SORBONNE UNIVERSITÉ

# C4. Introduction aux réseaux de neurones profonds

**Advanced Machine Learning (MLA)**

Kévin Bailly

kevin.bailly@sorbonne-universite.fr

http://people.isir.upmc.fr/bailly/

# Objectifs du cours

- Objectifs
  - Comprendre le principe de fonctionnement
  - Les mettre en œuvre dans des cas concrets à l'aide d'outils modernes (PyTorch)
  - Découvrir quelques techniques / architectures avancées ou couramment utilisées
  - Beaucoup de références pour aller plus loin

# Bibliographie


- I. Goodfellow , Y. Bengio , A. Courville. **Deep Learning**. MIT Press (2016)

- Fei-Fei Li et al. **Convolutional Neural Networks for Visual Recognition.** (depuis 2015)
  - http://cs231n.stanford.edu/

- M. Nielsen, **Neural networks and deep learning**, Determination press (2015)
  - http://neuralnetworksanddeeplearning.com/

# Plan

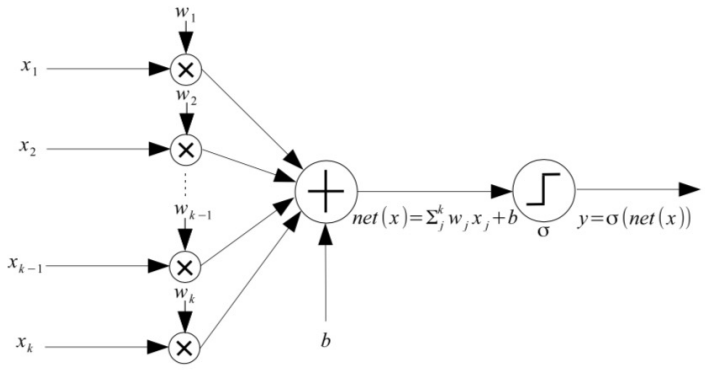
- Introduction (historique et applications)
- Qu’est-ce que l’apprentissage profond
- Comment créer un réseau de neurones avec PyTorch

# Plan

- Introduction (historique et applications)
- Qu’est-ce que l’apprentissage profond
- Comment créer un réseau de neurones avec Tenforflow

# Les 3 révolutions

- 1958 : le perceptron
  **[Rosenblatt, 58]**
  - Limitations : la fonction X- OR **[Minsky & Papert 69]**




- 1986 : La rétro-propagation
  **[LeCun, 86] [Rumelhart, 86]**

- ~2006 : L'apprentissage profond
  **[Hinton, 06 Science]**
  - 2019 : prix Turing **(**LeCun, Bengio, Hinton)

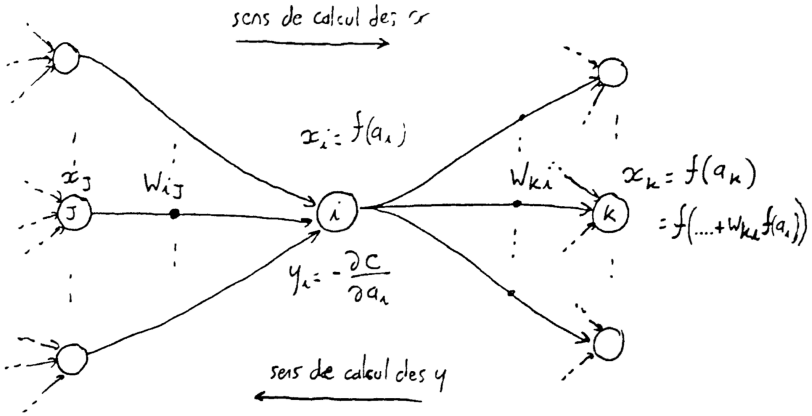



Figure 3. 4: Chaine des dépendances dans un réseau multicouche

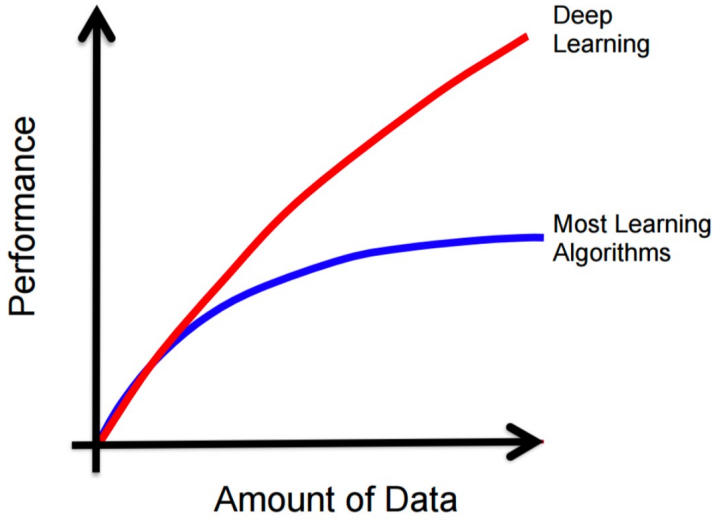
[LeCun, PhD 87]

http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/
http://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf

# Les facteurs de la révolution

# Les facteurs de la révolution

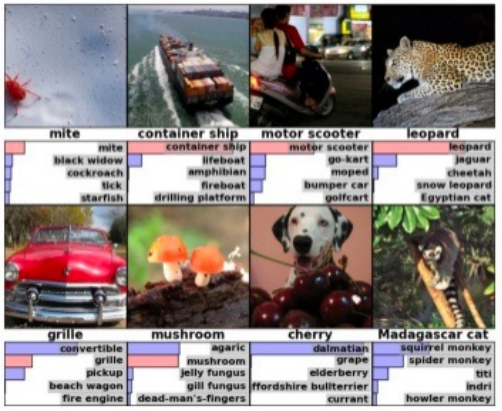
- Plus de données (labélisées)




**ImageNet Challenge**

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.




4

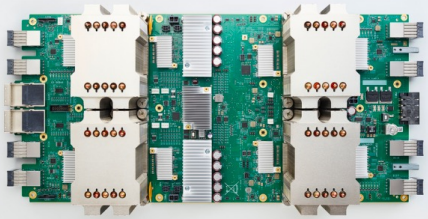
kaggle

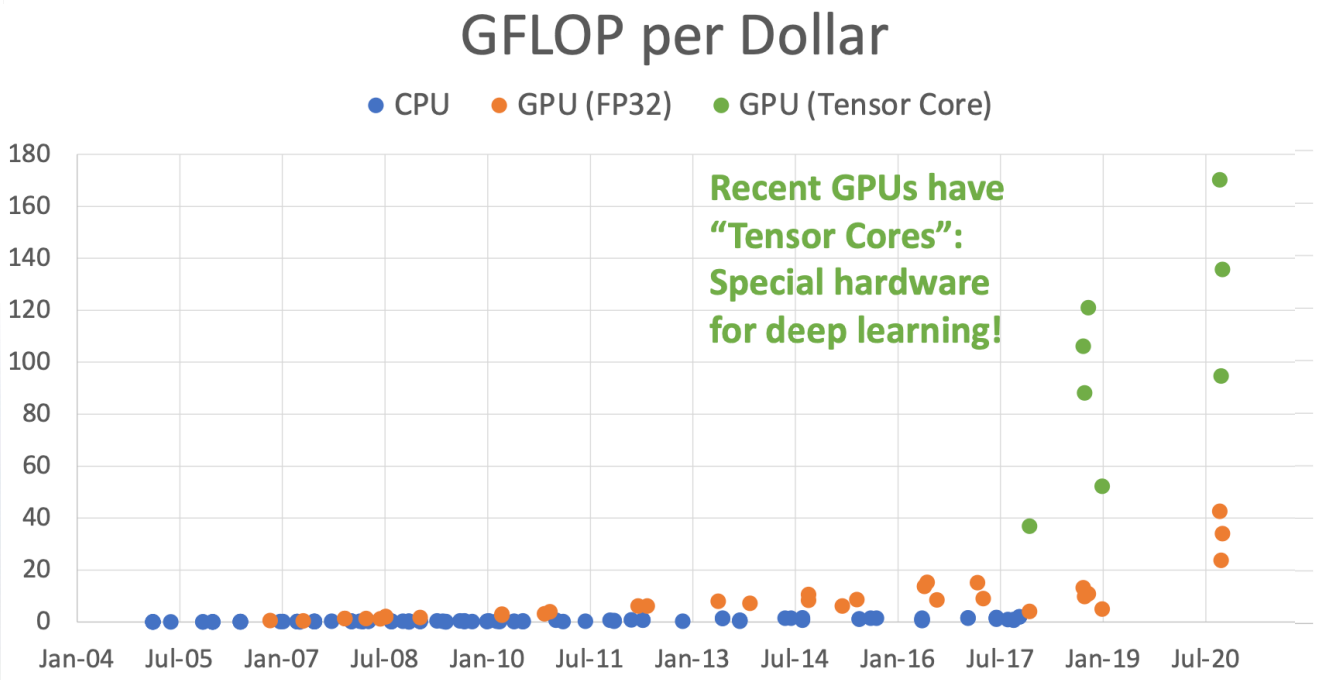
crowdAI

Sebastien Bubeck, Mark Sellke A Universal Law of Robustness via Isoperimetry NeurIPS 2021

# Les facteurs de la révolution

- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)




[Justin Johnson, EECS 498/598]

# Les facteurs de la révolution

- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)
- Des outils performants, flexibles et open-sources



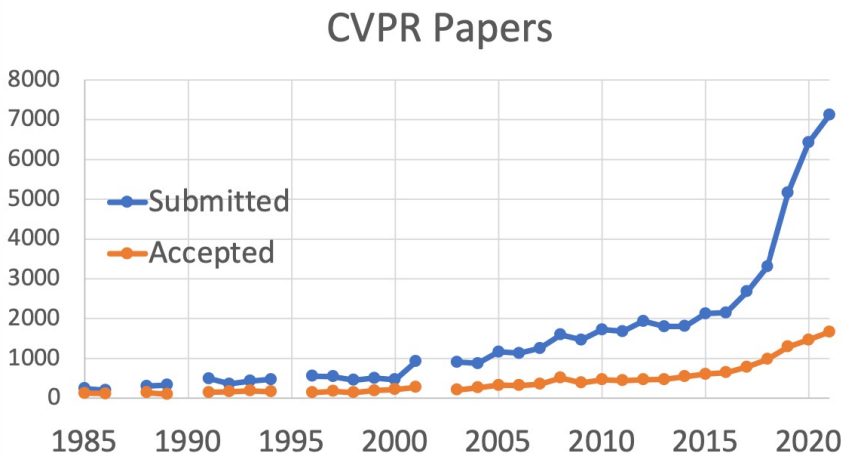
ONNX

# Les facteurs de la révolution

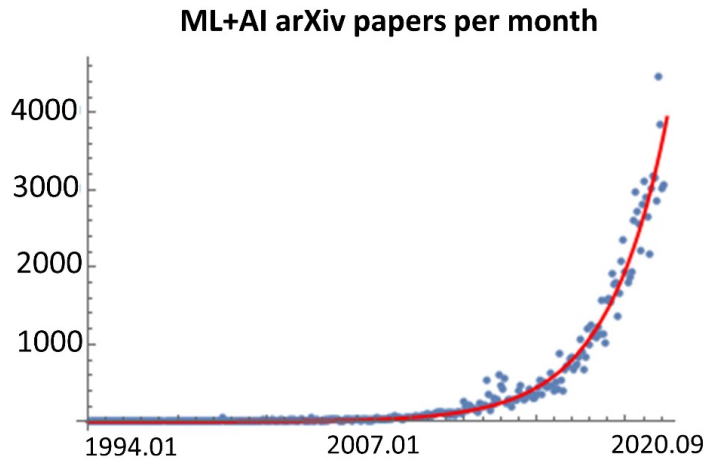
- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)
- Des outils performants, flexibles et open-sources
- Nouveaux algorithmes
  - Fonctions d'activations
  - Initialisation des paramètres
  - Méthodes d'optimisation (RMSProp, Adam)
  - Nouvelles architectures (CNN, ResNet, Inception)

