

Deep Learning et Intelligence Artificielle : mythes et réalités

Christian Wolf

Codeurs en Seine
24 Novembre, 2016





LIRIS

cnrs

INSA

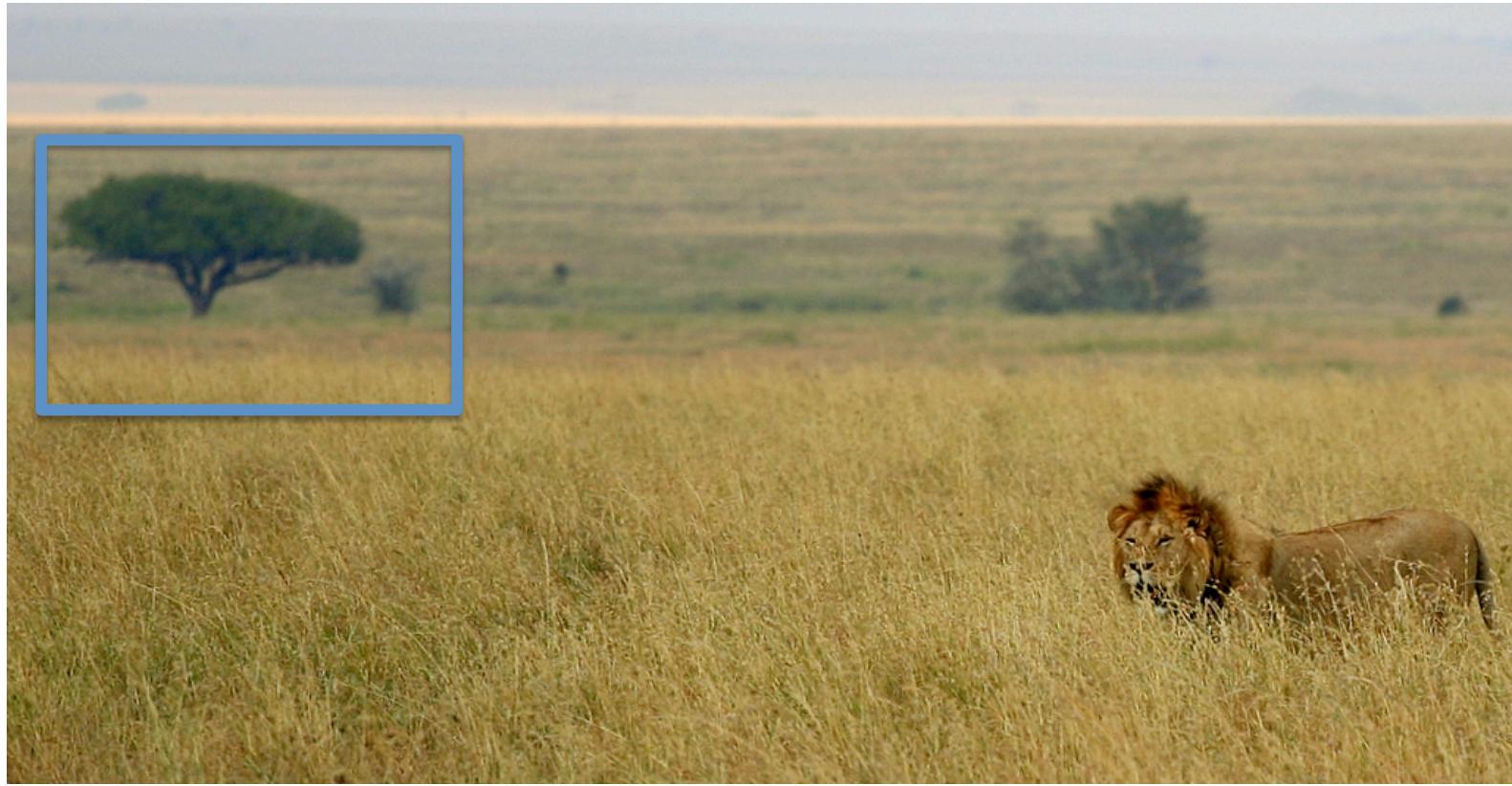


Interprétation



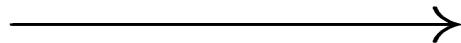


Ami ou pas?
Sourire ou Courir?

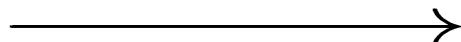


Où suis-je?

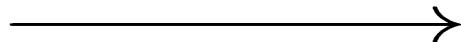
Apprendre à prédire



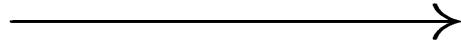
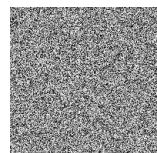
{chien, chat, pokemon, helicopter, ...}



{A. Turing, A. Einstein, S. Cooper, ...}



{0, 1, ... 24, 25, 26, ..., 98, 99, ...}

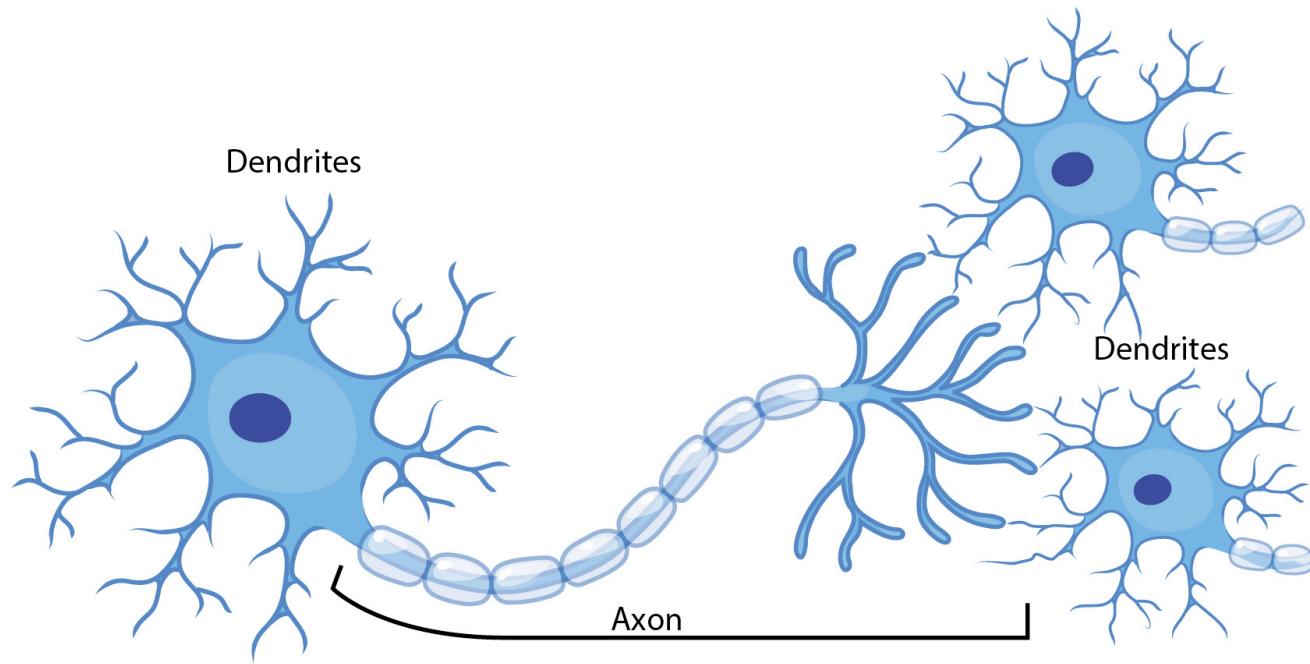


(*Manifold des Pokemons,
Par Damien Fourure*)

