SAISON 22/23

Formation

Introduction
Deep Learning

Séquence 2

Convolutional neural network (CNN) – 2/2
An example of a complete workflow implementation

FIDLE

https://fidle.cnrs.fr

Cette session va être enregistrée.
Retrouvez-nous sur notre chaine YouTube :-)

This session will be recorded.
Find us on our YouTube channel :-)

https://fidle.cnrs.fr/youtube

SAISON 22/23

Formation

Introduction Deep Learning

Séquence 2

Convolutional neural network (CNN) – 2/2

An example of a complete workflow implementation

FIDLE

https://fidle.cnrs.fr

Cette session va être enregistrée.
Retrouvez-nous sur notre chaine YouTube :-)

This session will be recorded.
Find us on our YouTube channel :-)

https://fidle.cnrs.fr/youtube

Resources

https://fidle.cnrs.fr

Powered by CNRS CRIC, and UGA DGDSI
of Grenoble, Thanks !

Course materials (pdf)

Practical work environment*

Corrected notebooks

Videos (YouTube)

(*) Procedure via Docket or pip
Remember to get the latest version !

Resources

You can also subscribe to :

http://fidle.cnrs.fr/listeinfo

https://listes.services.cnrs.fr/wws/info/devlog[1]

https://listes.math.cnrs.fr/wws/info/calcul[2]

(1) List of ESR* developers,
(2) List of ESR* « calcul » group
Where ESR is Enseignement Supérieur et Recherche, french universities and public academic research organizations

Program

20 Séquences
du 17 novembre
au 14 mai 2023

1 ✓
History, Fundamental Concepts
Basic Regression DNN
Basic Classification DNN

2 3
Hight Dimensionnal Data
CNN

4
Demystify **mathematics** for neural networks.

5
Training strategies
Evaluation
Sparse data (text)
Embedding

6
Sequences data
RNN

7
PyTorch
A small detour with **PyTorch**.

8
«Attention is All You Need»
Transformers

9 New !
Graph Neural Network
GNN

10
Autoencoder networks
AE

11
Variational Antoencoder
VAE

12
Project session
«My project in 180 s»

13
Generative Adversarial Networks
GAN

14 New !
Diffusion Model
Text to image

15
AI, Law, Society and **Ethics**

16 New !
Model and training optimization
Resource efficiency

17
Jean-Zay
GPU acceleration

18 New !
Physics-Informed Neural Networks
PINNS

19
Deep Reinforcement Learning
RL

20
What will be tomorrow's AI
Review & perspectives !

FIDLE

SAISON 22/23

Roadmap

2 3

High 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

3.1 **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

Previously on Fidle...