# Les facteurs de la révolution

- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)
- Des outils performants et open-sources
- Intensification de la recherche (publique et **privée**)

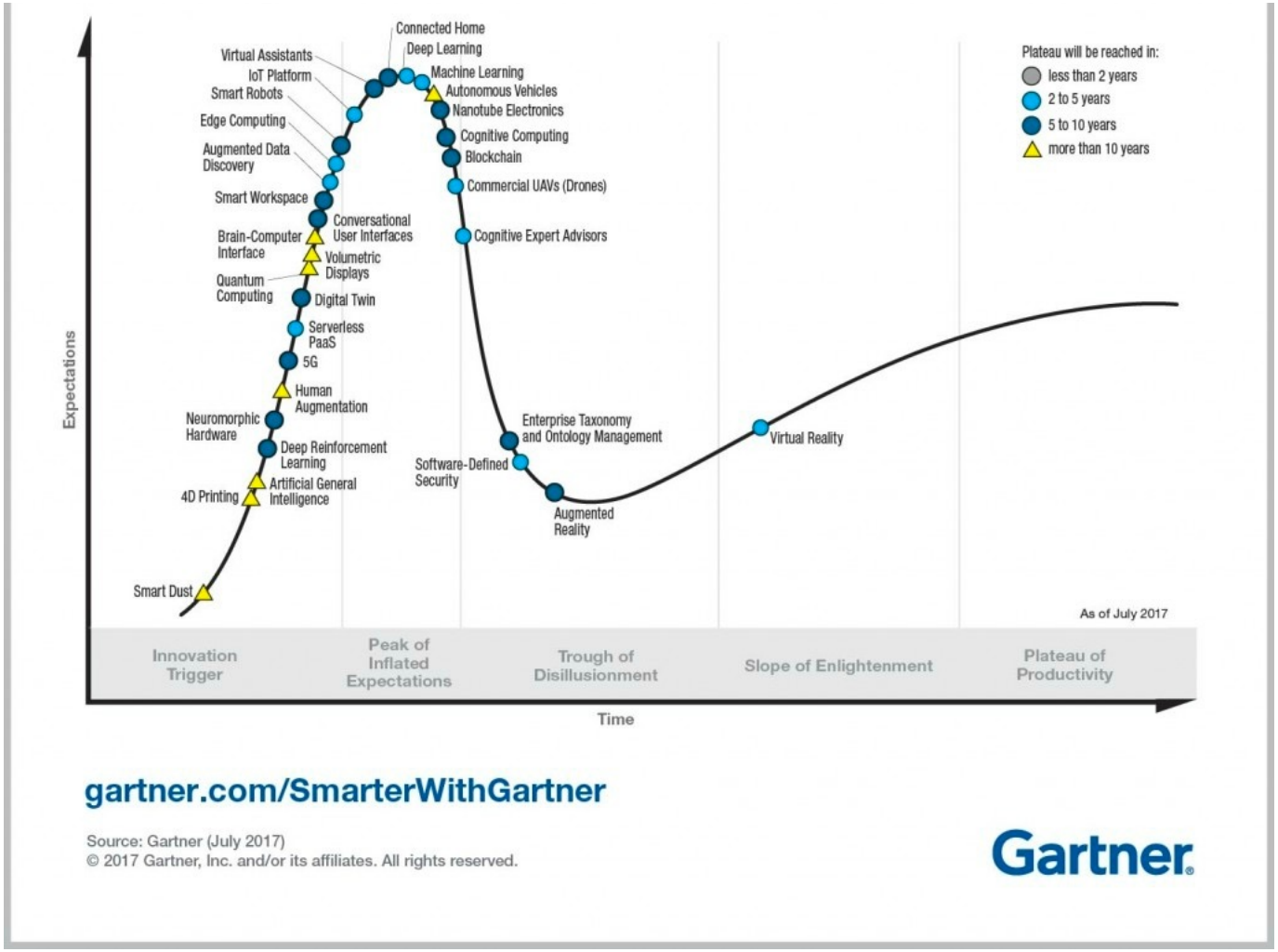




Publications at top Computer Vision conference



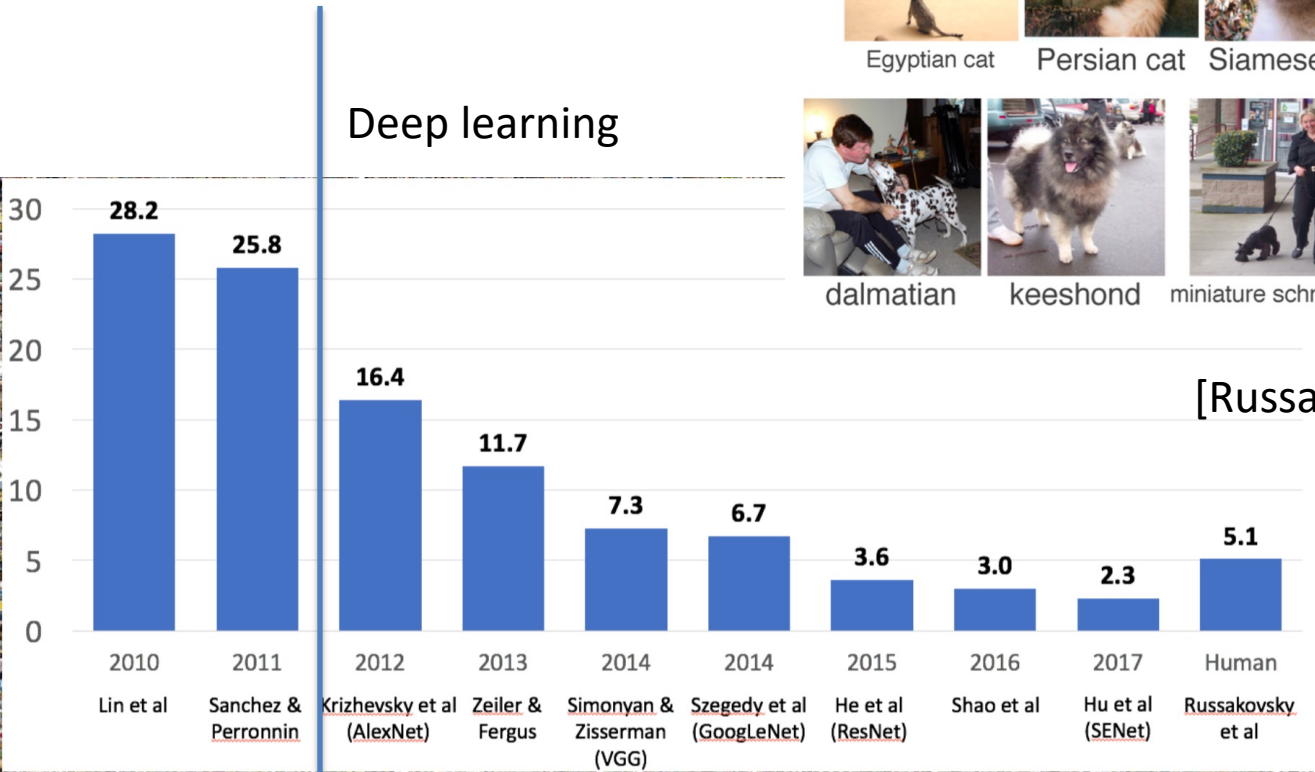

arXiv papers per month (source)

# Gartner Hype Cycle 2017



# Applications

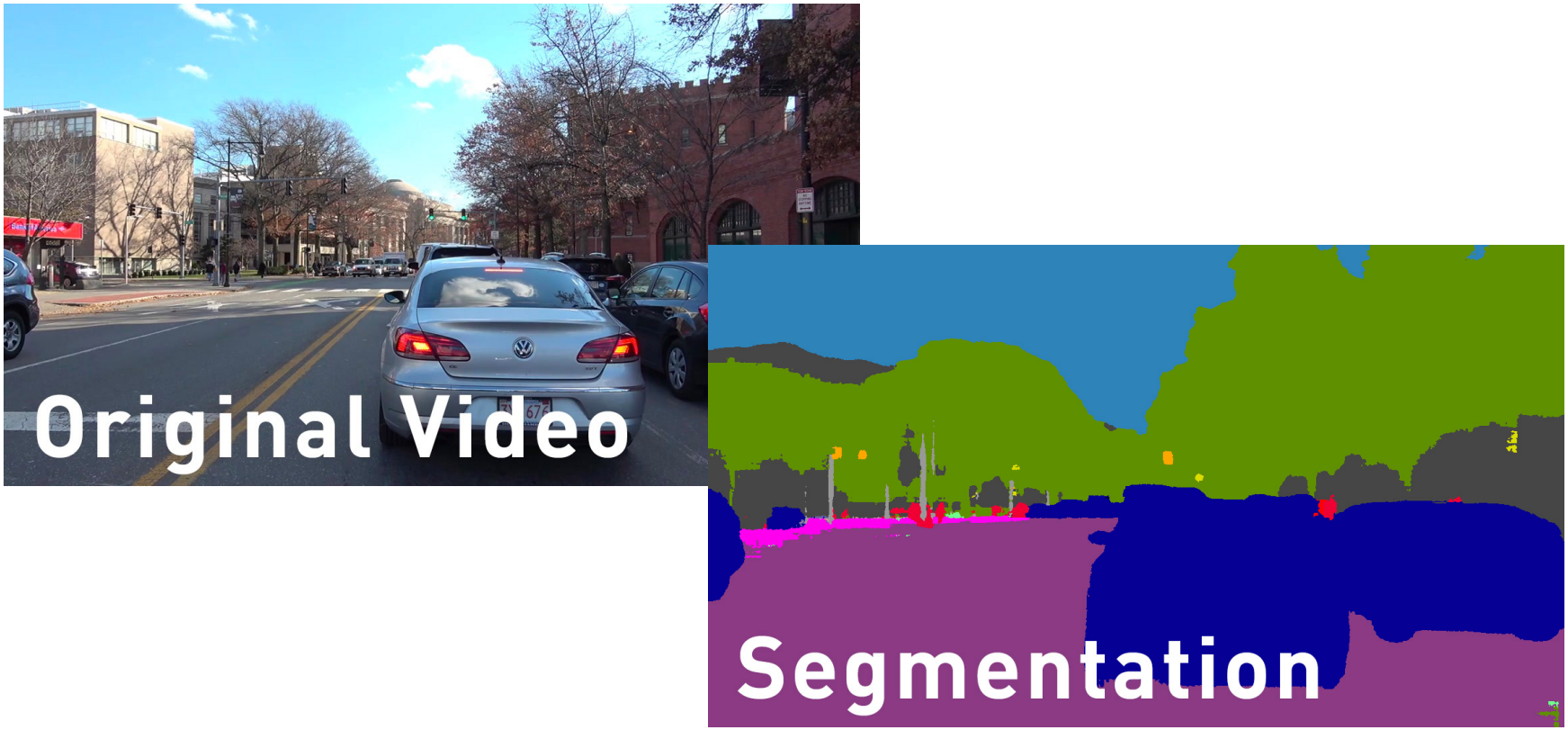
ImageNet Challenge
1000 classes
>1M d'images