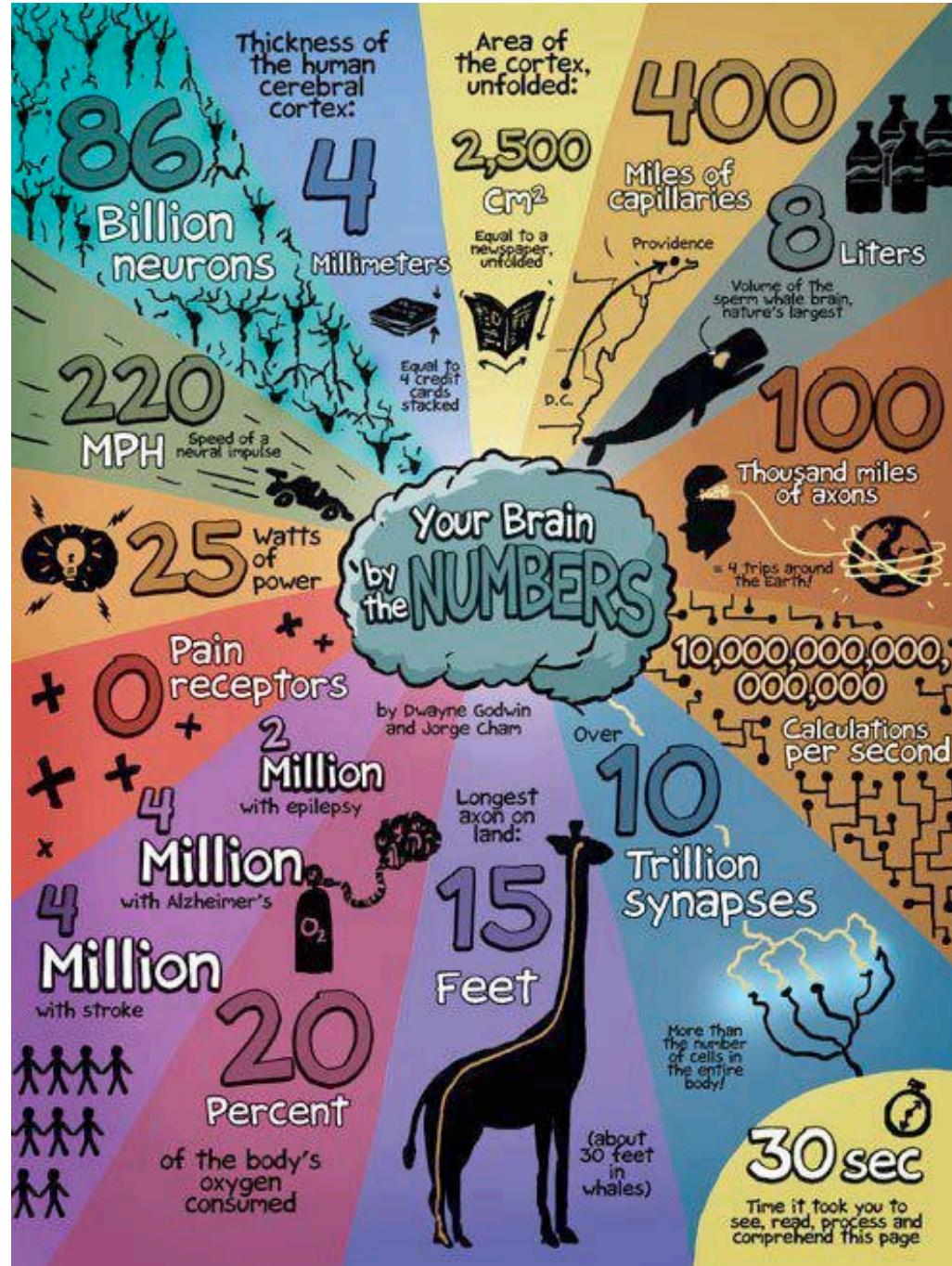
Apprendre à prédire t à partir d'une entrée x

$$y = f(x, \theta)$$

Un neurone en « VO »



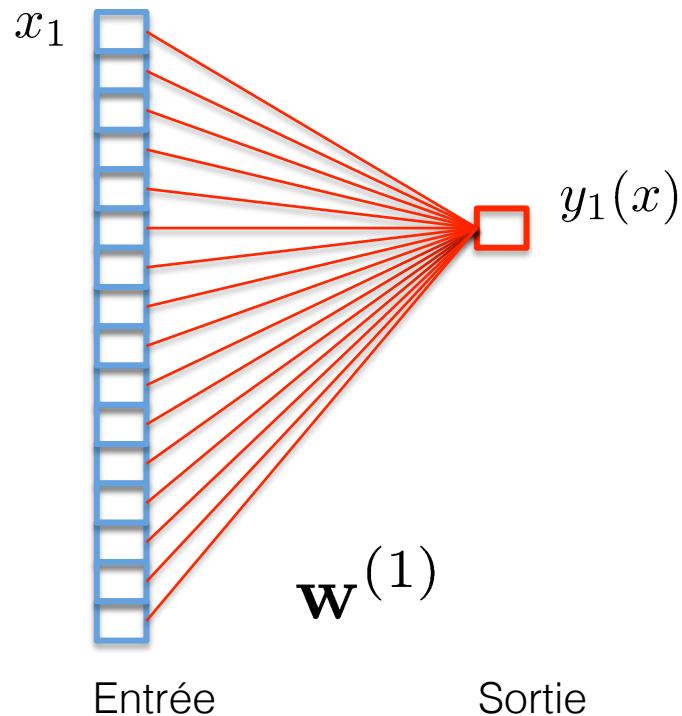
Devin K. Phillips



Réseaux de neurones

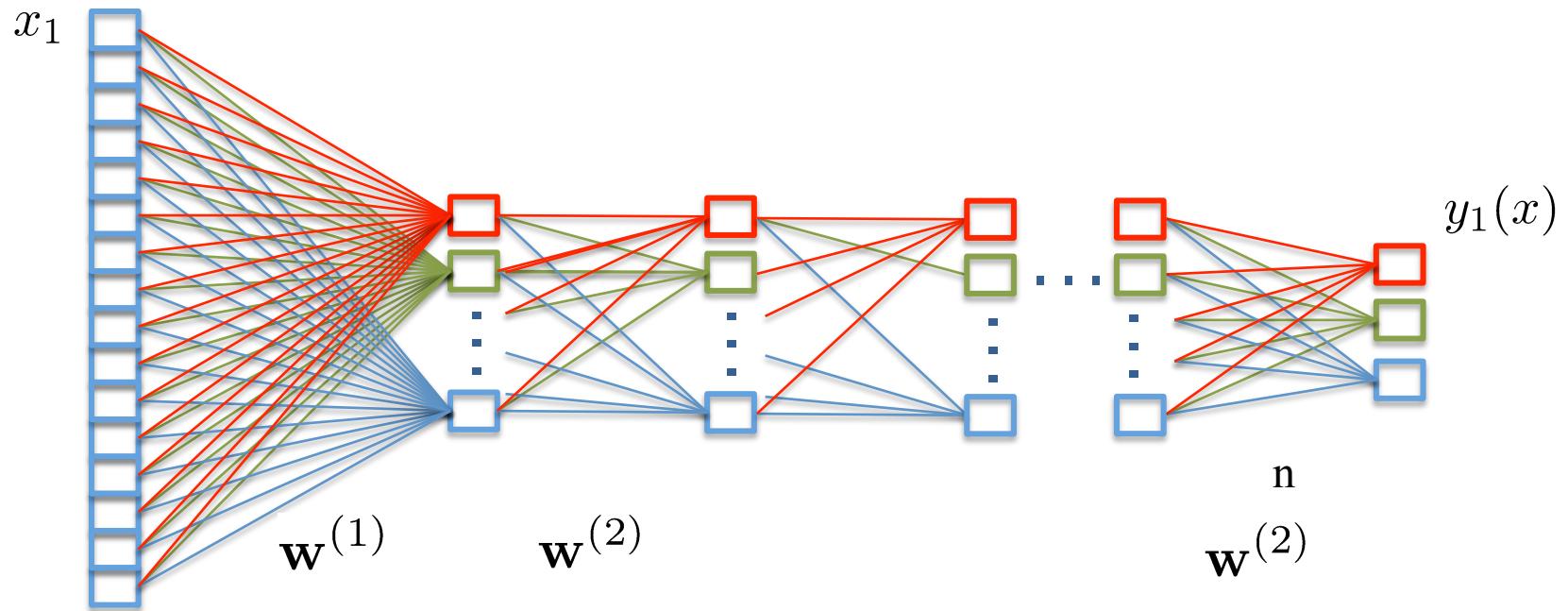
« Perceptron »

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^D w_i x_i$$



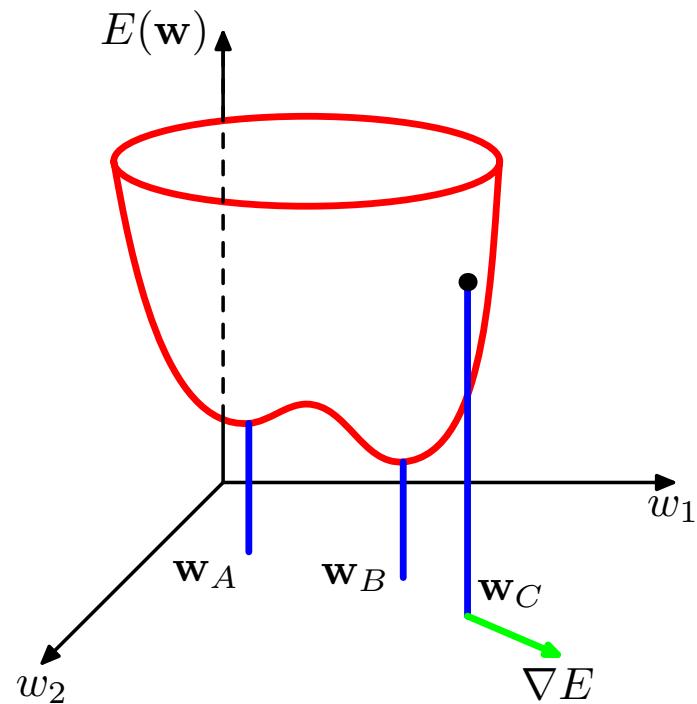


Réseaux « profonds »



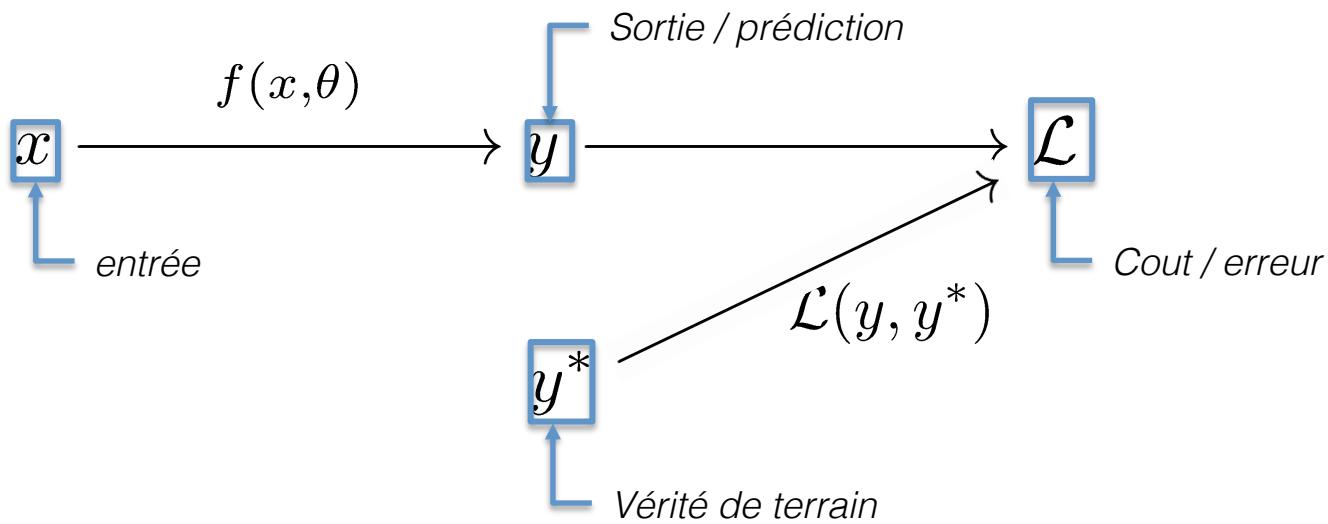
Descente de gradients

Minimisation de l'erreur commise sur des données connues (étiquetées).