Few little things and concepts to keep in mind

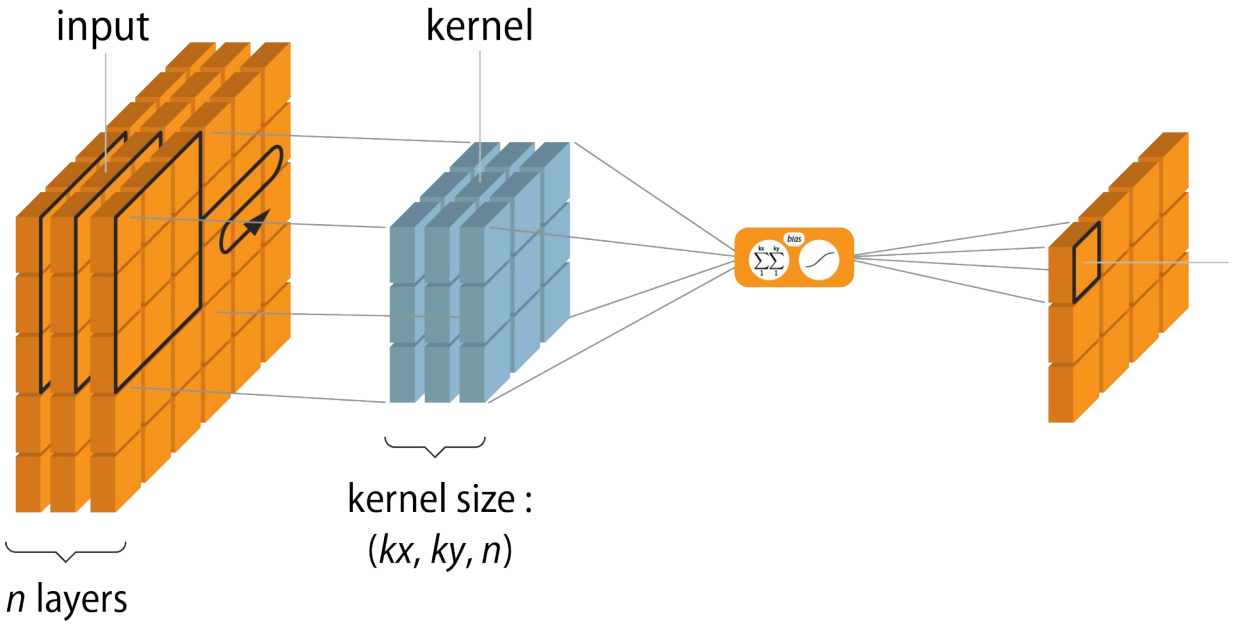
2 3

Hight Dimensionnal Data

CNN

- Convolution neuron
- Convolution network
- Training and test/validation
- Dropout
- Pooling
- Epochs and Batchs
- Metrics
- Correlation matrix
- Numpy shape

Previously on Fidle



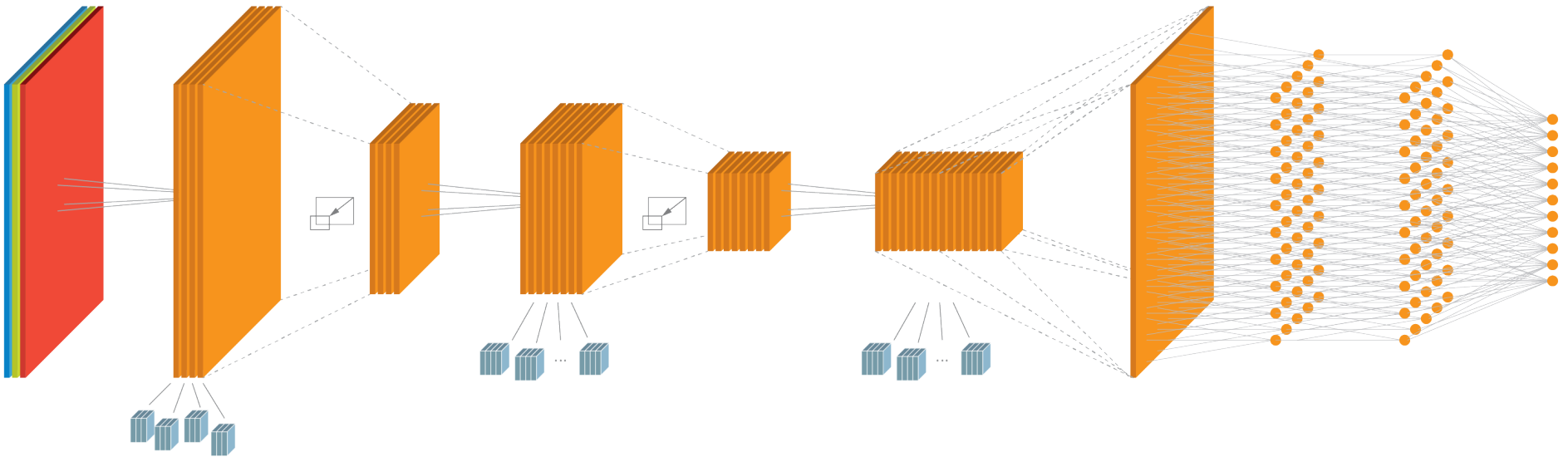
$$y = \sigma(\sum \text{input} \otimes \text{kernel} + b)$$

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

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

The weights constituting our kernel will be progressively shaped during the learning process, as in the case of "classical" neurons.