[Russakovsky et al. 2015, IJCV]

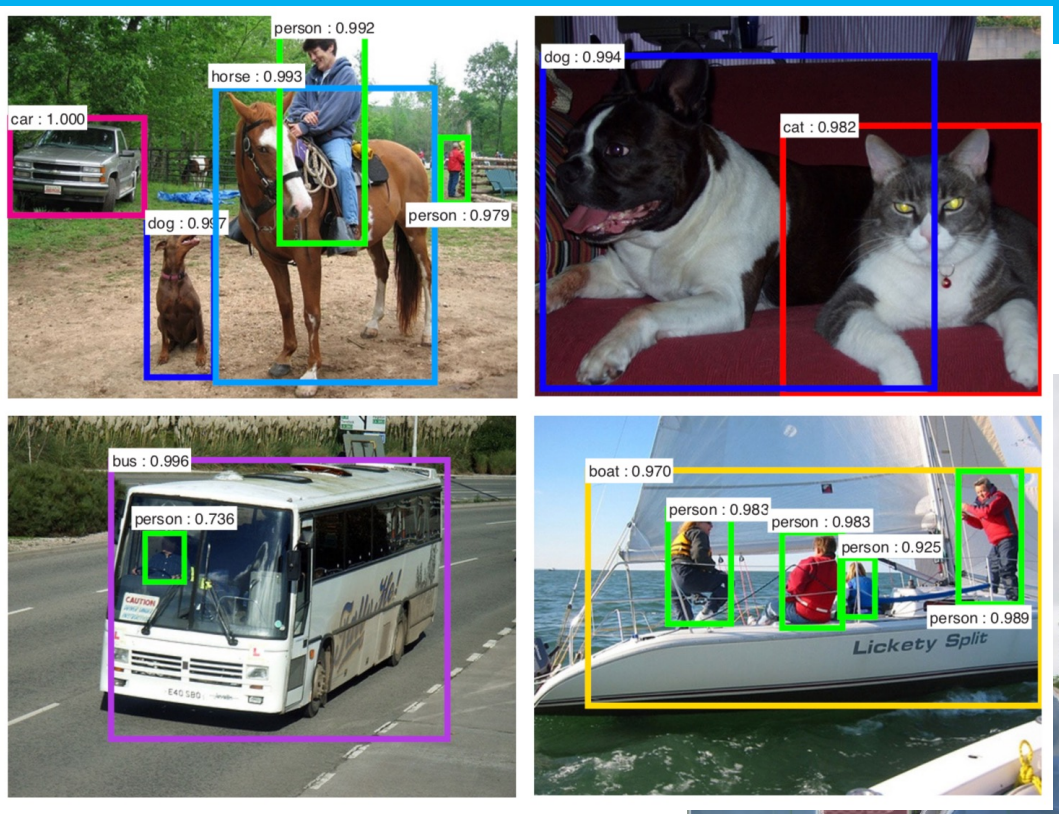
# Applications : vision

- Segmentation sémantique

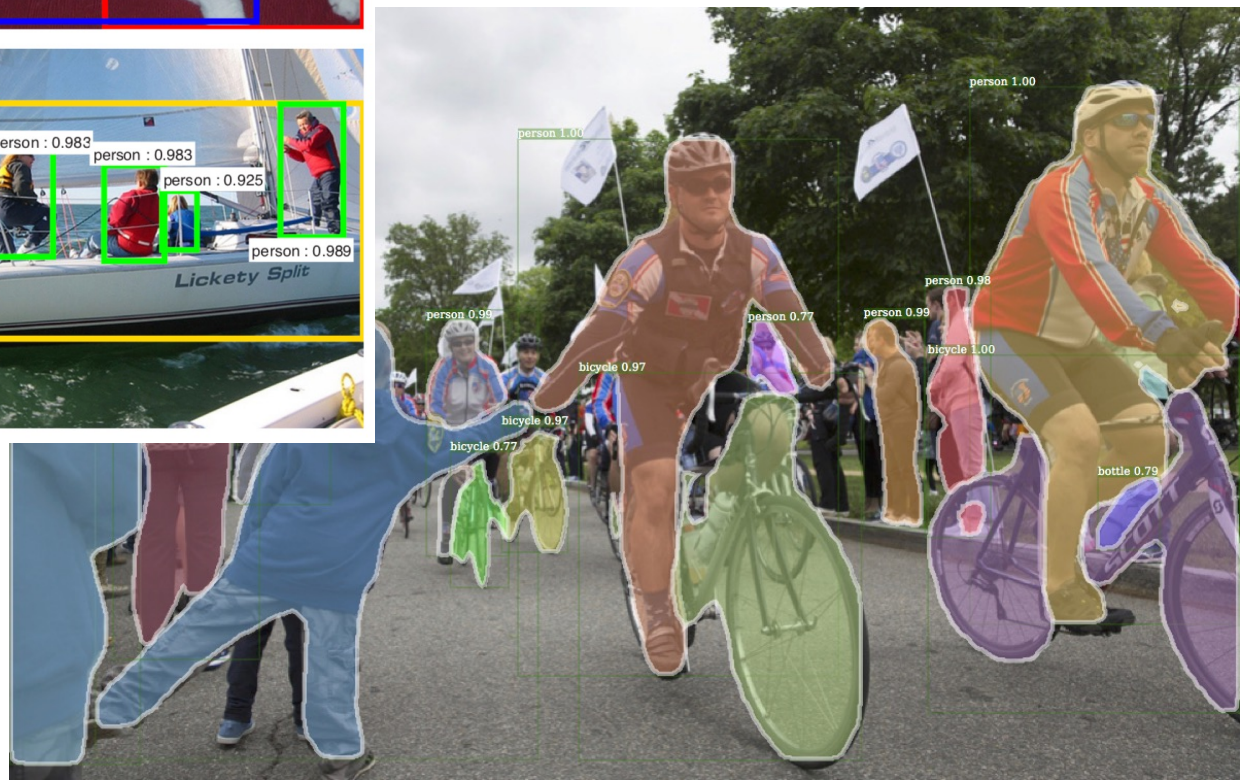



https://selfdrivingcars.mit.edu/segfuse/

# Applications




Faster R-CNN [S. Ren et al. 2015]




Mask R-CNN [K. He et al. 2017, ICCV]

# Applications

- Génération automatique de légendes

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

A. Karpathy, L. Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Description, CVPR, 2015

# Applications

- Visual Question Answering

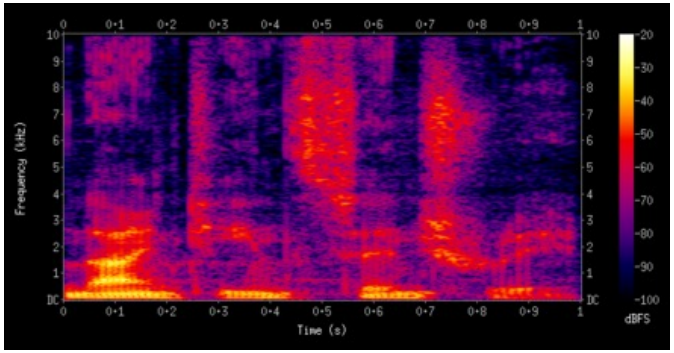
| | | |
|---|---|---|
| What kind of store is this? | bakery<br>bakery<br>pastry | art supplies<br>grocery<br>grocery |
| Is the display case as full as it could be? | no<br>no<br>no | no<br>yes<br>yes |

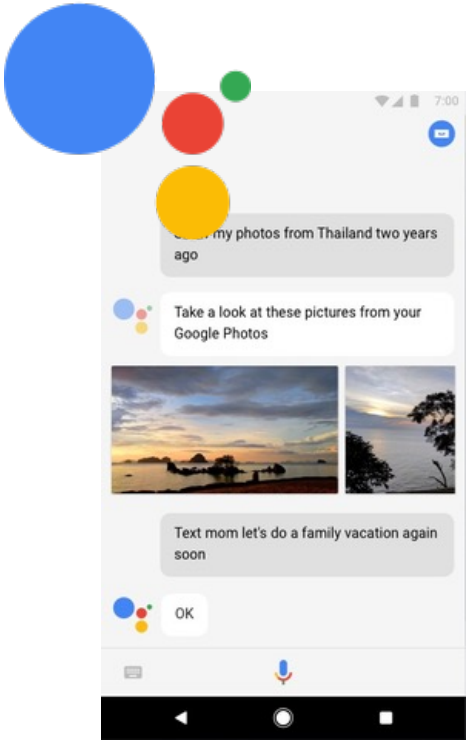

| | | |
|---|---|---|
| How many bikes are there? | 2<br>2<br>2 | 3<br>4<br>12 |
| What number is the bus? | 48<br>48<br>48 | 4<br>46<br>number 6 |

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, D. Parikh, VQA: Visual Question Answering, CVPR, 2015 SUNw:Scene Understanding workshop

# Applications

- Assistants personnels : reconnaissance vocale -- NLP (Natural Language processing) -- synthèse vocale