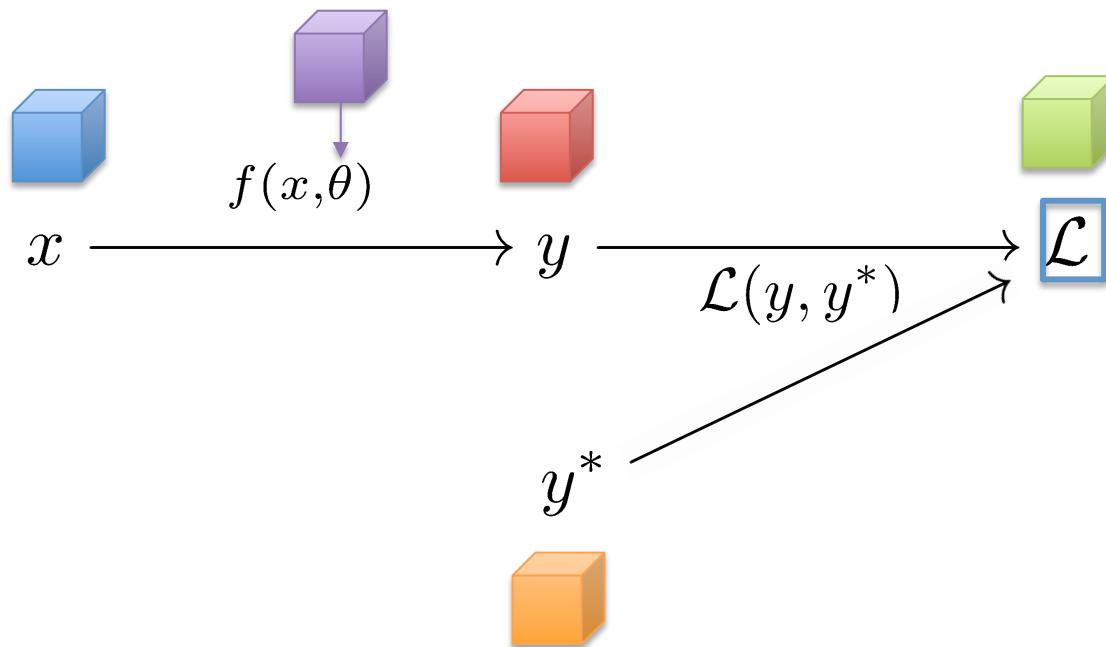
Blocage possible
dans un minimum
local (et on s'en fiche)

Mapping fonctionnel

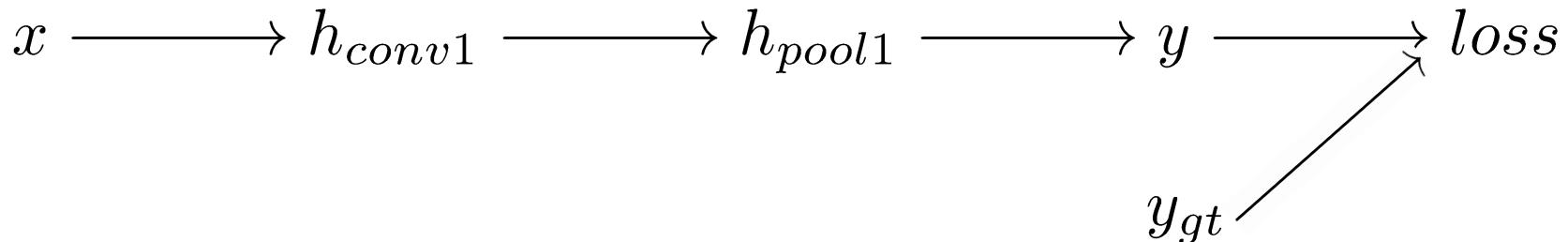


Quand les tenseurs coulent ...

Entrées, sorties et paramètres sont des tenseurs de plusieurs dimensions.



Google Tensorflow



Définition d'un graphe symbolique (!!) de calcul :

```
1 # The input tensors
2 x = tf.placeholder(tf.float32, [None, 224, 224, 1])
3
4 # The target labels
5 y_gt = tf.placeholder(tf.float32, [None, NO_CLASSES])
6
7 # Layer 1 : convolutional
8 W_conv1 = weight_variable([5, 5, 1, 32])
9 b_conv1 = bias_variable([32])
10 h_conv1 = tf.nn.relu(tf.nn.conv2d(x, W_conv1, strides=[1, 1, 1, 1], padding='SAME') + b_conv1)
11 h_pool1 = tf.nn.max_pool(h_conv1, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')
12
13 # Layer 2 (output) : fully connected
14 W_fc2 = weight_variable([32 * 7 * 7, 1024])
15 b_fc2 = bias_variable([1024])
16 y = tf.matmul(h_pool1, W_fc2) + b_fc2
17
18 # The loss function
19 loss = tf.reduce_sum(tf.squared_difference(y, y_gt))
```

Calcul automatique des dérivées des erreurs par rapport aux paramètres :

```
22 # Define gradients of loss with respect to some weights
23 grads_input = tf.gradients(loss, W_conv1)[0]
24
25 # Alternatively, define gradients of loss with respect to the input
26 grads_input = tf.gradients(loss, x)[0]
```

1960s



Alexey Ivakhnenko travaille sur les réseaux de neurones

1986



G. Hinton propose l'algorithme de retro-propagation de l'erreur

AI Winter



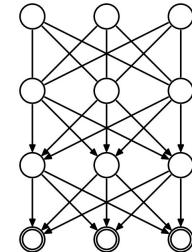
Pas de réseaux de neurones
Structures plates
Features conçues manuellement

2006

2006



G. Hinton's propose les *deep belief networks*
GPUs
Les données commencent à arriver



11/2011



Avancées en reconnaissance de parole par Microsoft,
avec des réseaux de neurones

10/2012

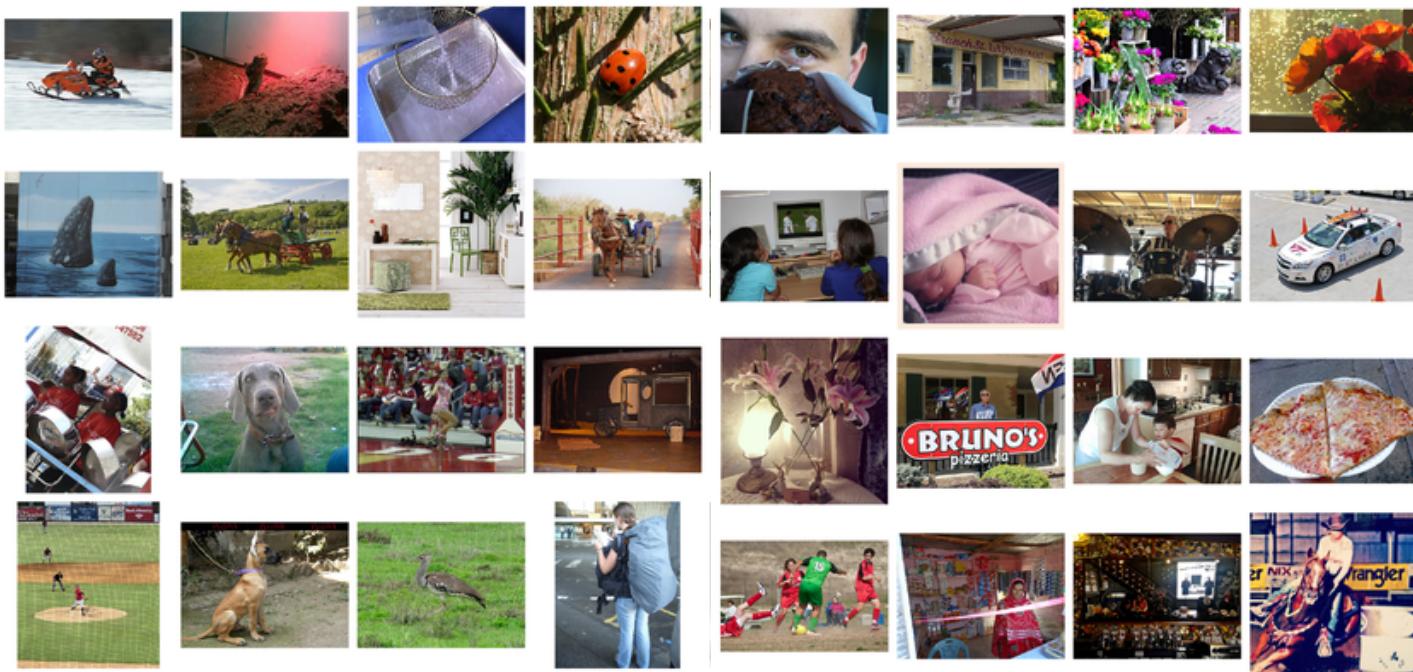


Le groupe de G. Hinton gagne ImageNet / ILSVRC

Rank	Name	Error rate	Description
1	U. Toronto	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	Bottleneck.

