



Bienvenue!

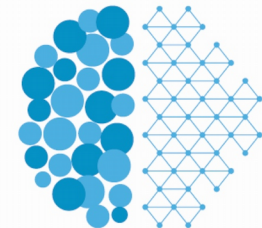
**ÉCOLE D'ÉTÉ FRANCOPHONE
EN APPRENTISSAGE PROFOND**

21-25 août 2017



IVADO

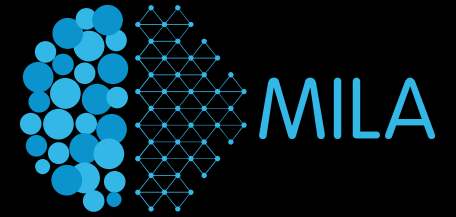
HEC Montréal
Polytechnique Montréal
Université de Montréal



MILA

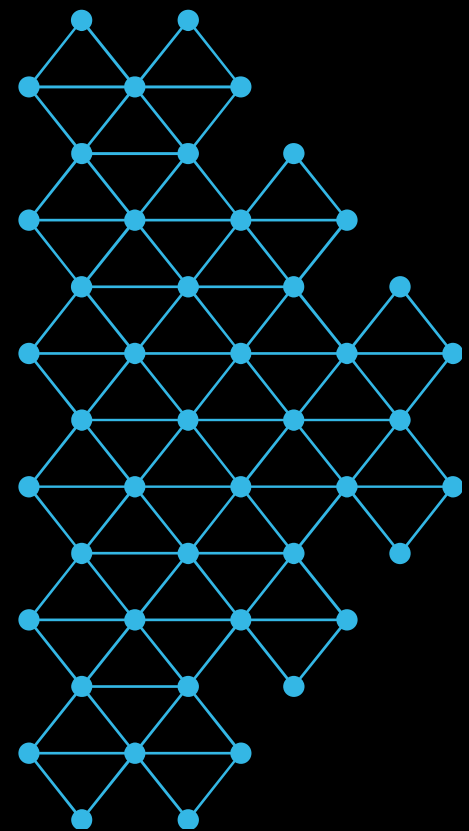


Institut
des algorithmes
d'apprentissage
de Montréal



En pratique : bibliothèques

Gaétan Marceau Caron



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

Les raisons pour utiliser une bibliothèque existante

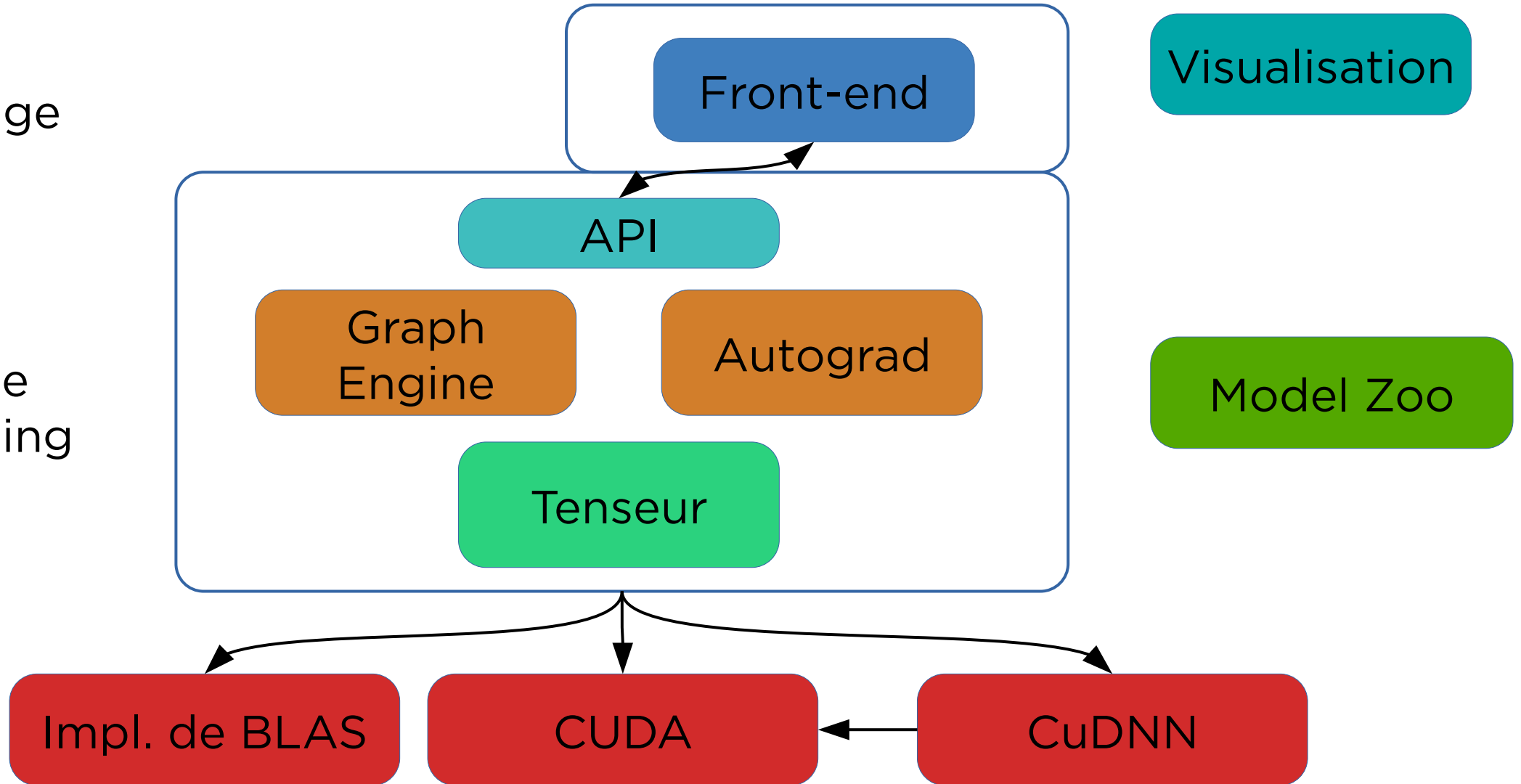
- Une hiérarchie d'outils
- Utilisation de matériel de calcul spécifique
- Support important à la recherche
- Passage de la recherche à la production
- Support de la communauté d'utilisateurs

Une hiérarchie d'outils

Modèles et apprentissage

Bibliothèque Deep Learning

Calcul matriciel



Une hiérarchie d'outils

Modèles et
apprentissage



Front-end

Une hiérarchie d'outils

Modèles et
apprentissage

Front-end

Script
d'apprentissage
Pour ConvNet

Exemple Lasagne

<https://github.com/Lasagne/Lasagne>

Exemple

```
import lasagne
import theano
import theano.tensor as T

# create Theano variables for input and target minibatch
input_var = T.tensor4('X')
target_var = T.ivector('y')

# create a small convolutional neural network
from lasagne.nonlinearities import leaky_rectify, softmax
network = lasagne.layers.InputLayer((None, 3, 32, 32), input_var)
network = lasagne.layers.Conv2DLayer(network, 64, (3, 3),
                                     nonlinearity=leaky_rectify)
network = lasagne.layers.Conv2DLayer(network, 32, (3, 3),
                                     nonlinearity=leaky_rectify)
network = lasagne.layers.Pool2DLayer(network, (3, 3), stride=2, mode='max')
network = lasagne.layers.DenseLayer(lasagne.layers.dropout(network, 0.5),
                                    128, nonlinearity=leaky_rectify,
                                    W=lasagne.init.Orthogonal())
network = lasagne.layers.DenseLayer(lasagne.layers.dropout(network, 0.5),
                                    10, nonlinearity=softmax)

# create loss function
prediction = lasagne.layers.get_output(network)
loss = lasagne.objectives.categorical_crossentropy(prediction, target_var)
loss = loss.mean() + 1e-4 * lasagne.regularization.regularize_network_params(
    network, lasagne.regularization.l2)

# create parameter update expressions
params = lasagne.layers.get_all_params(network, trainable=True)
updates = lasagne.updates.nesterov_momentum(loss, params, learning_rate=0.01,
                                             momentum=0.9)

# compile training function that updates parameters and returns training loss
train_fn = theano.function([input_var, target_var], loss, updates=updates)

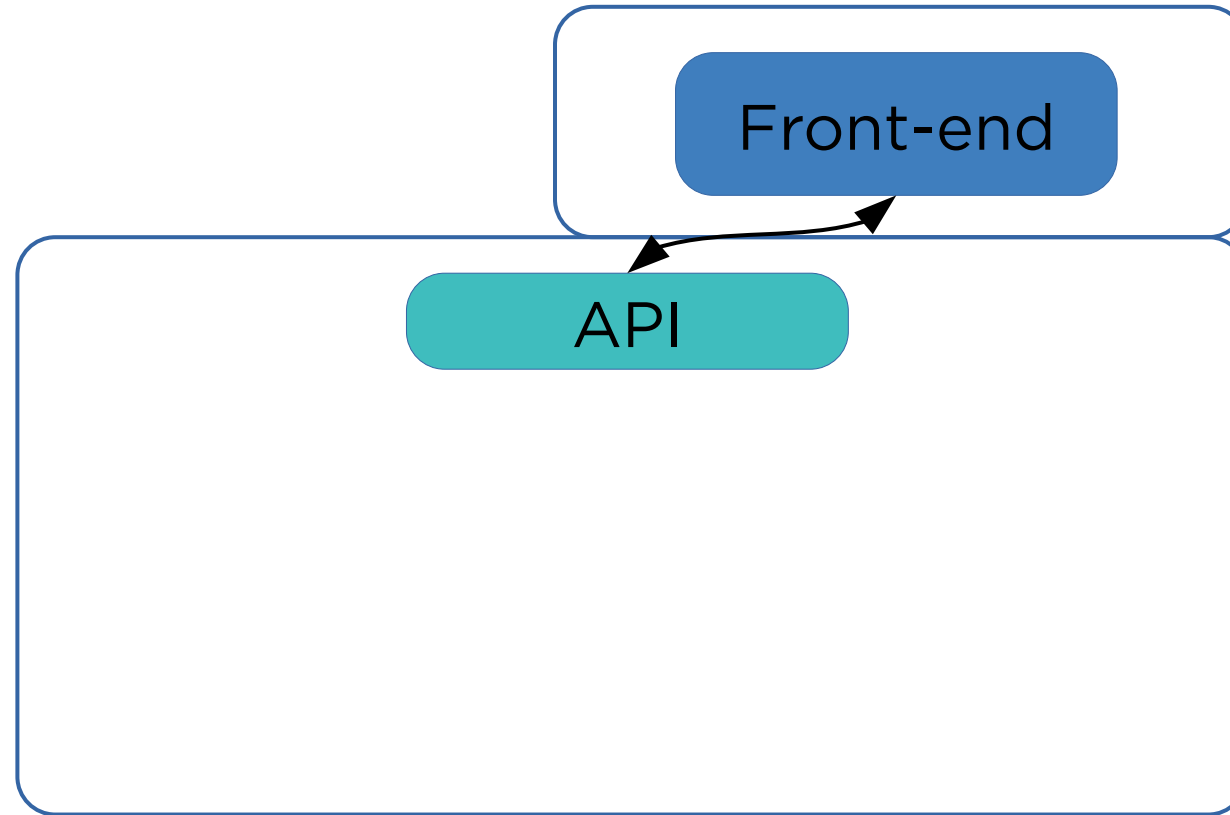
# train network (assuming you've got some training data in numpy arrays)
for epoch in range(100):
    loss = 0
    for input_batch, target_batch in training_data:
        loss += train_fn(input_batch, target_batch)
    print("Epoch %d: Loss %g" % (epoch + 1, loss / len(training_data)))

# use trained network for predictions
test_prediction = lasagne.layers.get_output(network, deterministic=True)
predict_fn = theano.function([input_var], T.argmax(test_prediction, axis=1))
print("Predicted class for first test input: %i" % predict_fn(test_data[0]))
```