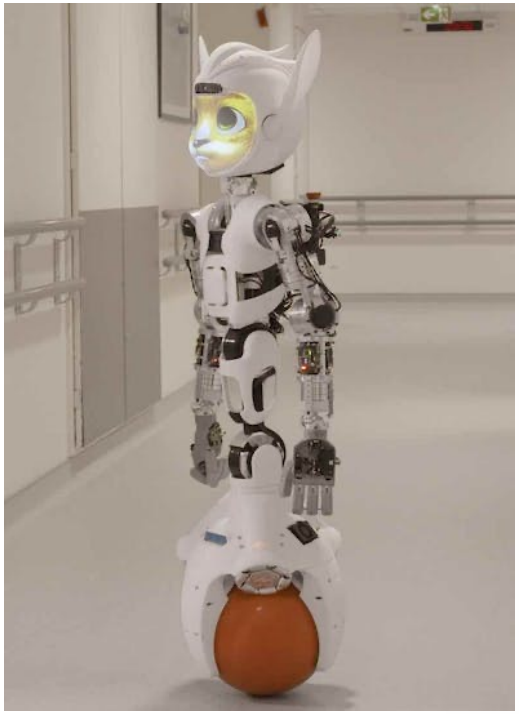





Images: Wikipédia

# Applications

- Robotique
- Véhicules intelligents/autonomes

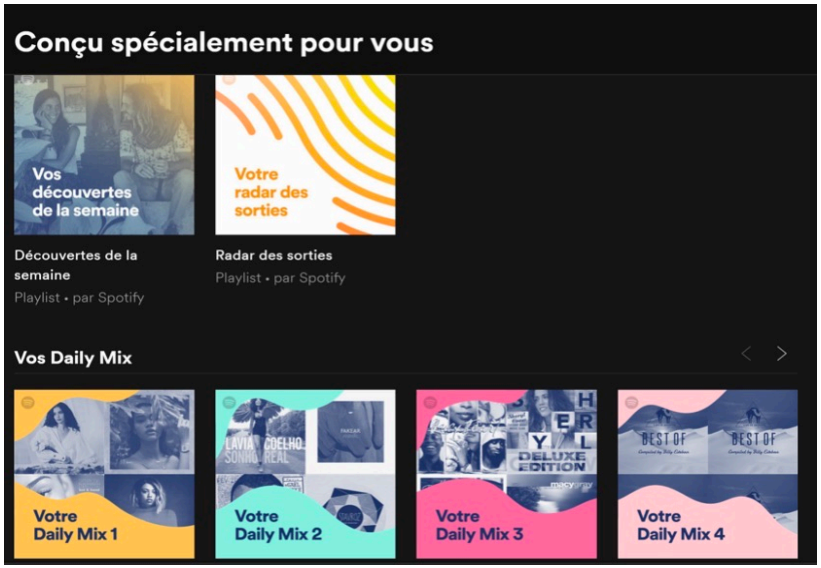


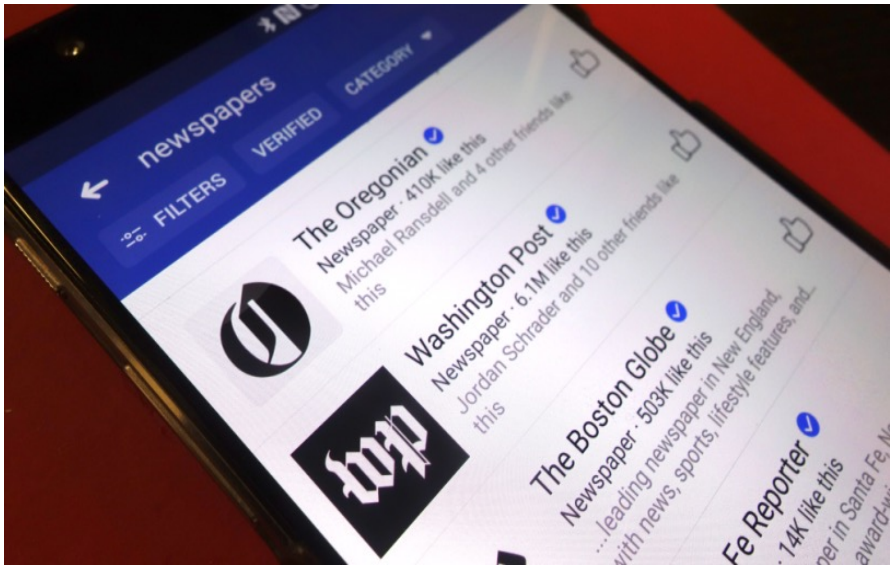

[ https://enchanted.tools/robot ]

[https://waymo.com]

# Applications

- Personnalisation des contenus
- Ciblage de la publicité
- Fake news



# Applications

DeepL | Traducteur | Linguee | DeepL Pro | Blog | Infos

Traduire **français** (langue identifiée)

Le cours d'apprentissage profond

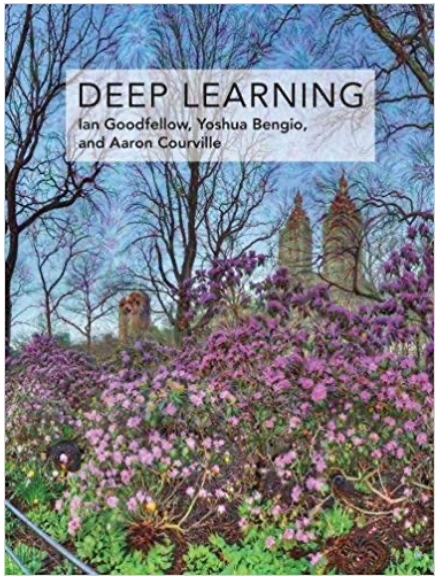
Traduire le document

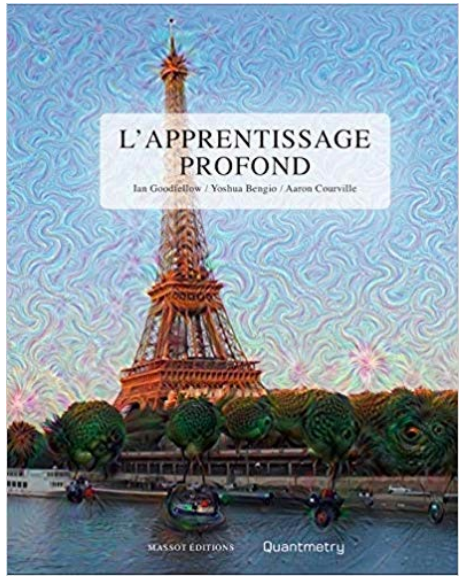
Traduire en **anglais**

The deep learning course

Autres traductions :

The profound learning course



# Applications

Figure 10: Top: Our CELEBA-HQ results. Next five rows: Nearest neighbors found from the training data, based on feature-space distance. We used activations from five VGG layers, as suggested by Chen & Koltun (2017). Only the crop highlighted in bottom right image was used for comparison in order to exclude image background and focus the search on matching facial features.

Génération automatique d'images

[Karras et al. ICLR 2018]

https://research.nvidia.com/publication/2017-10_Progressive-Growing-of

# Applications

- Transfert de style

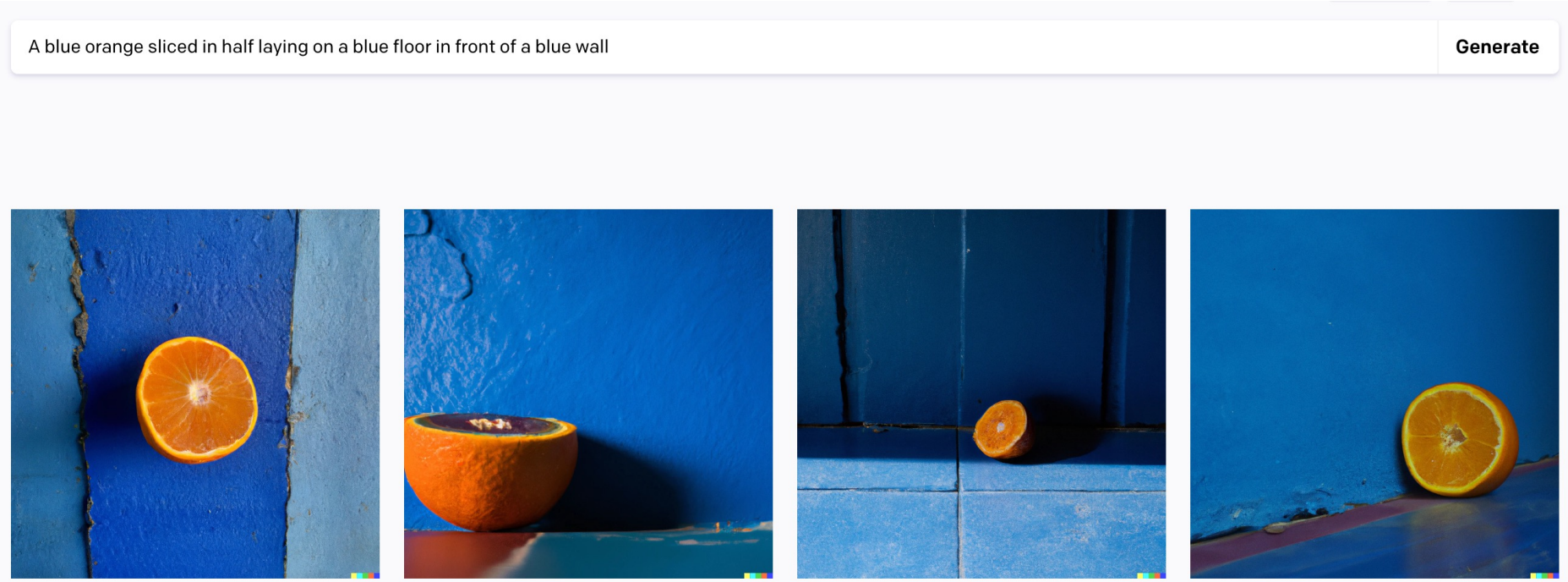

[Gatys et al. CVPR 2016]

# Applications

https://openai.com/dall-e-2/

# Applications

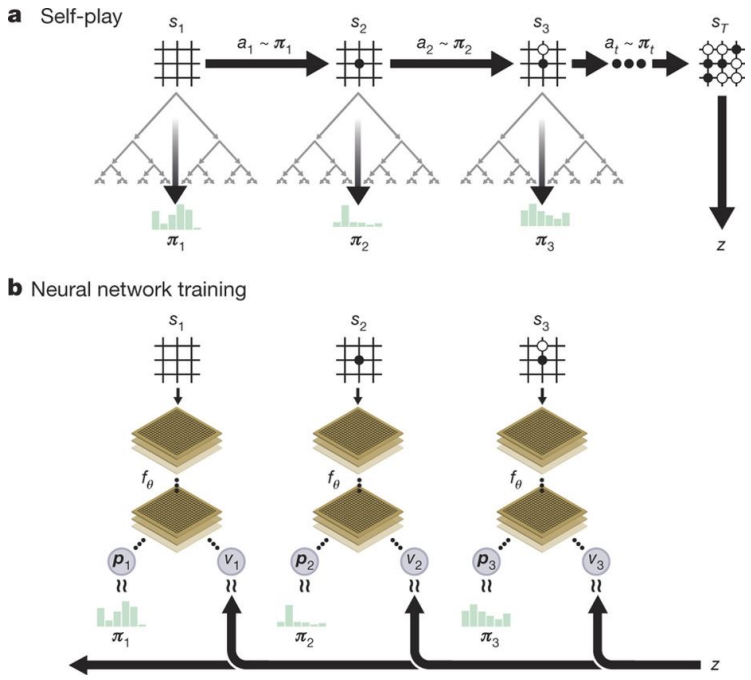
- Génération d'images à partir d'une description textuelle



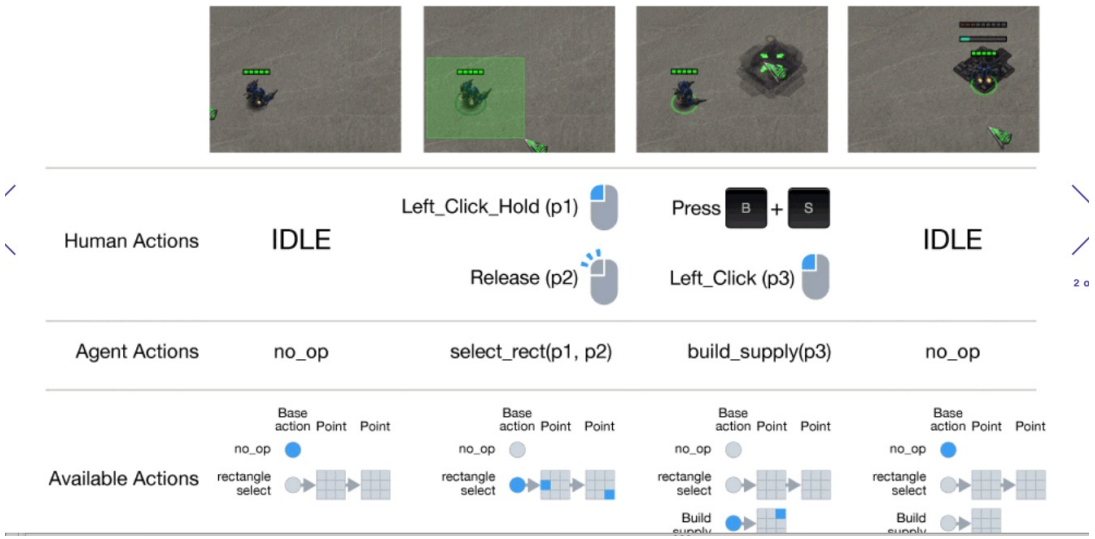

https://openai.com/dall-e-2/

# Applications

- Jeux (jeu de Go, Starcraft, Poker…)