IMAGENET Large Scale Visual Recognition Challenge

- 1,461,406 images dans ILSVRC 2010
- 1000 classes
- Annotation manuelle et vérification
- Crowd-sourcing (Amazon Mechanical Turk)



03/2013



G. Hinton rejoint Google



09/2013



Clarifai gagne ImageNet / ILSVRC (PhD @ NYU)

Classification task:

First 20 entries: deep learning

Rank	Name	Error rate	Description
1	NYU	0.11197	Deep learning
2	NUS	0.12535	Deep learning
3	Oxford	0.13555	Deep learning

Detection task:

Rank	Name	Mean Average Precision	Description
1	UvA-Euvision	0.22581	Hand-crafted features
2	NEC-MU	0.20895	Hand-crafted features
3	NYU	0.19400	Deep learning

12/2013



Y. LeCun dirige Facebook AI Research



01/2014



Google achète DeepMind pour 400M\$

05/2014



A. Ng dirige Baidu AI



09/2014



Google gagne ImageNet / ILSVRC

Classification task:

First 20 entries: deep learning

Rank	Name	Error rate	Description
1	Google	0.06656	Deep learning
2	Oxford	0.07325	Deep learning
3	MSRA	0.08062	Deep learning

Detection task:

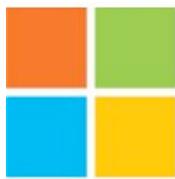
Rank	Name	Mean Average Precision	Description
1	Google	0.43933	Deep learning
2	CUHK	0.40656	Deep learning
3	DeepInsight	0.40452	Deep learning
4	UvA-Euvision	0.35421	Deep learning
5	Berkley Vision	0.34521	Deep learning

09/2014



Papier (NIPS) de Hinton mentionne un jeu de données de Google avec 100 000 000 images (annotées!) et 15 000 classes

12/2015



Microsoft research gagne ImageNet 2015

12/2015



Google introduit le « vision API »

12/2015



Creation de « Open AI », organisation non-profit pour l'IA.
Director: Ilya Sutskever (a travaillé avec Hinton, Ng, Google)

02/2016



Y. LeCun est nominé au *Collège de France*



COLLÈGE
DE FRANCE
1530

03/2016



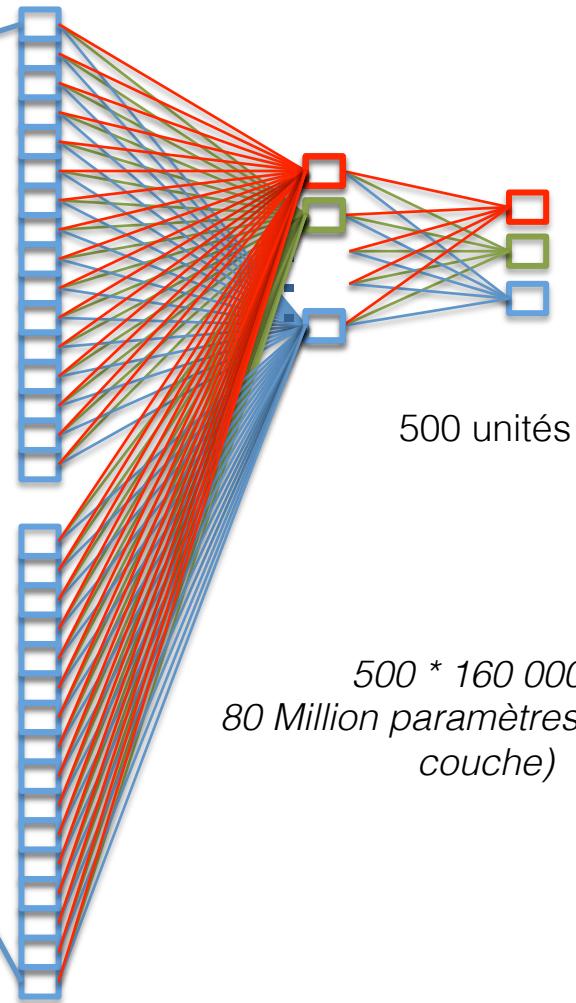
L'IA de Deepmind gagne contre le champion du jeu de Go, avec des réseaux de neurones (entre autres)

Comment ça fonctionne?

Le problème du « fully-connected »



400x400 pixels



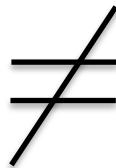
Mis à plat dans
160 000 valeurs

500 unités

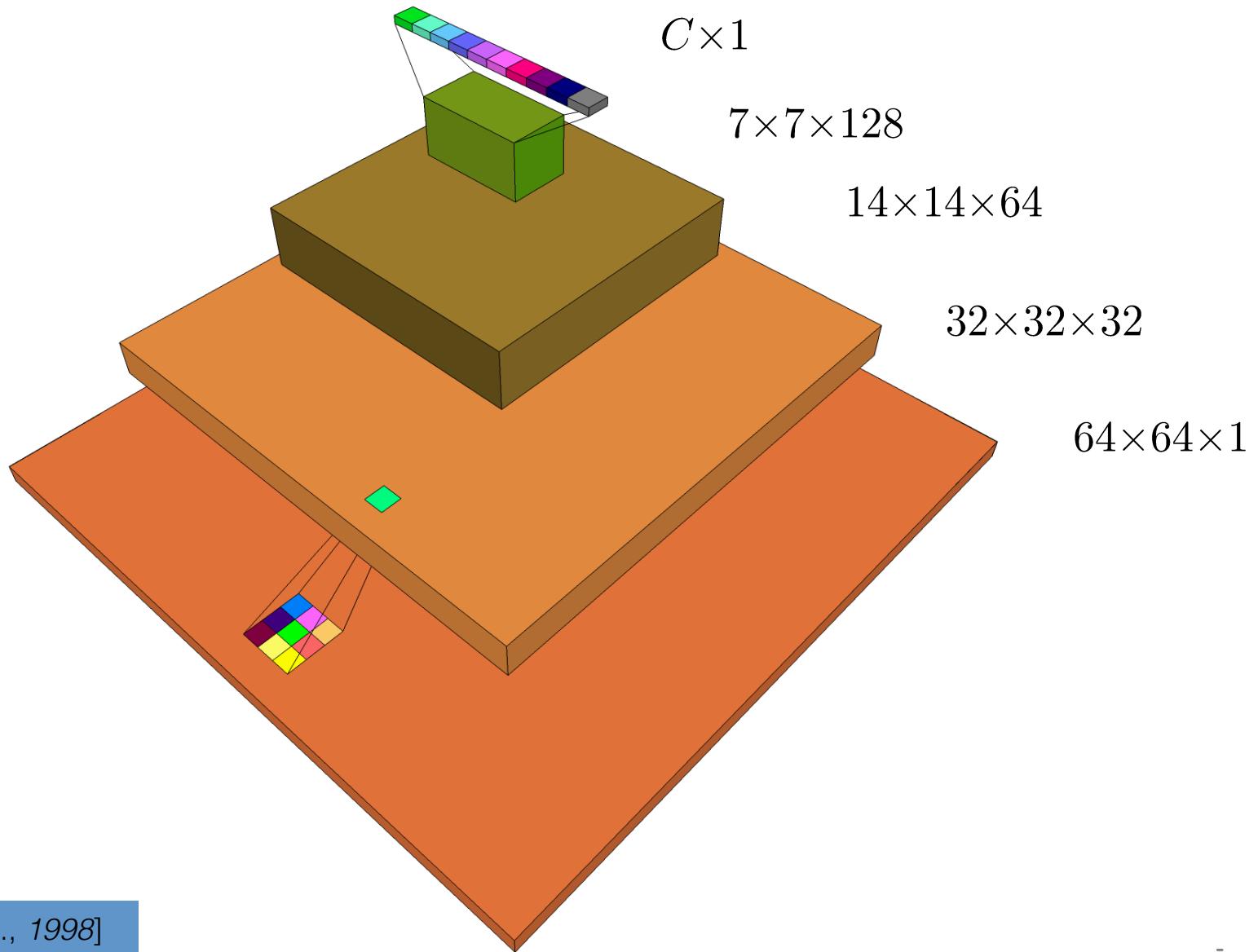
$500 * 160\,000 =$
80 Million paramètres (pour une
couche)

Le problème du « fully-connected »

Les paramètres appris pour une partie de l'image ne généralisent pas pour une autre partie



« LeNet »