Une hiérarchie d'outils

Modèles et
apprentissage

Bibliothèque
Deep Learning



Une hiérarchie d'outils

API

Définition de la
régression
logistique en
Theano

```
# Construct Theano expression graph
p_1 = 1 / (1 + T.exp(-T.dot(x, w) - b)) # Probability that target = 1
prediction = p_1 > 0.5 # The prediction thresholded
xent = -y * T.log(p_1) - (1-y) * T.log(1-p_1) # Cross-entropy loss function
cost = xent.mean() + 0.01 * (w ** 2).sum() # The cost to minimize
gw, gb = T.grad(cost, [w, b]) # Compute the gradient of the cost
# w.r.t weight vector w and
# bias term b
# (we shall return to this in a
# following section of this tutorial)

# Compile
train = theano.function(
    inputs=[x,y],
    outputs=[prediction, xent],
    updates=((w, w - 0.1 * gw), (b, b - 0.1 * gb)))
predict = theano.function(inputs=[x], outputs=prediction)
```

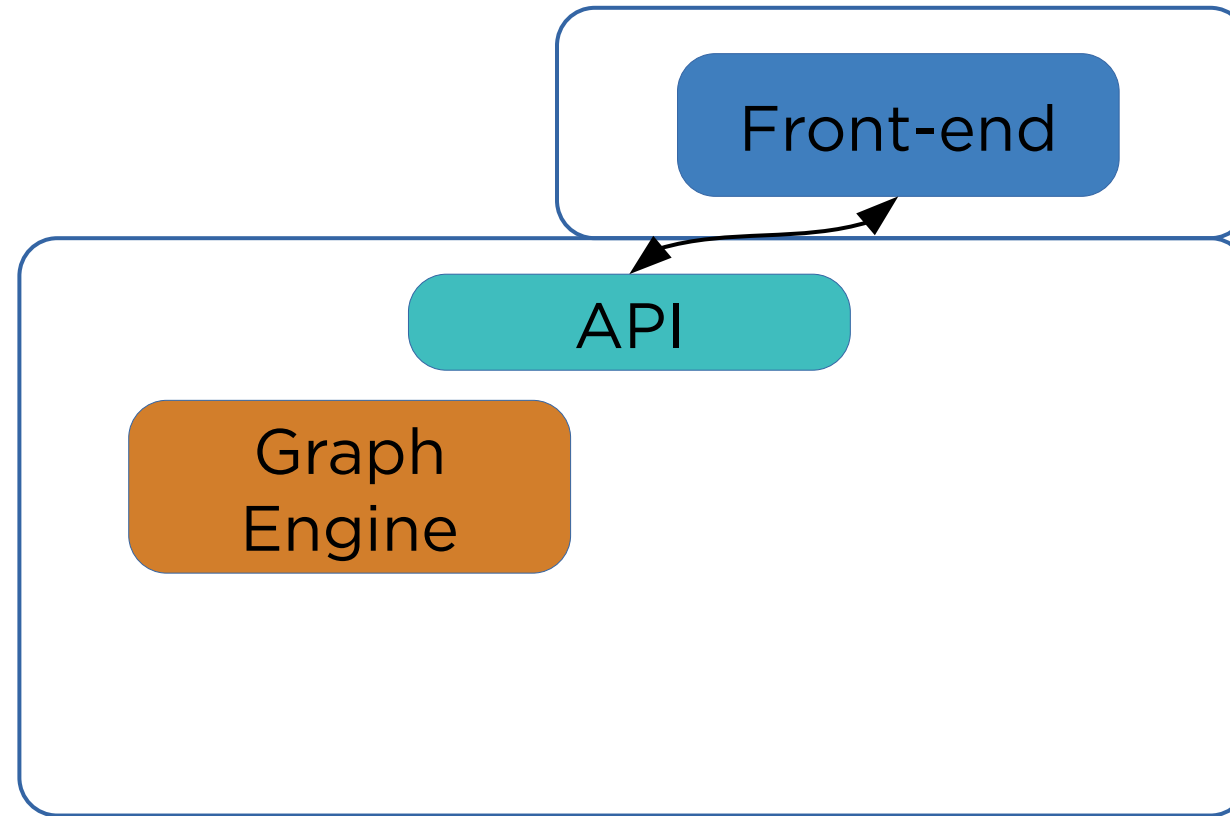
Tutoriel Theano

<http://deeplearning.net/software/theano/tutorial/examples.html#logistic-function>

Une hiérarchie d'outils

Modèles et
apprentissage

Bibliothèque
Deep Learning



L'approche modulaire

- Concept central: le graphe computationnel



Google Pipes Datacenter | by Jorge Jorquera

<https://creativecommons.org/licenses/by-nc-nd/2.0/>

L'approche modulaire

Définition d'un module

1. Spécification des entrées
2. Spécification des opérations



L'approche modulaire

Définition d'un module

1. Spécification des entrées



Image RGB HxL pixels

L'approche modulaire

Définition d'un module

2. Spécification des opérations

2.1 Définition des paramètres

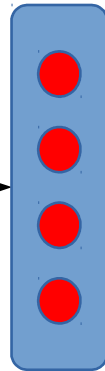


Image RGB HxL pixels

L'approche modulaire

Définition d'un module

2. Spécification des opérations

2.1 Définition des paramètres

2.2 Définition du forward

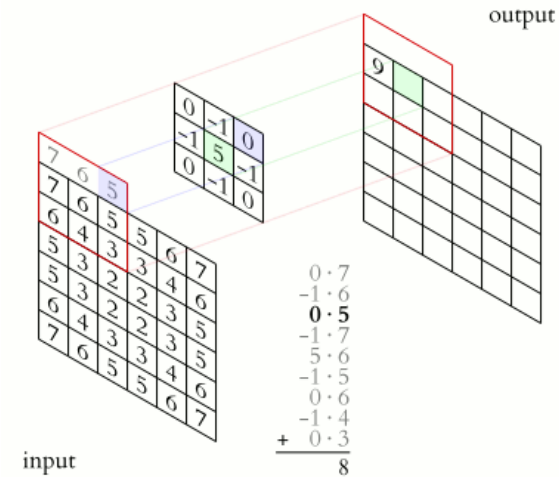
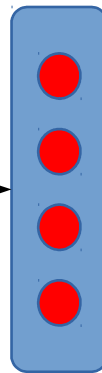


Image RGB HxL pixels



Attention à l'initialisation!