Previously on Fidle



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

Roadmap

2 3

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

3.1 **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

Roadmap

2 3

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

3.1 **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

Roadmap

2 3

Hight Dimensionnal Data

CNN

Sure that it will remain easy

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

- Understanding what a CNN is
- Identify use cases

2.2 **Example 1 : MNIST**

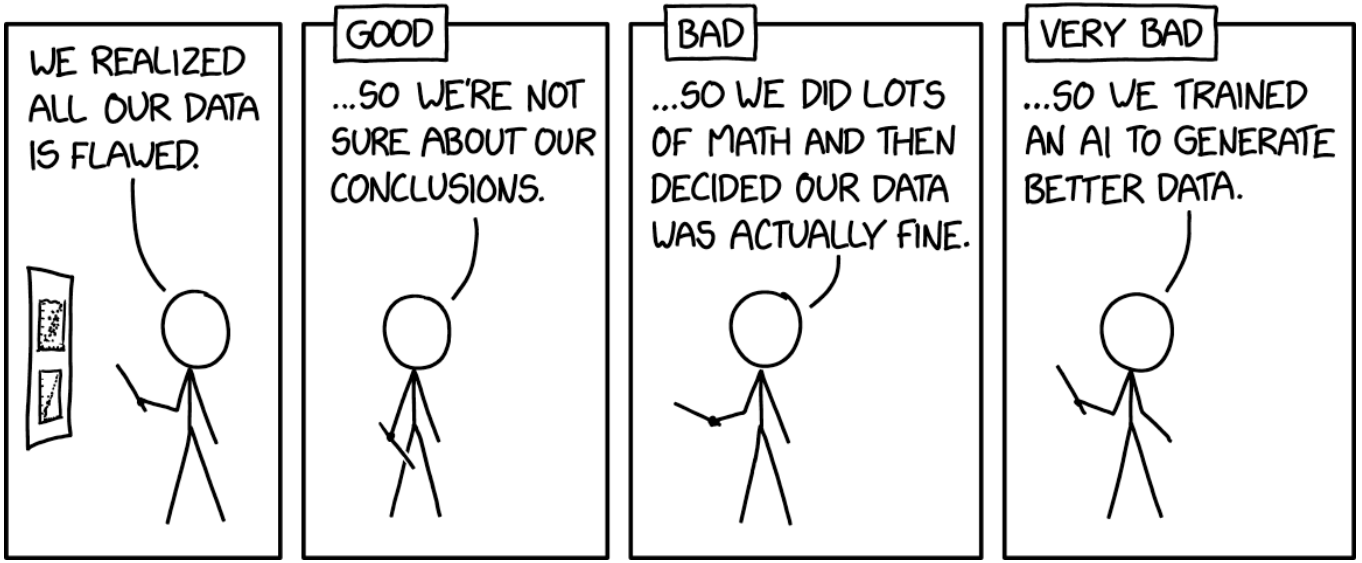
- Implementation of a simple case

3.1 **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

Our objective is to implement a real and complete example of processing !

Deep Learning uses data...
...preparing and using data is not always easy !



https://imgs.xkcd.com/comics/flawed_data.png

CNN with GTSRB dataset

Notebooks : [GTSRB1-7]

Objective :
Recognizing trafic signs

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

CNN with GTSRB dataset

Notebooks : [GTSRB1-7]

GTSRB1 : Data analysis and creation of a usable dataset

GTSRB2 : First convolutions and first results

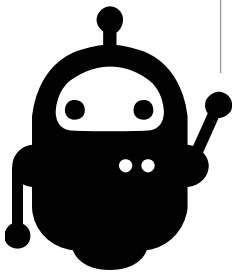
GTSRB3 : Monitoring training, managing checkpoints

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 :-)

For production,
with a GPU !

For development,
without GPU...

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 :-)

For production, with a GPU !

For development, without GPU...

Enhanced dataset location :

