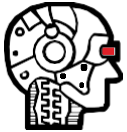
Deep Learning

Seq2- Hight Dimensionnal Data
CNN

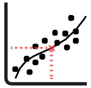
Previously

Few little things and concepts to keep in mind

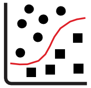
1



History, Fundamental Concepts



Basic Regression DNN



Basic Classification DNN

- Regression vs. Classification
- Data normalization
- Training and validation
- Epochs and Batchs
- Activation functions
- Loss function
- Optimization and gradient descent
- Metrics
- Softmax and Argmax function
- Numpy shape

Roadmap

2

Hight Dimensionnal Data
CNN

2.1 What is a Convolutional Neuron Network (CNN)?

- Understanding what a CNN is
- Identify use cases

2.2 Example 1 : MNIST

- Implementation of a simple case

2.3 Example 2 : GTSRB

- The devil is also hiding in the data
- How to work with « large » dataset
- Monitoring the training phase and managing our models
- Improve our results with data augmentation
- Datasets and models: how to automat testing
- How to go from notebook to HPC

https://www.youtube.com/watch?v=581X9wsnWJs

Roadmap

2

Hight Dimensionnal Data

CNN

2.1 What is a Convolutional Neuron Network (CNN)?

- → Understanding what a CNN is
- → Identify use cases

2.2 Example 1 : MNIST

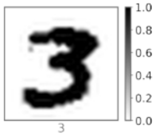
- → Implementation of a simple case

2.3 Example 2 : GTSRB

- → The devil is also hiding in the data
- → How to work with « large » dataset
- → Monitoring the training phase and managing our models
- → Improve our results with data augmentation
- → Datasets and models: how to automate testing
- → How to go from notebook to HPC

Convolutional Neural Networks (CNN)

For a fully connected layer of (only) 1000 neurons, we would need to

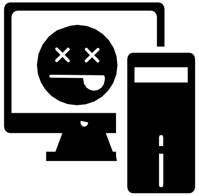


0.0008 M pixels
28x28, 8 bits

784.000 params

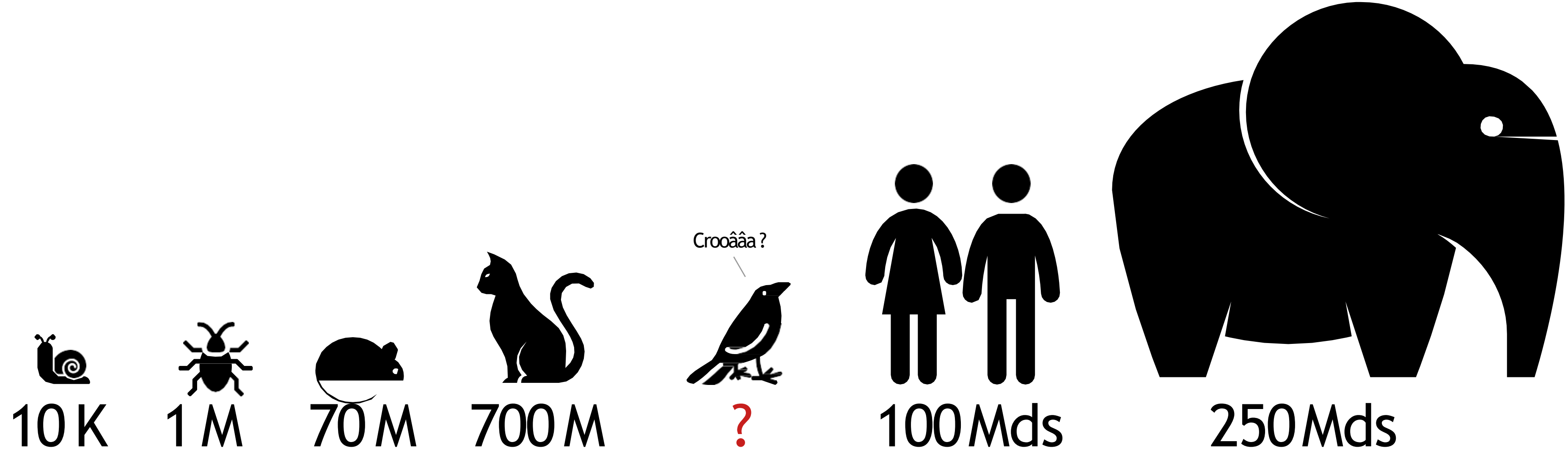
24 M pixels
(r,v,b) 3x8 bits

72.10E9 params...



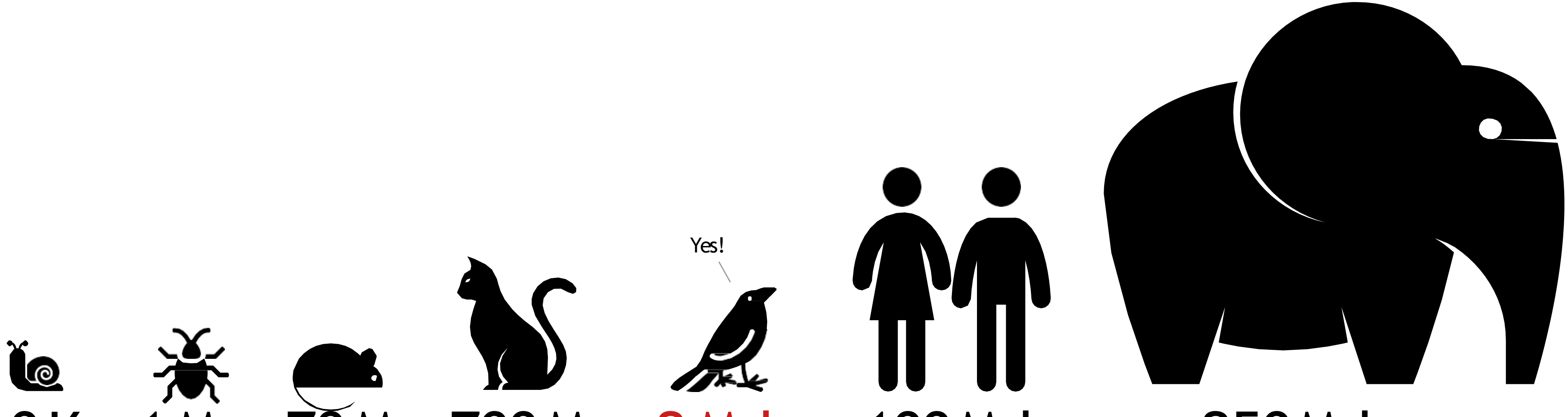
Short Digression !

One neuron is good… but more than one is better !



Short Digression !

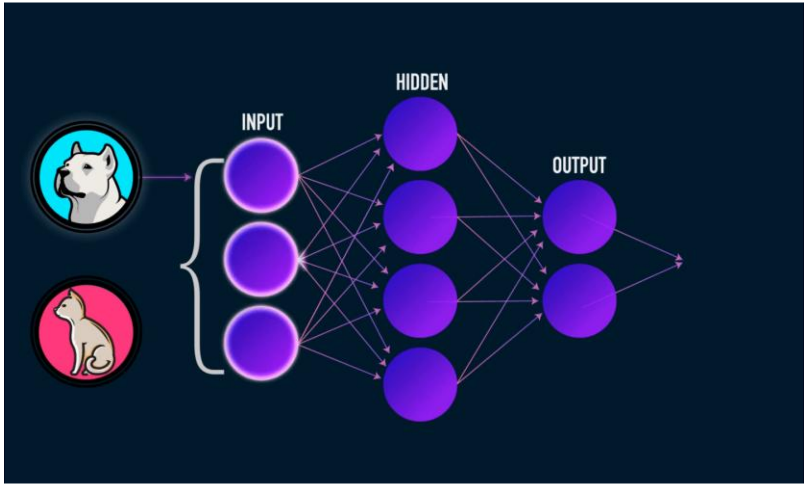
One neuron is good… but more than one is better !



10 K | 1 M | 70 M | 700 M | 2 Mds | 100 Mds | 250 Mds

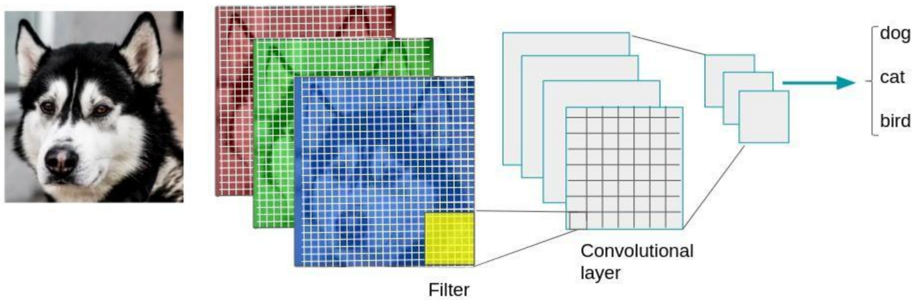
CNN for what?

- These networks are used mainly for cases involving image or video
- Main applications using CNN are :
  - face recognition
  - image classification
- Bai Du, Snapchat use CNN for face recognition/identification

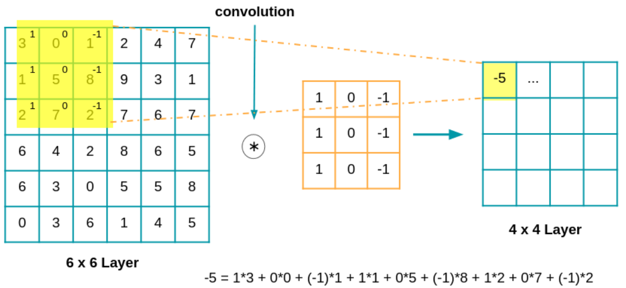


What is convolution for?