[AlphaGo Nature 2017]

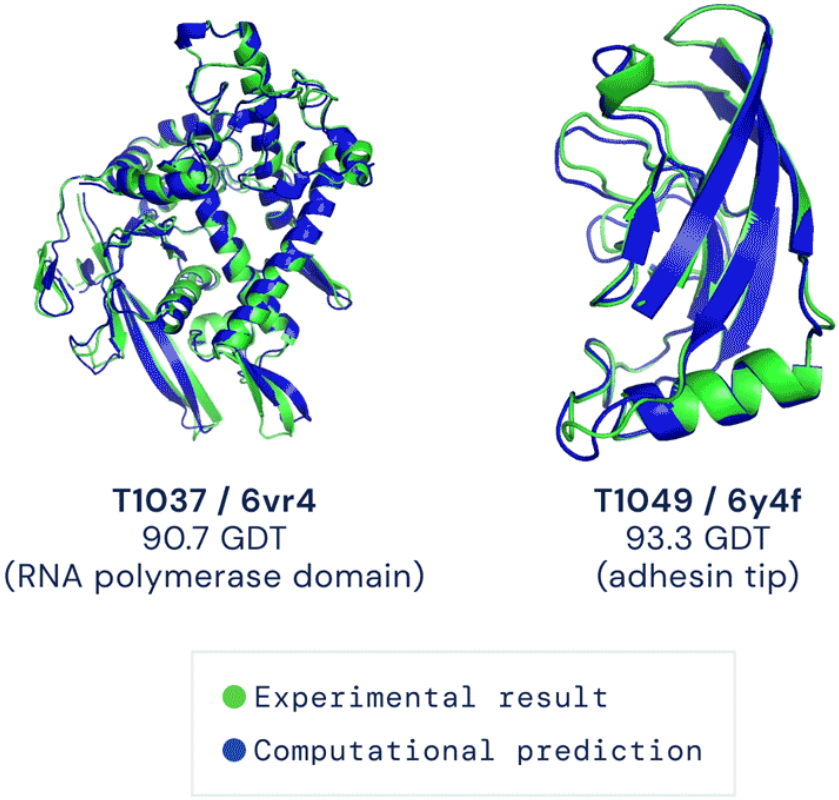



Starcraft 2

[ https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/ ]

# Applications

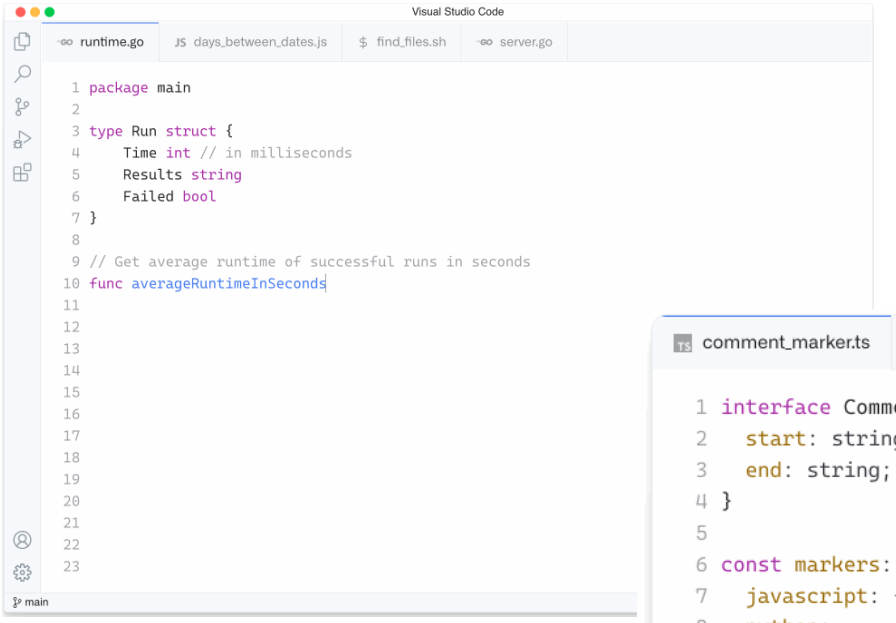
- Prédiction de la structure des protéines à partir de la sequence d'acides aminés

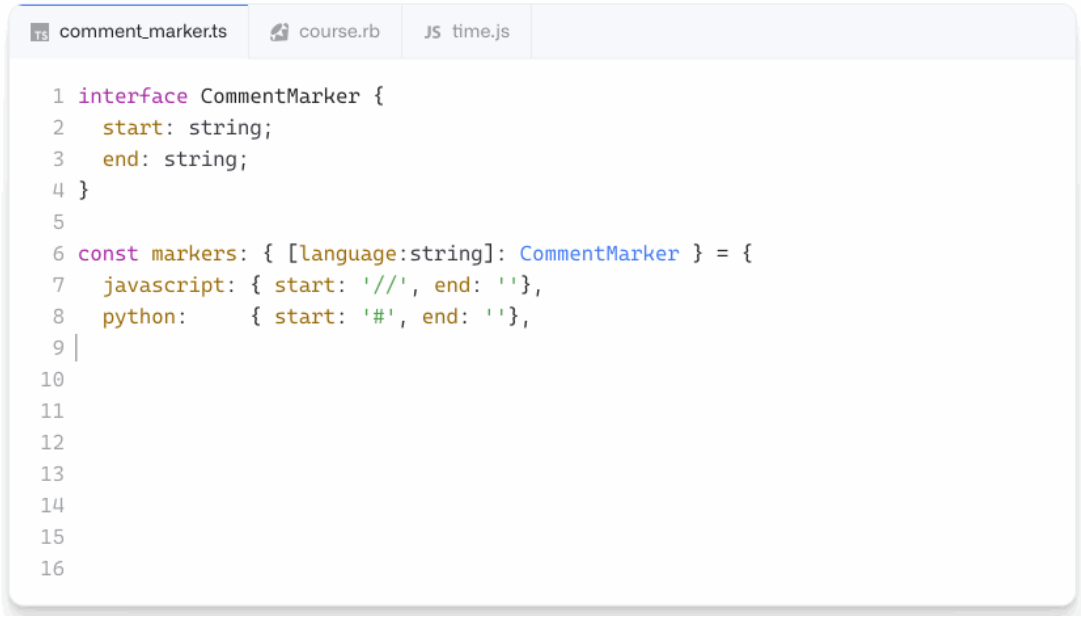



https://github.com/deepmind/alphafold

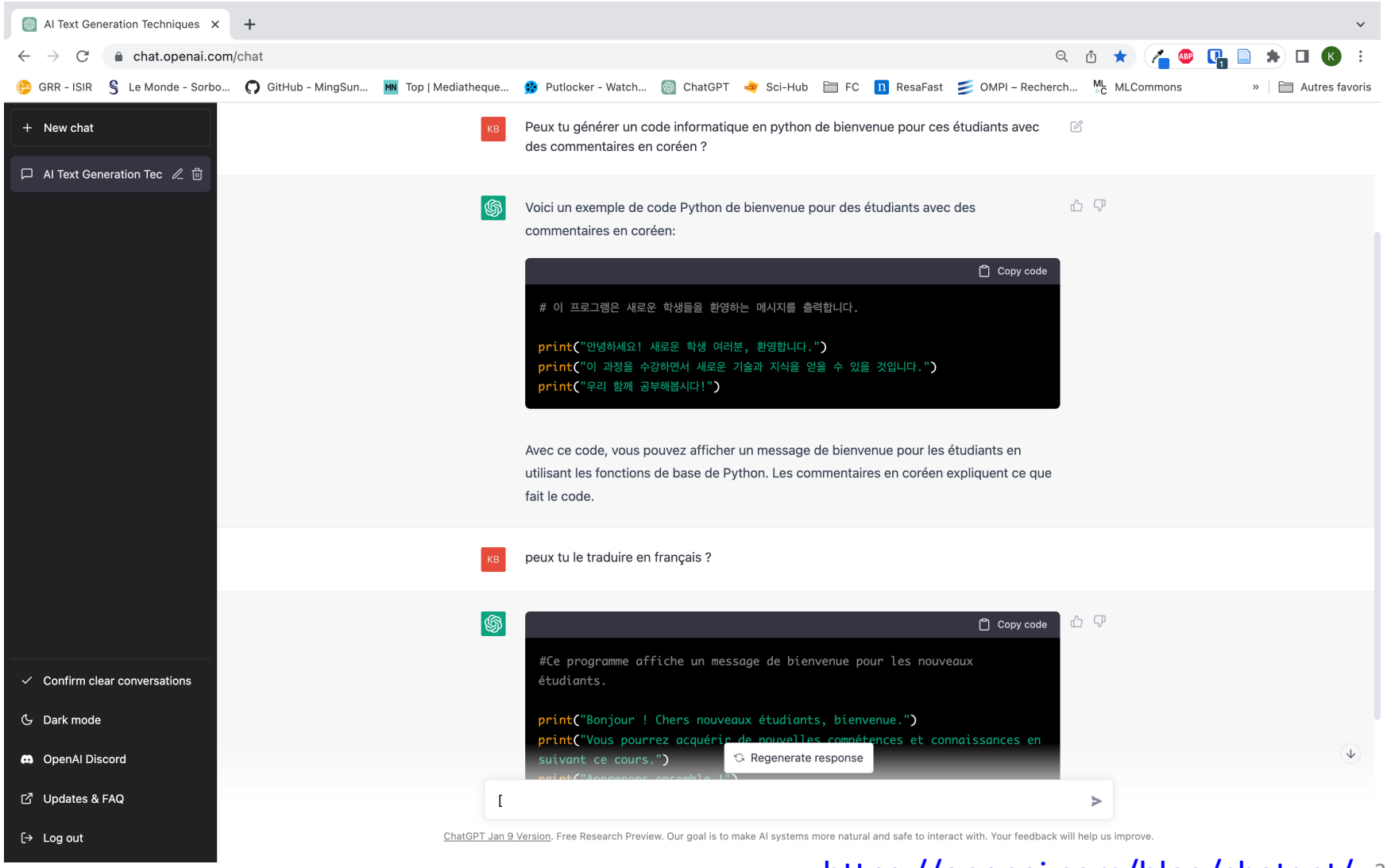
# Applications

- Github Copilot : génération automatique de code



# Applications



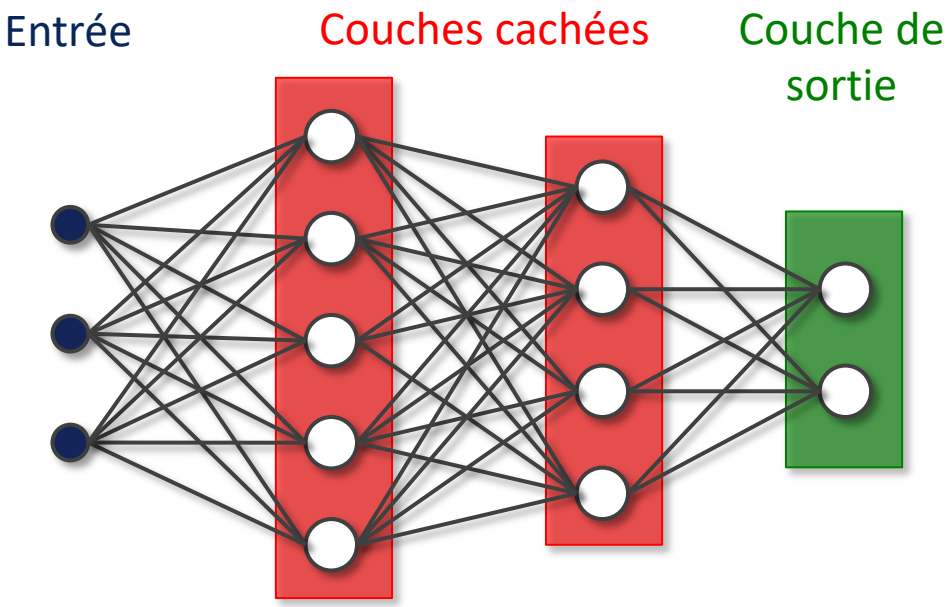

https://openai.com/blog/chatgpt/

# Plan

- Introduction (historique et applications)
- Qu’est-ce que l’apprentissage profond
- Comment créer un réseau de neurones avec Tenforflow

