracy Pathologies in training Intuition about regularization Set your hyperparameters
Specific networks	Convolution neural networks Recurrent neural networks

Focus on specific deep neural networks

Artificial intelligence (AI)

Machine learning (ML)

Neural Networks (NN)

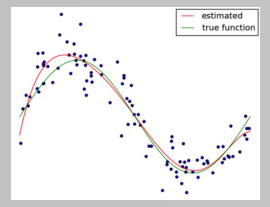
Deep learning (DL)

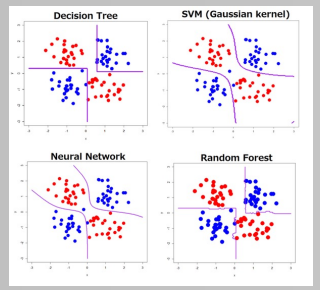
Reproduction of human intellectual activities

Programs that can sense, reason, act and adapt.

Learning of rules and patterns hidden in data

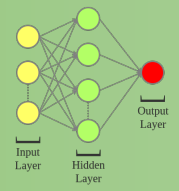
Algorithms whose performance improve as they are exposed to more data over time.





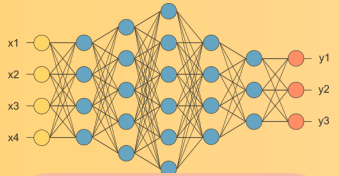
- Email spam filter
- Netflix reco
- Google page rank
- IBM Watson
- …

Brain activities emulation

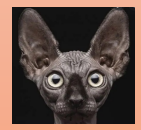


Multi-layer NN architecture

Subset of machine learning in which multilayered neural networks learn from vast amounts of data.



Convolutionnal Neural Netwoks



Computer vision, image processing

Recurrent Neural Netwoks

Natural language processing, time series

Convolutional neural networks (CNN)

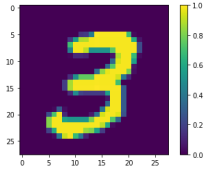
Why CNNs ?

Consider a network with one hidden layer

- n_i: input size
- n_h: size of the hidden layer
- n_o: output size

$$n_{param} = (n_i \times n_h + n_h \times n_o) + (n_h + n_o)$$

MNIST Handwritting dataset (28x28)



for example

FCNN with ~1000 neurons → >8.10^5 parameters to fit

HiRise Camera images (39876x29903, RGB)

NASA/JPL/UArizona

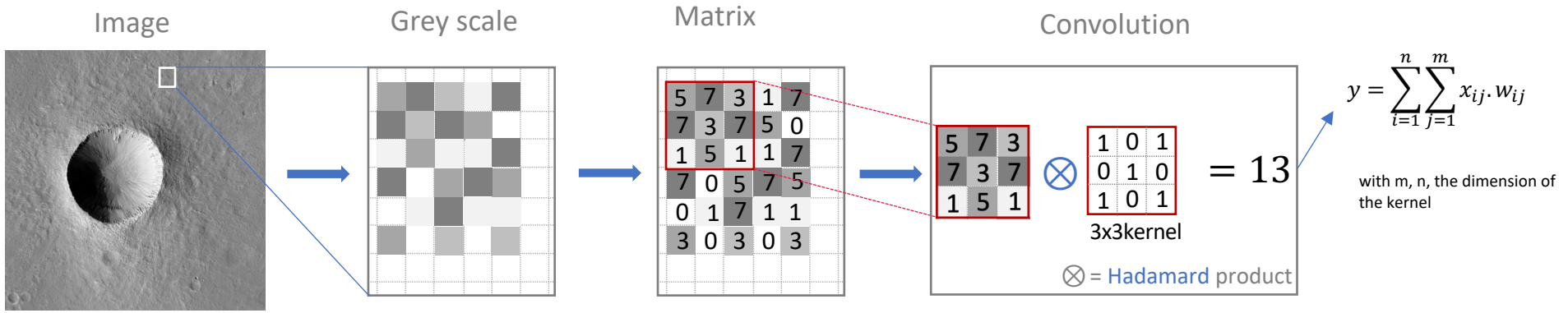
FCNN with ~1000 neurons → >> 10^{12} parameters to fit 😱

CNN are able to reduce the number of parameters.

CNN – Convolution intuitions

2D Convolution

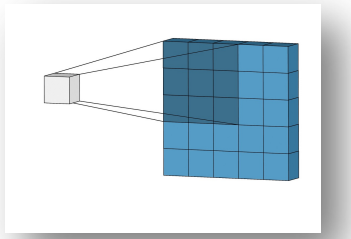
Convolutional layers = filters to extract the features of the images



$$y = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}.w_{ij}$$

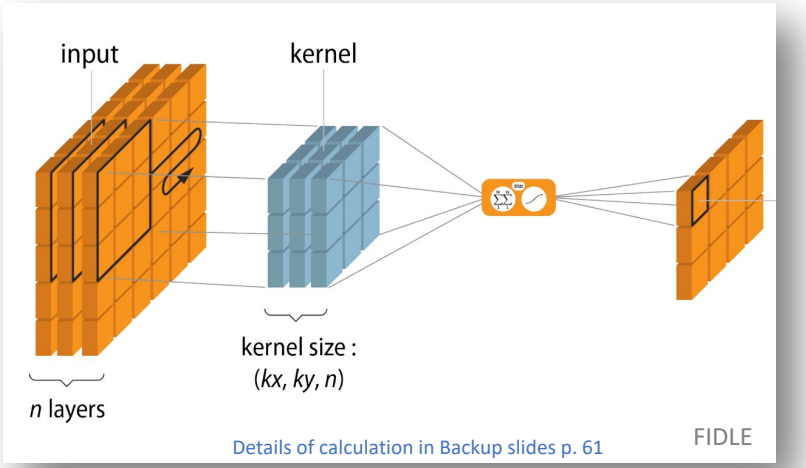
with m, n, the dimension of the kernel

Shift along the image



(stride and padding not mentioned)

Generalisation



Also usable for more than 2D dataset, off course.