- The name “convolution” comes from a mathematical operation: convolution between functions
- The convolution applies a filter to the input image
- The filter parameters are learned through the learning
- A learnt filter will be able of detecting features in an image; for example angles, and use them to classify at best the image



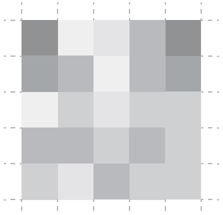
The image is decomposed into 3 channels (R, G, B)



Example of a convolution with a **6 x 6** size matrix

Principle of image convolutions

By Jan Kroon, from Pexels.com



5	2	1	3	5
4	3	2	3	4
0	2	1	2	2
3	3	2	3	2
2	1	3	2	2

Image piece, Kernel 3x3:

$$\underbrace{\begin{bmatrix} 5 & 2 & 1 \\ 4 & 3 & 2 \\ 0 & 2 & 1 \end{bmatrix}}_{X} \otimes \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{\omega} = \underbrace{\boxed{10}}_{y}$$

$$\begin{aligned} y = {} & 5\text{x}1 + 2\text{x}0 + 1\text{x}1 \\ & + 4\text{x}0 + 3\text{x}1 + 2\text{x}0 \\ & + 0\text{x}1 + 2\text{x}0 + 1\text{x}1 = 10 \end{aligned}$$

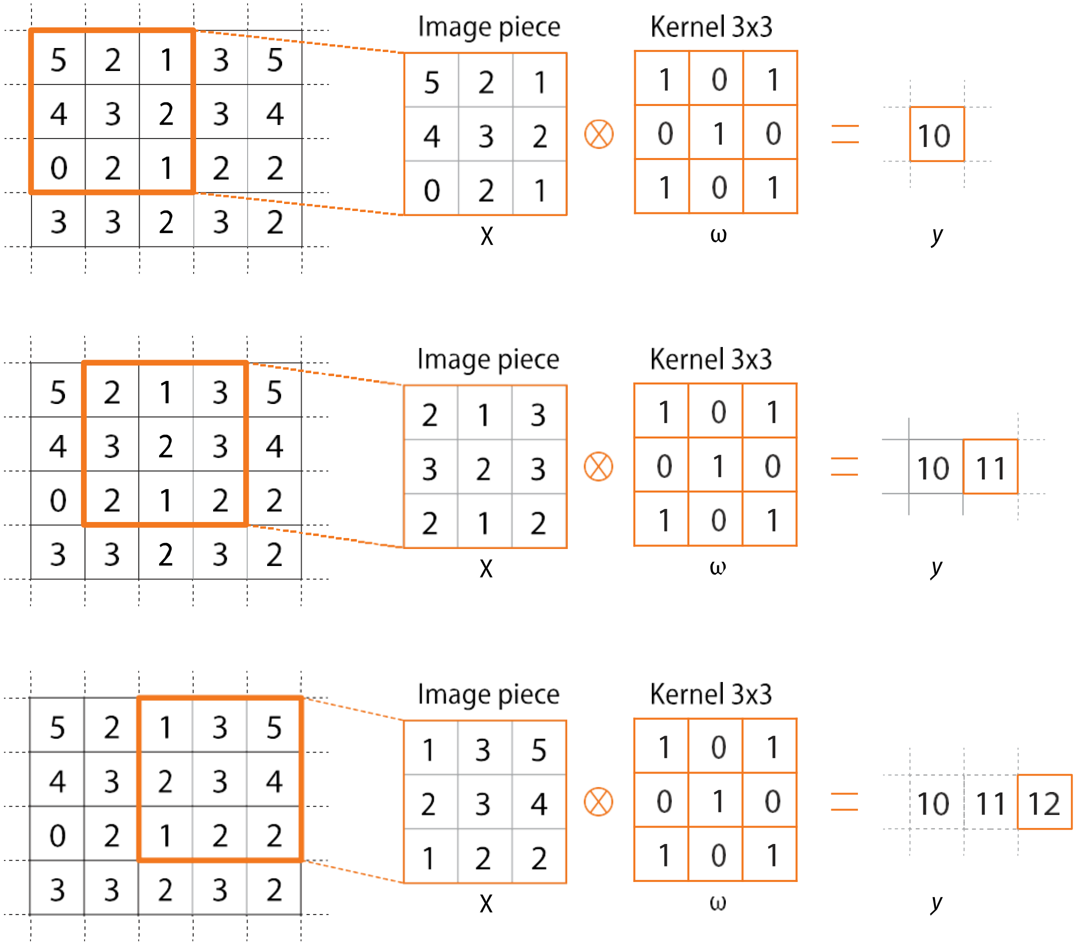
$$y = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} \cdot \omega_{i,j} \quad \text{with} \begin{cases} n & \text{kernel width} \\ m & \text{kernel height} \end{cases}$$

2D convolution

$\otimes$ is Hadamard product

Principle of image convolutions

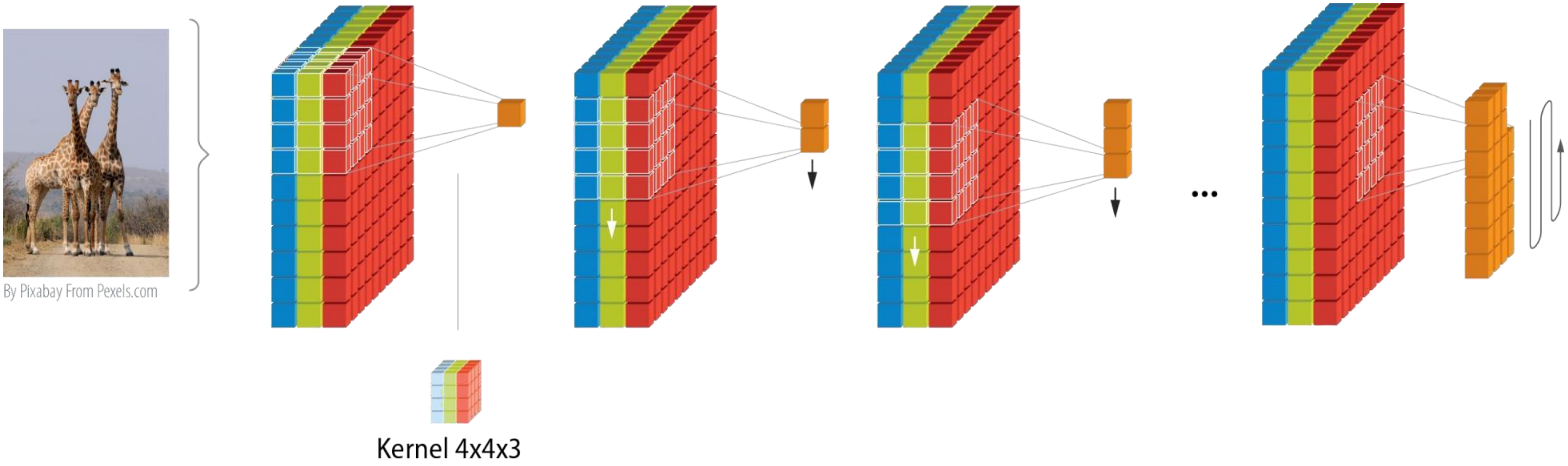
By Jan Kroon, from Pexels.com



> We can perform convolutions in 1,2,3 ...or n-dimensional spaces !

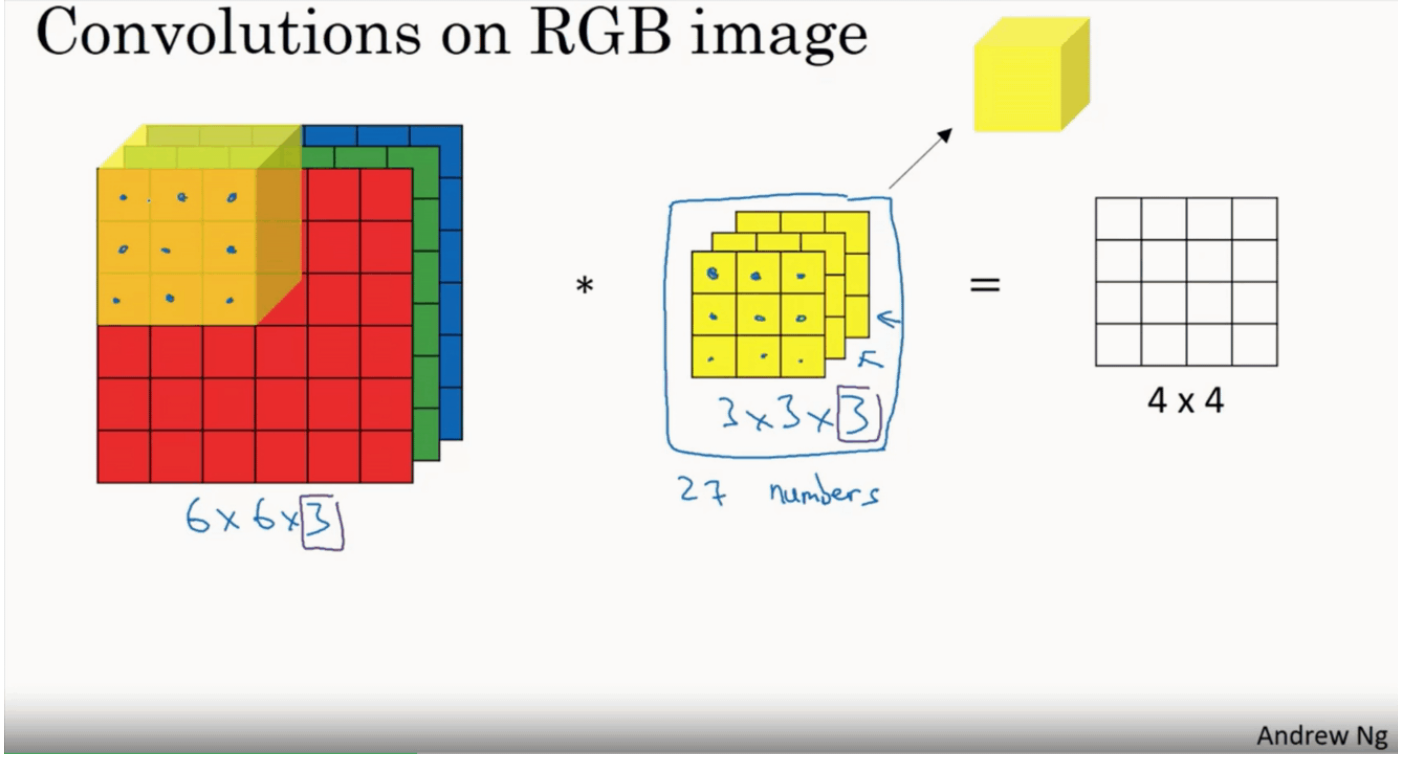
$\otimes$ is Hadamard product

Principle of image convolutions



3D convolution

Principle of image convolutions



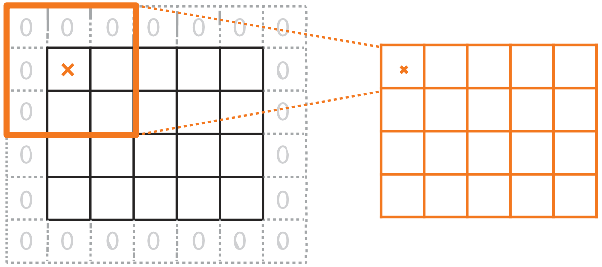
3D convolution

[Andrew Ng](#)

Principle of image convolutions

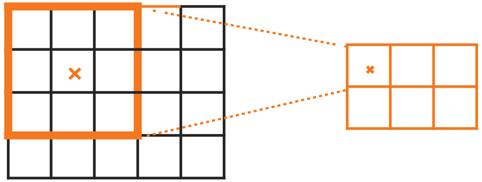
Some parameters of a convolution :

padding



`padding = 'same'`

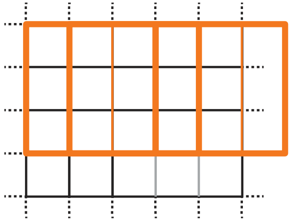
The use of a padding allows to keep the size of the image.