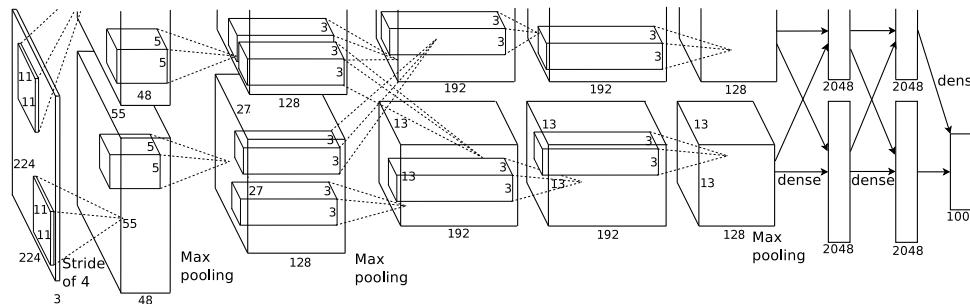
[LeCun et al., 1998]

Augmentation des données

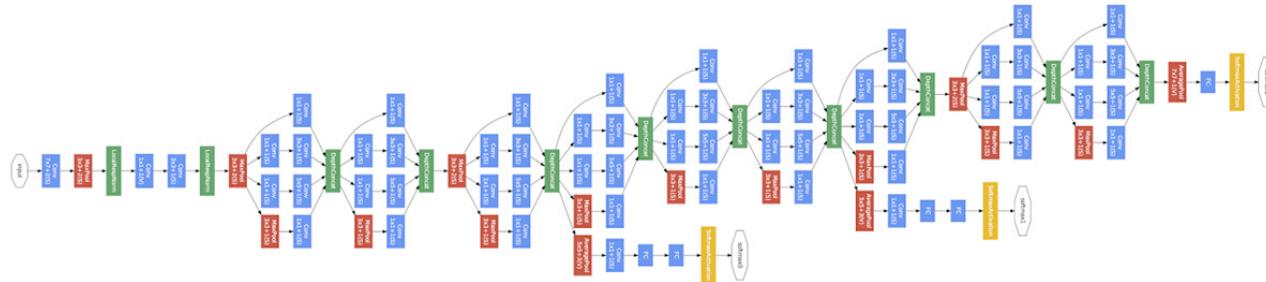


Le deep du deep

2012 : AlexNet, 8 couches entraînées sur 2 GPUs. Nouvelle technique : dropout



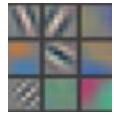
2014 : GoogLeNet, 20 couches. Nouvelle technique : intermediate supervision



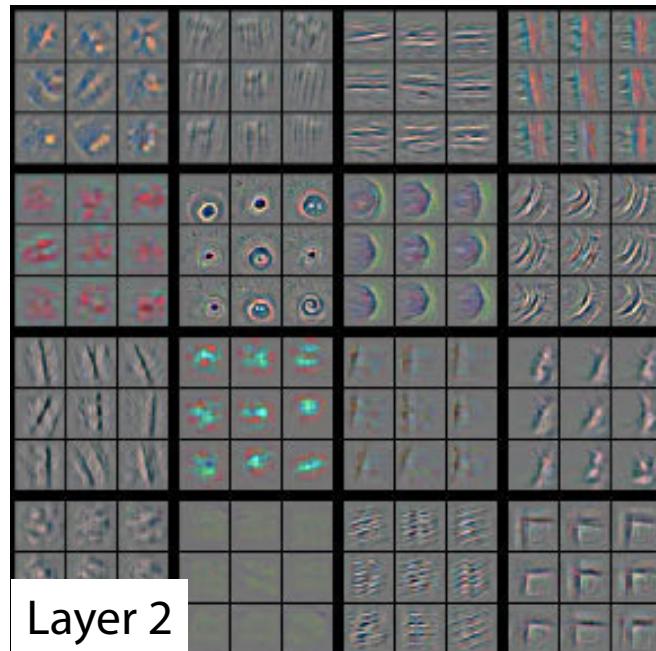
2015 : Microsoft research, 150 couches (!!). Nouvelle technique : residual learning