`output enhanced_dir`

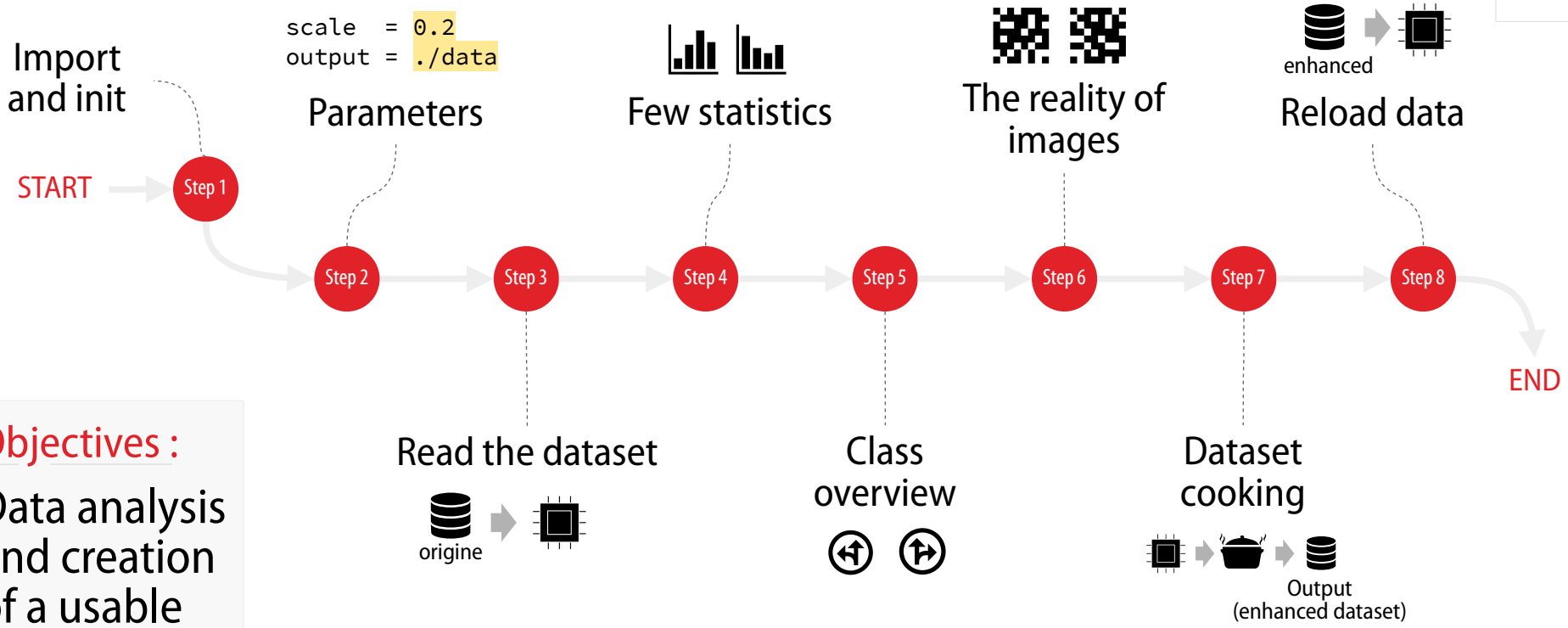
- `./data`
- `f'{datasets_dir}/GTSRB/enhanced'`

Notebooks outputs (run_dir) :

`./run`

Dataset analysis and preparation

± 3 minutes



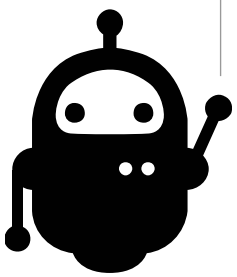
Objectives :

Data analysis and creation of a usable enhanced dataset

CNN with GTSRB dataset

Notebook : [GTS1-7]