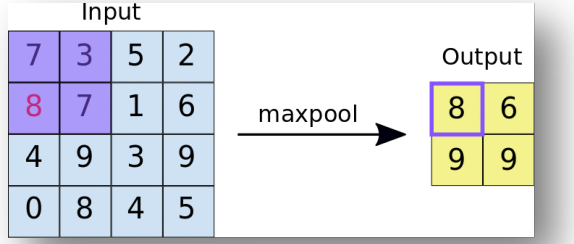
CNN – Pooling / Padding / Stride

Pooling

Pooling layers = under sampling to reduce processing time, memory use, number of parameters

2 main types of pooling:

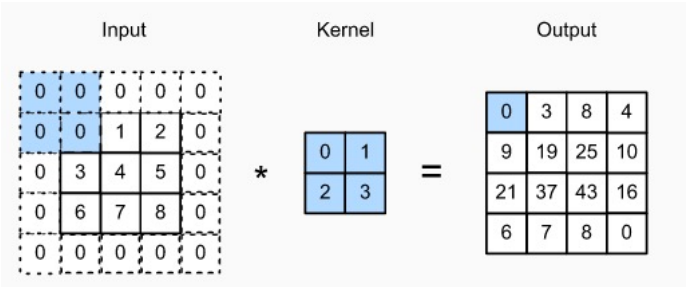
- Maximum pooling -> take the maximum value of the mask
- Average pooling -> take the average value of the mask



Example of maximum pooling

Padding

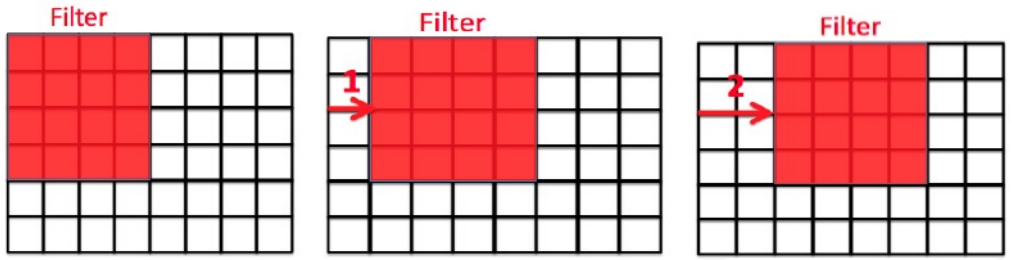
Used to avoid to get a smaller image after convolution



So if a $n*n$ matrix convolved with an $f*f$ matrix the with padding p then the size of the output image will be (n + 2p — f + 1) * (n + 2p — f + 1) where p =1 in this case.

Stride

Stride is the number of pixels shifts over the input matrix

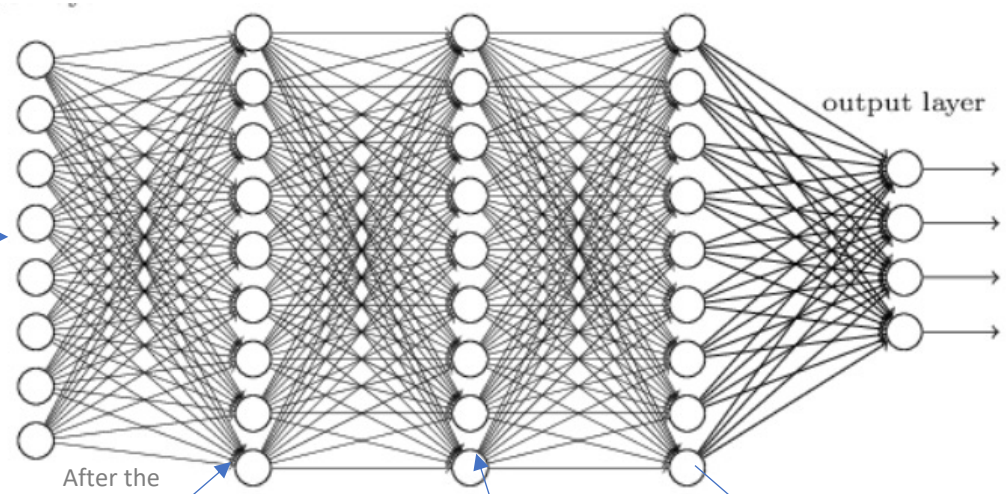


Stride = 0 Stride = 1 Stride = 2

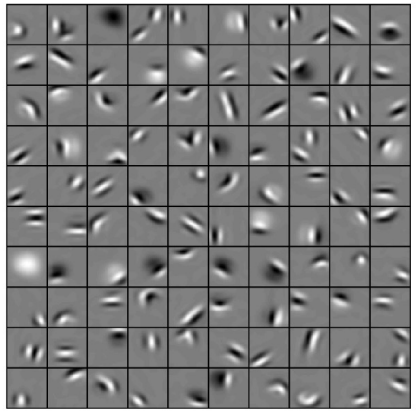
For padding p, filter size $f*f$ and input image size $n * n$ and stride 's' our output image dimension will be [$\{(n + 2p - f + 1) / s\} + 1] * [\{(n + 2p - f + 1) / s\} + 1]$.

CNN – Effects of convolution & pooling

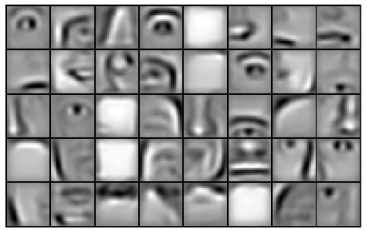
Examples with faces



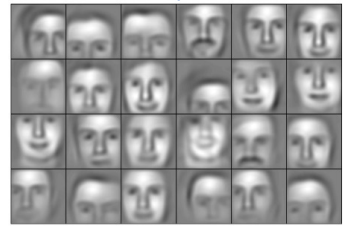
After the 1st block



After the 2nd block



After the 3rd block



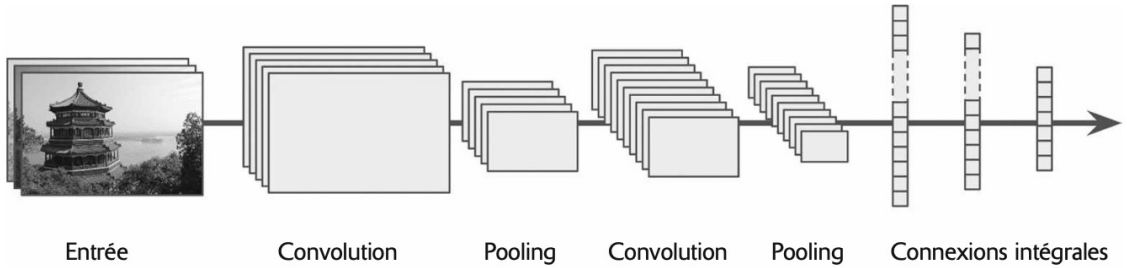
Examples of convolutions

Image from https://blog.engineering.publicissapient.fr/2018/07/03/echo-des-tos-n2-how-deep-is-your-love/

CNN – Few architectures

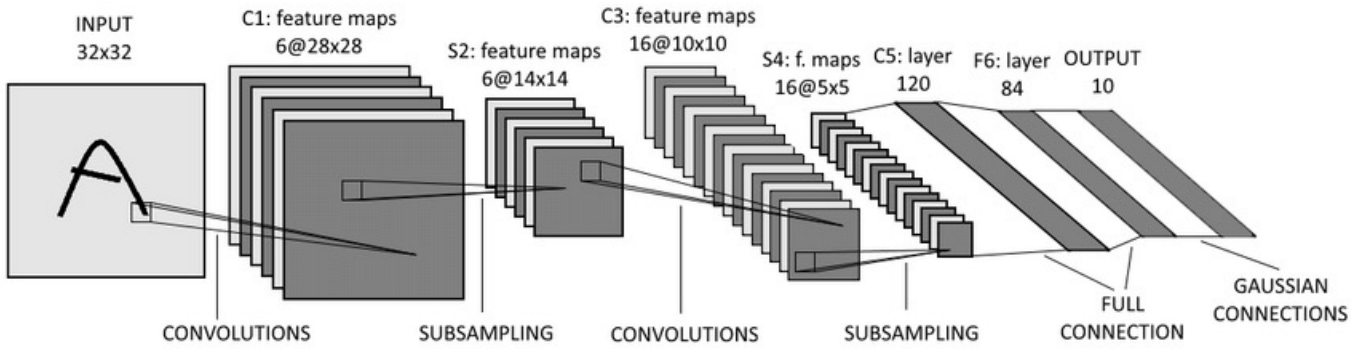
Typical architecture

A typical CNN is often a stack of convolution and pooling blocks.



