



Bienvenue!

**ÉCOLE D'HIVER FRANCOPHONE
EN APPRENTISSAGE PROFOND**

5 - 9 mars 2018



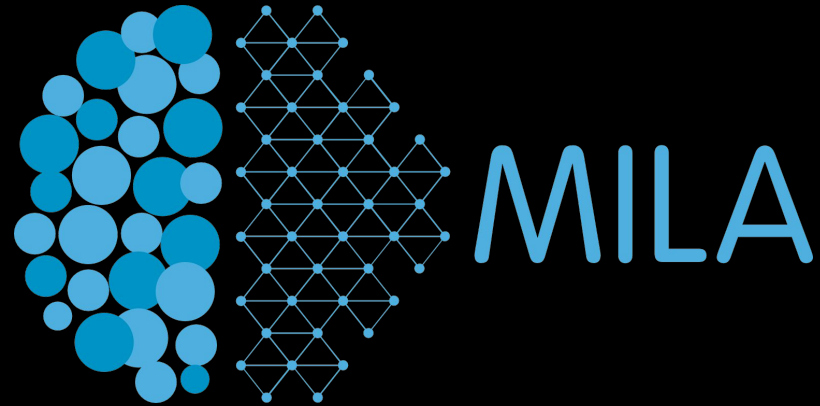
IVADO

HEC Montréal
Polytechnique Montréal
Université de Montréal



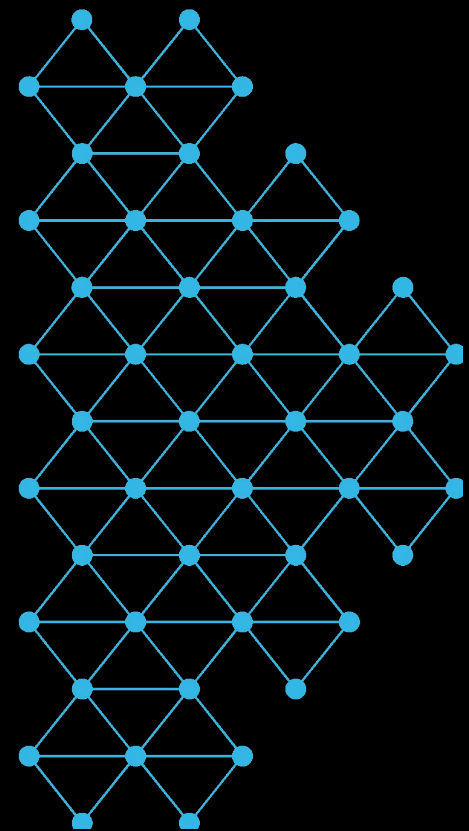
MILA

Institut
des algorithmes
d'apprentissage
de Montréal



Introduction: Bibliothèques & Installation Party

Gaétan Marceau Caron
gaetan.marceau.caron@rd.mila.quebec



Éviter de réinventer la
roue... en moins bien.

Librairies existantes

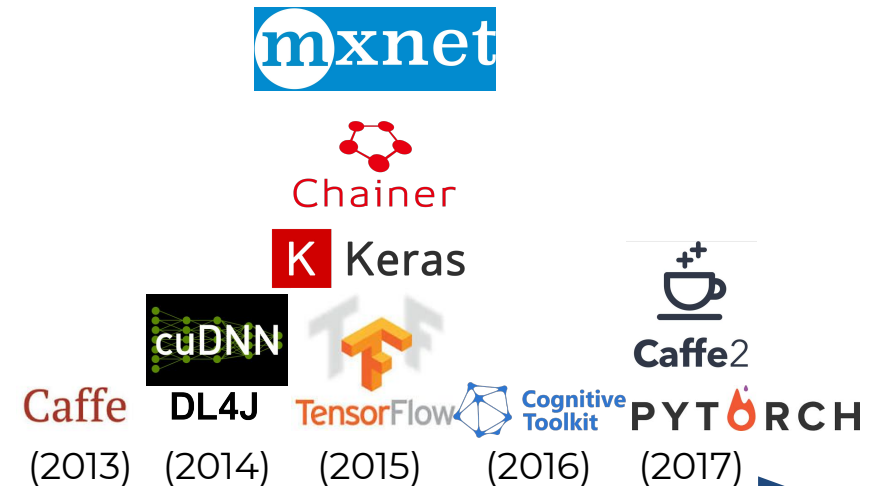


Torch
(2002)



theano

(2008)



2017/11/15: Release of Theano 1.0.0

Arrêt du développement logiciel par le MILA
Précurseur à beaucoup d'idées qui se
retrouvent dans les librairies plus récentes.