L'approche modulaire

- Définition d'un module
- 2. Spécification des opérations
 - 2.1 Définition des paramètres
 - 2.2 Définition du forward
 - 2.3 Définition du backward / mise à jour des paramètres

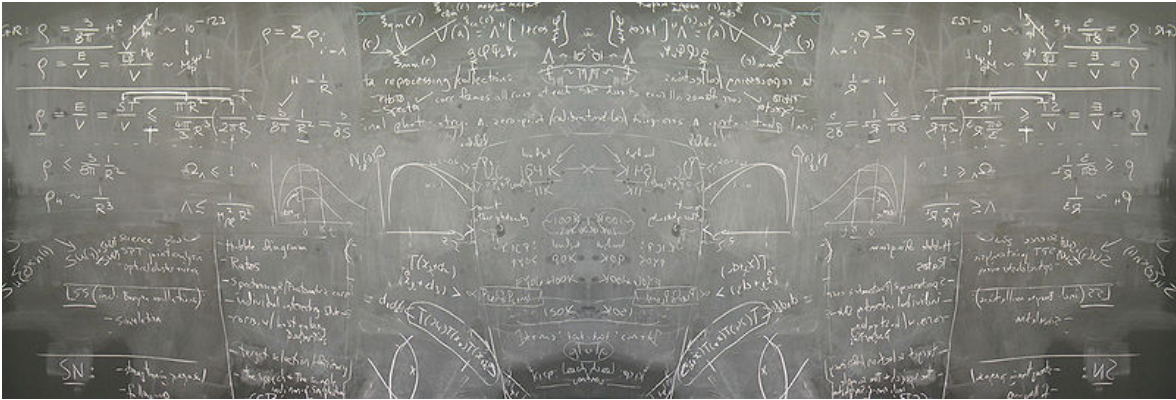
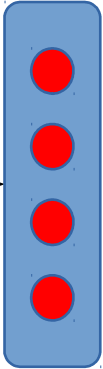
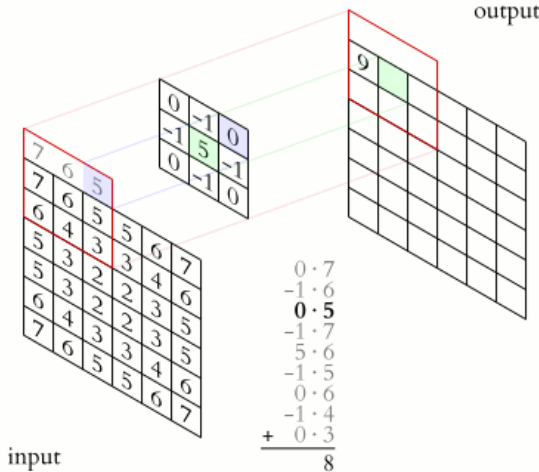


Image RGB HxL pixels

L'approche modulaire

Définition d'un module

2. Spécification des opérations

2.1 Définition des paramètres

2.2 Définition du forward

~~2.3 Définition du backward /
mise à jour des paramètres~~

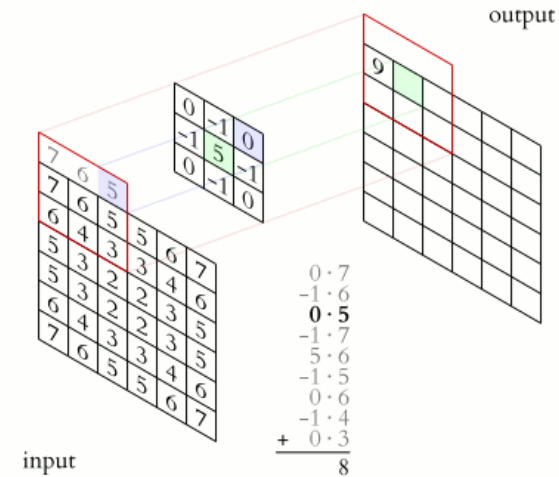
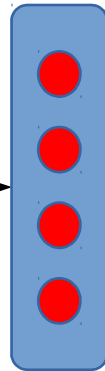
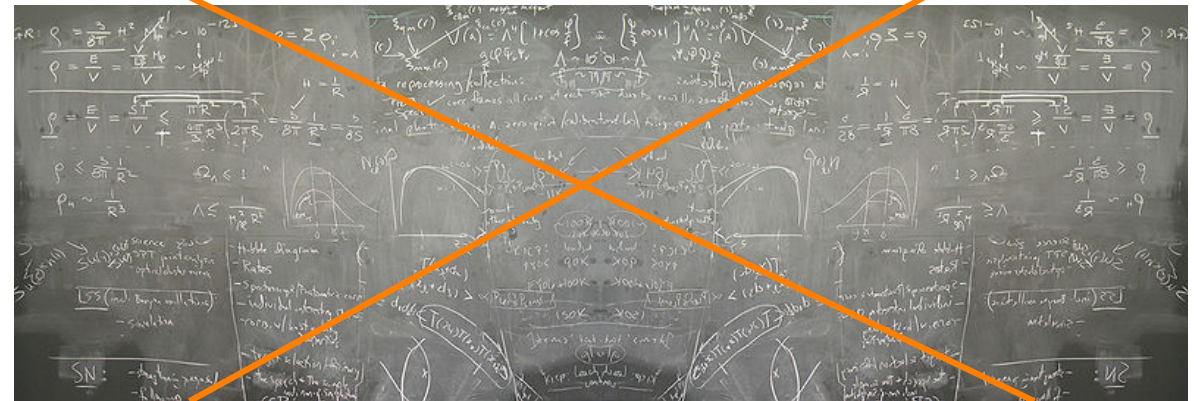


Image RGB HxL pixels



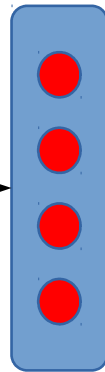
Différentiation automatique



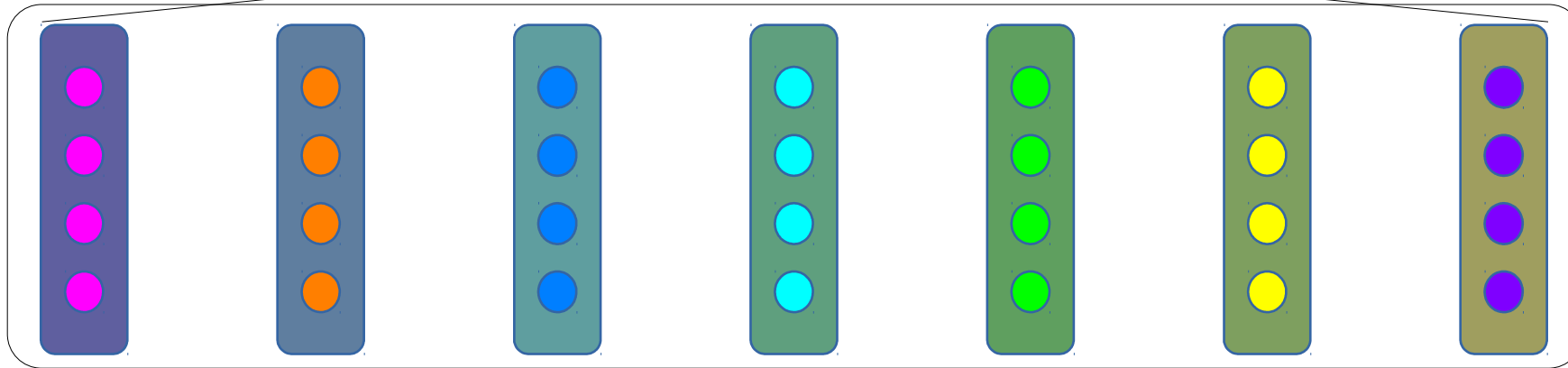
L'approche modulaire

Définition d'un module

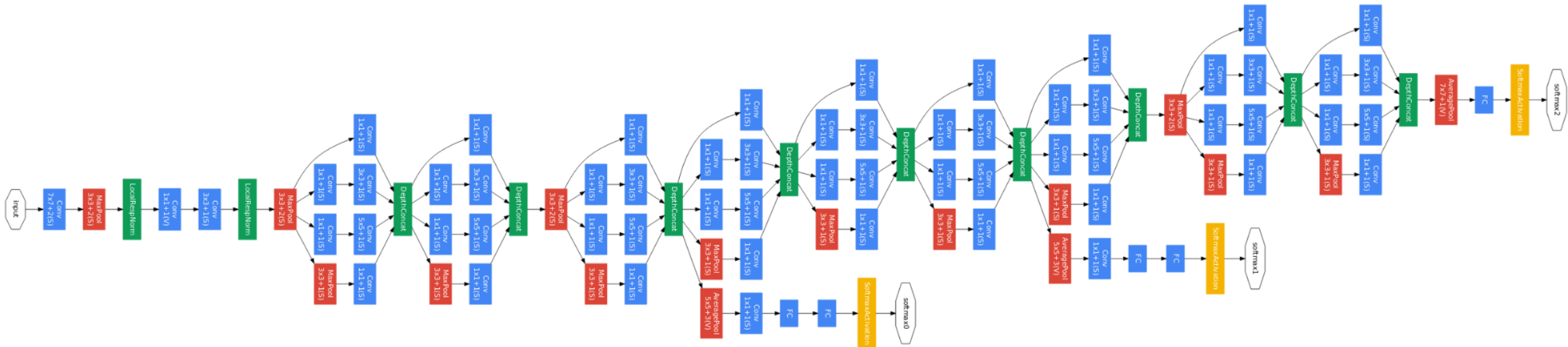
1. Spécification des entrées
2. Spécification des opérations



Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%



L'approche modulaire: exemple

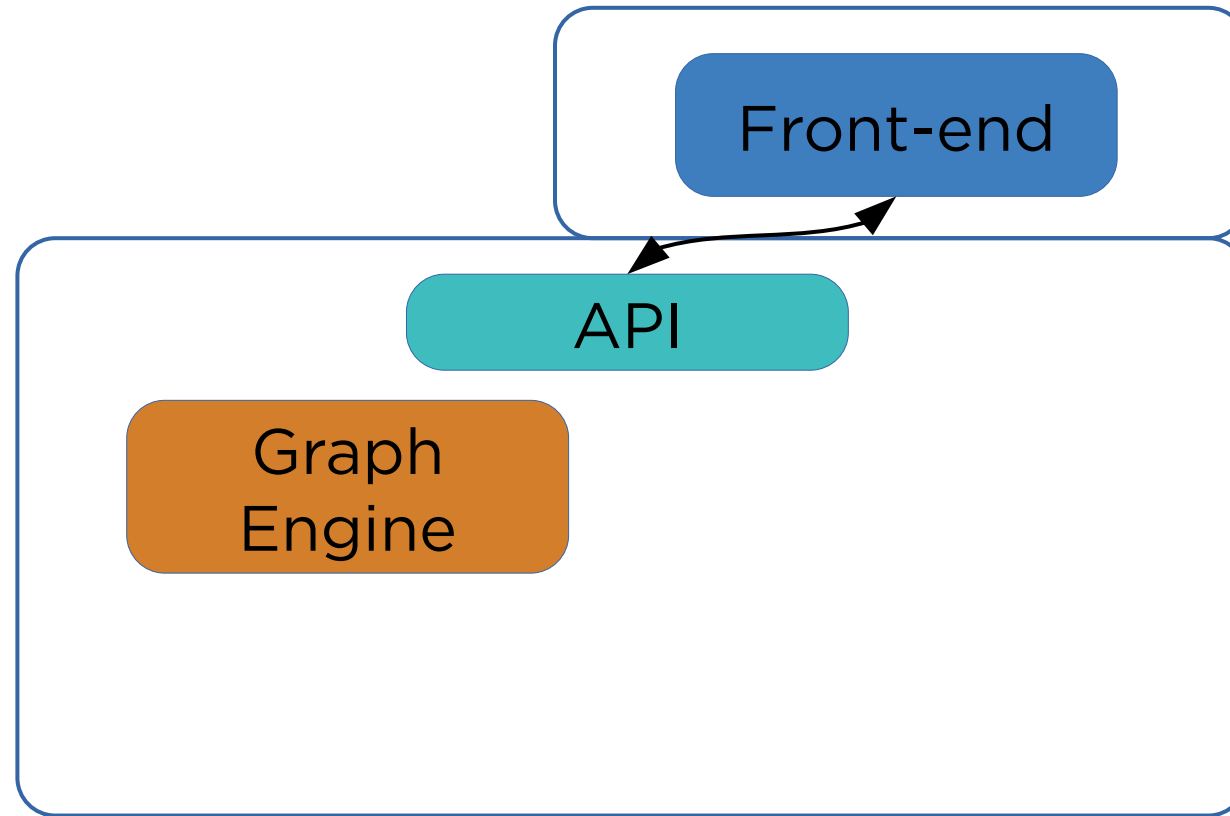


Szegedy, Christian, et al. "Going deeper with convolutions."
Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

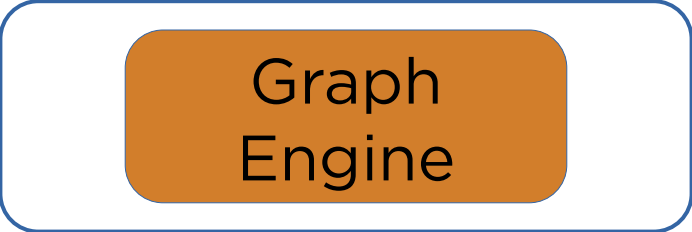
Une hiérarchie d'outils

Modèles et
apprentissage

Bibliothèque
Deep Learning



Une hiérarchie d'outils

A blue rounded rectangle containing an orange rounded rectangle with the text "Graph Engine" in white.

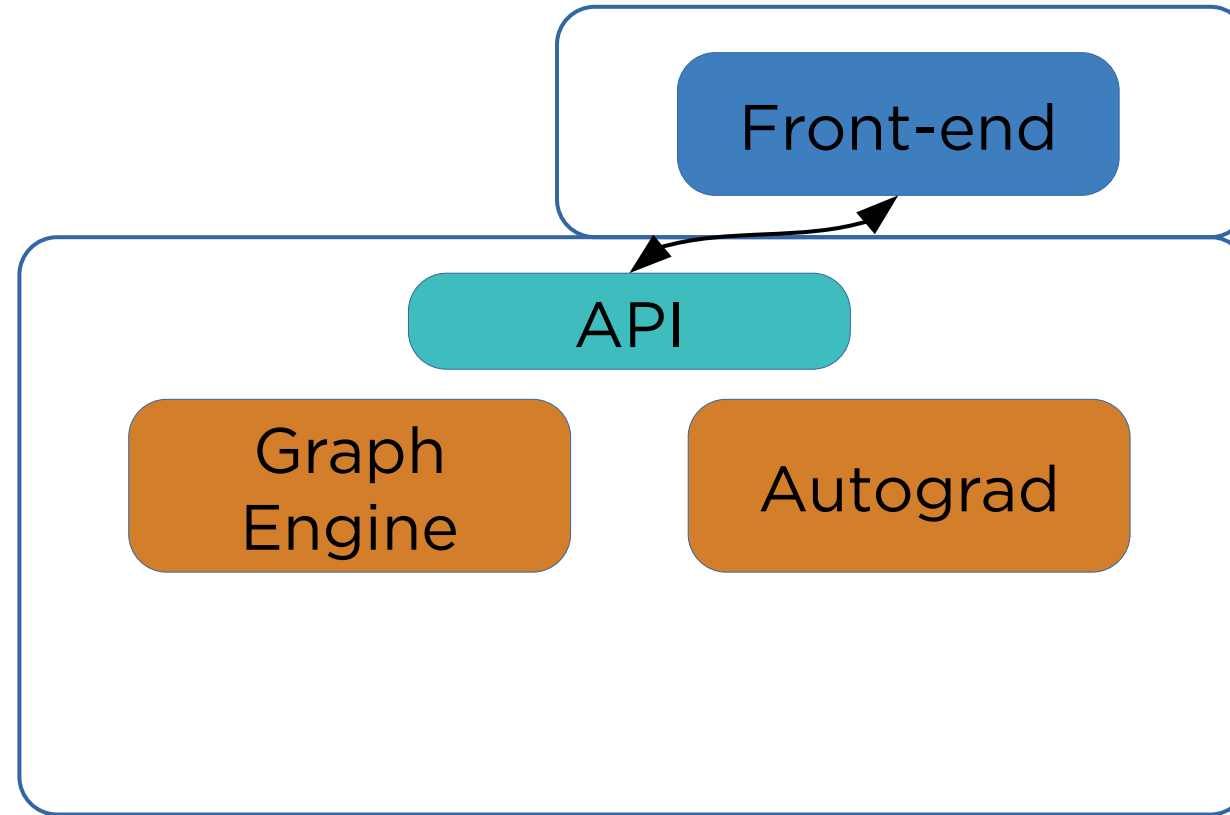
Graph
Engine

- Construit et optimise le graphe computationnel
- Optimise en terme de vitesse et de stockage
- Gère les instabilités numériques
- Permet la différentiation automatique (e.g. backprop)

Une hiérarchie d'outils

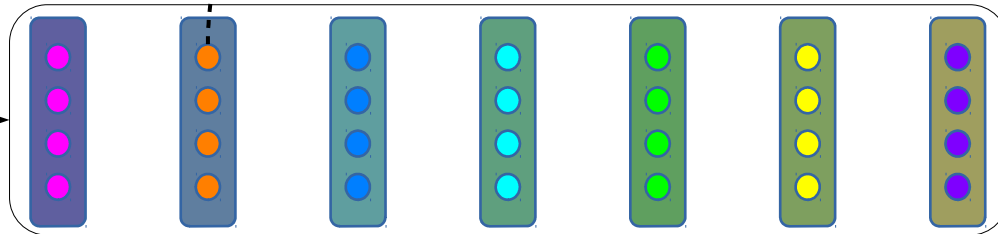
Modèles et
apprentissage

Bibliothèque
Deep Learning



Différentiation automatique

Quel est l'effet
d'une perturbation
de ce paramètre
sur le coût ?



Coût: 0.2

Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%

Différentiation automatique

Quel est l'effet
d'une perturbation
de ce paramètre
sur le coût ?

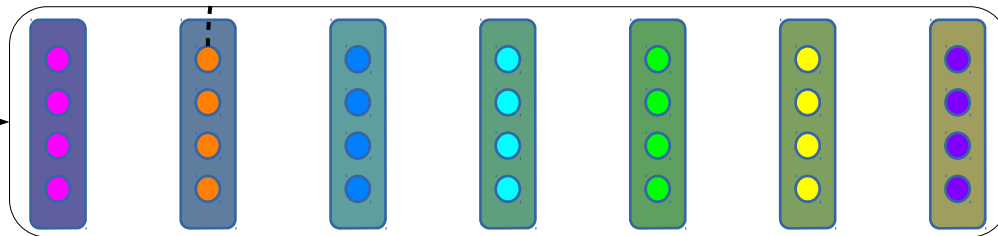
Leibniz
(1676)



Utiliser ma règle
de chaînage!

Coût: 0.2

Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%



Différentiation automatique

For concreteness, consider the function

$$y = e^{\sin x^2}.$$

This can be decomposed as the composite of three functions:

$$y = f(u) = e^u,$$

$$u = g(v) = \sin v,$$

$$v = h(x) = x^2.$$

Their derivatives are:

$$\frac{dy}{du} = f'(u) = e^u,$$

$$\frac{du}{dv} = g'(v) = \cos v,$$

$$\frac{dv}{dx} = h'(x) = 2x.$$

In Leibniz notation, this is:

$$\frac{dy}{dx} = \frac{dy}{du} \Big|_{u=g(h(a))} \cdot \frac{du}{dv} \Big|_{v=h(a)} \cdot \frac{dv}{dx} \Big|_{x=a},$$

or for short,

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dv} \cdot \frac{dv}{dx}.$$

The derivative function is therefore:

$$\frac{dy}{dx} = e^{\sin x^2} \cdot \cos x^2 \cdot 2x.$$

https://en.wikipedia.org/wiki/Chain_rule#Composites_of_more_than_two_functions

Différentiation automatique

For concreteness, consider the function

$$y = e^{\sin x^2}.$$

This can be decomposed as the composite of three functions:

$$y = f(u) = e^u,$$

$$u = g(v) = \sin v,$$

$$v = h(x) = x^2.$$

Their derivatives are:

$$\frac{dy}{du} = f'(u) = e^u,$$

$$\frac{du}{dv} = g'(v) = \cos v,$$

$$\frac{dv}{dx} = h'(x) = 2x.$$

https://en.wikipedia.org/wiki/Chain_rule#Composites_of_more_than_two_functions

In Leibniz notation, this is:

$$\frac{dy}{dx} = \frac{dy}{du} \Big|_{u=g(h(a))} \cdot \frac{du}{dv} \Big|_{v=h(a)} \cdot \frac{dv}{dx} \Big|_{x=a},$$