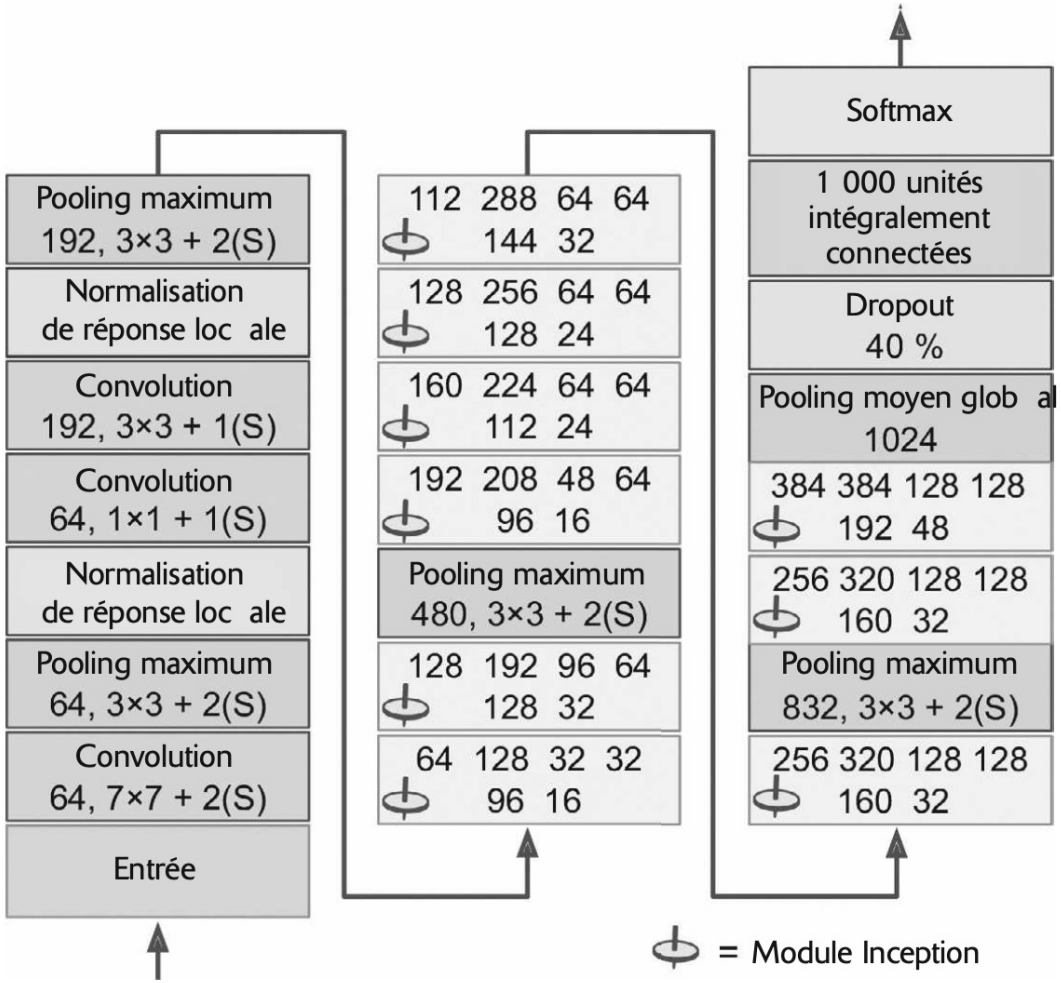
LeNet-5

By Yann LeCun (1998)

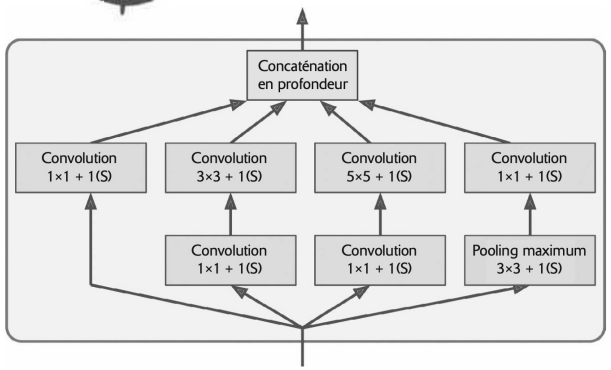


CNN – Few architectures

GoogleLeNet



Inception Module

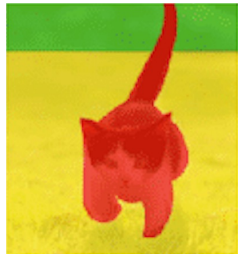


The Inception module kind on supercharged convolution layer, capable of outputting feature maps that identify complex patterns at different scales.