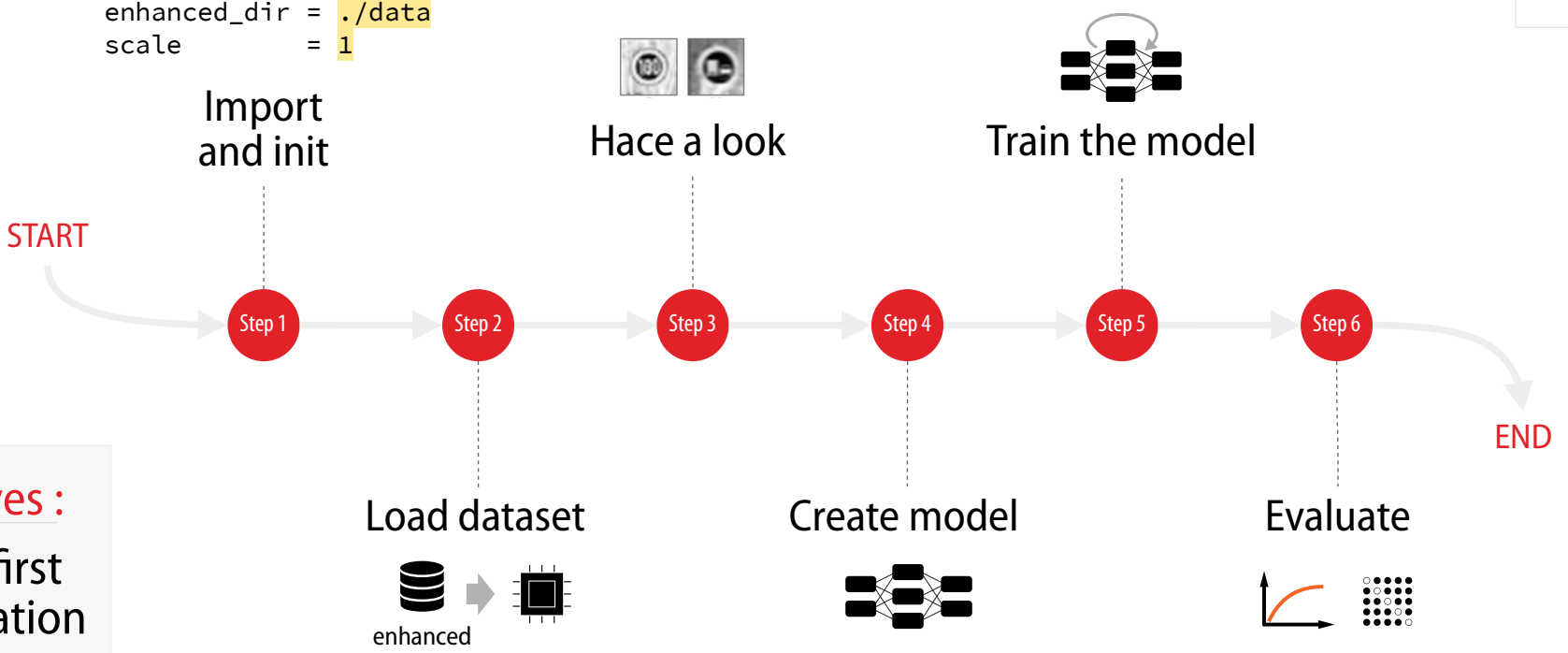
GTSRB1 :	Data analysis and creation of a usable dataset
GTSRB2 :	First convolutions and first results
GTSRB3 :	Monitoring training, managing checkpoints
GTSRB4 :	Improving the results with data augmentation
GTSRB5 :	Combine lots of models and lots of datasets
GTSRB6 :	Run Full convolution notebook as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



First Convolutions

1m50s
(5 epochs)

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



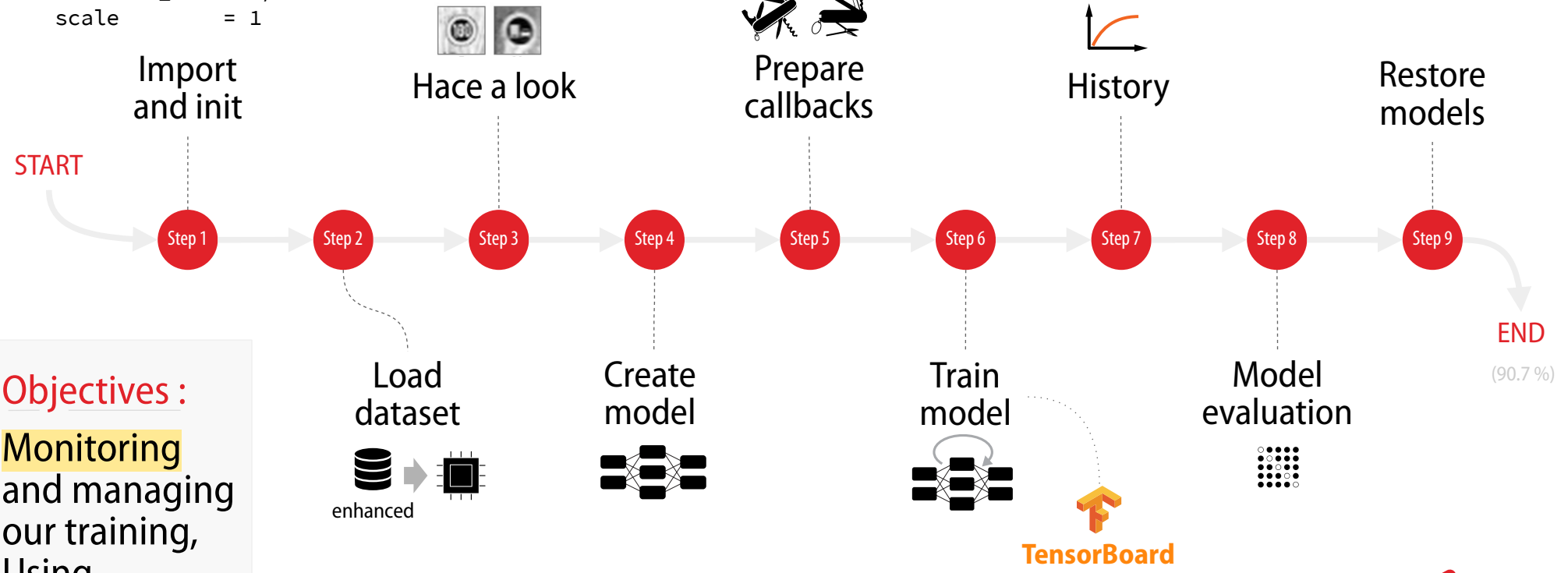
Objectives :