# Qu’est-ce que l’apprentissage profond

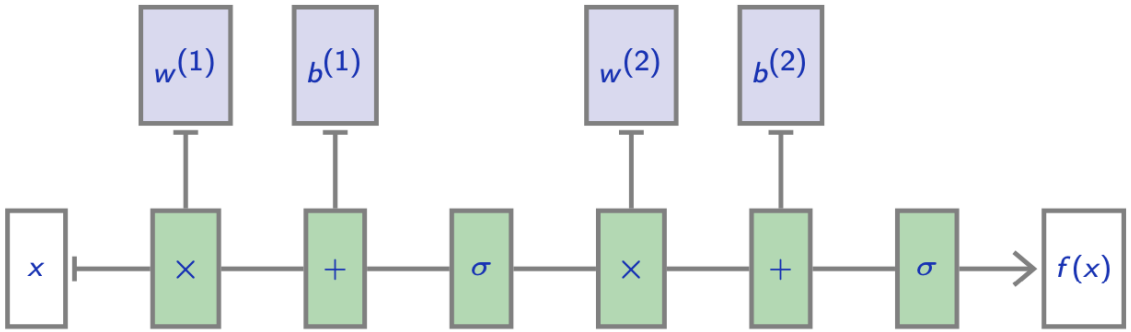
- Principe général : augmenter le nombre de couches dans le réseau
  - Projections **non linéaires** successives
  - Apprentissage conjoint de l’espace de **représentation** et de la **fonction de décision**



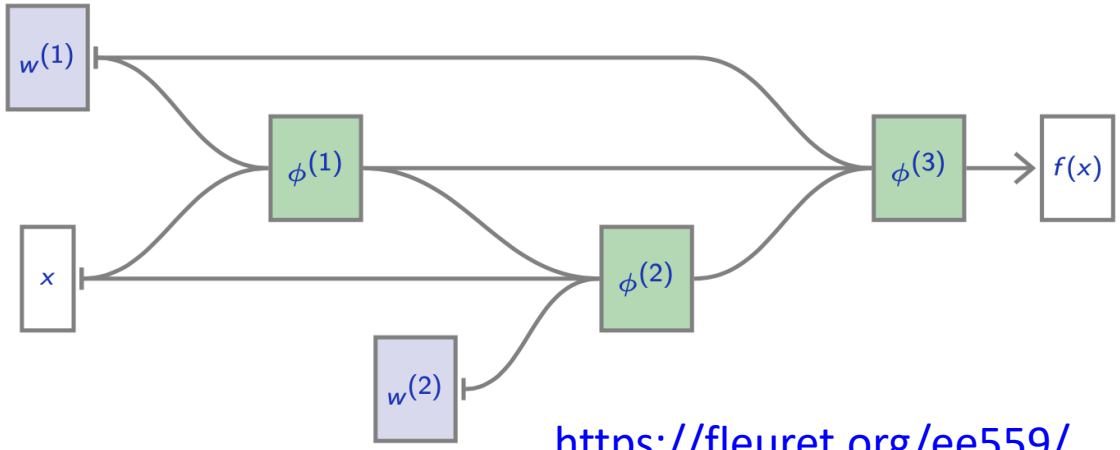
# Qu'est-ce que l'apprentissage profond

- Graphe Orienté Acyclique (DAG)