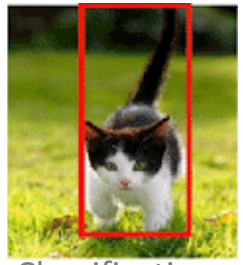
CNN – Usages

Computer vision

Object classification

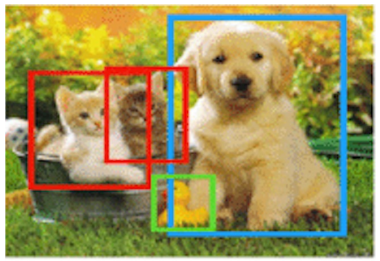


Semantic segmentation



Classification + Localisation

Single object



Object detection

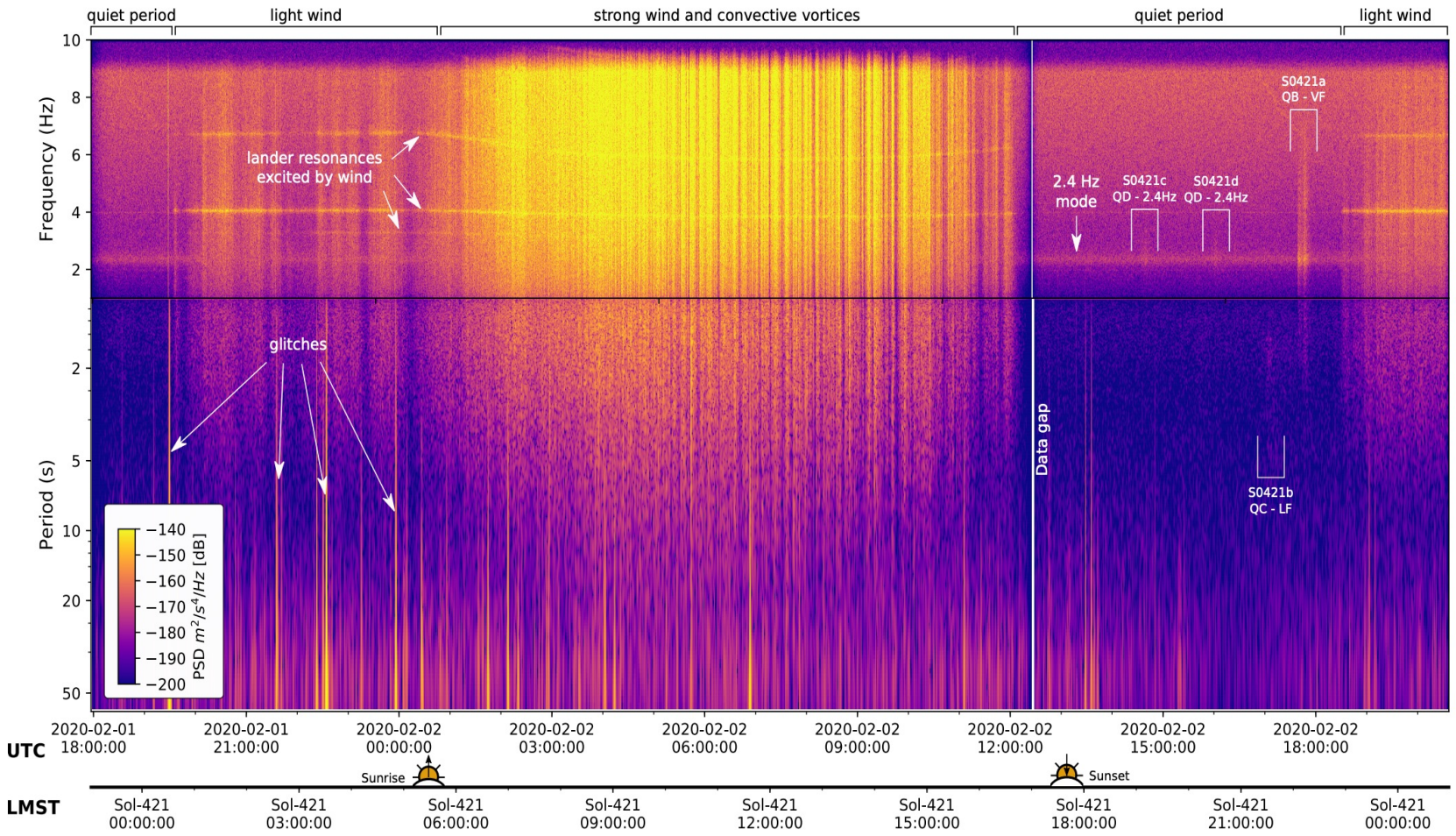
Instance segmentation

Multiple objects

CNN – Some other “species” of images

Spectrograms

a visual representation of the spectrum of frequencies of a signal as it varies with time.

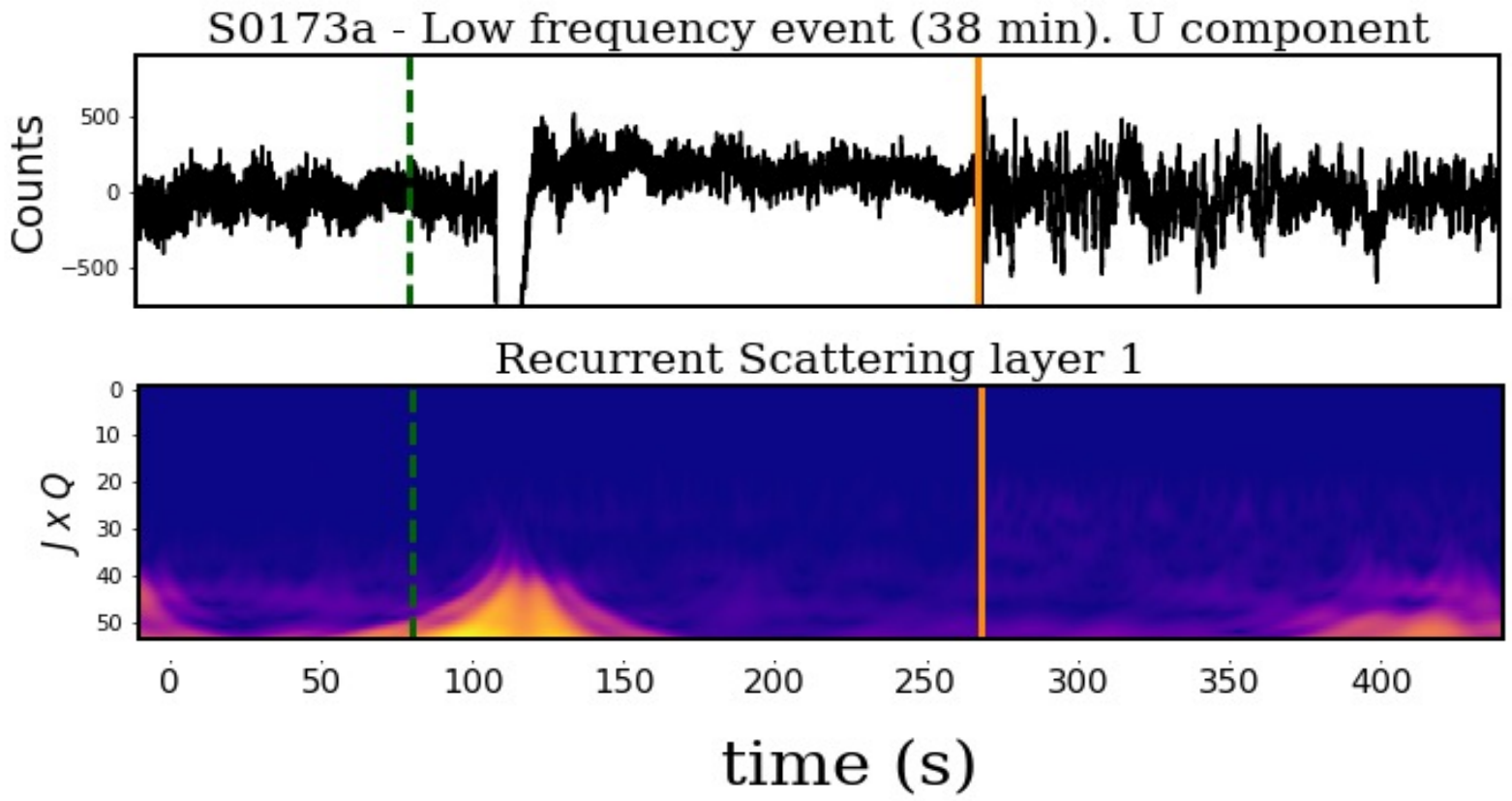


InSight VBB Data, sol 421

CNN – Some other “species” of images -2

Scalograms

Visual representation of a wavelet transform, having axes for time, scale, and coefficient value



From S. Barkaoui, PhD Defence, 2021, Dec 13th

Notebook – Deep Neural Network

Goal

- Train a CNN with different hyperparameters
- Predict handwriting numbers after training with the famous MNIST Dataset
- Have a look on the statistical results

Advices

- This is just a quick look so…

Don’t hesitate to play with the notebook.
Look the documentation which contain many explanations about the numerous parameters.
Modify the structure of the network to improve the result
If two heavy, use Google Collab for example

Let’s go…

Outlines

Generality	Machine learning, deep learning, IA… Supervised vs unsupervised Artificial network Gradient descent Artificial neural network
Deep Learning	Deep network Training concept Loss and accuracy Pathologies in training Intuition about regularization Set your hyperparameters
Specific networks	Convolution neural networks Recurrent neural networks

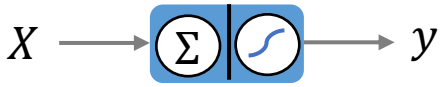
Recurrent neuron

Motivation

How to deal with the time dependency of the elements ?