Make a first classification via a convolution al network

Training and monitoring

1m 40s
(10 epochs)

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



Objectives :

Monitoring and managing our training, Using checkpoints.

Data augmentation

3m 30s
(20 epochs)

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

START

Step 1 — Import and init

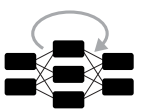
Step 2 — Load dataset

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 !

Full convolutions

3m 40s
(2 % datasets)

30' on a V100
(full datasets)



Models Datasets Hyperp.

Scale = 0.1

Import and init

Dataset loader

Multi run function

START

END

Start infos

Models collection

Run !

Run report
(json)

Objectives :

Combine lots of models and lots of datasets

Full convolutions as a batch

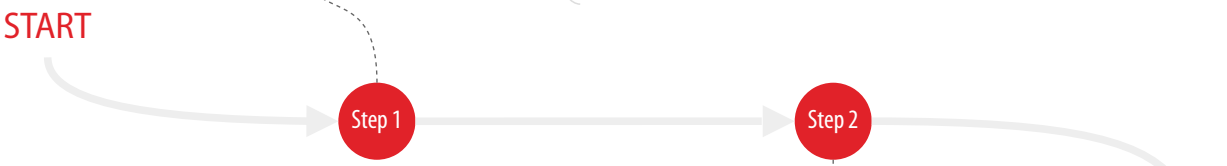
How to run a notebook in a command line ?

As a notebook !

```
$ jupyter nbconvert --ExecutePreprocessor.timeout=-1  \
                    --to notebook  \
                    --execute <notebook>
```

As a python script !