`padding = 'valid'`

Means no padding

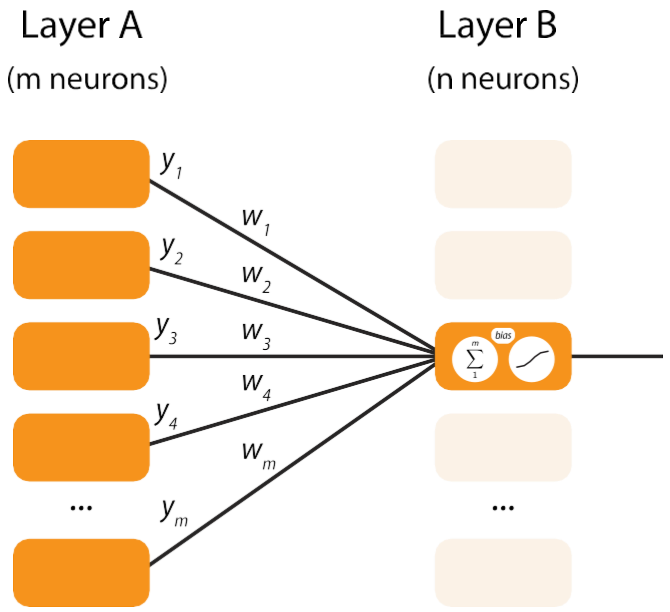
strides



`strides = (dx,dy)`

Strides of the convolution along the height and width

Reminder: Principle of a fully connected layer

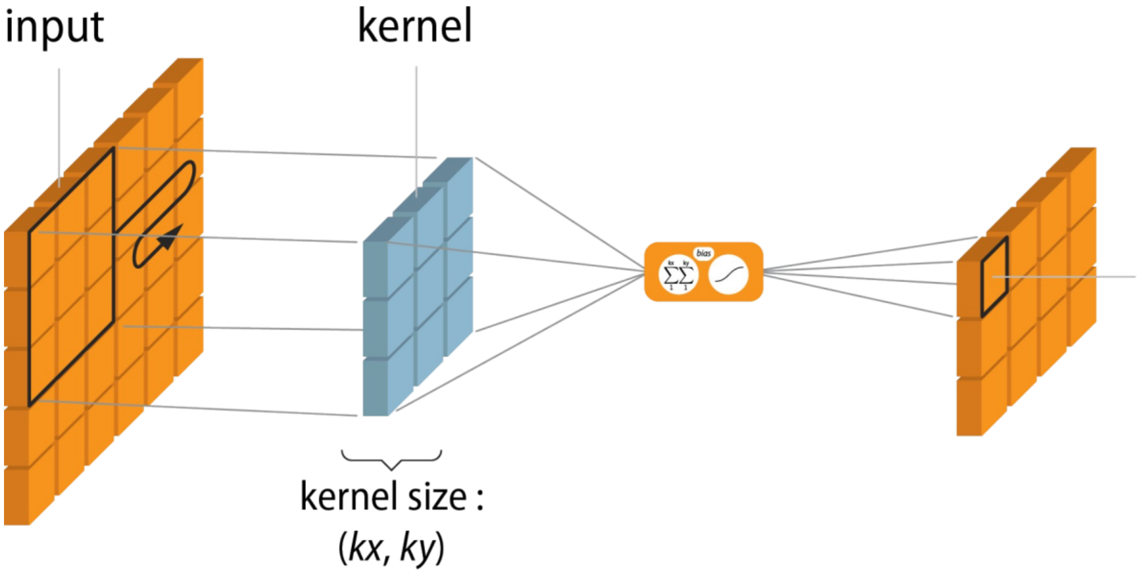


$$y = \sigma(\sum_{1}^{m} y_i \cdot w_i + b)$$

Where :
- σ : is an activation function
- b : is a bias

Number of parameters for a convolutional layer : $n\ (m + 1)$

Convolutional layers

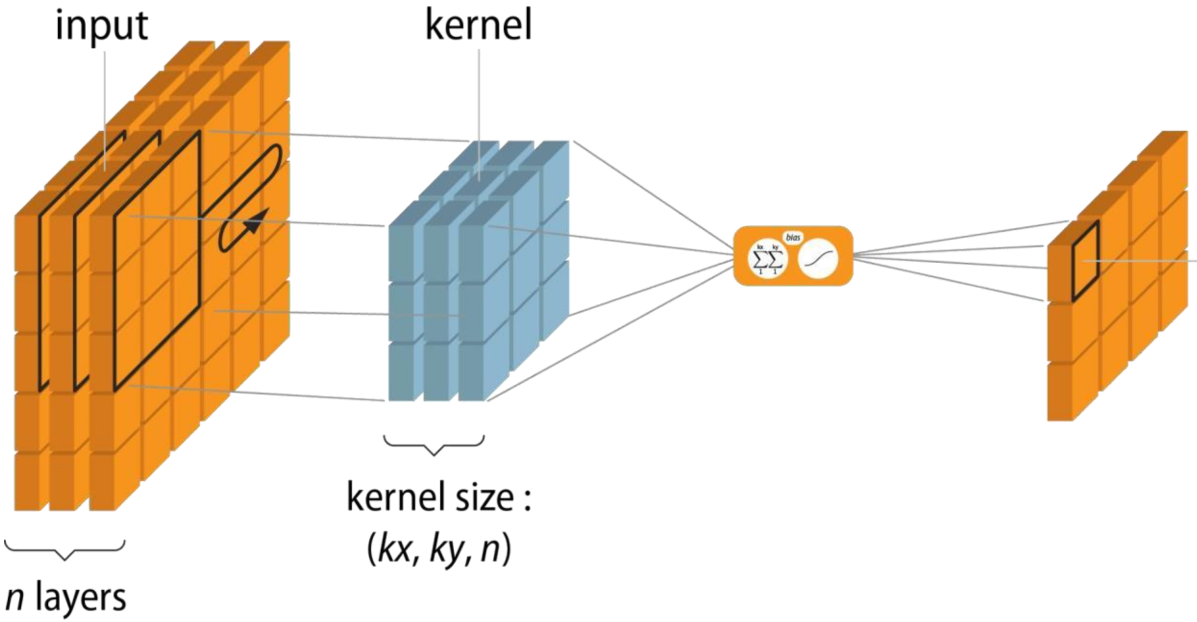


$$y = \sigma\left(\sum \text{input} \otimes \text{kernel} + b\right)$$

Where :
σ :is an activation function
b :is a bias

Number of parameters for a convolutional layer : $kx \,.ky + 1$

Convolutional layers



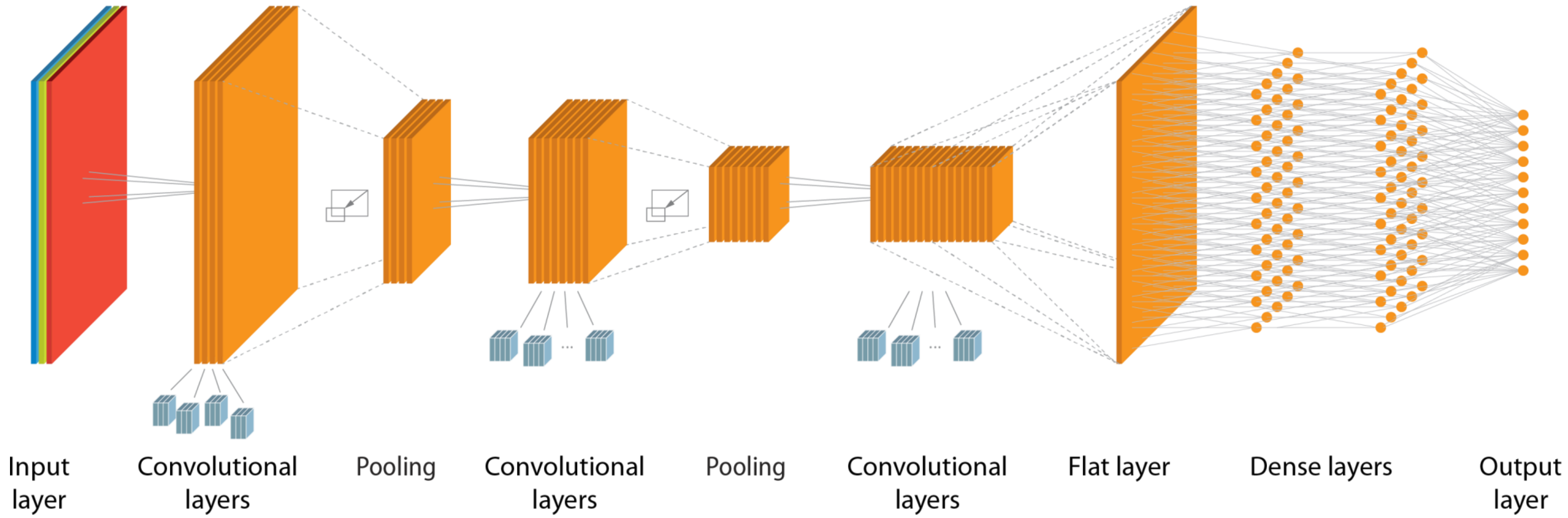
$$y = \sigma\left(\sum \text{input} \otimes \text{kernel} + b\right)$$

Where :
- σ : is an activation function
- b : is a bias

Number of parameters for a convolutional layer : $n \,.kx \,.Ky + 1$

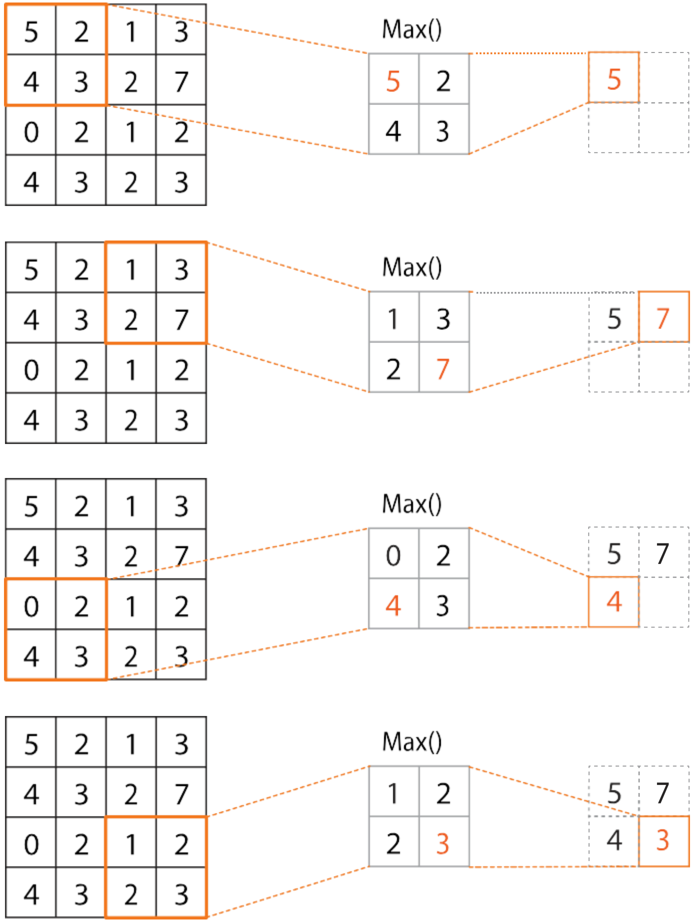
If we want to generate m convolutional layers, we will need m convolutional neurons

Convolutional Neural Networks (CNN)



Convolutional Neural Networks (CNN)

Principle of MaxPooling :