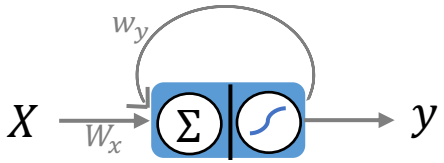
Time series

Classical neuron



$$\hat{y} = \sigma(W_x^T . X + b)$$

Recurrent neuron

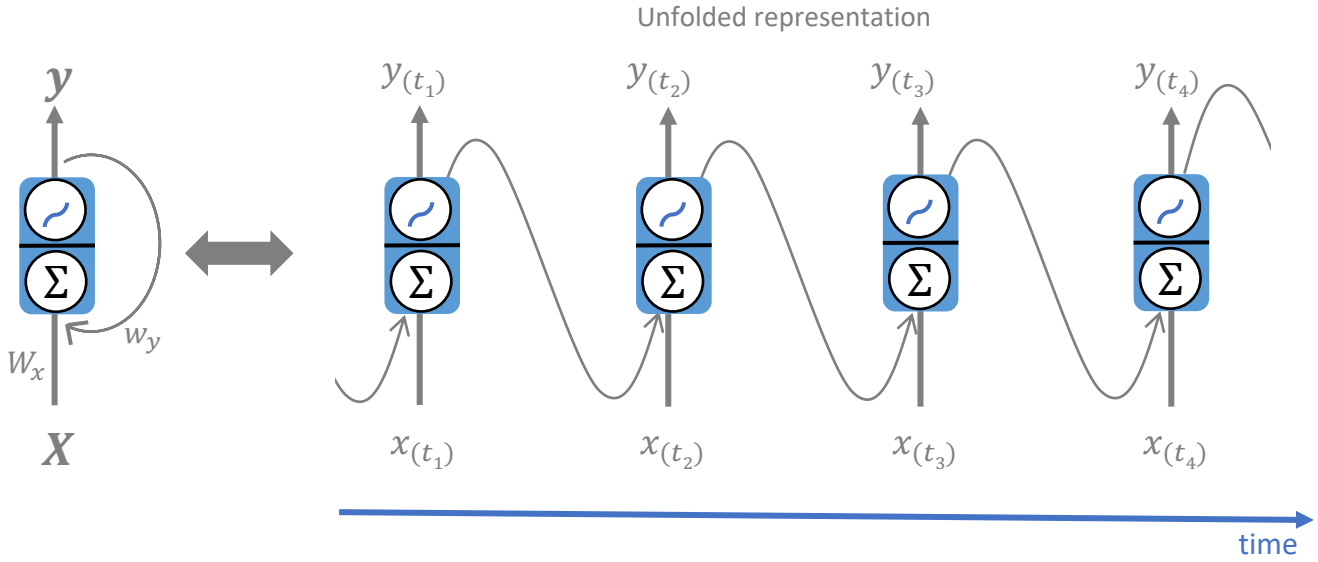


$$\hat{y} = \sigma(W_x^T . X_{(t)} + w_y . y_{(t-1)} + b)$$

This is the way to take into account the time dependency between information of the time series.

Recurrent neuron -2

Notation



Literally, the information of our time series $x(t_1)$ enter into the network and gives the output $y(t_1)$ which is injected as input in the next neuron with the input $x(t_2)$…

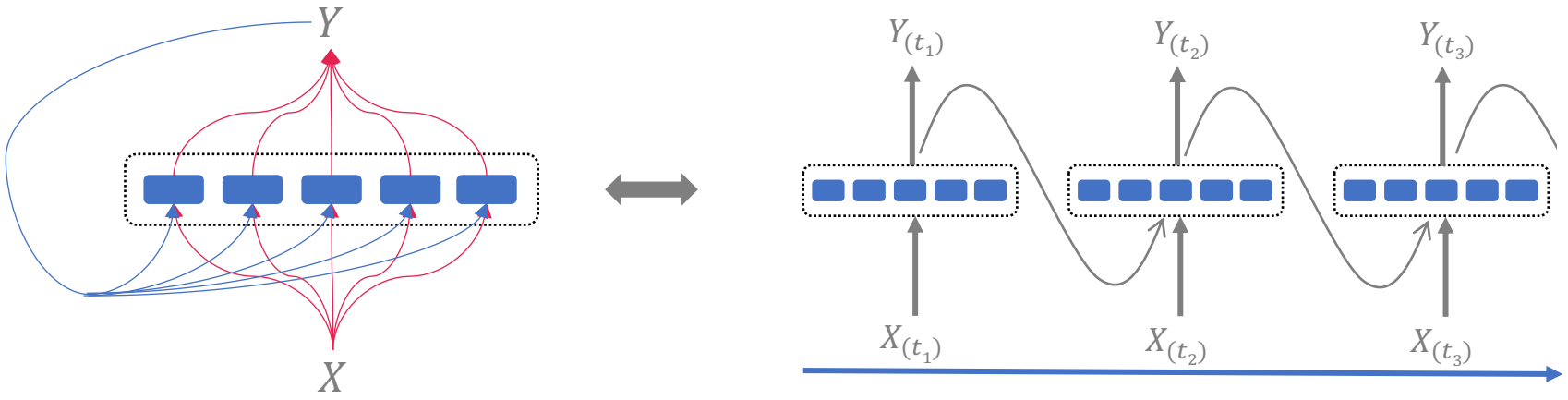
Mathematical expression

$$\hat{y} = \sigma(W_x^T . X_{(t)} + w_y . y_{(t-1)} + b)$$

with $\begin{cases} X \text{ is a vector} \\ W_x \text{ is a vector} \\ y \text{ is a scalar} \\ w_y \text{ is a scalar} \end{cases}$