or for short,

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dv} \cdot \frac{dv}{dx}.$$

The derivative function is therefore:

$$\frac{dy}{dx} = e^{\sin x^2} \cdot \cos x^2 \cdot 2x.$$

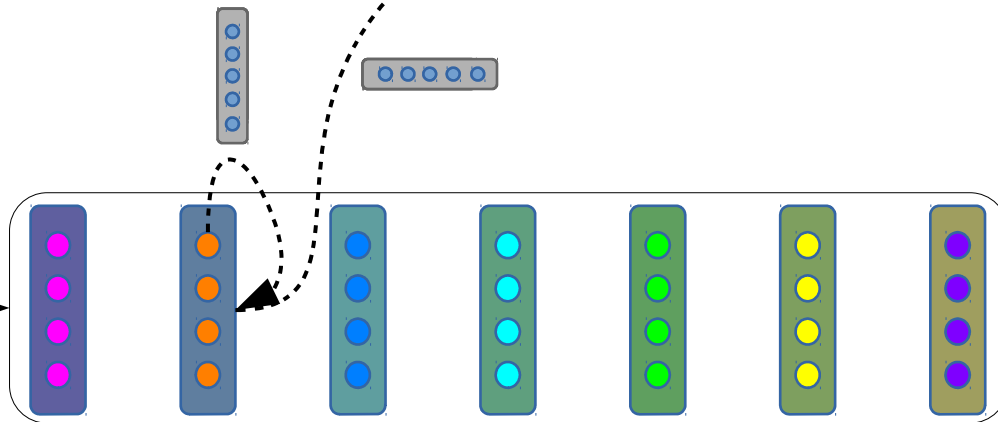
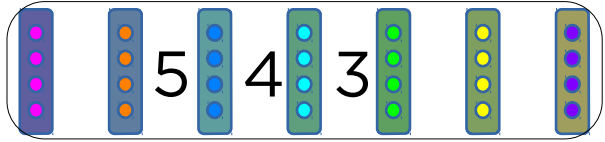
Leibniz
(1676)



En exercice!

Différentiation automatique

Nombre de sorties

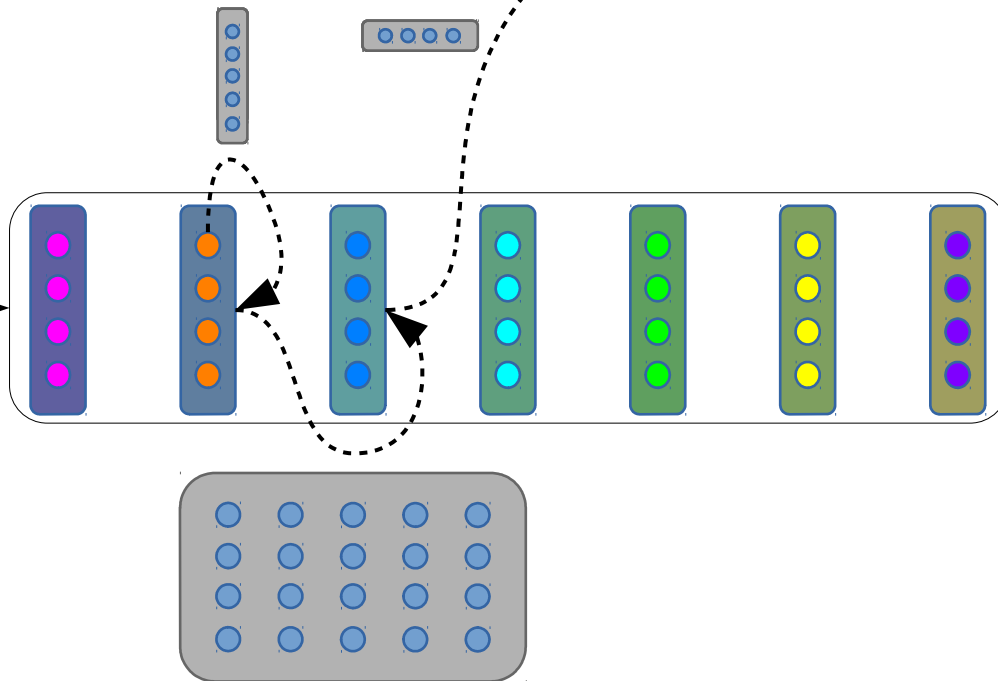
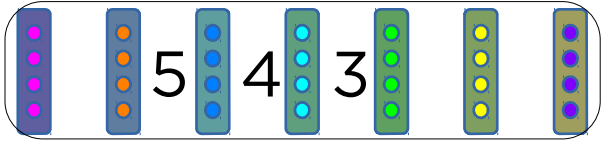


Coût: 0.2

Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%

Différentiation automatique

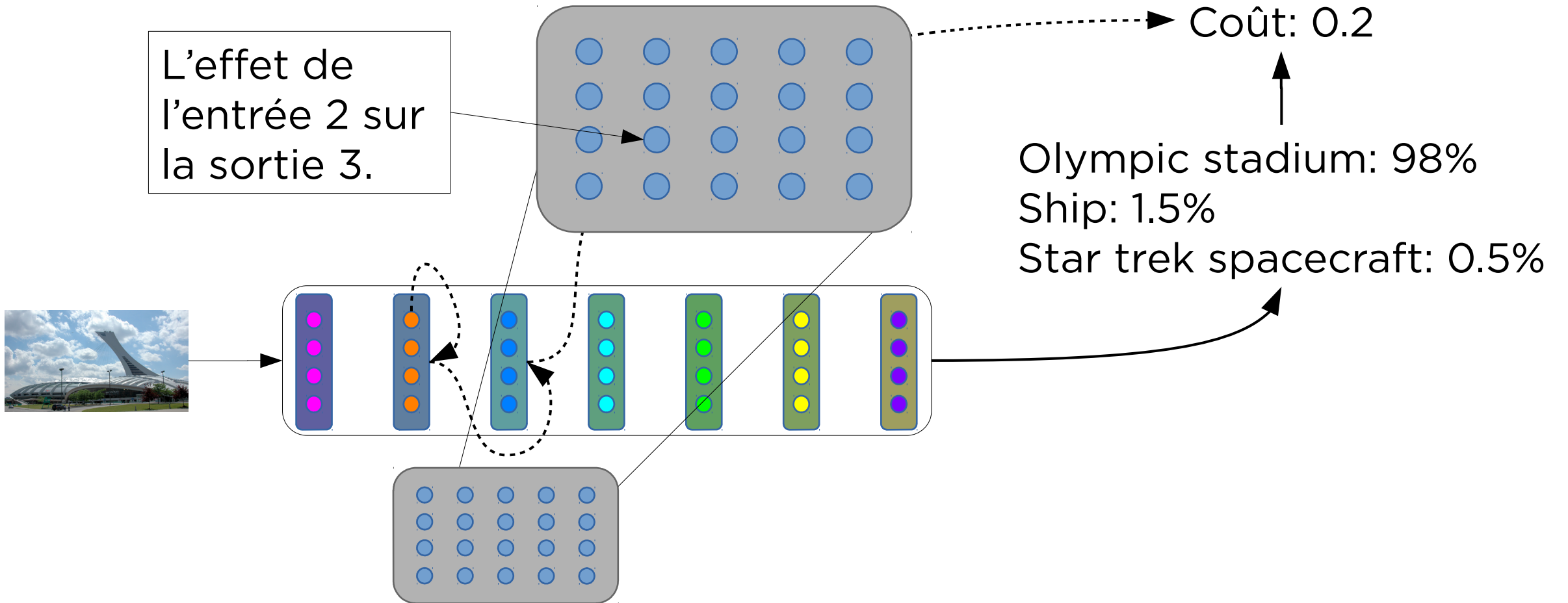
Nombre de sorties



Coût: 0.2

Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%

Différentiation automatique



Différentiation automatique

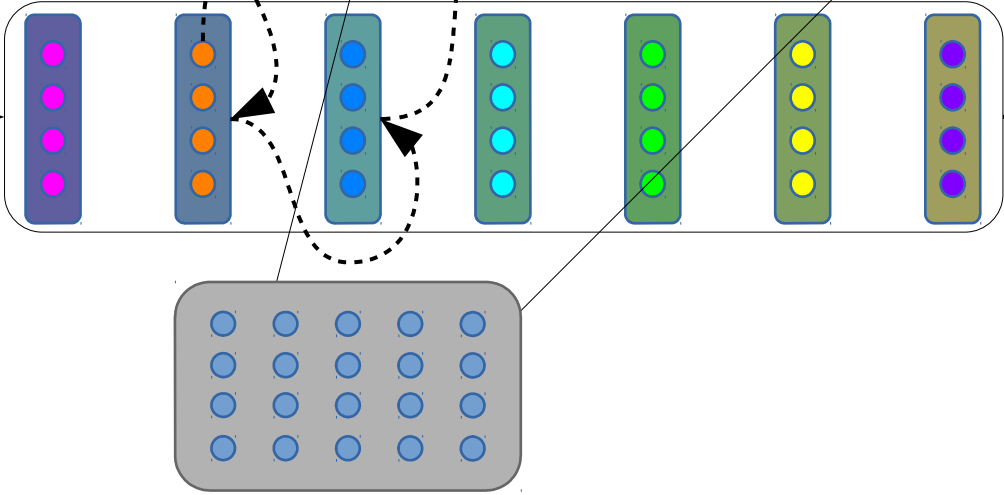
L'effet de l'entrée 2 sur la sortie 3.