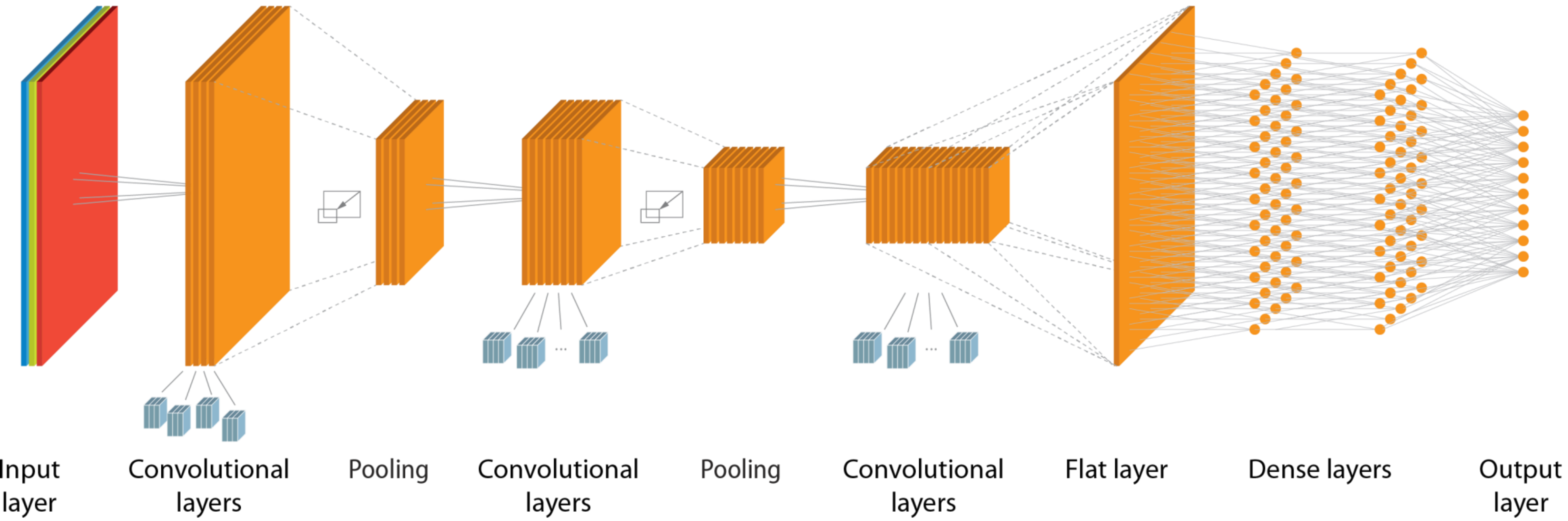


It is possible to set the window size, padding mode and strides.

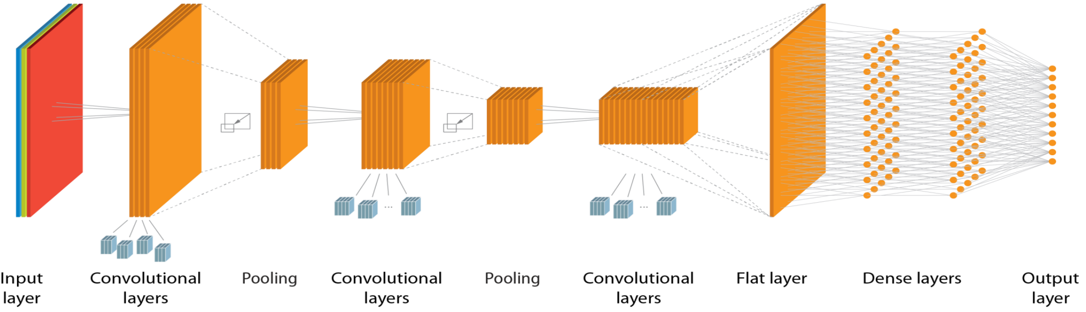
By default, the strides correspond to the size of the window.

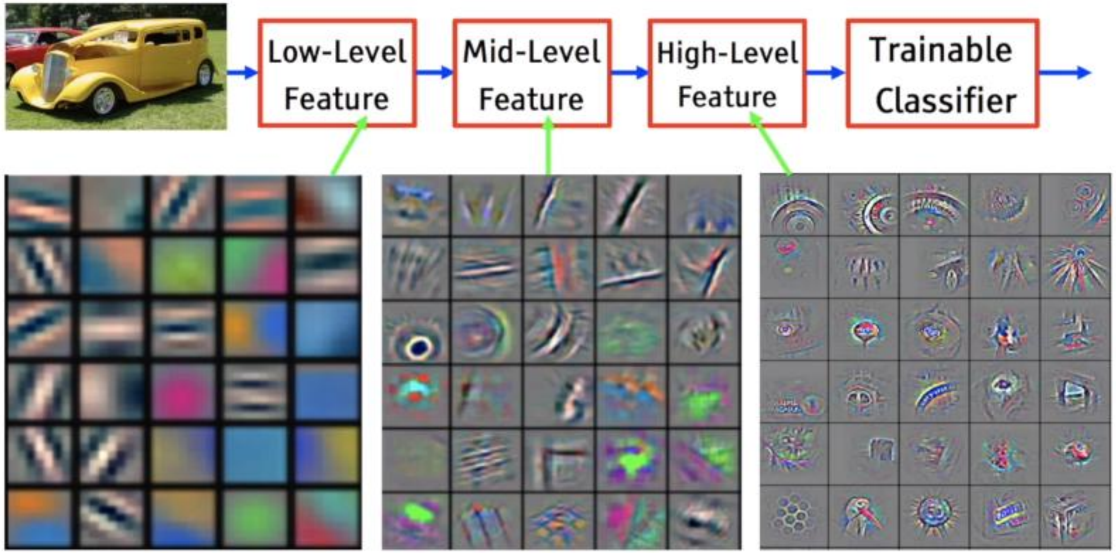
A window (2,2) generates an image twice as small.

Convolutional Neural Networks (CNN)

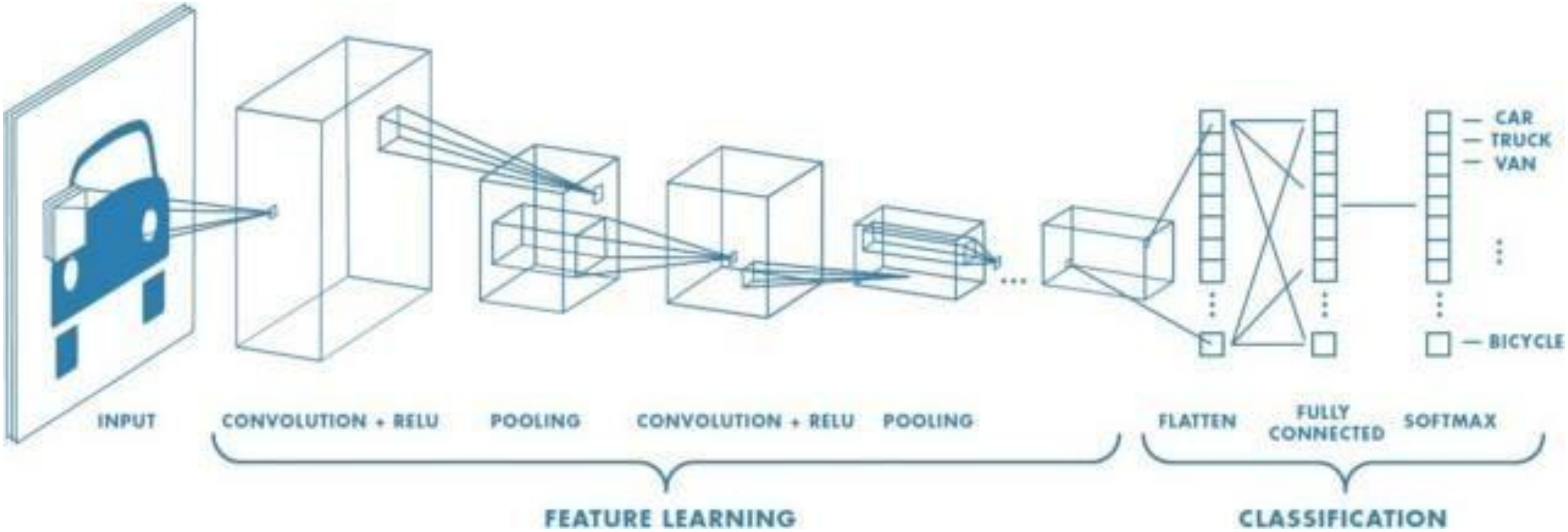


Convolutional Neural Networks (CNN)

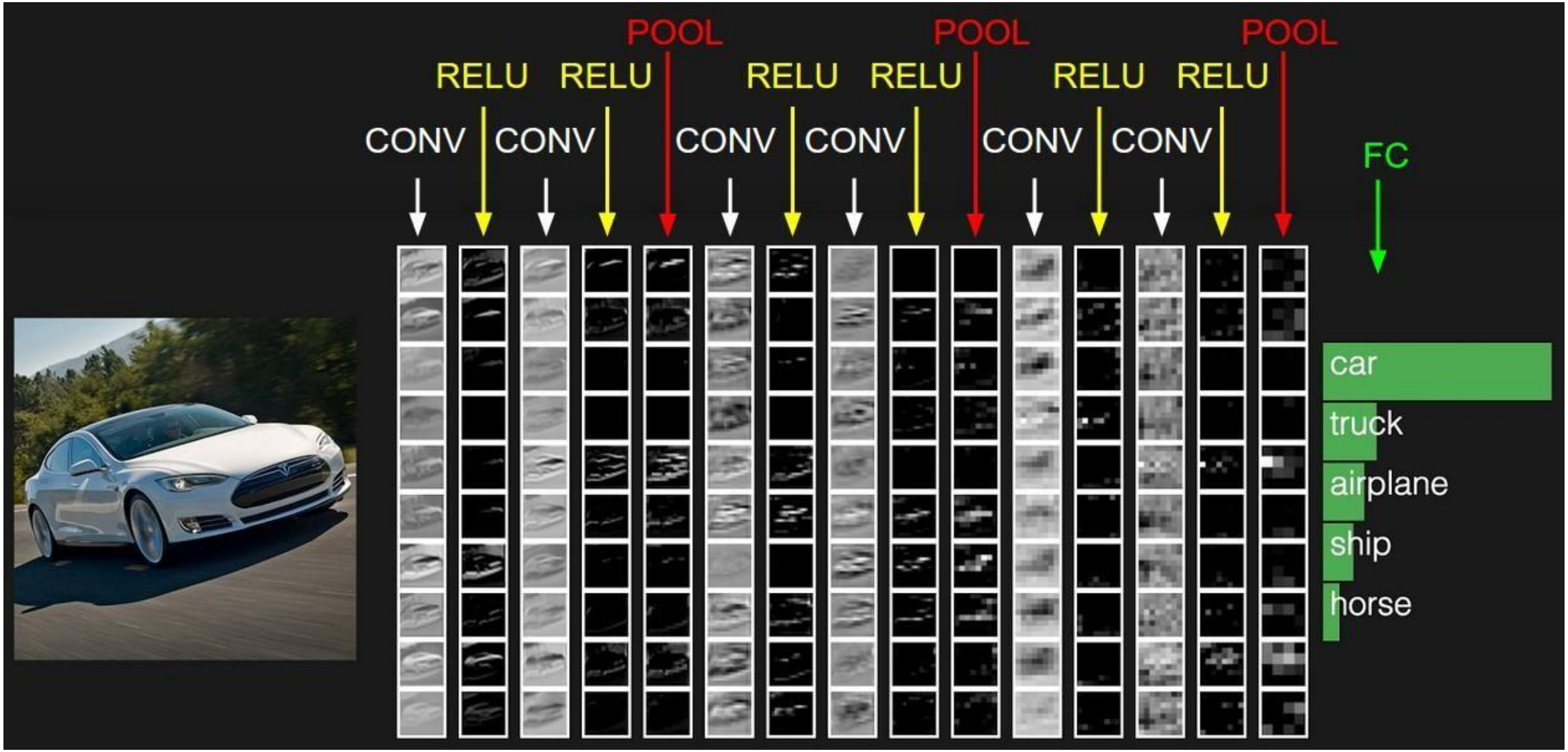




Convolutional Neural Networks (CNN)