Recurrent layer - explanation

Cell of recurrent neurons



Advantage

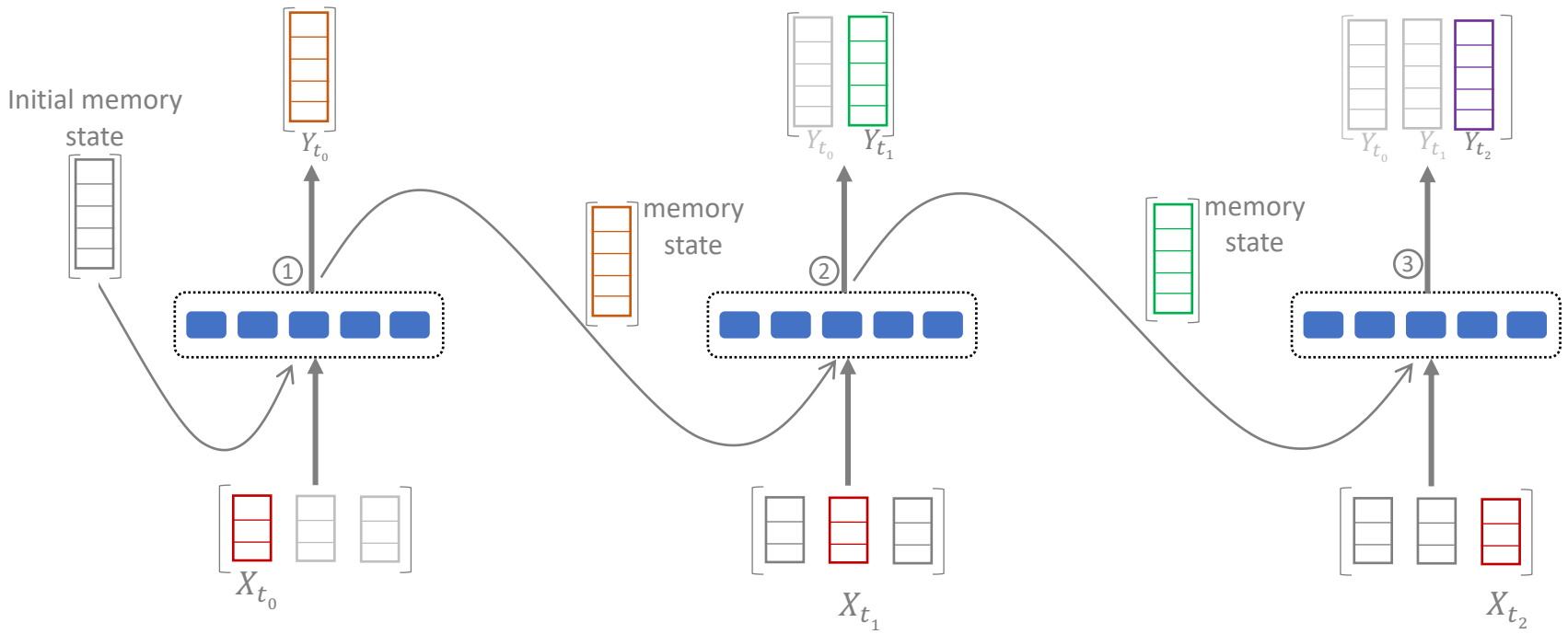
It allows recurrent neurons to work in group of units
Output Y is now a vector

Mathematical expression

$$Y_{(t)} = \phi(W_x^T . X_{(t)} + W_y^T . Y_{(t-1)} + b)$$

Recurrent layer - illustration

Example



How does it work ?

- *At first iteration, memory state is a null vector.*
- *Memory state and X_{t0} is injected into the recurrent cell and gives Y_{to}*
- *Memory state becomes Y_{t0} and the network starts the second step…*
- *Output has the same size as the number of units*

Recurrent layer - limitations

Unfortunately

- RNN are converging very slowly
- RNN have very short… euh, ah yes very short memory !
- RNN are often facing vanishing or exploding gradient

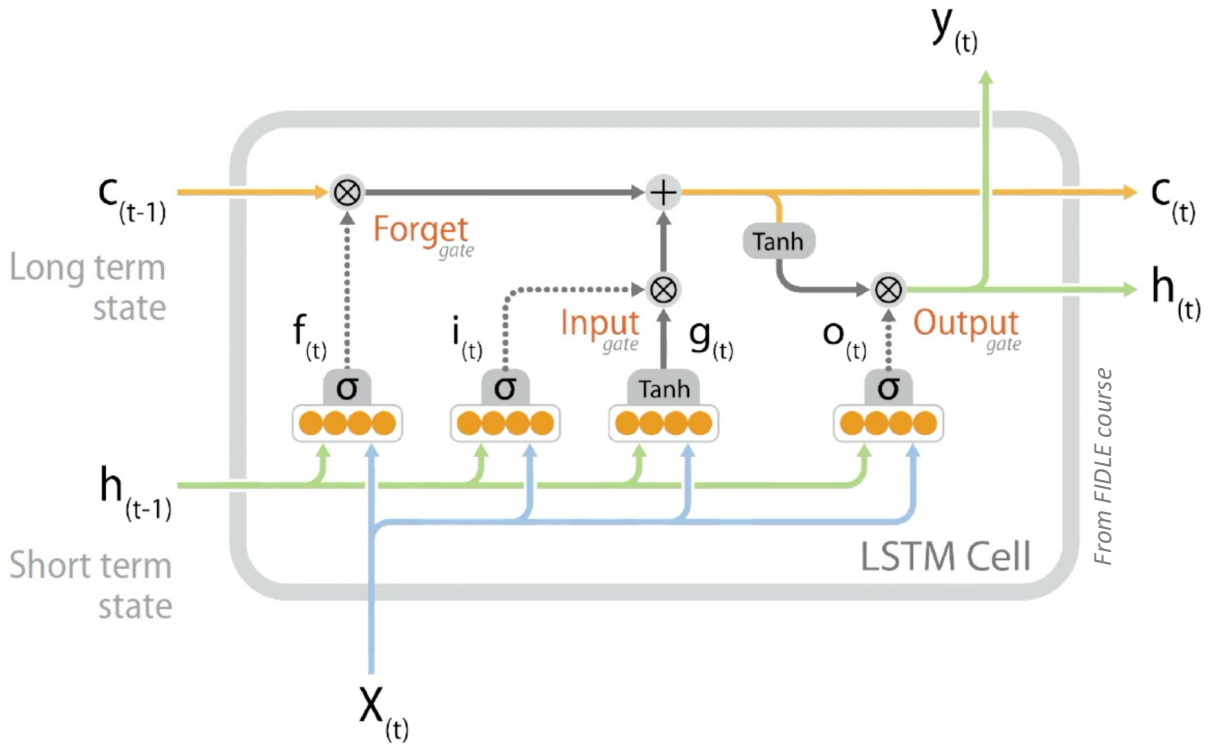
Solution ?

- Be able to keep in memory short and long term informations
- Such recurrent neural networks exist:
    - Long Short Term Memory (LSTM)
    - Gated Recurrent Unit (GRU)

Long Short Term Memory (LSTM) - Principle

Working principle

Hochreiter S., Schmidhuber. J., **Long short-term memory** *Neural computation* 9.8 (1997): 1735-1780.



$$f_{(t)} = \sigma(W_{xf}^T X_{(t)} + W_{hf}^T h_{(t-1)} + b_f)$$
$$i_{(t)} = \sigma(W_{xi}^T X_{(t)} + W_{hi}^T h_{(t-1)} + b_i)$$
$$g_{(t)} = \tanh(W_{xg}^T X_{(t)} + W_{hg}^T h_{(t-1)} + b_g)$$
$$o_{(t)} = \sigma(W_{xo}^T X_{(t)} + W_{ho}^T h_{(t-1)} + b_o)$$
$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}$$
$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})$$

How does it work ?