Matrice Jacobienne

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

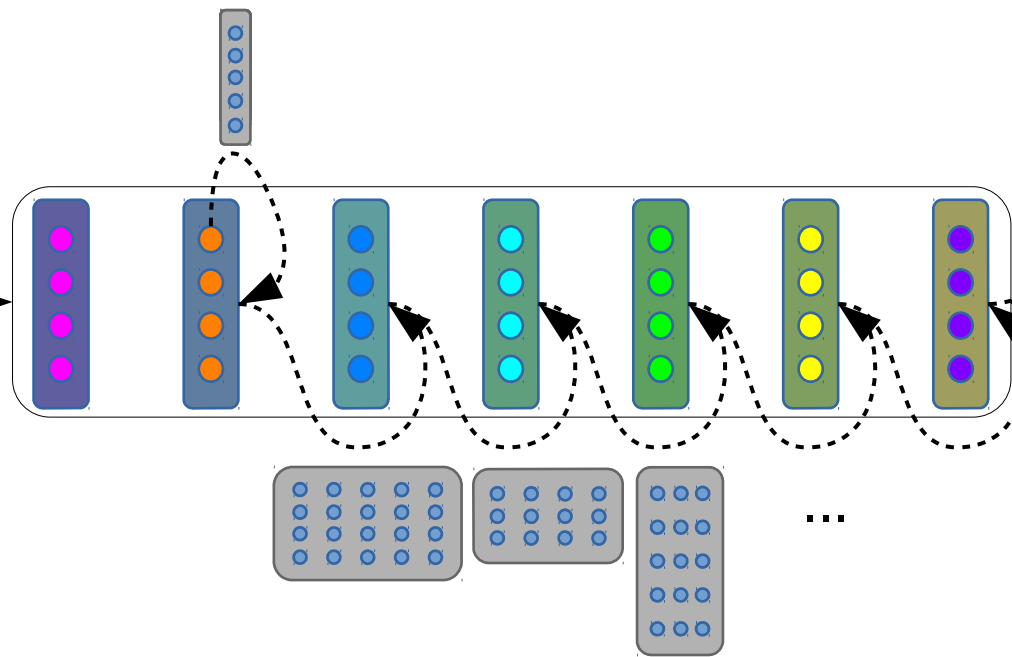
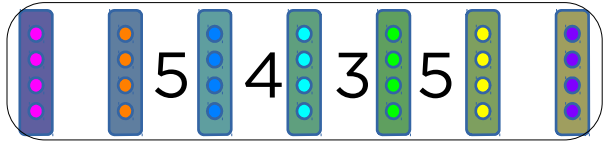
Coût: 0.2

Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%



Différentiation automatique

Nombre de sorties

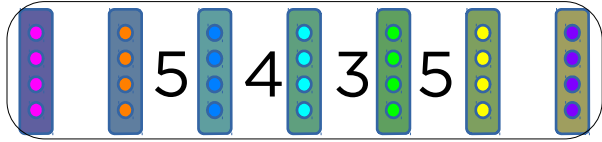


Coût: 0.2

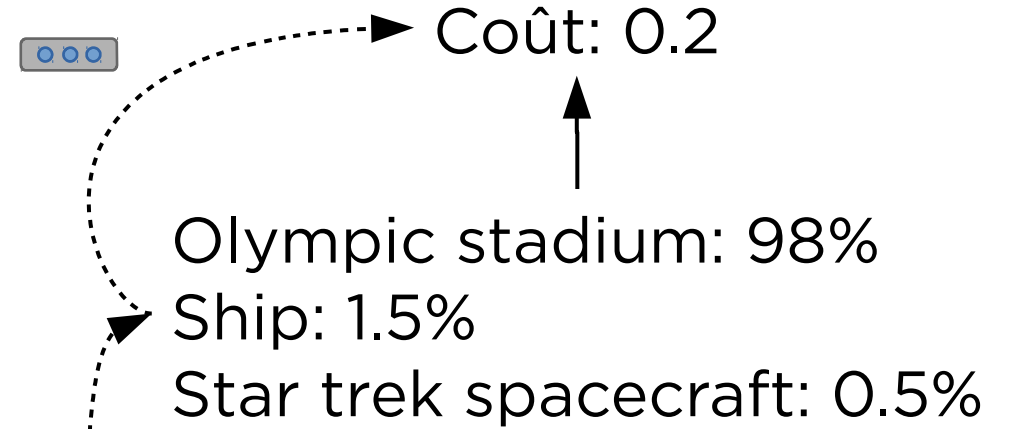
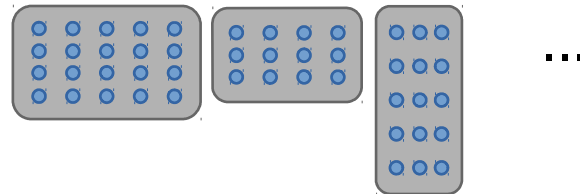
Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%

Différentiation automatique

Nombre de sorties



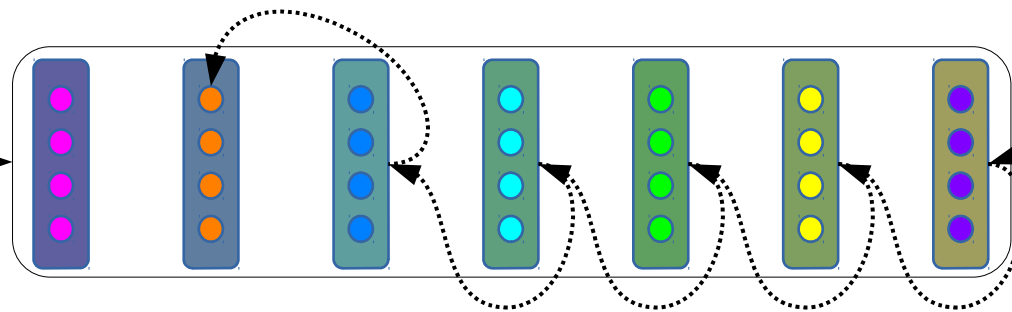
L'ordre des multiplications est important!
Exercice: démontrer sur un exemple simple



Différentiation automatique

Quel est le bon ordre pour calculer la règle de chaînage ?

Backpropagation:
de la sortie vers l'entrée
Implique que des produits matrice-vecteur



Coût: 0.2
Olympic stadium: 98%
Ship: 1.5%
Star trek spacecraft: 0.5%

On généralise ce principe à toutes opérations mathématiques!

Une hiérarchie d'outils



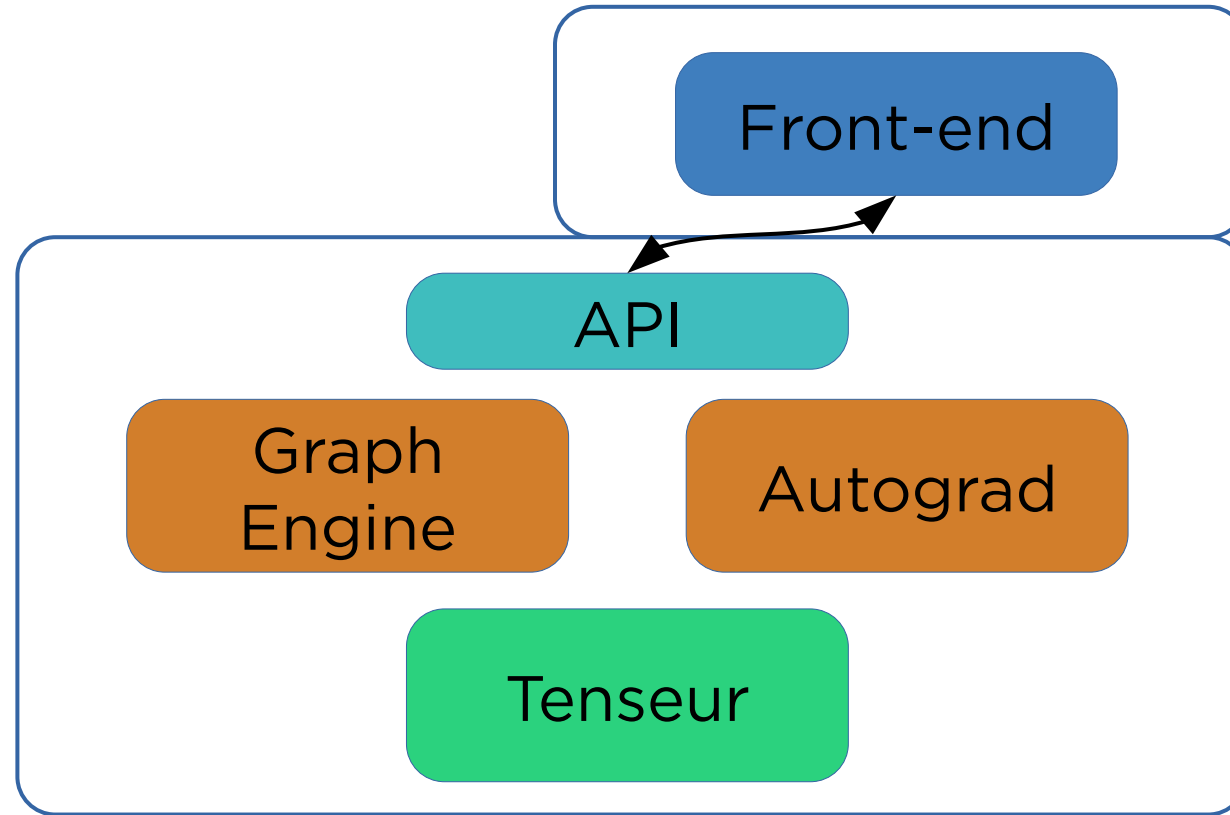
Autograd

- Technologie existante en calcul scientifique
- Permet le calcul automatique du gradient de la fonction de coût en fonction des paramètres
- Ne dit pas ce que l'on doit faire avec le gradient (rôle de l'optimiseur)