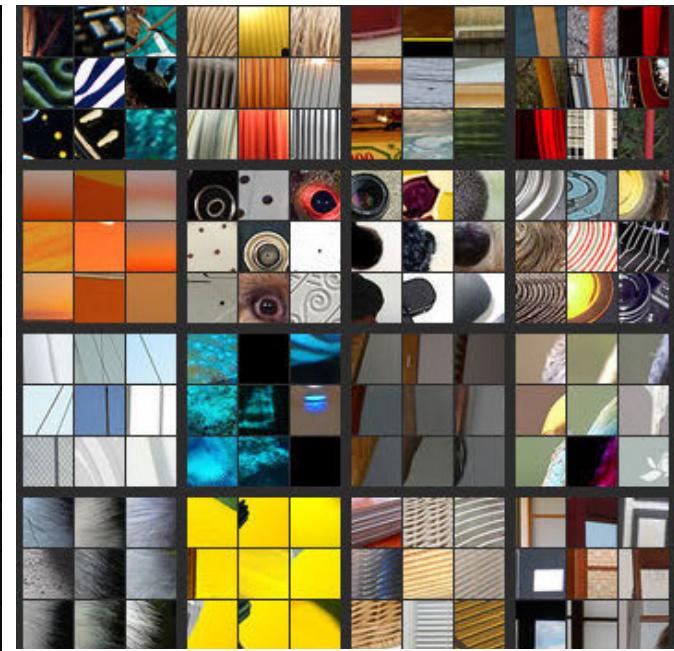
Visualisation



Layer 1

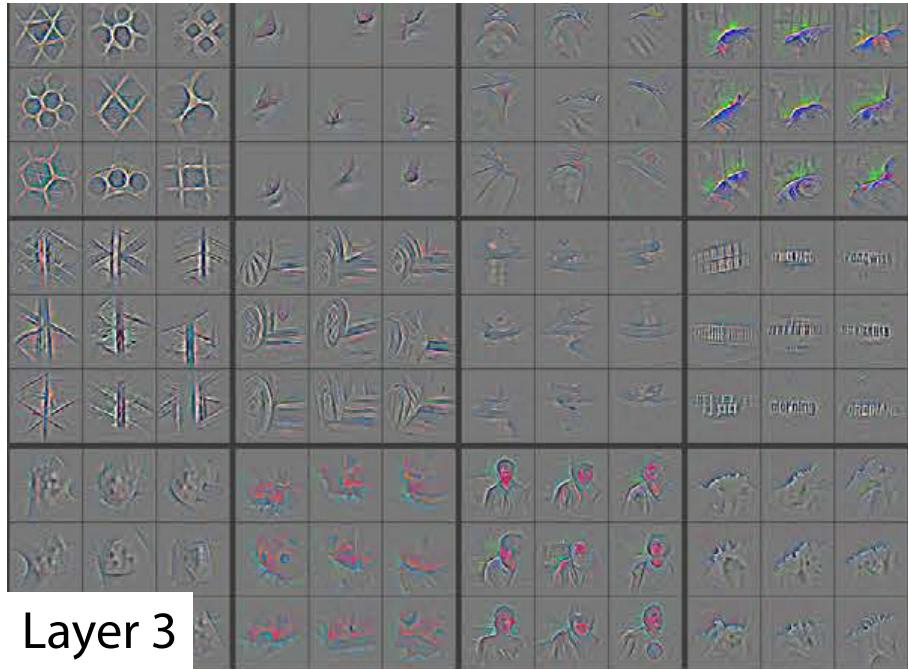


Layer 2



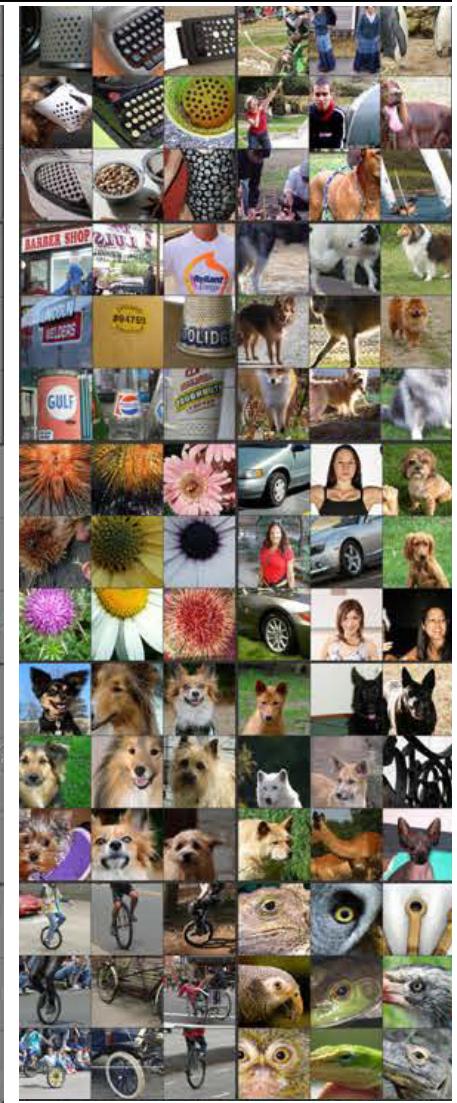
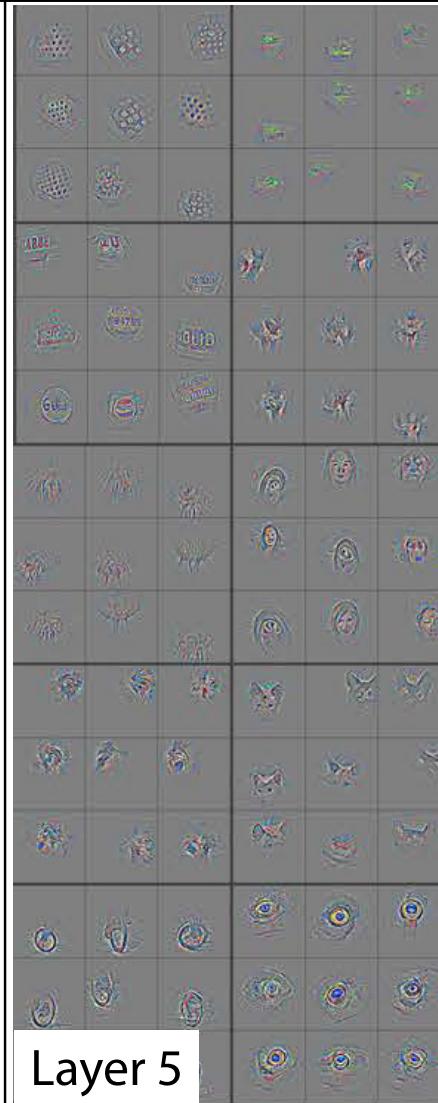
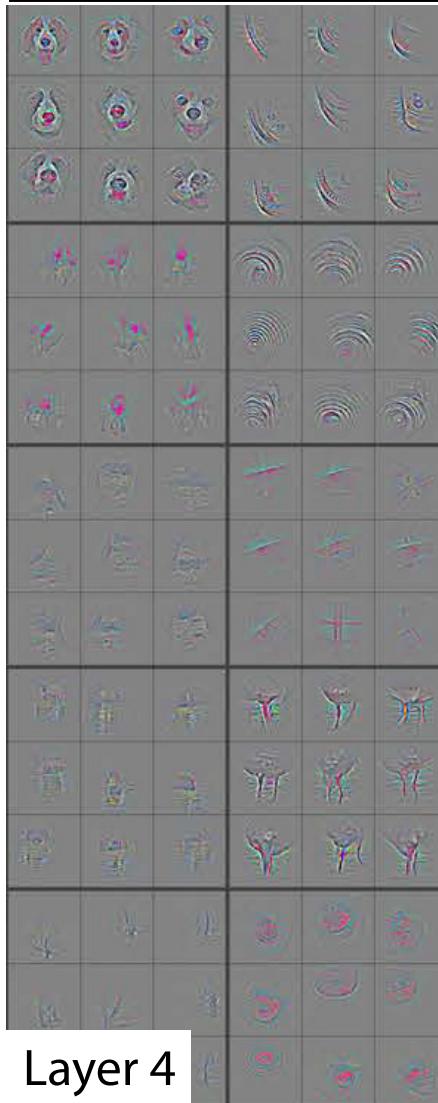
[Zeiler and Fergus,
ECCV 2014]

Visualisation



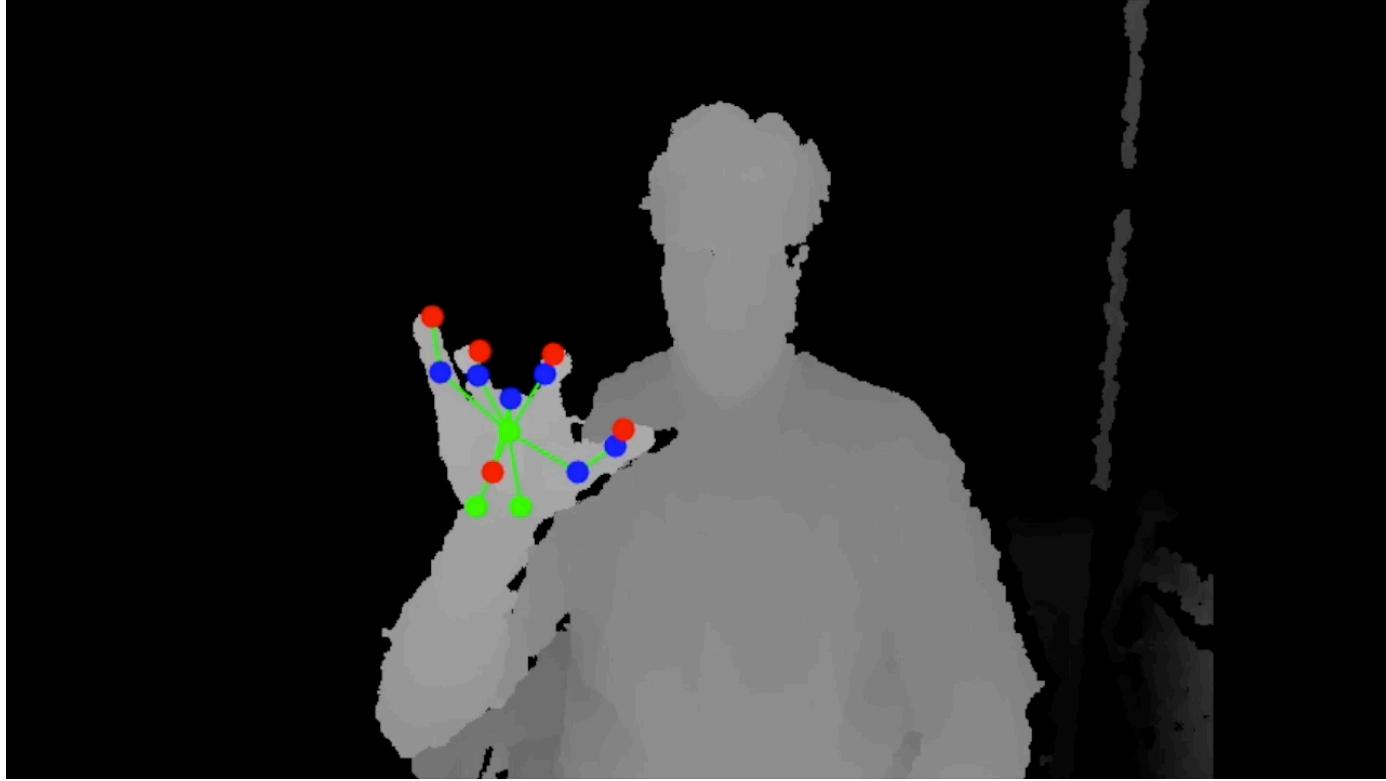
[Zeiler and Fergus, ECCV 2014]

Visualisation



Autres applications

Estimation de la posture articulée



Travail de Natalia Neverova
LIRIS (maintenant à Facebook AI
Research)

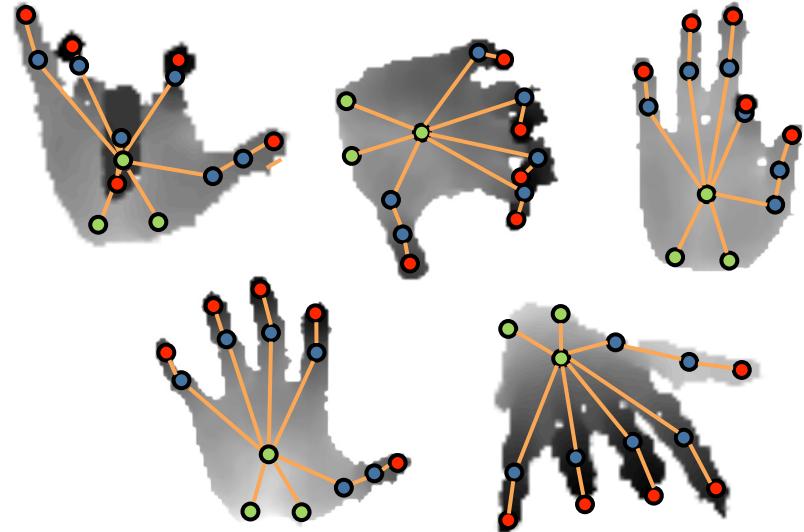


Avec Graham W. Taylor,
University of Guelph, Canada



Awabot
— Beyond Robotics —

Entrainement : big data



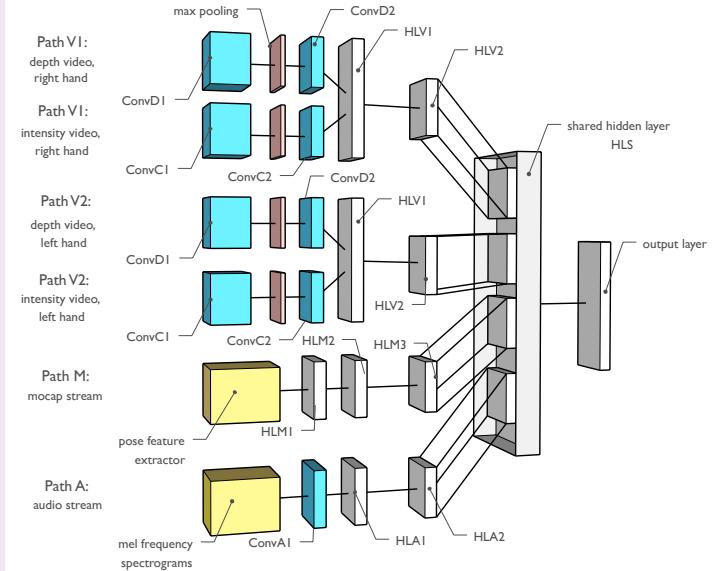
Dataset 1: données réelles (NYU)



Dataset 2: nos données synthétiques

[Neverova, Wolf, Taylor, Nebout, arxiv 2015]

Reconnaissance de gestes



Travail de Natalia Neverova
LIRIS (maintenant à Facebook AI
Research)