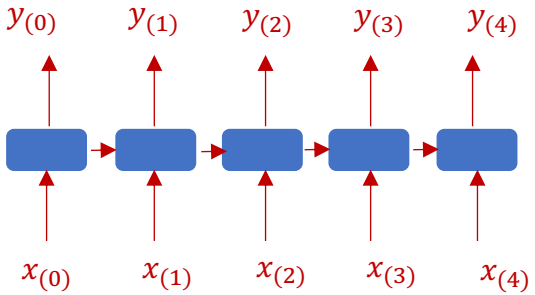
In blue, as before, the input data
In green, as before, the short term memory
In orange, the long term memory

The output Y_t take into account the input data X_t, the long term memory C_{t-1} and the short term memory h_{t-1}

Long Short Term Memory (LSTM) - Usages

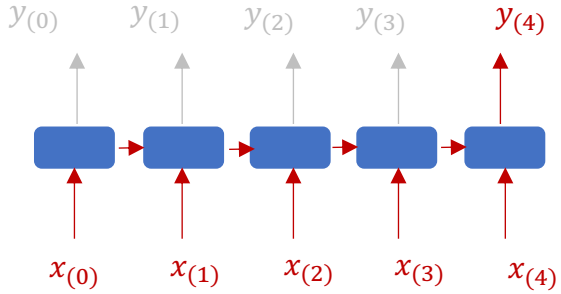
Serie-to-serie

Eg: Time series prediction
(regression)



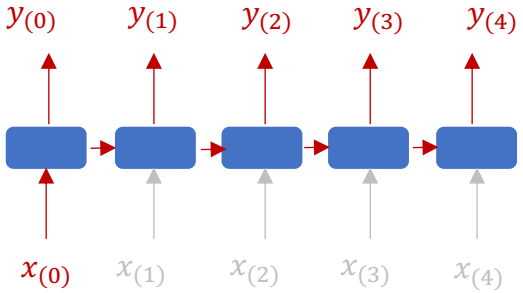
Serie-to-vector

Eg: Sentiment analysis
(classification)



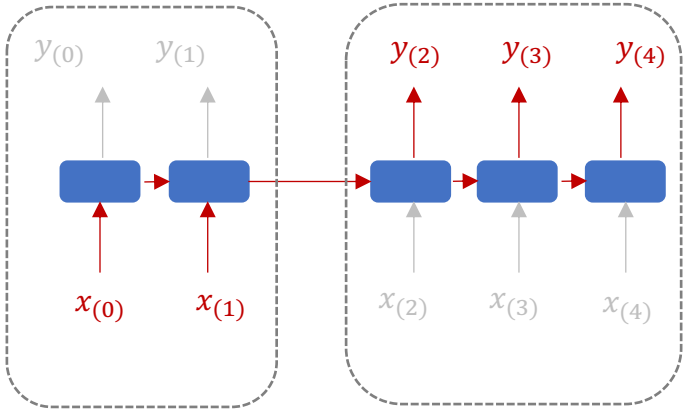
Vector-to-serie

Eg: Image annotation (cf FB)
(labelling)



Encoder-Decoder

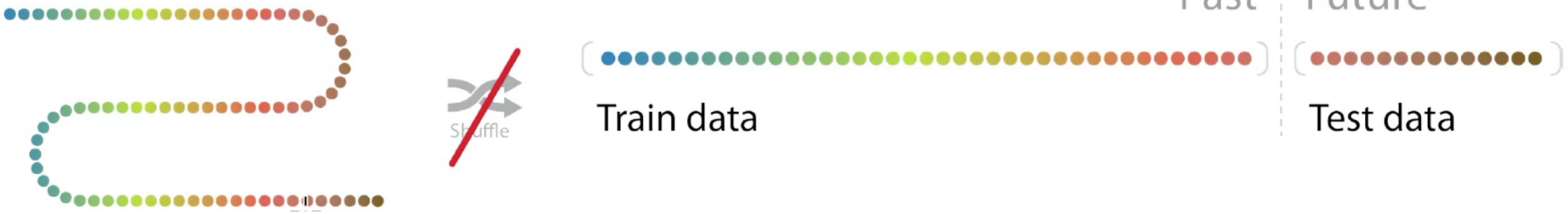
Language translator
(cf: deepl, google translate…)



Long Short Term Memory (LSTM) - Dataset

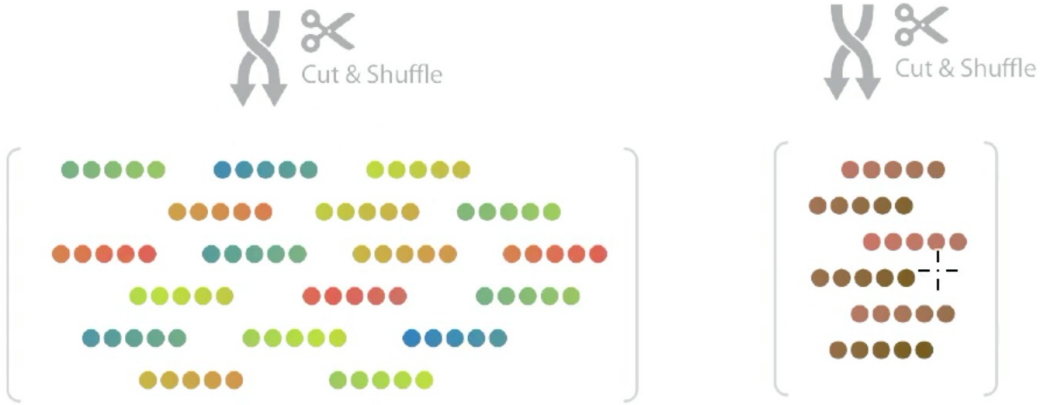
Ordered dataset

Time series, texts are some ordered dataset, impossible to shuffle the unit



Data preparation

In practice, we will use a data generator from Keras to help us to prepare the data.



From FIDLE course

Quick conclusion

Deep learning

- Fashion tool used in almost every fields of science today;
- Adapted for large datasets either time series or images;
- A lot of framework “easy” to learn (Keras, PyTorch, Caffé…).

Advices

But

- Understand your data
- Garbagge in – Garbagge out -> Bad data -> bad/wrong results
- Think about the bias
- There is no magic behind deep learning, just algorithms
- What do you expect from a DNN ?
- Don’t forget the computer cost and the GPU needs

Have fun !

Earth Data Science

course

Crash course of

deep learning

2021/2022 – Grégory Sainton

THANK YOU FOR YOUR ATTENTION

Back up slides

The 3 laws of robotics in case of strong AI ?

First law

A robot may not injure a human being or, through inaction, allow a human being to come to harm.