Convolutional Neural Networks (CNN)



[Typical architecture of CNN]

Convolutional Neural Networks (CNN)

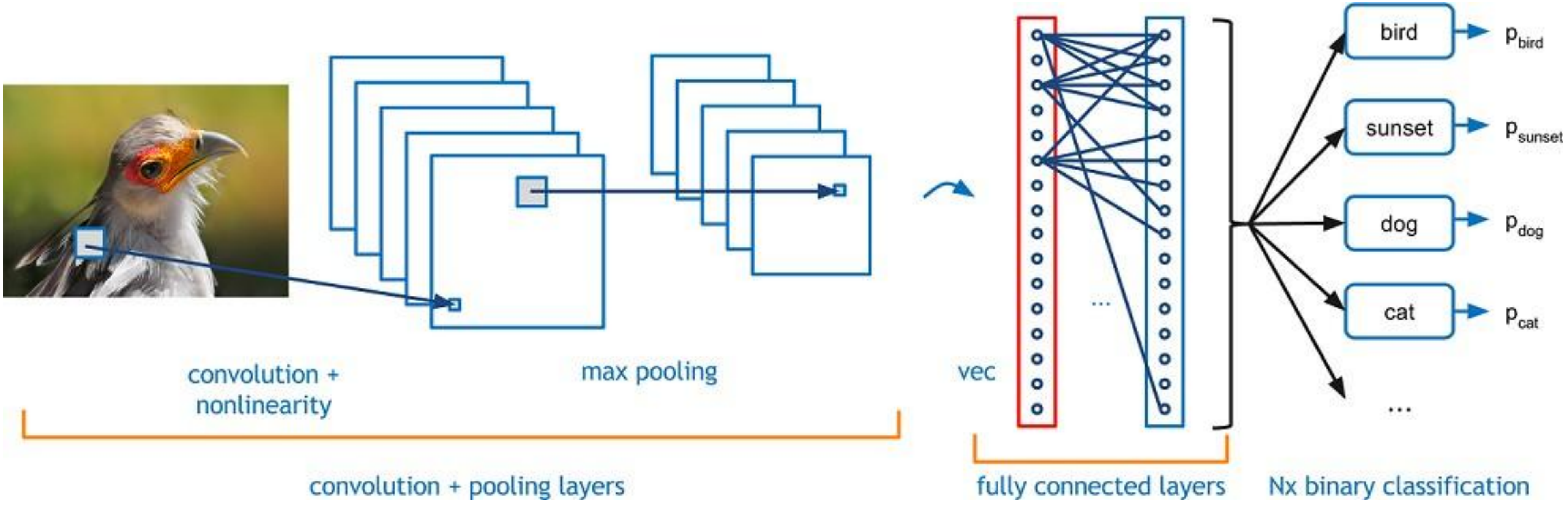


Illustration credit

Convolutional Neural Networks (CNN)

https://www.youtube.com/watch?v=V9ZYDCnItr0&ab_channel=AnimatedAI

Convolutional Neural Networks (CNN)

https://www.youtube.com/watch?v=YSNLMNnlNw8&ab_channel=AnimatedAI

Roadmap

Episode :S01E01

2

Hight Dimensionnal Data

CNN

2.1 **What is a Convolutional Neuron Network (CNN)?**

- Understanding what a CNN is
- Identify use cases

2.2 **Example 1 :MNIST**

- Implementation of a simple case

2.3 **Example 2 :GTSRB**

- The devil is also hiding in the data
- How to work with « large » dataset
- Monitoring the training phase and managing our models
- Improve our results with data augmentation
- Datasets and models: how to automate
- testing How to go from notebook to HPC

Image classification with CNN

Notebook : [MNIST2]

Objective :
Recognizing handwritten numbers

Dataset :
Modified National Institute of Standards and Technology (MNIST)

97.7 %

Previously with a DNN

START

Step1 — Import and init

Step2 — Retrieve data

keras

Step3 — Preparing the data

Train set

Test set

Step4 — Build a model

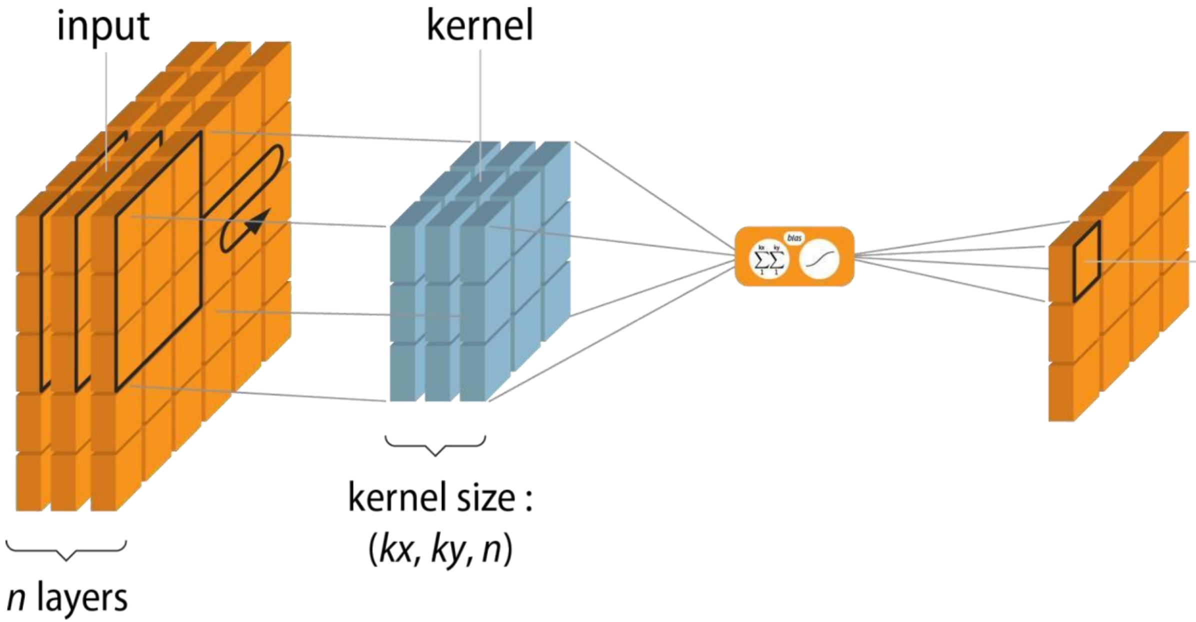
Step5 — Train the model

Step6 — Evaluate

END

Objectives :

Classify handwritten numbers (MNIST dataset) via a CNN



$$y = \sigma\left(\sum \text{input} \otimes \text{kernel} + b\right)$$

Where :
σ :is an activation function
b :is a bias

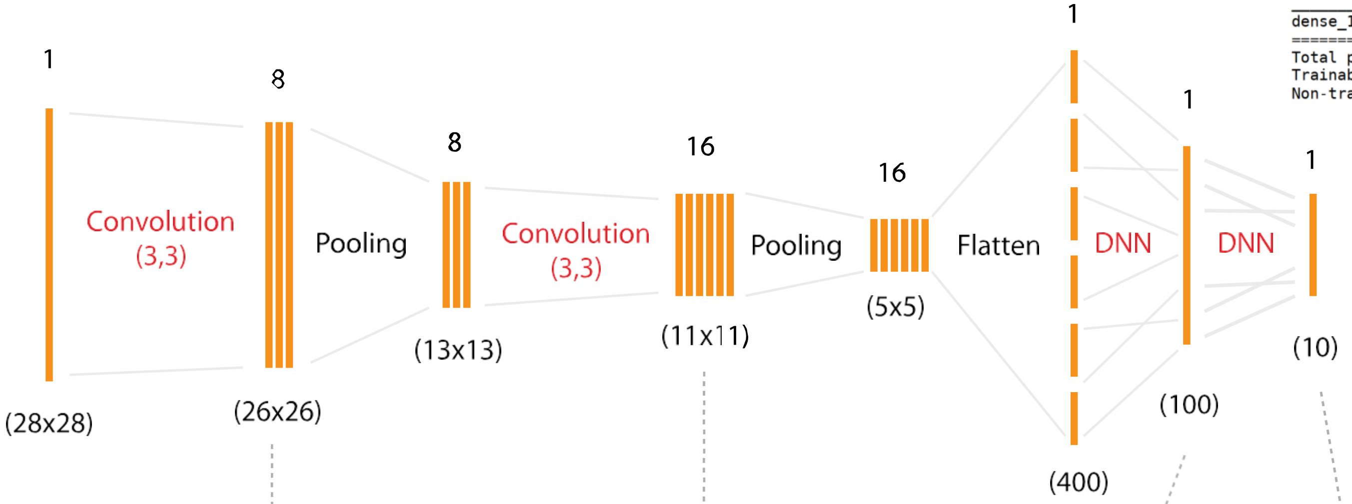
Number of parameters for a convolutional layer : n .kx .Ky + 1

If we want to generate m convolutional layers,we will need m convolutional neurons

[MNIST2]

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 8)         80
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 8)         0
_________________________________________________________________
dropout (Dropout)            (None, 13, 13, 8)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 16)        1168
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 5, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 400)               0
_________________________________________________________________
dense (Dense)                (None, 100)               40100
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1010
=================================================================
Total params: 42,358
Trainable params: 42,358
Non-trainable params: 0
```

Understand how it works by understanding where the parameters are...



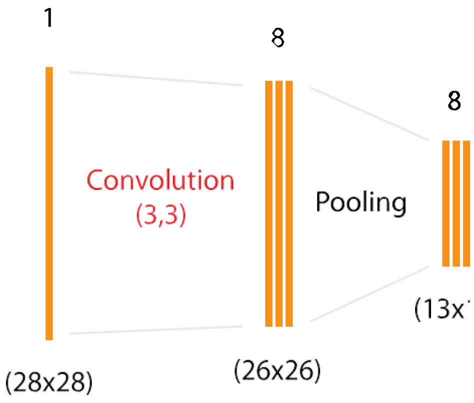
(3x3 + 1)*8 — 80 params

(8x3x3 + 1)*16 — 1168 params

100*(400 + 1) — 40100 params

10*(100 + 1) — 1010 params

Understand how it worl
where the parametersa



(3x3 + 1)*8

80 params

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 8)         80
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 8)         0
_________________________________________________________________
dropout (Dropout)            (None, 13, 13, 8)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 16)        1168
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 16)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 5, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 400)               0
_________________________________________________________________
dense (Dense)                (None, 100)               40100
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1010
=================================================================
Total params: 42,358
Trainable params: 42,358
Non-trainable params: 0
```