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



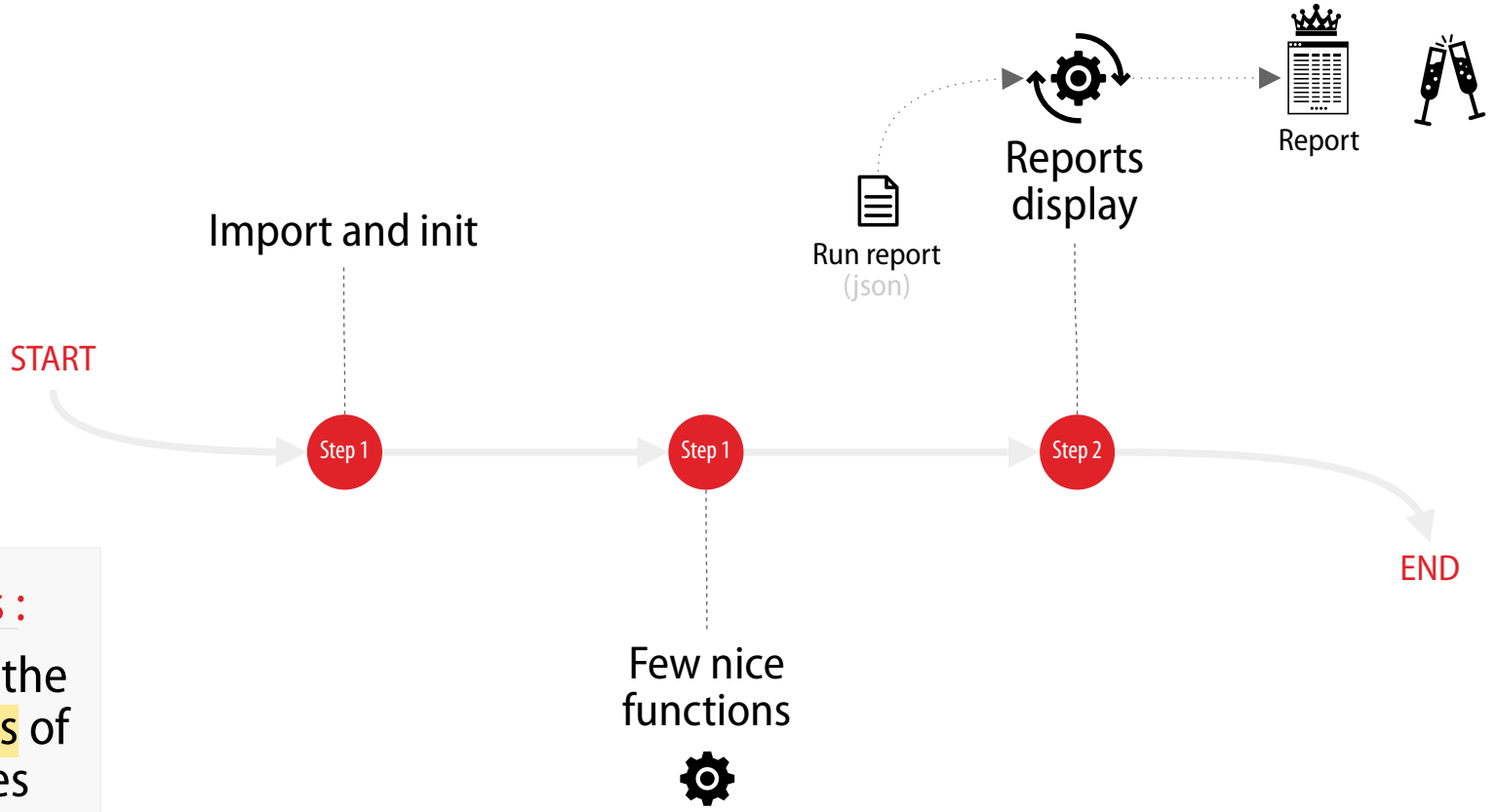
How to run a notebook in a batch ?

slurm workload manager

Objectives :

How do I switch from notebook to HPC?

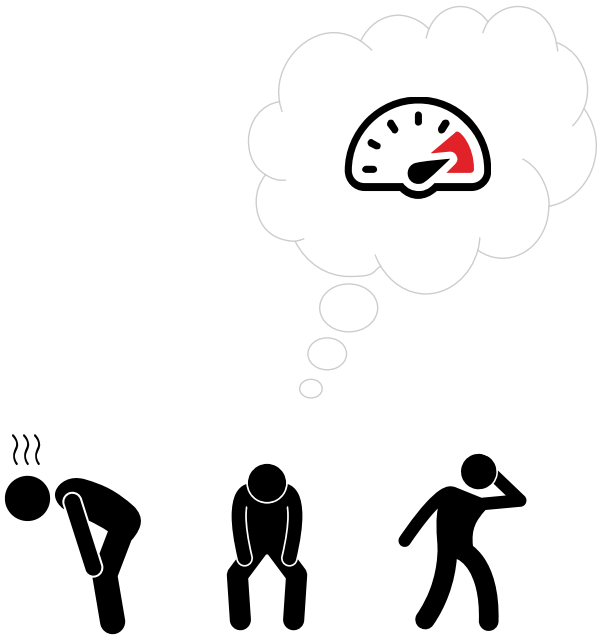
Batch reports



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

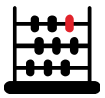
$$\exp^{-1}\left(2^{-8}\right) = \overline{\frac{1}{\mathbf{g}^{(z)}}} \cap z\left(-\infty^{1}\right).$$