Caractéristiques

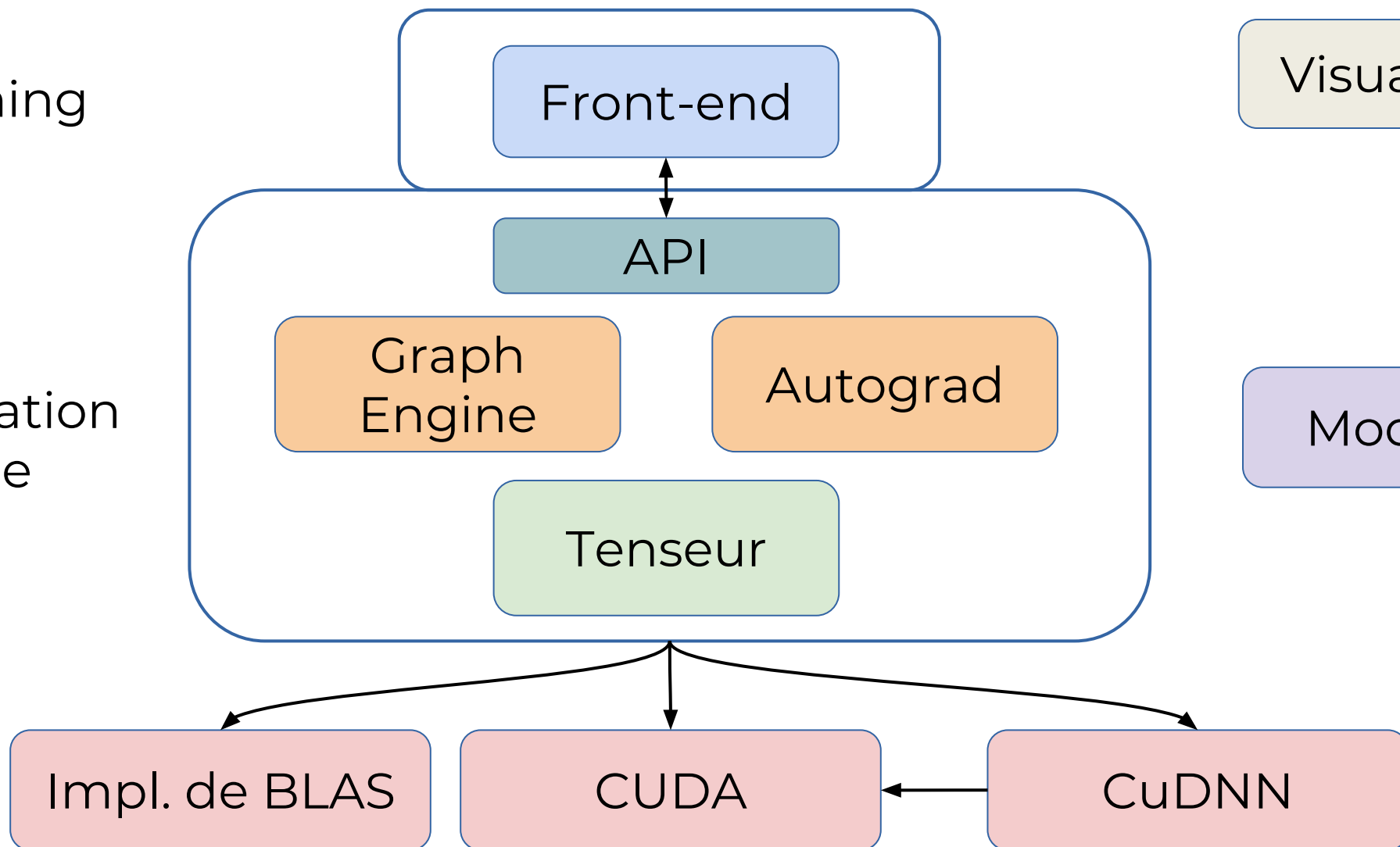
- Une **hiérarchie** d'outils
- Utilisation de **matériel de calcul spécifique** (GPU-TPU)
- Support important à la **recherche**
- Passage de la recherche à la **production**
- Support de la **communauté** d'utilisateurs

Une hiérarchie d'outils

Deep Learning

Programmation différentielle

Calcul matriciel



Visualisation

Model Zoo

Une hiérarchie d'outils

Deep Learning

Front-end

Permet de se concentrer sur les concepts de deep learning.
Le front-end peut appartenir à la librairie, e.g., en Pytorch.



Une hiérarchie d'outils

Programmation
différentielle

API

L'API permet de programmer des concepts mathématiques sur les tenseurs afin de créer de nouveaux modules.