MLP Classique

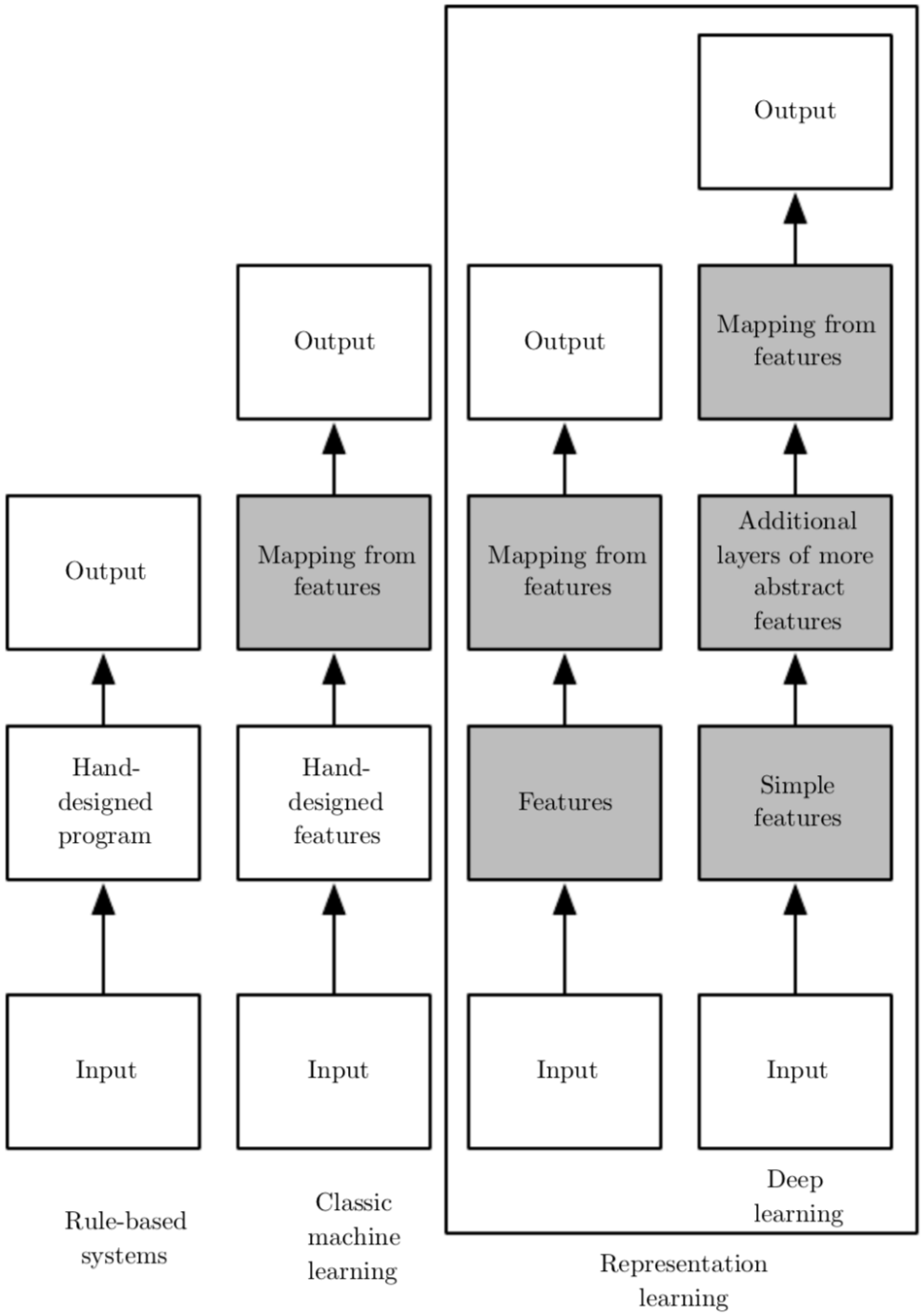



DAG



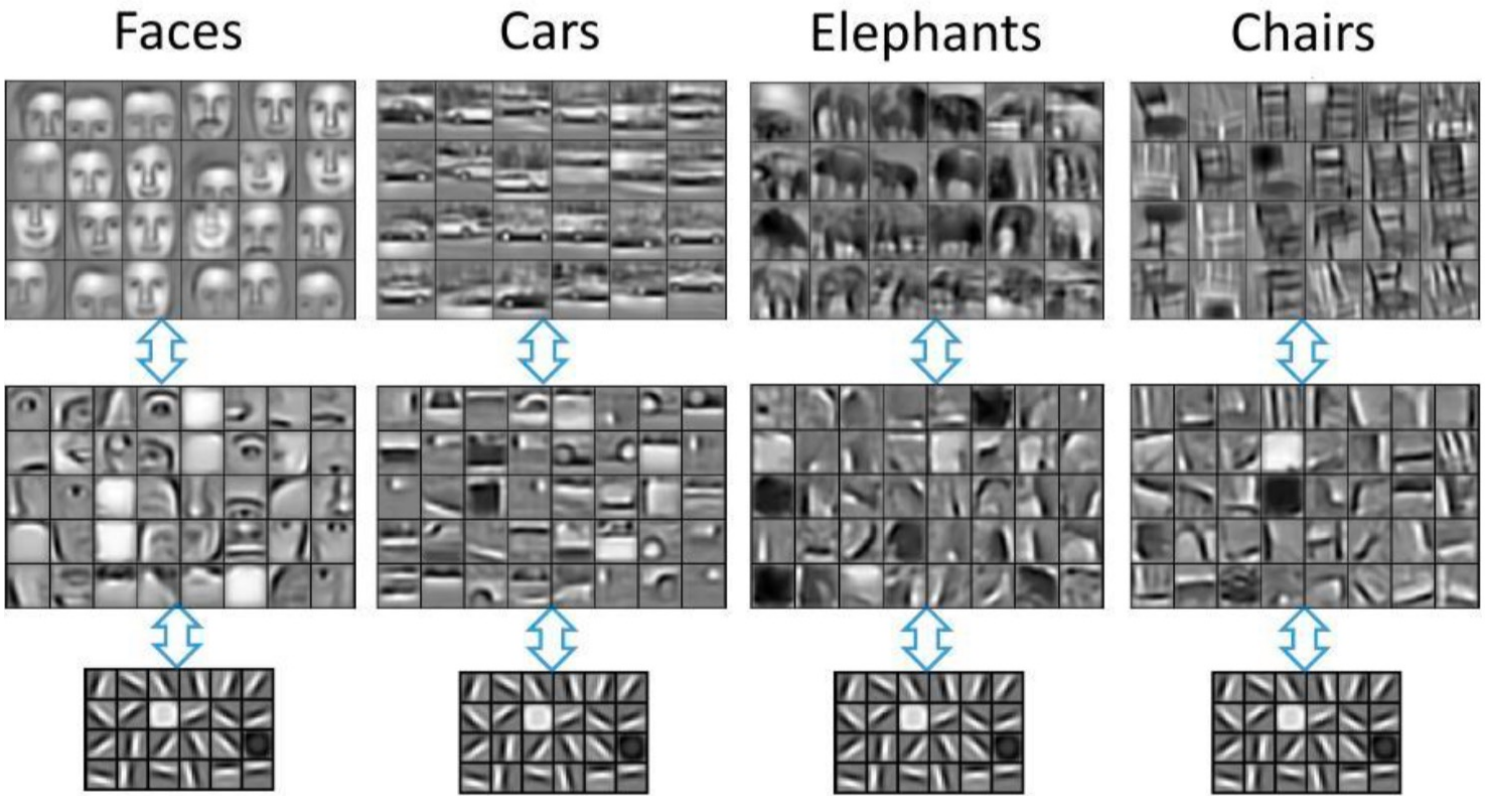

https://fleuret.org/ee559/

# L'apprentissage profond



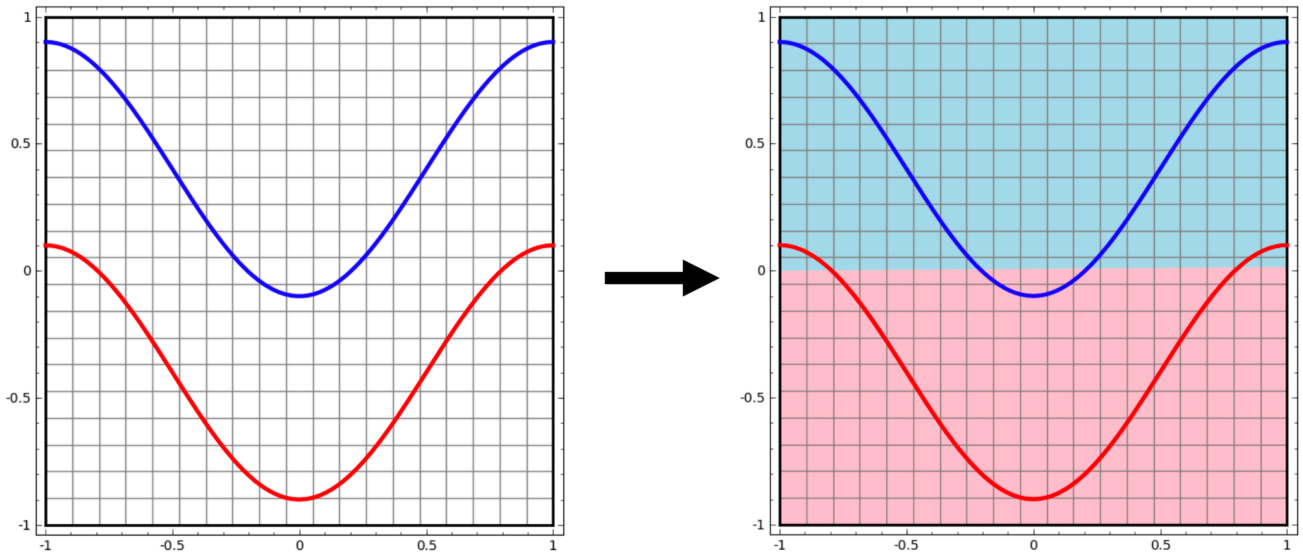

[Goodfellow et al 2016]

# Représentation hiérarchique

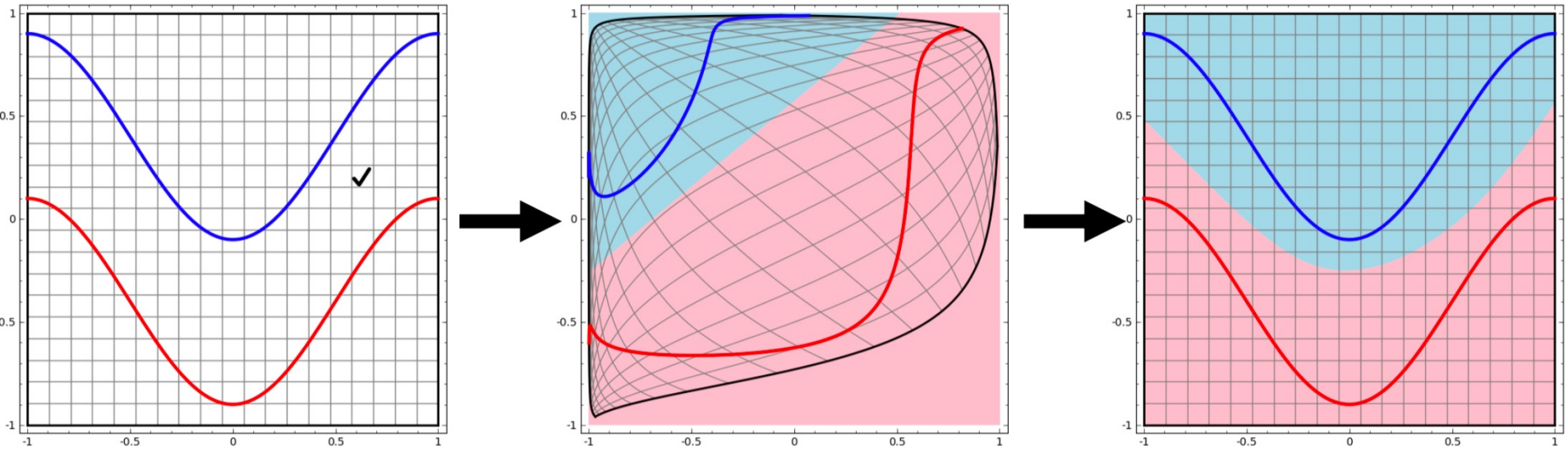



[Lee et al. ICML 2009]

# Importance de la représentation



# Importance de la représentation



# Les modèles linéaires sont limités

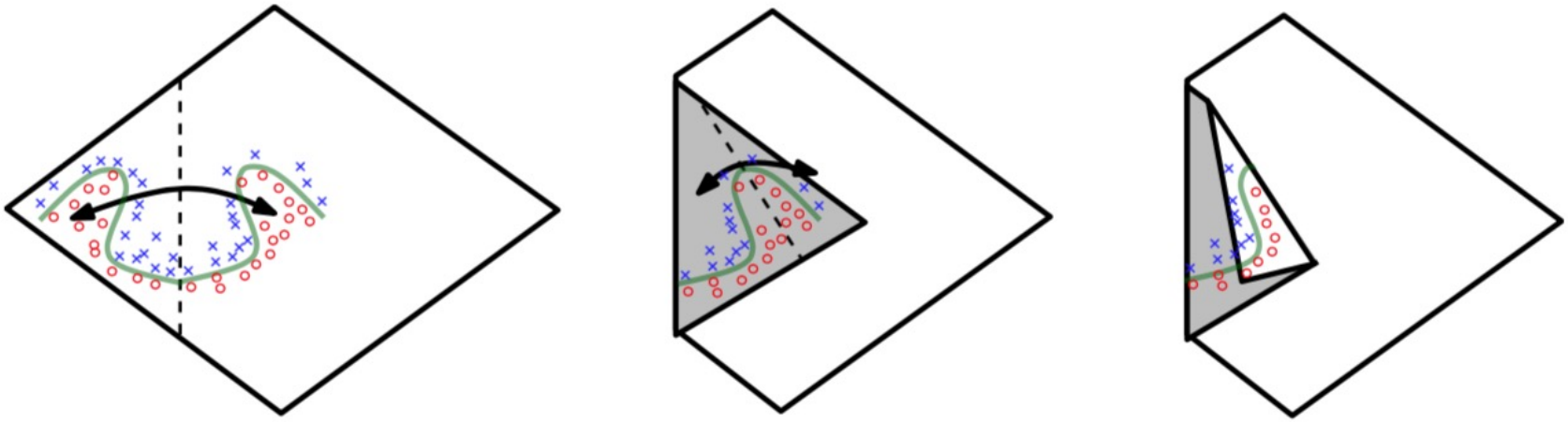
- Avantages des modèles linéaires
  - Stabilité
  - Facilité de calcul
  - Gradients constants
- Mais représentation limitée.
  - Empiler des modèles linéaires ne marche pas, pourquoi ?

# Les modèles linéaires sont limités

- Avantages des modèles linéaires
  - Stabilité
  - Facilité de calcul
  - Gradients constants
- Mais représentation limitée.
  - Empiler des modèles linéaires ne marche pas
    - La composition de fonctions linéaire est une fonction linéaire (produit de matrice)
    - Il faut introduire des non-linéarités entre les couches

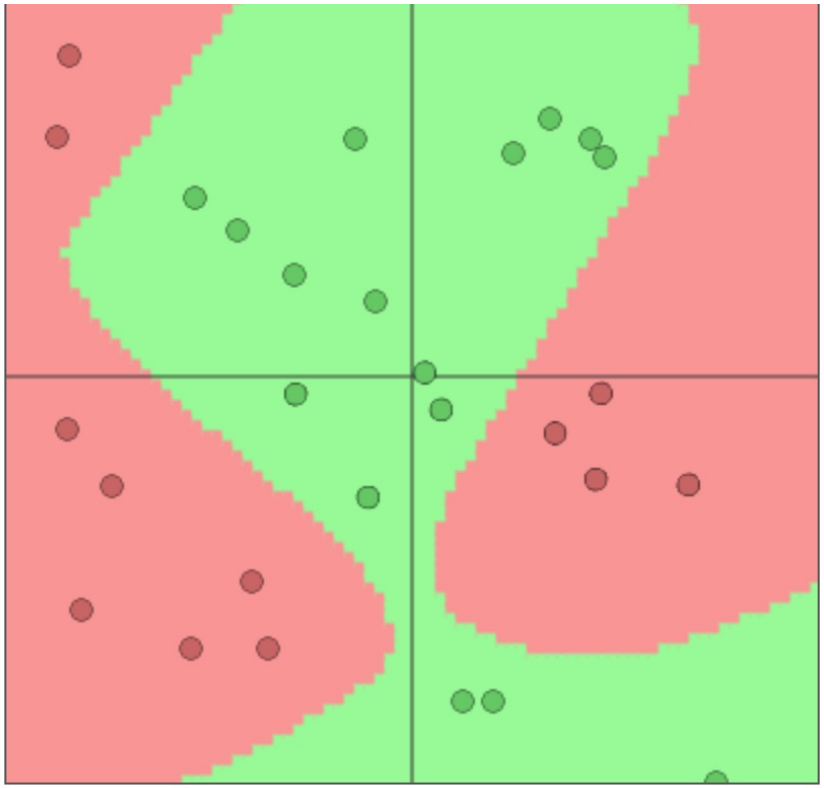
# Pourquoi des réseaux profonds ?