Une hiérarchie d'outils

Modèles et
apprentissage

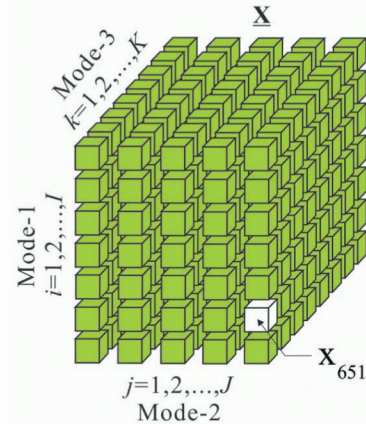
Bibliothèque
Deep Learning



Une hiérarchie d'outils

Tenseur 3D

Tenseur



Associer un sens
aux dimensions

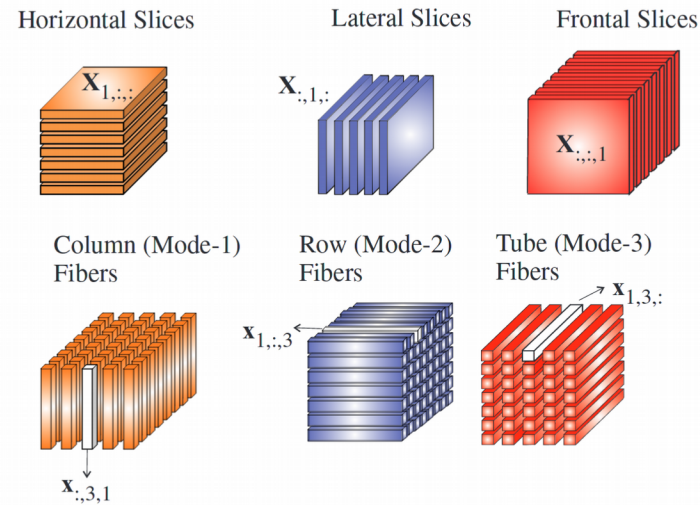


Image from
Cichocki, Andrzej.
"Era of big data processing:
A new approach via tensor networks
and tensor decompositions."
arXiv preprint arXiv:1403.2048 (2014).

Une hiérarchie d'outils



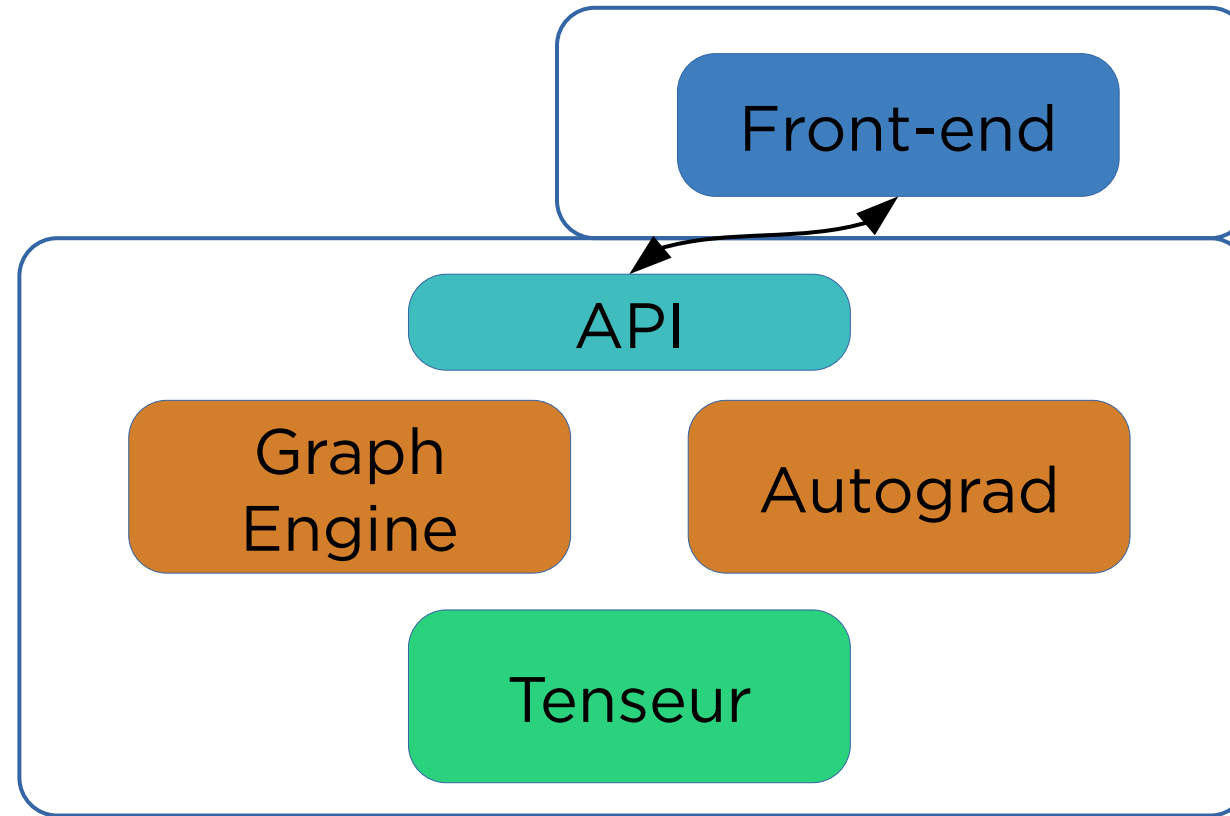
Tenseur

- Généralisation des matrices (3-4-5 dimensions)
- Représentation flexible et commode des données et des calculs
- Permet de gérer efficacement la mémoire
- Implémentation des opérations mathématiques élémentaires (somme, multiplication, produit scalaire, etc.)

Une hiérarchie d'outils

Modèles et
apprentissage

Bibliothèque
Deep Learning



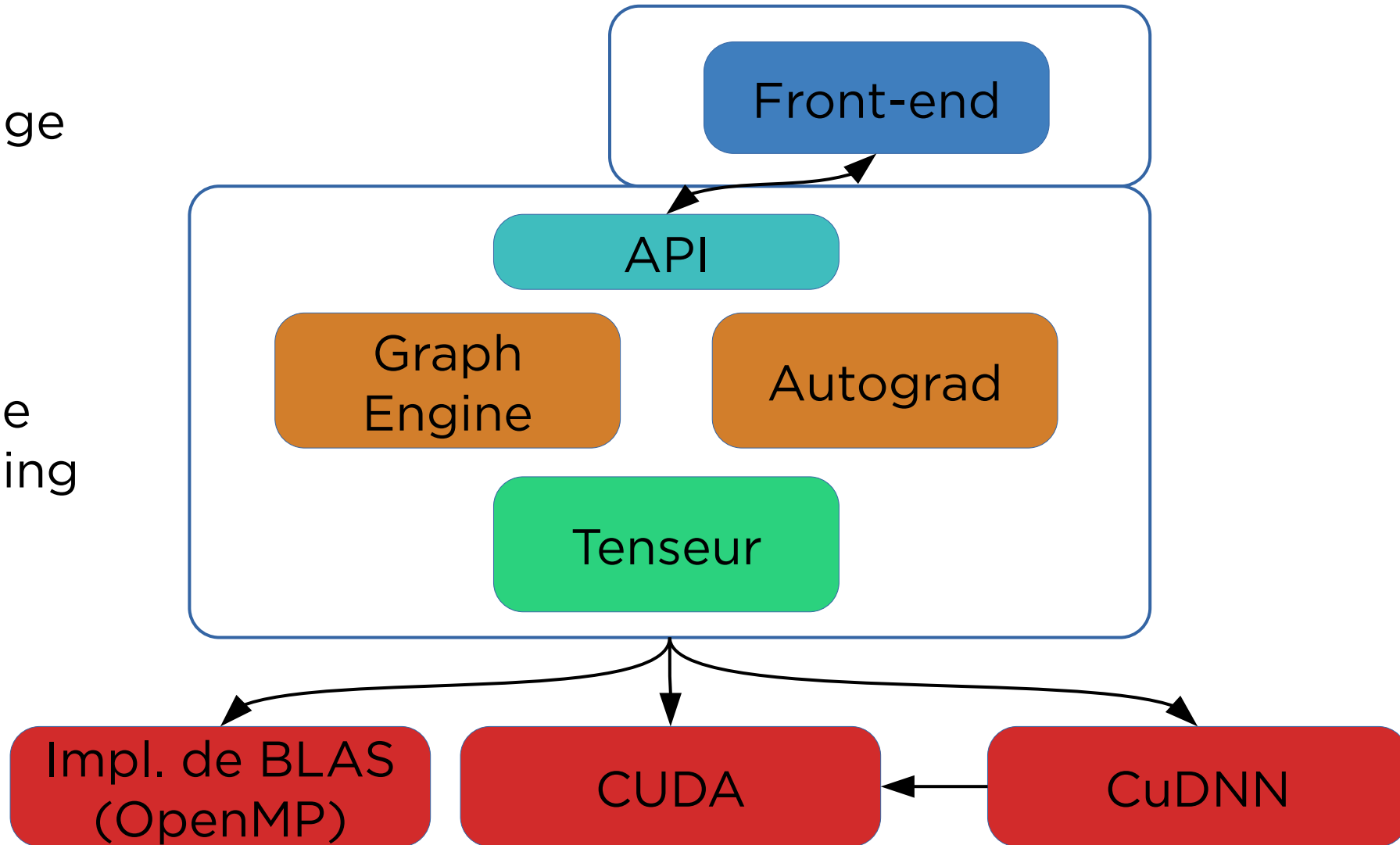
Implémentation effective des calculs ???