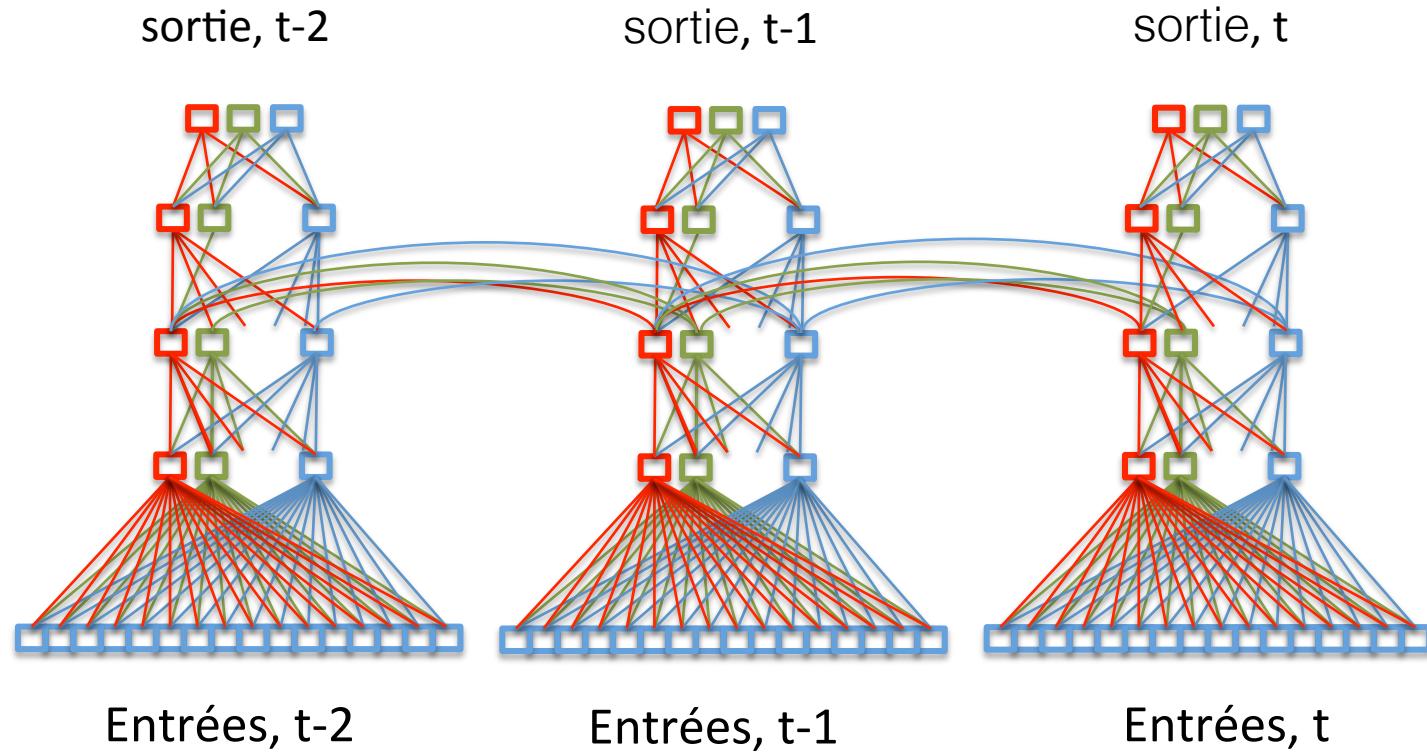
Avec Graham W. Taylor,
University of Guelph, Canada



Awabot
— Beyond Robotics —

[Neverova, Wolf, Taylor, Nebout,
IEEE PAMI 2016]

Apprendre des séquences



Apprendre à expliquer les images

Dataset of images and sentence descriptions

training image

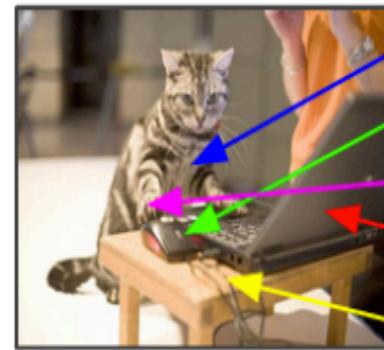


"A Tabby cat is leaning on a wooden table, with one paw on a laser mouse and the other on a black laptop"



Inferred correspondences

training image



"Tabby cat is leaning"
"laser mouse"
"paw"
"black laptop"
"wooden table"

[Karpathy et al, CVPR 2015]

Apprendre à expliquer les images



[Karpathy et al, 2015]

Déverrouiller un smartphone est pénible



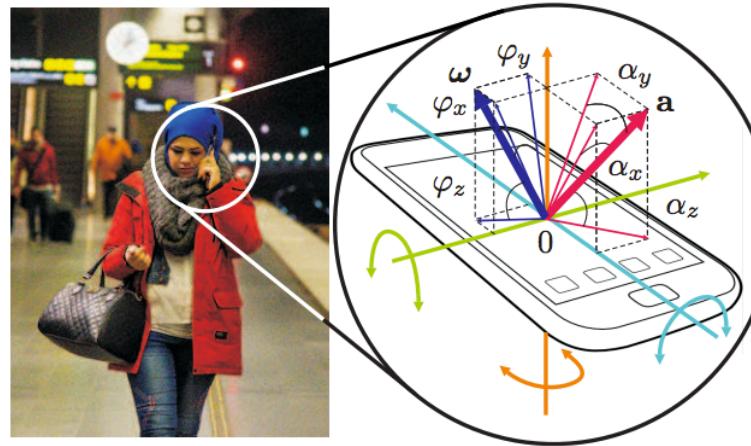
© 2005 Scott Adams, Inc. / Dist. by UFS, Inc.



Apprendre à déverrouiller mon smartphone à partir de mon comportement (mes mouvements).

Projet "Abacus » (Google)

- 1500 volontaires, 1500 Nexus 5 smartphones
- Plusieurs mois d'usage quotidien (27.6 TO de données)
- Plusieurs capteurs : camera, touchscreen, GPS, bluetooth, wifi, antenne, inertiel, gyro, magnetomètre
- Notre travail : capteurs inertIELS à 200Hz



Travail de Natalia Neverova
INSA-Lyon, LIRIS (maintenant à
Facebook AI Research)

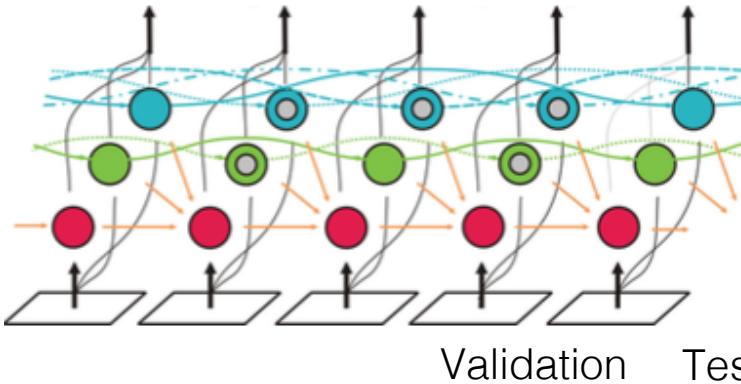


Avec Graham W. Taylor,
University of Guelph, Canada



[Neverova, Wolf, Lacey, Fridmann, Chandra, Barbello, Taylor, IEEE Access 2016]

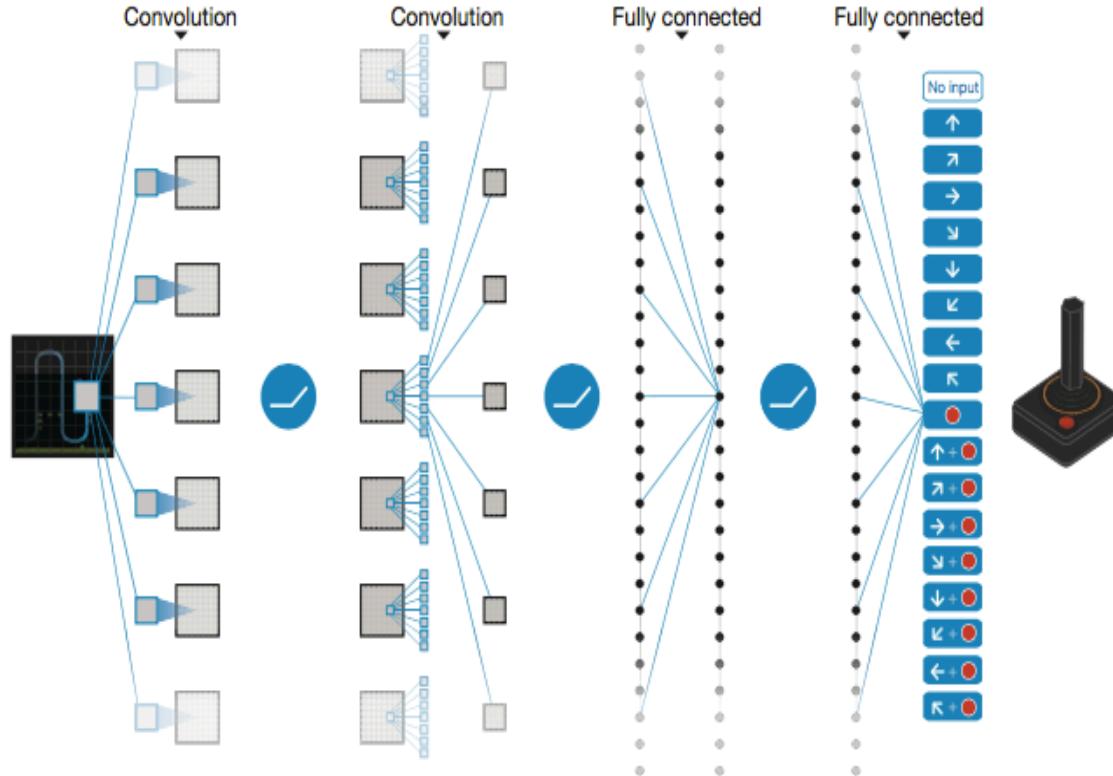
Dense clockwork recurrent networks



Model	EER, %	HTER, %	
Raw features	36.21	42.17	10sec
ST Convnet	32.44	34.89	1 sec
LT Convnet	28.15	29.01	10sec
Conv-RNN	22.32	22.49	10sec
Conv-CWRNN	21.52	21.92	10sec
Conv-LSTM	21.13	21.41	10sec
Conv-DCWRNN	20.01	20.52	10sec
Conv-DCWRNN, zt-norm	18.17	19.29	10sec
<i>Conv-DCWRNN (per device)</i>	15.84	16.13	10sec
<i>Conv-DCWRNN (per session)</i>	8.82	9.37	10sec