Roadmap

Episode :S01E01

2

Hight Dimensionnal Data

CNN

2.1 What is a Convolutional Neuron Network (CNN)?

- → Understanding what a CNN is
- → Identify use cases

2.2 Example 1 : MNIST

- → Implementation of a simple case

2.3 Example 2 : GTSRB

- → The devil is also hiding in the data
- → How to work with « large » dataset
- → Monitoring the training phase and managing our models
- → Improve our results with data augmentation
- → Datasets and models: how to automate testing
- → How to go from notebook to HPC

CNN with GTSRBdataset

Notebooks : [GTSRB1-7]

Objective :
Recognizing traficsigns

Dataset :
German Traffic Sign Recognition Benchmark (GTSRB) is a dataset with more than 50,000 photos of road signs from about 40 classes

CNNwith GTSRBdataset

Notebooks : [GTSRB1-7]

GTSRB1 :	Data analysis and creation of a usable dataset
GTSRB2 :	First convolutions and first results
GTSRB3 :	Monitoring training,managing checkpoints
GTSRB4 :	Improving the results with data augmentation
GTSRB5 :	Combine lots of models and lots of datasets
GTSRB6 :	Run Full convolution notebook as a batch
GTSRB7:	Displaying the reports of the different jobs

Calculation scale :

`scale`

- `scale = 1` Use 100 % of the dataset !
- `scale = 0.1` Use 10 % of the dataset :-)

Enhanced dataset location:

`output enhanced_dir`

- `./data`
- `f'{datasets_dir}/GTSRB/enhanced'`

Notebooks outputs (run_dir) :

`./run`

Calculation scale :

`scale`

- `scale = 1` — Use 100 % of the dataset !
- `scale = 0.1` — Use 10 % of the dataset :-)

Enhanced dataset location:

`output enhanced_dir`

- `./data`
- `f'{datasets_dir}/GTSRB/enhanced'`

Notebooks outputs (run_dir) :

`./run`

START

Step1 — Import and init

Step2 — Parameters

```
scale  = 0.1
output = ./data
```

Step3 — Read the dataset

origine

Step4 — Few statistics

Step5 — Class overview

Step6 — The reality of images

Step7 — Dataset coocking

enhanced

Step8 — Reload data

enhanced

END

Objectives :

Data analysis and creation of a usable enhanced dataset

Roadmap

Episode :S01E01

2

Hight Dimensionnal Data

CNN

2.1 What is a Convolutional Neuron Network (CNN)?

- → Understanding what a CNN is
- → Identify use cases

2.2 Example 1 : MNIST

- → Implementation of a simple case

2.3 Example 2 : GTSRB

- → The devil is also hiding in the data
- → How to work with « large » dataset
- → Monitoring the training phase and managing our models
- → Improve our results with data augmentation
- → Datasets and models: how to automate testing
- → How to go from notebook to HPC

While we talk...

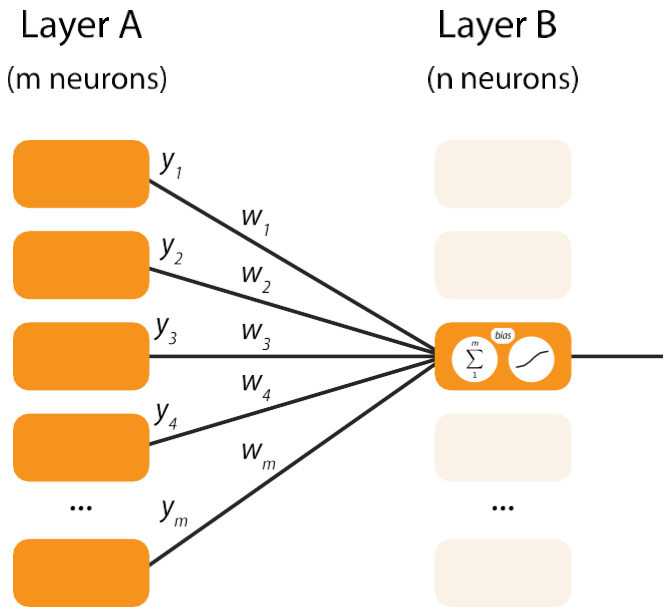
First of all,
let's prepare data with the GTSRB1 notebook.

With :

```
scale  = .2
output = ./data
```

1 - 2
minutes

Principle of a fully connected layer

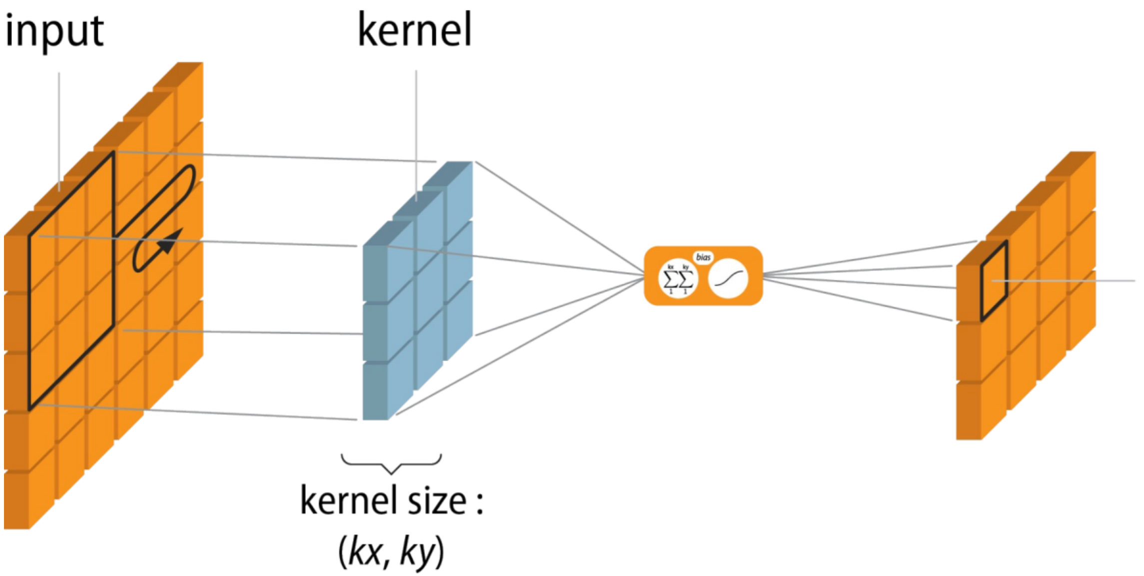


$$y = \sigma(\sum_{1}^{m} y_i \cdot w_i + b)$$

Where :
- σ :is an activation function
- b :is a bias

Number of parameters for a convolutional layer : $n\ (m + 1)$

Previously

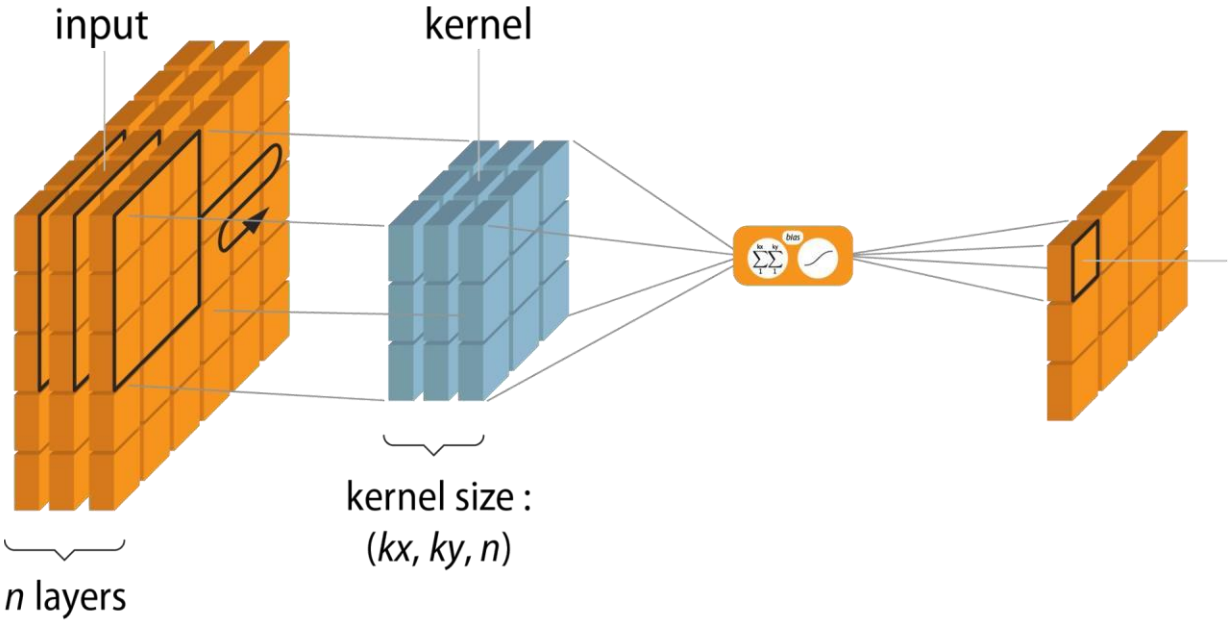


$$y = \sigma(\sum \text{input} \otimes \text{kernel} + b)$$

Where :
σ :is an activation function
b :is a bias

Number of parameters for a convolutional layer : $kx\,.ky + 1$

Previously



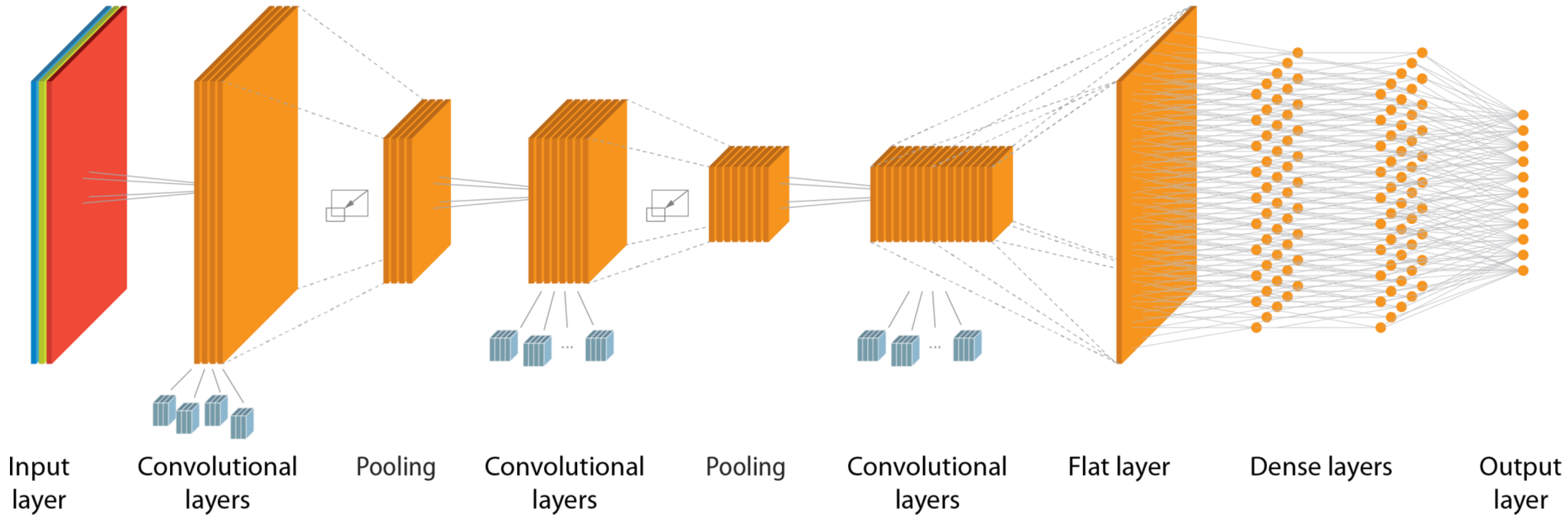
$$y = \sigma(\sum \text{input} \otimes \text{kernel} + b)$$