Une hiérarchie d'outils

Modèles et
apprentissage

Bibliothèque
Deep Learning

Calcul
matriciel



Une hiérarchie d'outils



BLAS

- BLAS est un standard (interface) de programmation pour le calcul matriciel
- Plusieurs bibliothèques implémentent BLAS (INTEL MKL, OpenBLAS)
- Support au calcul parallèle avec OpenMP

Une hiérarchie d'outils



CUDA

CuDNN

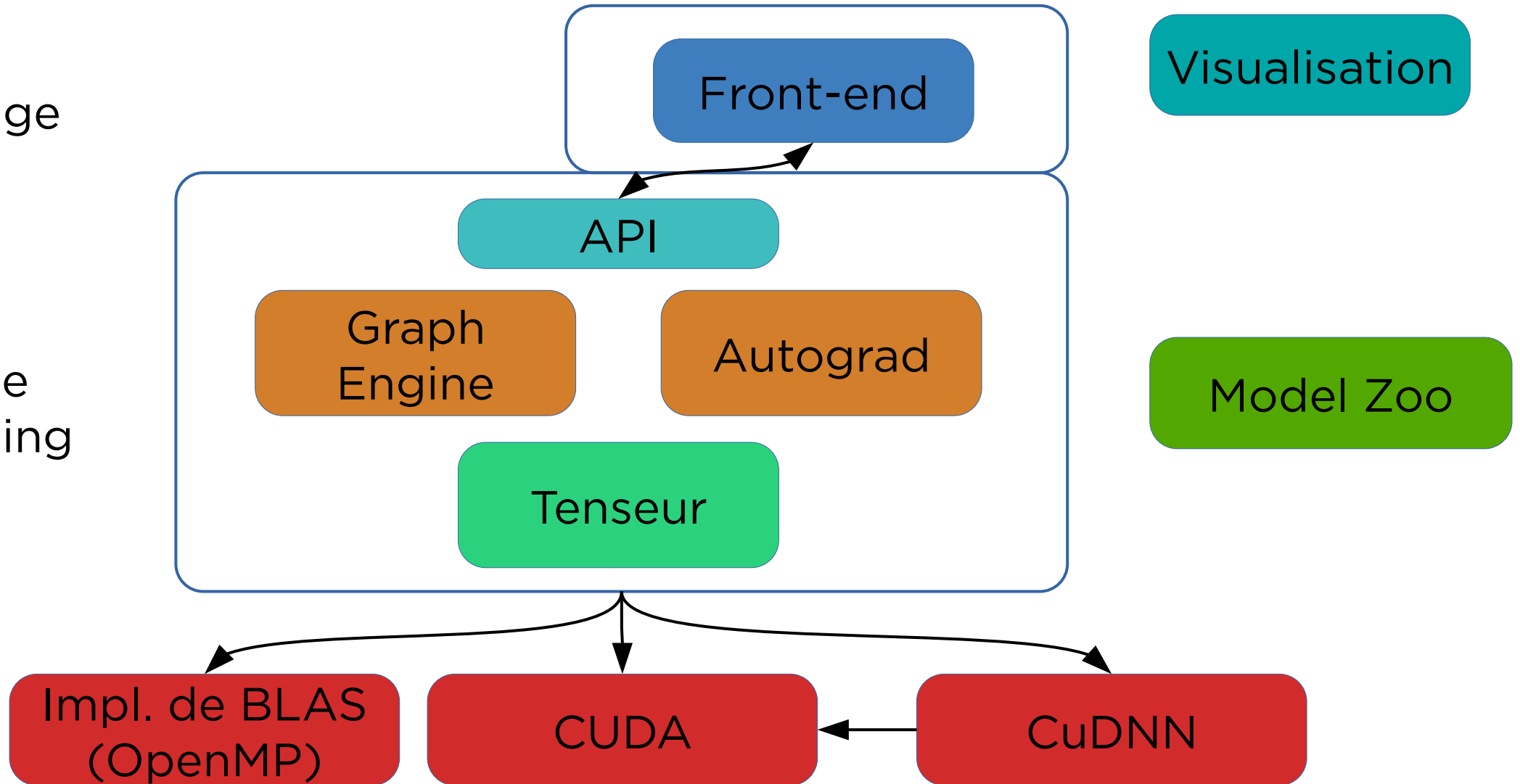
- Calcul scientifique et primitive de deep learning pour les cartes graphiques NVIDIA
- À installer avant l'installation d'une bibliothèque de deep learning
- N'oubliez pas après l'installation
`CUDA_ROOT=/path/to/cuda/root`

Une hiérarchie d'outils

Modèles et apprentissage

Bibliothèque Deep Learning

Calcul matriciel

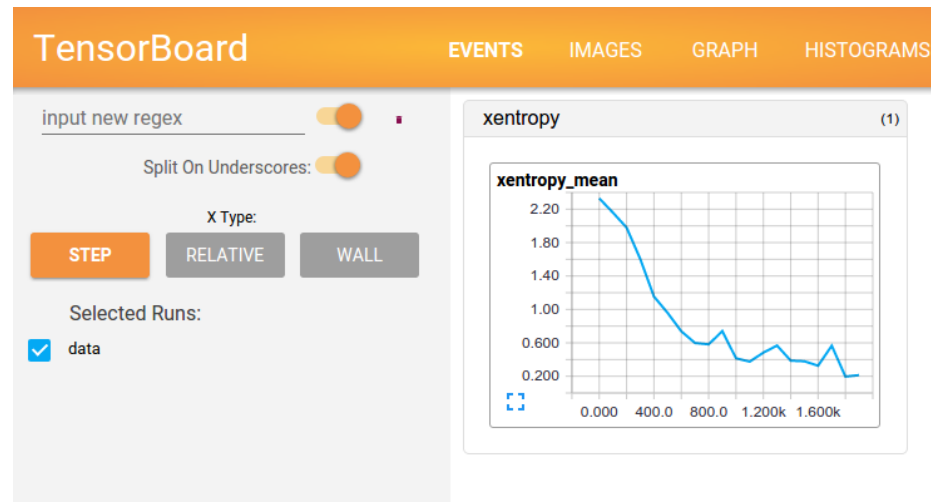


Une hiérarchie d'outils

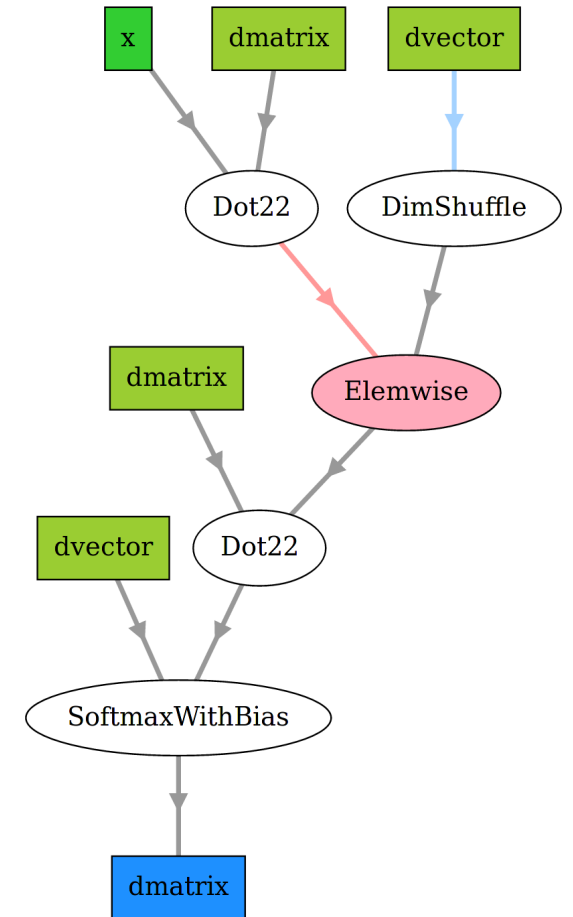
<http://deeplearning.net/software/theano/library/d3viz/index.html>

Visualisation

matplotlib



https://www.tensorflow.org/get_started/summaries_and_tensorboard



Fonctionnalités

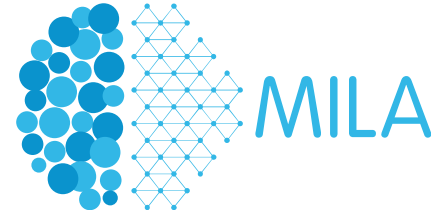
- Support pour CUDA et CuDNN
- Implémentation de modules (e.g. RNN)
- Implémentation d'optimiseurs
- Facilité d'utilisation (e.g. front-end)
- Différenciateur automatique
- Graphe statique ou dynamique
- Vitesse de calculs
- Multi-gpu (éventuellement cluster)
- Model serialization
- Model zoo
- Visualisation
- Taille de la communauté
- Support au FP16
- Licence: ouverte

Les raisons pour utiliser une bibliothèque existante

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

Les bibliothèques existantes

- Theano (Université de Montréal)
- Tensorflow (Google)
- Pytorch et Caffe2 (Facebook)
- CNTK (Microsoft)
- MXNet



Theano

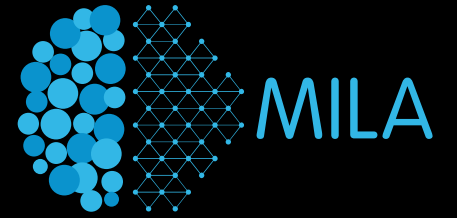
- Projet stable: développé et utilisé depuis janvier 2008 (9 ans)
- Theano 0.9 en mars 2017 (version 0.10 en dev.)
- Associé à plus de 1000 articles scientifiques
- 123 contributeurs pour theano 0.9
- Mailing list avec participants du monde entier
- Utilisé pour l'enseignement universitaire
- Utilisé dans l'industrie
- Theano: deeplearning.net/software/theano/
- Deep Learning Tutorials: deeplearning.net/tutorial

Remerciements

À toutes les personnes qui travaillent ou ont travaillé au MILA (anciennement LISA) et spécialement aux contributeurs de Theano:

Reyhane Askari Hemmat, Frédéric Bastien, Yoshua Bengio, James Bergstra, Arnaud Bergeron, Steven Bocco, Philemon Brakel, Olivier Breuleux, Pierre Luc Carrier, Mathieu Germain, Ian Goodfellow, Pascal Lamblin, Simon Lefrançois, Razvan Pascanu, Joseph Turian, David Warde-Farley, et tous les autres.

Institut
des algorithmes
d'apprentissage
de Montréal



Merci pour votre attention!