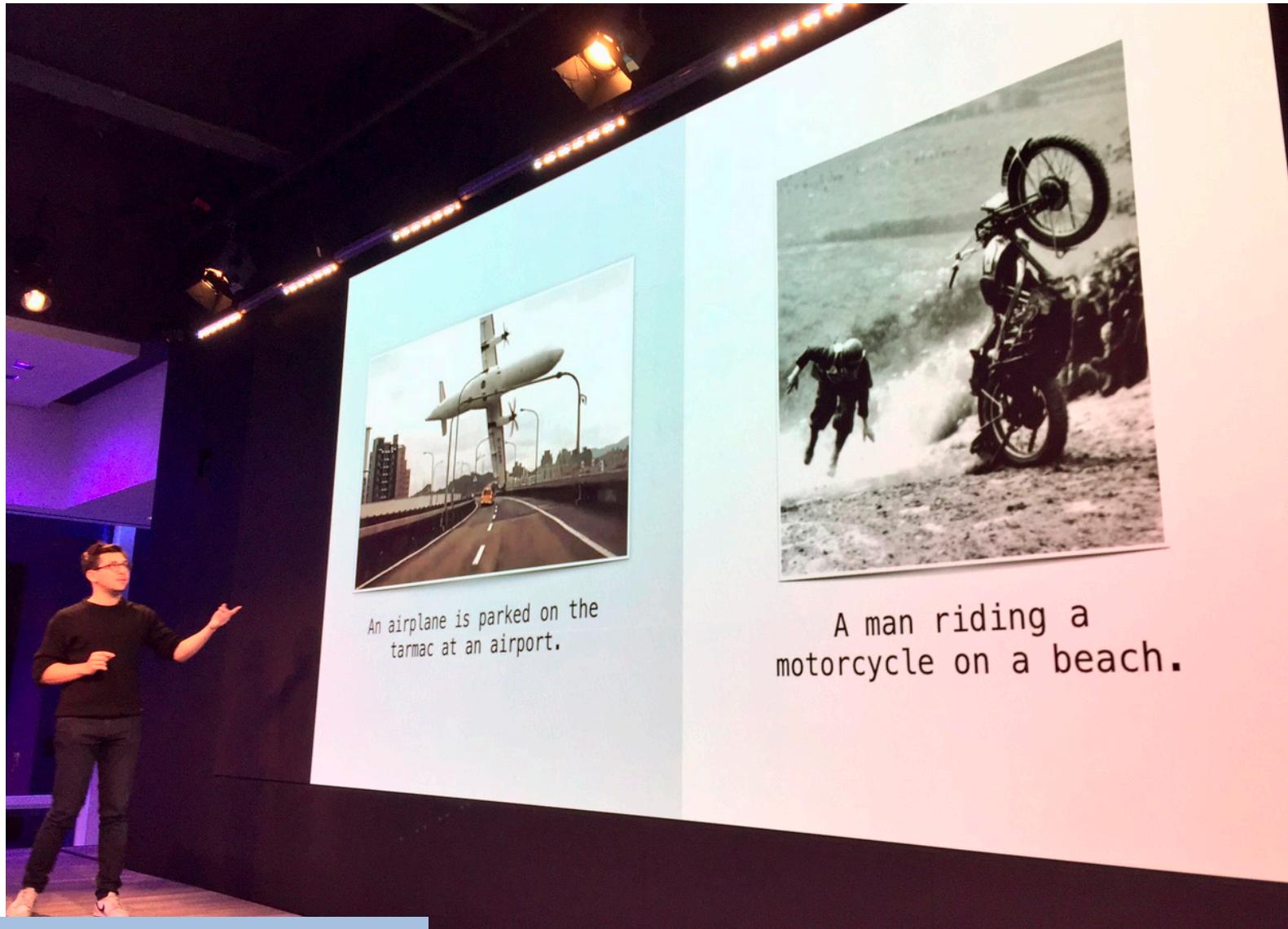
Reinforcement deep learning?



[Mnih et al, Nature 2015]

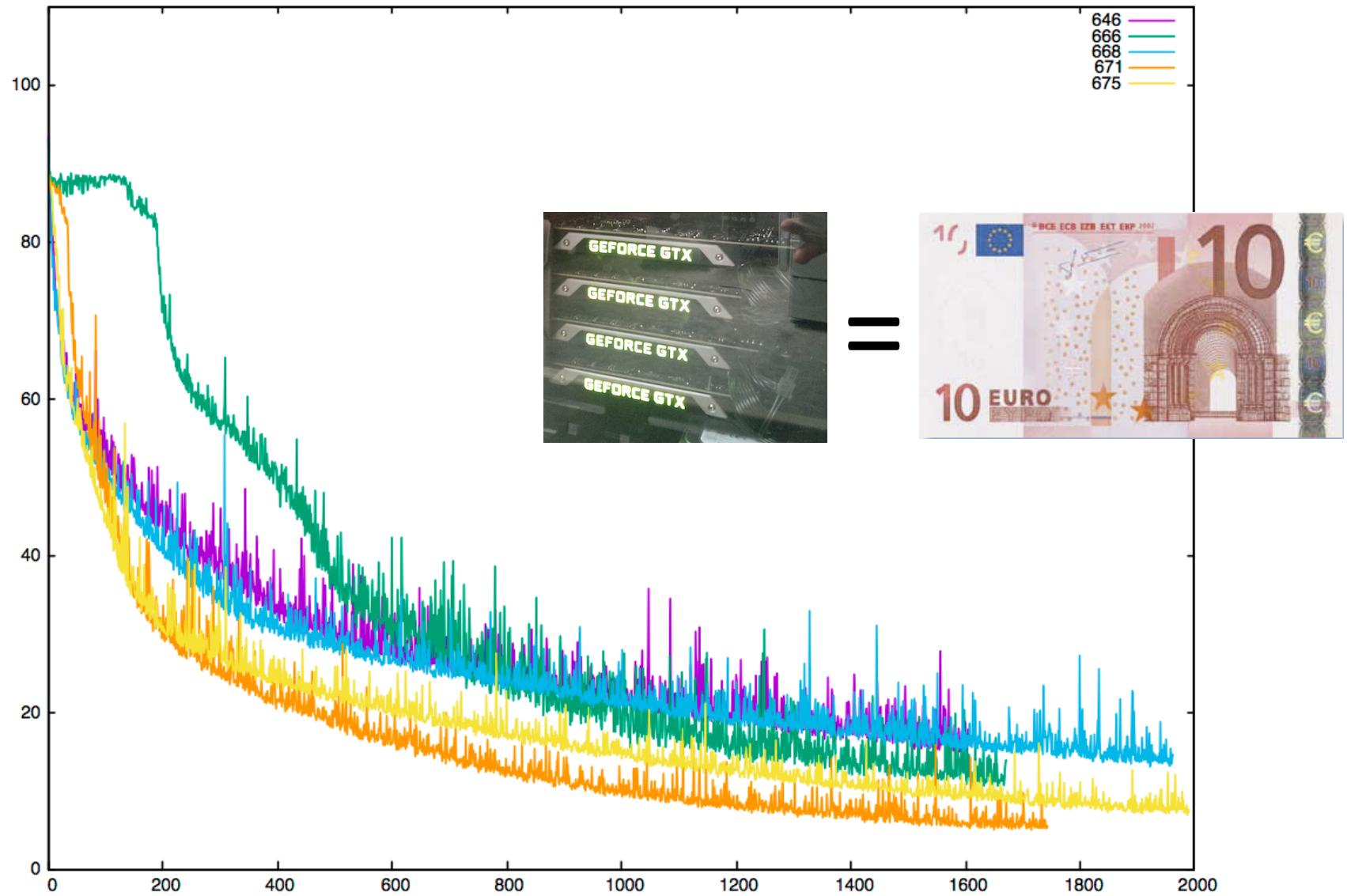
Google DeepMind's Deep Q-learning

Cas d'échec



[Antoine Bordes, Facebook AI Research, 2016]

Les couts du deep learning



Conclusion

- Le *Deep Learning* nécessite beaucoup de données (annotées) : **big data!**
- L'ingénierie des architectures est devenu la nouvelle ingénierie des *features*
- Gourmand de ressources (GPU, clusters, GPU, clusters de GPUs, FPGA, ASIC).
- Est-ce que j'ai parlé de GPU?
- Nécessite une expertise pour l'entraînement (“*babysitting*”).
- Classification, régression, génération de données ... pour quand IA “forte”?