Next, on Fidle :

$$H^{(\rho)}\left(\emptyset^{-6}, -1^{6}\right) \geq \left\{ \bar{P}(t) \times \pi : \overline{-\gamma} \leq \frac{\cosh\left(\left\|\hat{\lambda}\right\|\right)}{M_{\Lambda,\psi}\left(1 - -\infty, \ldots, - - \infty\right)} \right\}$$

$$\supset \frac{p''\left(\frac{1}{\pi}\right)}{Y\left(\infty^{9}, \ldots, \psi \vee \varphi_{\mathcal{C}}\right)} - O\left(\frac{1}{1}, \hat{W} \pm \mathcal{N}\right).$$

4



Demistify **mathematics** for neural networks

Jeudi 8 décembre, 14h00

Séquence 4 :

Démystifier les outils mathématiques

Présentation mathématique de l'optimisation par descente de gradient

Principe de la rétropropagation

Mise en lumière de l'intérêt pratique des méthodes stochastiques.

Exemple proposé :

Programmation manuelle de l'apprentissage d'un réseau convolutif

Durée : 2h00

$$\mathcal{H}\left(\left\|\mu\right\|, \alpha^{(h)}\right) \geq \frac{Y1}{\mathcal{V}\left(P^{-9}, \frac{1}{\mathbf{c}^{(h)}}\right)}.$$

$$\bar{\Psi}\left(a1, \ldots, \eta \cup \iota\right) = \bigotimes_{\tilde{\tau}=2}^{\emptyset} \tan\left(\pi^{-3}\right),$$

$$\hat{f}^{-1}\left(-\infty e\right) = \int_{-\infty}^{\sqrt{2}} \overline{e^{-2}}\, d\bar{T} + \cdots \mathbf{u}\left(-\left\|\mathfrak{q}\right\|, 2^{3}\right)$$

$$< \frac{\log\left(-1\right)}{\sinh^{-1}\left(\hat{\mathbf{c}}^{8}\right)} + \cdots \cup \log^{-1}\left(\tilde{\rho}\right),$$

$$\hat{f}^{-1}\left(-\infty e\right) = \int_{-\infty}^{\sqrt{2}} \overline{e^{-2}}\, d\bar{T} + \cdots \mathbf{u}\left(-\left\|\mathfrak{q}\right\|, 2^{3}\right)$$

$$< \frac{\log\left(-1\right)}{\sinh^{-1}\left(\hat{\mathbf{c}}^{8}\right)} + \cdots \cup \log^{-1}\left(\tilde{\rho}\right),$$

$$G\left(-\pi\right) \neq \left\{ 1^{7} : \delta^{-1}\left(\pi^{5}\right) \geq \int_{\emptyset}^{0} \Theta\left(-\infty, \ldots, \frac{1}{e}\right) dK \right\}.$$

Next on Fidle :

4

Demistify **mathematics** for neural networks

Jeudi 8 décembre, 14h00

Séquence 4 :

Démystifier les outils mathématiques

Programmation manuelle de l'apprentissage d'un réseau convolutif

$$\hat{f}^{-1}(-\infty e) = \int_{-\infty}^{\sqrt{2}} \overline{e^{-2}}\, d\bar{T} + \cdots \mathbf{u}\left(-\|\mathfrak{q}\|, 2^{3}\right)$$
$$< \frac{\log(-1)}{\sinh^{-1}(\dot{\mathbf{c}}^{8})} + \cdots \cup \log^{-1}(\tilde{\rho}),$$
$$G(-\pi) \neq \left\{ 1^{7} : \delta^{-1}(\pi^{5}) \geq \int_{\emptyset}^{0} \Theta\left(-\infty, \ldots, \frac{1}{e}\right) dK \right\}.$$
$$H^{(\rho)}\left(\emptyset^{-6}, -1^{6}\right) \geq \left\{ \bar{P}(t) \times \pi : \overline{-\gamma} \leq \frac{\cosh\left(\|\hat{\lambda}\|\right)}{M_{\Lambda,\psi}\left(1 - -\infty, \ldots, - - \infty\right)} \right\}$$
$$\supset \frac{p''\left(\frac{1}{\pi}\right)}{Y\left(\infty^{9}, \ldots, \psi \vee \varphi_{\mathcal{C}}\right)} - O\left(\frac{1}{1}, \hat{W} \pm \mathcal{N}\right).$$

To be continued...

Contact@fidle.cnrs.fr
FIDLE https://fidle.cnrs.fr
YouTube https://fidle.cnrs.fr/youtube

FIDLE