A graph is created on the fly

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))

i2h = torch.mm(W_x, x.t())
h2h = torch.mm(W_h, prev_h.t())
next_h = i2h + h2h
next_h = next_h.tanh()
```


Une hiérarchie d'outils

Programmation
différentielle

Graph
Engine

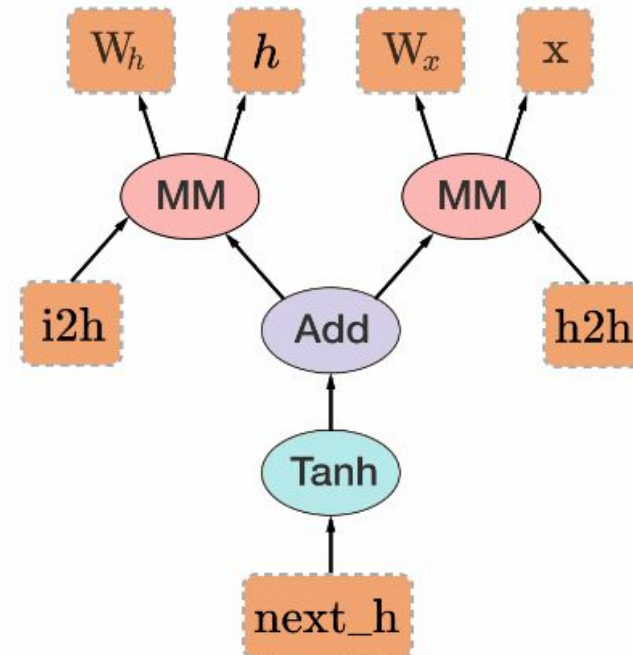
Le *graph engine* permet de construire et d'optimiser le graphe computationnel.

A graph is created on the fly

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))

i2h = torch.mm(W_x, x.t())
h2h = torch.mm(W_h, prev_h.t())
next_h = i2h + h2h
next_h = next_h.tanh()
```



Une hiérarchie d'outils

Programmation
différentielle

Autograd

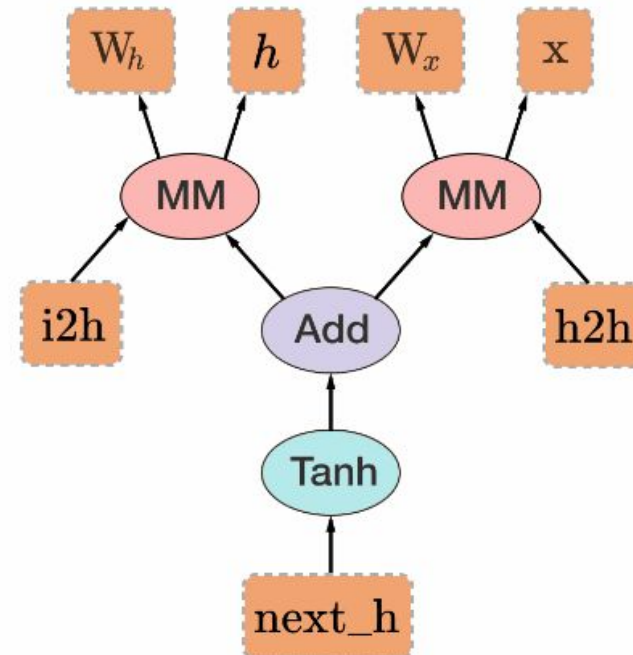
Le différentiateur automatique permet de calculer le gradient de la sortie par rapport aux paramètres.

A graph is created on the fly

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))

i2h = torch.mm(W_x, x.t())
h2h = torch.mm(W_h, prev_h.t())
next_h = i2h + h2h
next_h = next_h.tanh()
```



Une hiérarchie d'outils

Programmation
différentielle

Tenseur

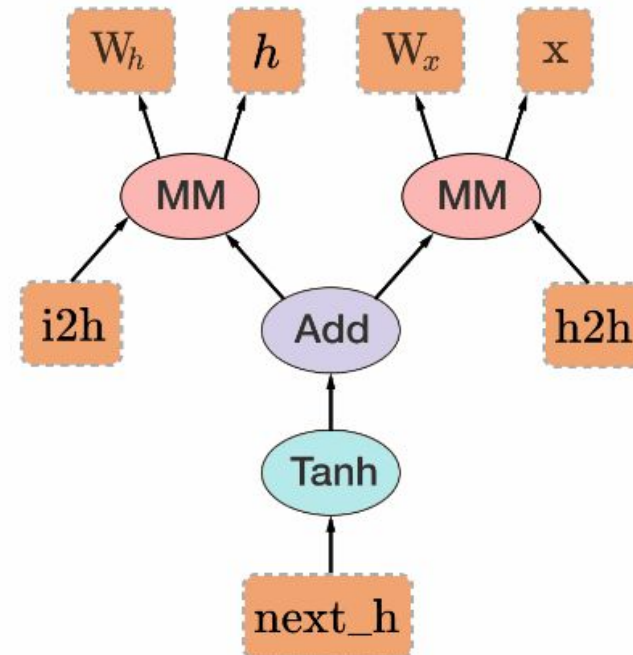
La structure de *Tenseur* permet de manipuler efficacement les données et de les représenter de différentes manières (1d, 2d, 3d,...)

A graph is created on the fly

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))

i2h = torch.mm(W_x, x.t())
h2h = torch.mm(W_h, prev_h.t())
next_h = i2h + h2h
next_h = next_h.tanh()
```