Un robot ne peut porter atteinte à un être humain ni, restant passif, laisser cet être humain exposé au danger.

Second law

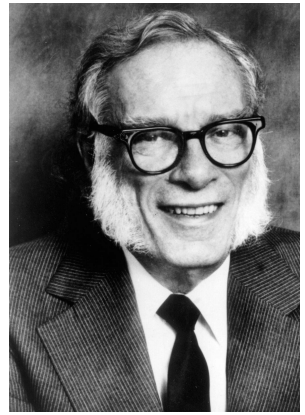
A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres entrent en contradiction avec la première loi.

Third law

A robot must protect its own existence as long as such protection does not conflict with the First or Second Law

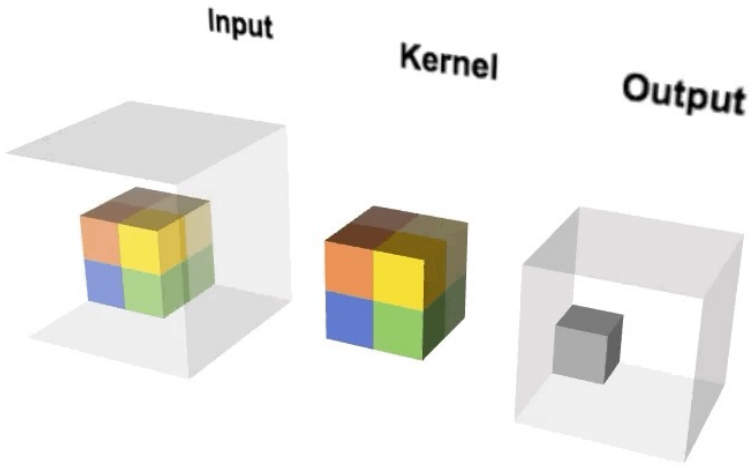
Un robot doit protéger son existence dans la mesure où cette protection n'entre pas en contradiction avec la première ou la deuxième loi.

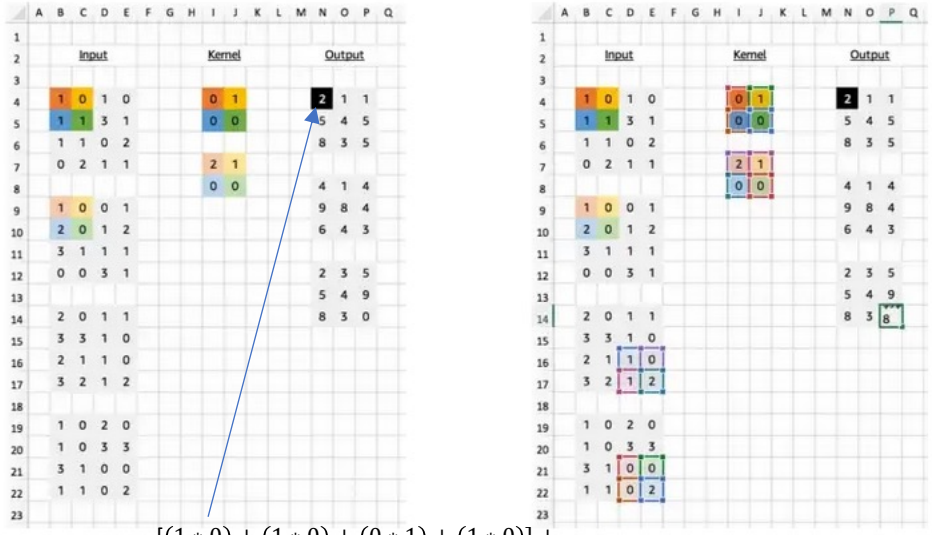


Isaac Azimov

Details about Conv 3D

How to estimate the output dimensions





$$[(1*0)+(1*0)+(0*1)+(1*0)]+$$
$$[(1*2)+(2*0)*(0*1)+(0*0)]=2$$

$$Out_i = \frac{(W_i - F_i + 2P)}{S} + 1$$

With :

- W_i =Input volume size
- F_i = filter size
- S = stride
- P = Padding

For exemple, with a 4x4x4 set of data with a 2x2x2 filters with 0 padding and stride equal to 1

$$Out_x = \frac{4-2+0}{1} + 1 = 3$$

$$Out_y = \frac{4-2+0}{1} + 1 = 3$$

$$Out_z = \frac{4-2+0}{1} + 1 = 3$$

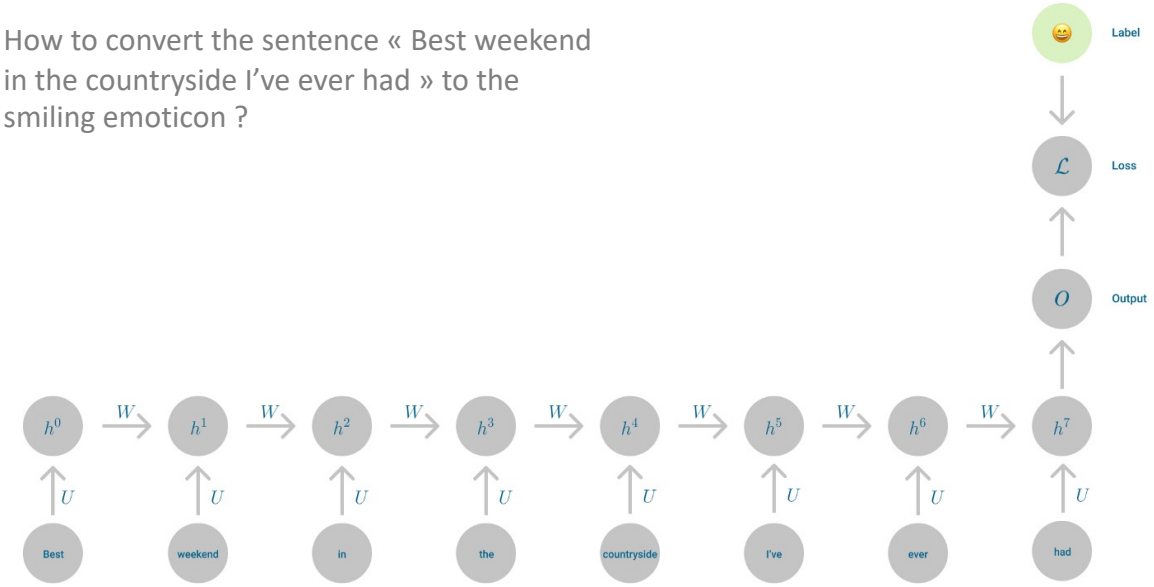
Output will be (3x3x3)

Recurrent network - example

https://towardsdatascience.com/recurrent-neural-networks-explained-with-a-real-life-example-and-python-code-e8403a45f5de

Serie to vector example

How to convert the sentence « Best weekend in the countryside I've ever had » to the smiling emoticon ?



- Many sentences labeled in the training set to be able to train properly the network.
- Don't expect good result with plain RNN du the gradient vanishing or exploding
- LSTM is much more efficient.