Where :
σ : is an activation function
b : is a bias

Number of parameters for a convolutional layer : $n . kx . Ky + 1$

If we want to generate m convolutional layers, we will need m convolutional neurons

Previously



Input layer
Convolutional layers
Pooling
Convolutional layers
Pooling
Convolutional layers
Flat layer
Dense layers
Output layer

Roadmap

Episode :S01E01

2

Hight Dimensionnal Data

CNN

2.1 **What is a Convolutional Neuron Network (CNN)?**

- Understanding what a CNN is
- Identify use cases

2.2 **Example 1 : MNIST**

- Implementation of a simple case

2.3 **Example 2 : GTSRB**

- The devil is also hiding in the data
- How to work with « large » dataset
- Monitoring the training phase and managing our models
- Improve our results with data augmentation
- Datasets and models: how to automate testing
- How to go from notebook to HPC

CNN with GTSRB dataset

Notebook : [GTS1-7]

GTSRB1 : Data analysis and creation of a usable dataset

GTSRB2 : First convolutions and first results

GTSRB3 : Monitoring training, managing checkpoints

GTSRB4 : Improving the results with data augmentation

GTSRB5 : Combine lots of models and lots of datasets

GTSRB6 : Run Full convolution notebook as a batch

GTSRB7 : Displaying the reports of the different jobs

50s
(5 epochs)

```
enhanced_dir = ./data
scale        = 1
```

START

Step1 — Import and init

Step2 — Load dataset

enhanced

Step3 — Hace a look

Step4 — Create model

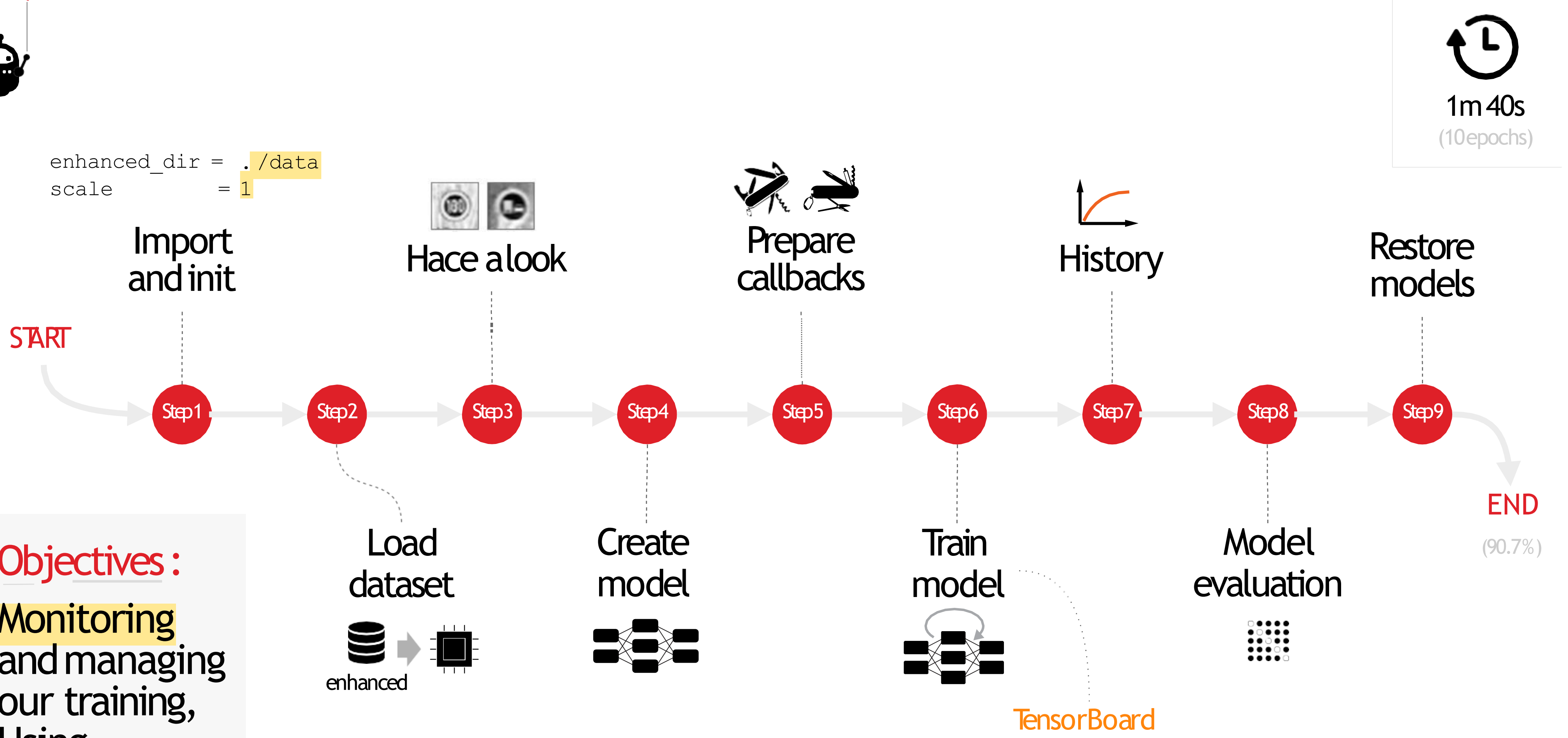
Step5 — Train the model

Step6 — Evaluate

END

Objectives :

Make a first classification via a convolutional network



Objectives :

Monitoring and managing our training, Using checkpoints.

3m 30s
(20 epochs)

```
enhanced_dir = ./data
scale        = 1
```

START

Step 1 — Import and init

Step 2 — Load dataset

enhanced

Step 3 — Create model

Step 4 — Prepare callbacks

Step 5 — Data generator

Step 6 — Train model

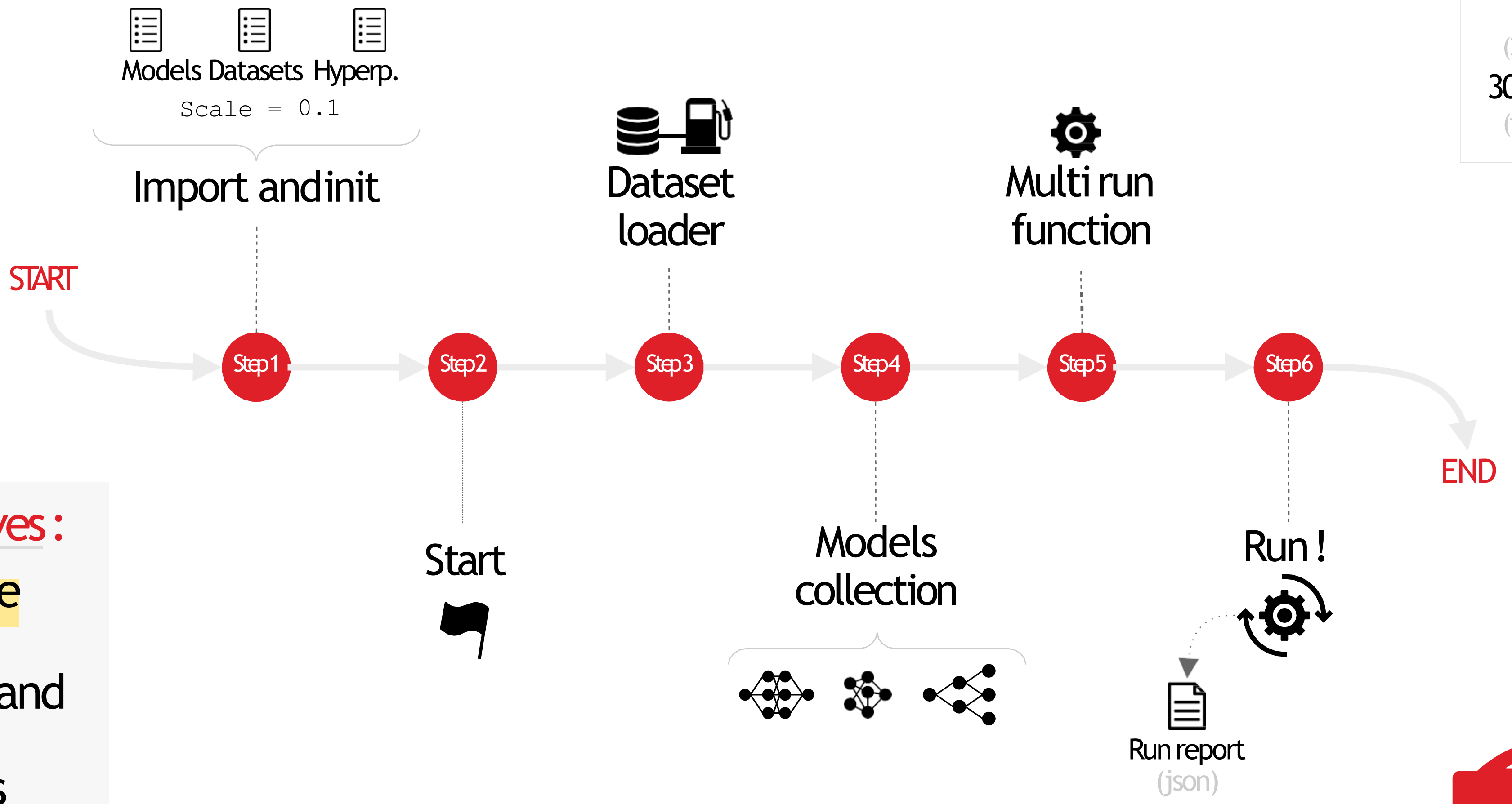
Step 7 — History

Step 8 — Model evaluation

END
(92.7%)

Objectives :

How can we have **more data** when we don't have more !



2m 50s
(2 % datasets)
30' on a V100
(full datasets)

Objectives :

Combine lots of models and lots of datasets

```
$ jupyter nbconvert (...) --to notebook --execute <notebook>
```

How to run a notebook
in a command line?