Une hiérarchie d'outils

Calcul matriciel

Impl. de BLAS

CUDA

CuDNN

Les calculs sur les tenseurs s'implémentent par des **calculs matriciels**.

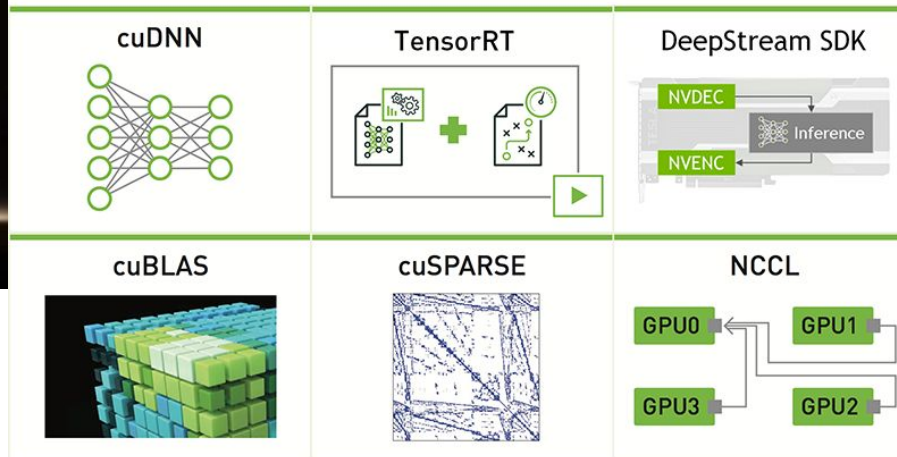
Une hiérarchie d'outils

Calcul matriciel

Impl. de BLAS

CUDA

CuDNN

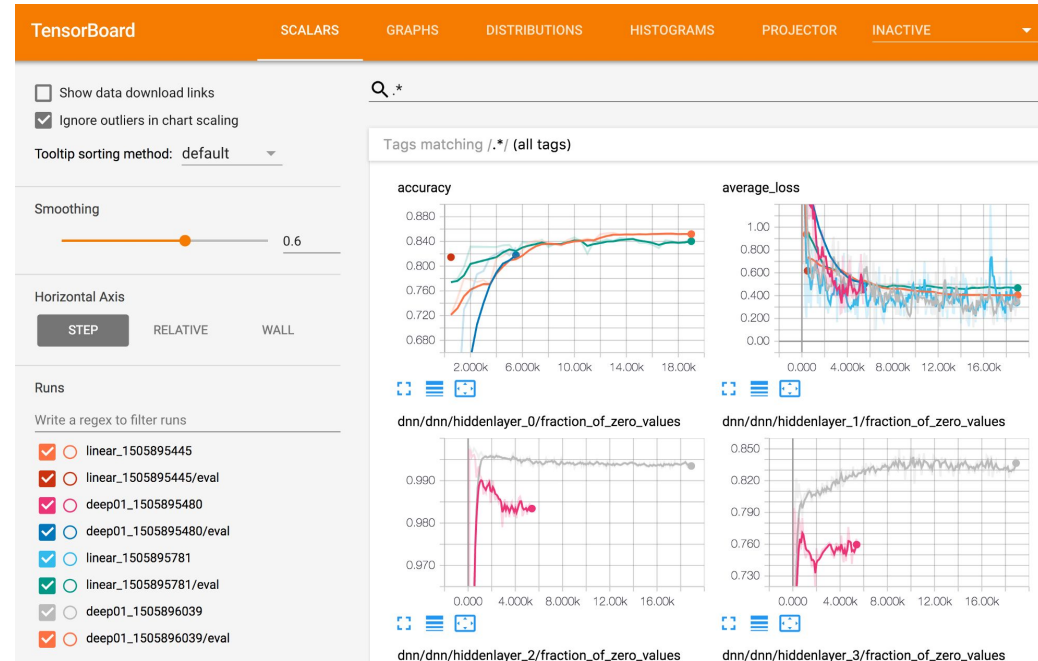
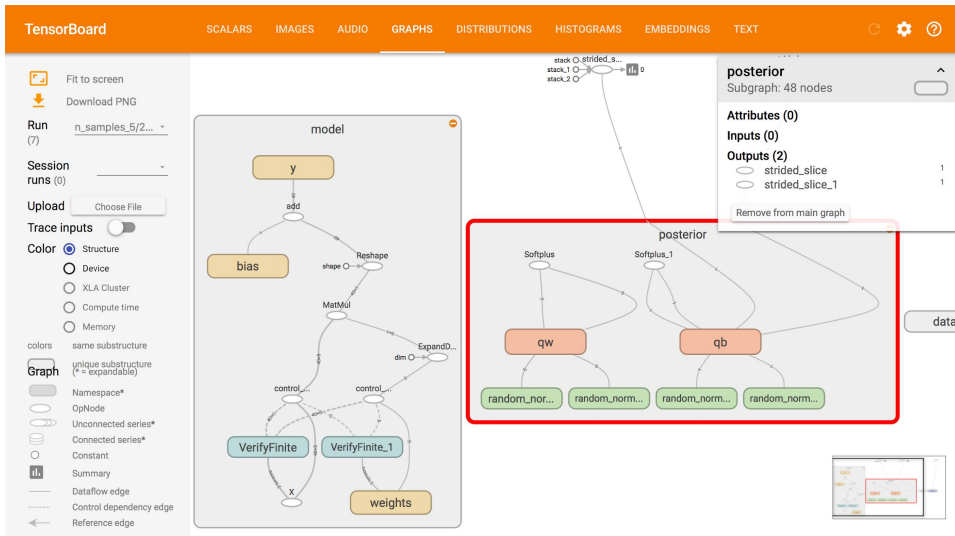


Une hiérarchie d'outils

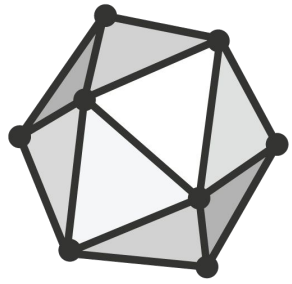
Programmation
différentielle

Visualisation

La visualisation est importante pour diagnostiquer les problèmes d'apprentissage.



Interopérabilité et Model Zoo



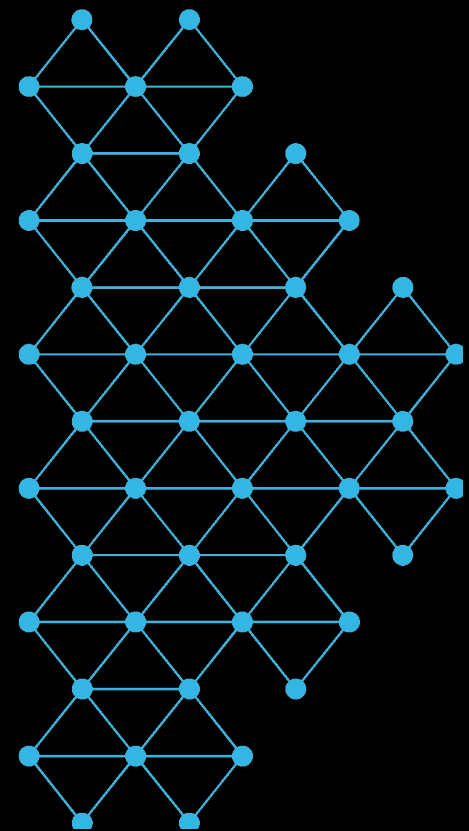
ONNX



Motivation pour Pytorch

- Courbe d'apprentissage faible
- Grande flexibilité
- Bonnes performances
- Communauté active

PYTORCH



Installation Party

Plan



- Suivre les instructions: <https://goo.gl/Z9nSBB>
- Voir la doc: <http://pytorch.org/docs/master/>
- Ouvrir les tutoriaux et exécuter la première ligne

Plan de secours



- Suivre les instructions: <https://goo.gl/Z9nSBB>
(Récupérer les tutoriaux à partir du Github)
- Ouvrir Google Colab: <https://colab.research.google.com>
- Doc: <http://pytorch.org/docs/master/>
- Ouvrir les tutoriaux et exécuter la première ligne