START

Step1

Step2

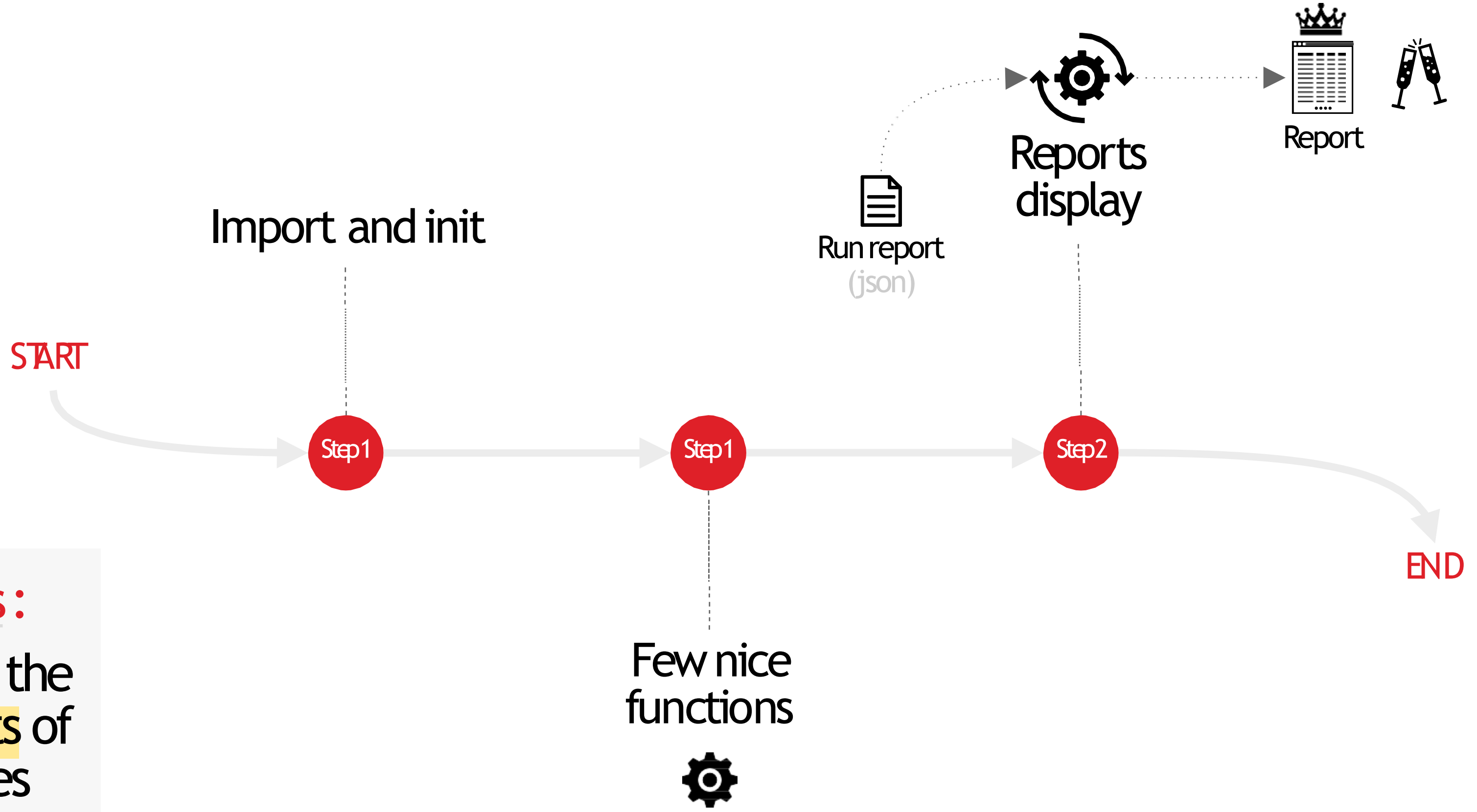
END

How to run a notebook
in a batch?

```
jupyter nbconvert --to script <notebook>
```

Objectives :

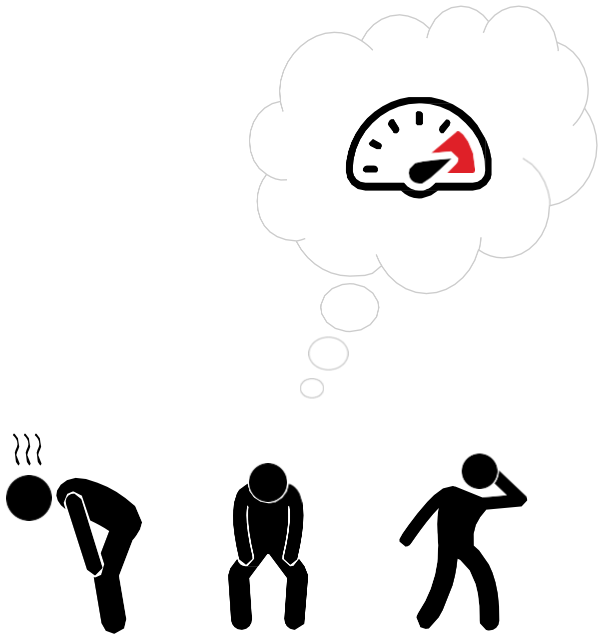
How do I switch from notebook to HPC?



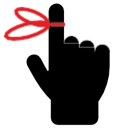
Objectives :

Display of the run reports of our batches

...and last but not least ;-)



...and Last but not least ;-)



Little things and concepts to keep in mind

- Understand the data !
- Organize and prepare our data
- Lots of small data = big problems
- Store our data, h5 files
- Finding the right model isn't easy
- Principle of hyperparameters
- Follow the training (Tensorboard...)
- Saving, retrieving and using recovery points
- Data augmentation
- Automate tests
- Batch mode submission

References

[JGRAY] Gray,J. (2001), from «The Fourth Paradigm: Data-Intensive Scientific Discovery »Tony Hey,Stewart Tansley,Kristin Tolle (2009).Published by Microsoft Research. ISBN:978-0-9825442-0-4

[MCPIT] McCulloch,Warren;Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity".Bulletin of Mathematical Biophysics. 5 (4):115–133.doi:10.1007/BF02478259

[DHEBB] Hebb,D.O.(1949). «The Org anization of Behavior:A Neuropsychological Theory.»NewYork:Wiley and Sons. ISBN9780471367277.

[FROS] Rosenblatt, Frank. (1958). «The perceptron:A probabilistic model for information storage and organization in the brain.» Psychological Review,65(6),386-408.

[MIPA] Minsky,Marvin;Papert,Seymour. (1969). « Perceptrons :An Introduction to Computational Geometry »,MIT Press

[DRUM] Rumelhart, David E.;Hinton, Geoffrey E.;Williams, Ronald J. (1986). « Learning representations by back-propagating errors ».Nature.323 (6088):533–536.doi:10.1038/323533a0.

[YLEC1] Y.LeCun,B.Boser,J. S. Denker,D.Henderson,R.E.Howard,W. Hubbard, L. D.Jackel, « Backpropagation Applied to Handwritten Zip Code Recognition»,AT&T Bell Laboratories

[LRDN] Dominique Cardon,Jean-Philippe Cointet,Antoine Mazieres. (2018).«La revanche des neurones »,Réseaux,La Découverte,5 (211),<10.3917/res.211.0173>. <hal-01925644>

[AMAZ] Antoine Mazieres (2016)Thèse :« Cartographie de l'apprentissage artificiel et de ses algorithmes » Université Paris 7 Denis Diderot, <hal-01771655>

[TOP500] Statistics on top 500 high-performance computers.(2018) « Exponential growth of supercomputing power as recorded by the TOP500 list ». https://www.top500.org

[WKP1] Wikipedia/en.(2018) « List of datasets for machine-learning research ».https://en.wikipedia.org

[WOS1] Core database :TS=("support vector machine*" OR("SVM" AND "classification") OR("SVM" AND"regression") OR("SVM" AND "classifier") OR"support vector network*" OR("SVM"AND"kernel trick*"))

[WOS2] Core database :TS=("deep learning" OR "deep neural network*" OR ("DNN" AND "neural network*") OR "convolutional neural network*" OR ("CNN" AND "neural network*") OR "recurrent neural network*" OR ("LSTM" AND "neural network*") OR ("RNN*" AND "neural network*"))

[ALEX] A.Krizhevsky, I. Sutskever, G.Hinton. (2012). « ImageNet Classification with Deep Convolutional Neural Networks » doi:10.1145/3065386

[ILSVRC] ImageNet Large Scale Visual Recognition Challenges http://image-net.org/challenges/LSVRC/<2012..2017>/results https://en.wikipedia.org/wiki/ImageNet

[MOBIN] Howard,Andrew G.et al. (2017) "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." https://arxiv.org/abs/1704.04861

References

[W2VEC] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013), « Distributed Representations of Words and Phrases and their Compositionality », https://arxiv.org/abs/1310.4546

[GLOVE] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014) « GloVe: Global Vectors for WordRepresentation », http://nlp.stanford.edu/projects/glove/

[P2VEC] Ehsaneddin Asgari, Mohammad R.K. Mofrad, (2016), « ProtVec: AContinuous Distributed Representation of Biological Sequences », https://arxiv.org/abs/1503.05140

[LSTM] Sepp Hochreiter, Jürgen Schmidhuber, (1997), « Long Short-Term Memory, https://doi.org/10.1162/neco.1997.9.8.1735

[GRU] Cho, Kyunghyun;van Merrienboer, Bart;Gulcehre,Caglar; Bahdanau,Dzmitry;Bougares, Fethi;Schwenk,Holger;Bengio, Yoshua (2014), « Learning Phrase Representations using RNN Encoder-Decoder for Statistical MachineTranslation ». https://arxiv.org/abs/1406.1078

[CARTP] AGBarto, RS Sutton and CWAnderson, (1983), « Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem »,IEEETransactionson Systems,Man, and Cybernetics, 1983

[GAN] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, (2014), « Generative Adversarial Networks » https://arxiv.org/abs/1406.2661

[WOS3] Core database : TS=('material' and ('design' or 'discovery' or 'optimization') and ('deep learning' or 'machine learning' or 'neurons'))

[AIDEX] AI Index. « Astarting point for informed conversations about progress in artificial intelligence. The report aggregates a diverse set of metrics, and makes the underlying data easily accessible to the general public». https://aiindex.org

[DLPW] Jeff Hale, « Deep Learning Framework Power Scores 2018 » http://bit.ly/33Wp14y and http://bit.ly/2NagcgH

[CNIL1] Comment permettre à l'homme de garder la main ? Synthèse du débat public animé parla cnil dans le cadre de la mission de réflexion éthique confiée par la loi pour une république numérique. https://www.cnil.fr/fr/comment-permettre-lhomme-de-garder-la-main-rapport-sur-les-enjeux-ethiques-des-algorithmes-et-de

[CNIL2] Reconnaissance faciale :pour un débat à la hauteur des enjeux 15 novembre 2019 https://www.cnil.fr/fr/reconnaissance-faciale-pour-un-debat-la-hauteur-des-enjeux

Illustrations

[POTATO]	From *Die Giftpflanzen Deutschlands,* Peter Esser, 1910, via iconspng.com
[CONVO]	An Introduction to different Types of Convolutions in Deep Learning https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d
[NEURON]	Wikimedia Commons, the free media repository.
Photos	pixels.com
Icons	thenounproject.com

GitLab

https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle

Fidle.Contact@grenoble.cnrs.fr