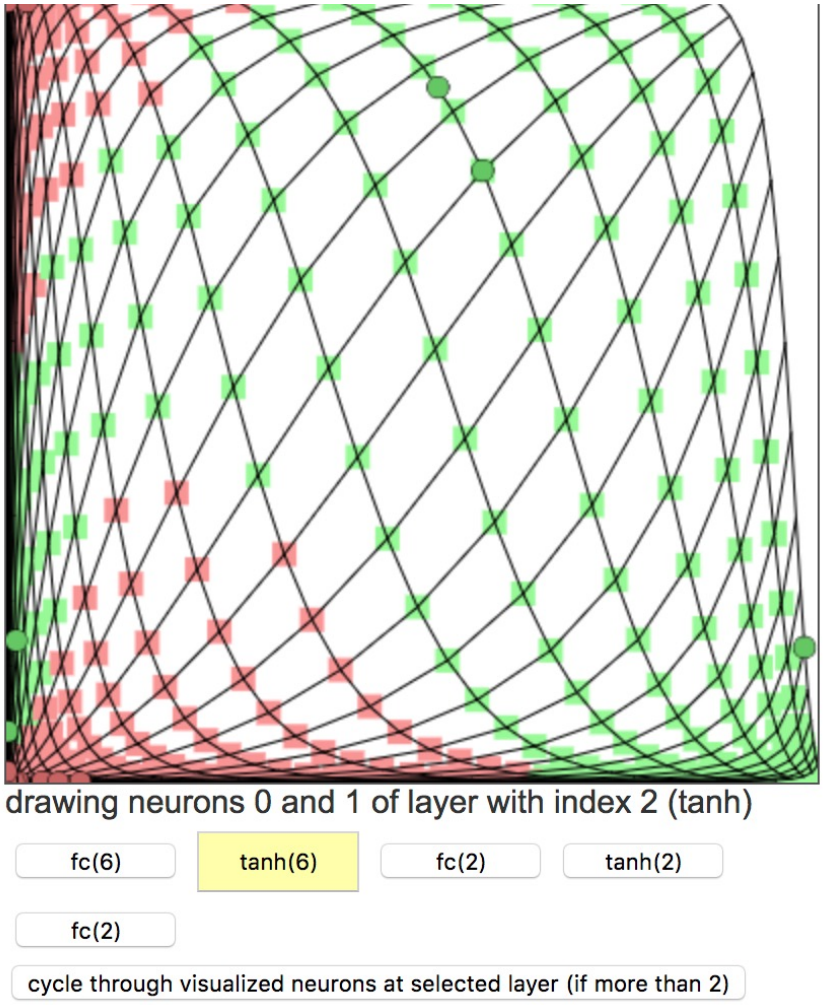
- Une illustration intuitive avec des fonctions d'activation « valeur absolue »



[Montufar et al. 2014]

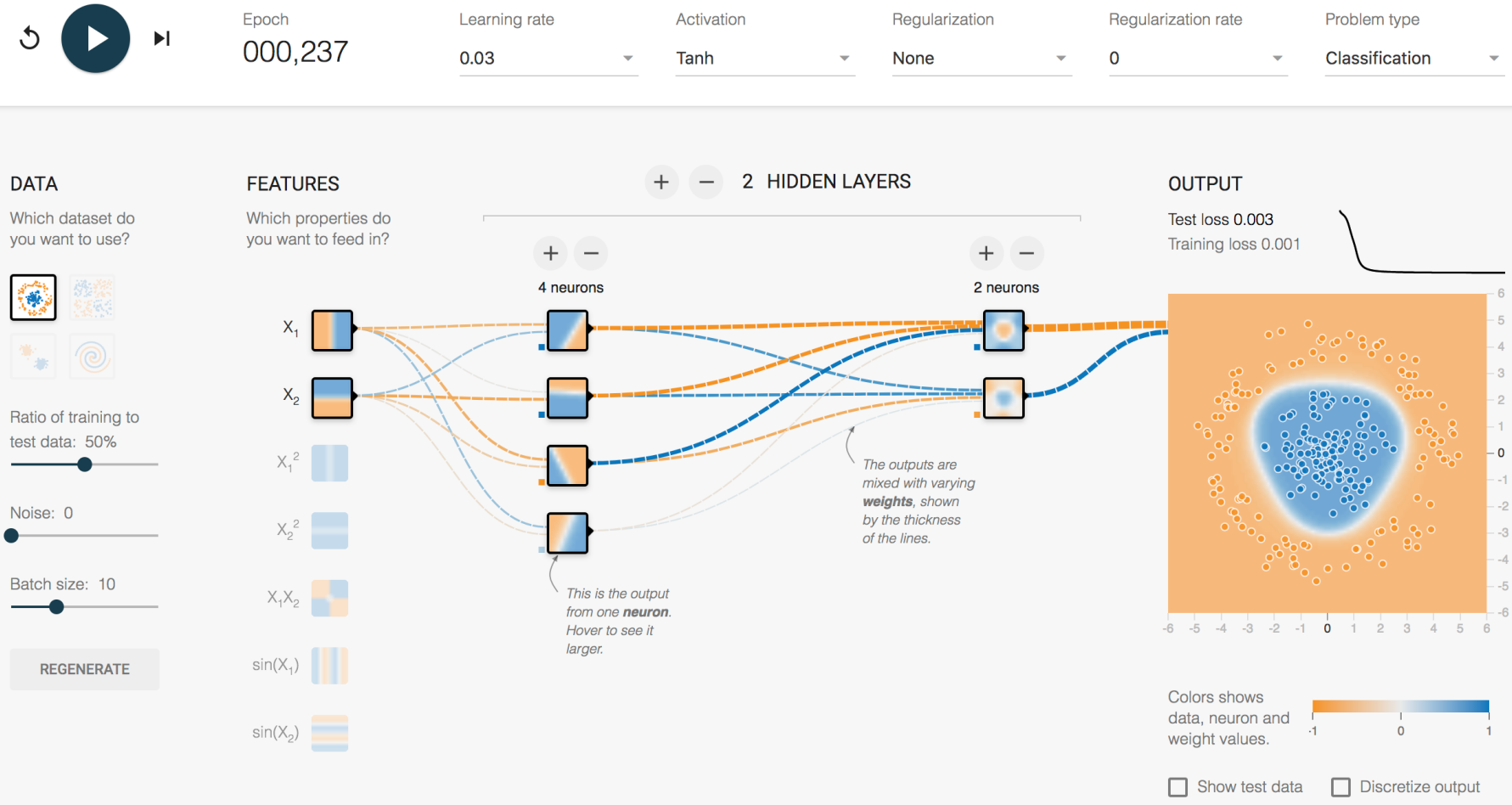
# Demonstration interactive




https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html

# Demonstration interactive




http://playground.tensorflow.org

# Plan

- Introduction (historique et applications)
- Qu’est-ce que l’apprentissage profond
- Comment créer un réseau de neurones avec TensorFlow

# Environnement




API


Edition

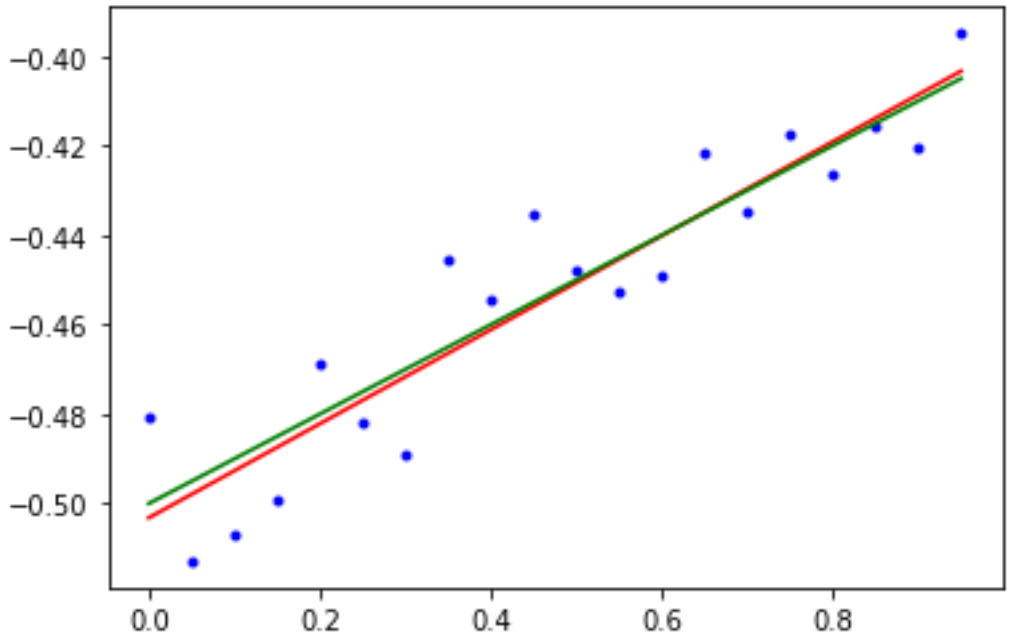


colab.research.google.com

Installation

# Premier réseau en Tensorflow

- Notebook : **01_regression_lineaire.ipynb**



# Calcul du gradient

- Méthodes de calcul du gradient

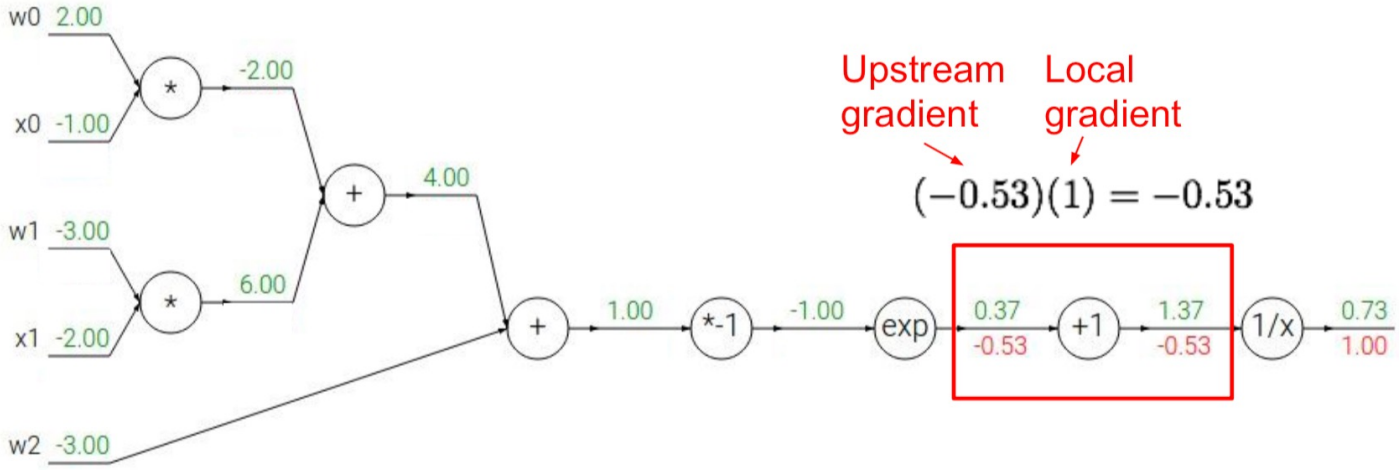
$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Numérique
  - Lent, approximation, implémentation facile
- Analytique
  - Rapide, exact, risque d'erreurs
- Différentiation automatique [1]
  - Rapide, exact, implémentation facile
  - Par exemple : AutoGrad

[1] A. G. Baydin, B. Pearlmutter, R. Andreyevich J. Siskind, (2018). Automatic differentiation in machine learning: A survey. JMLR. 18. 1-43.
Exemple simple en python : https://github.com/karpathy/nn-zero-to-hero/tree/master/lectures/micrograd

# Calcul du gradient

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$




$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Tiré de: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture04.pdf

# Exercices

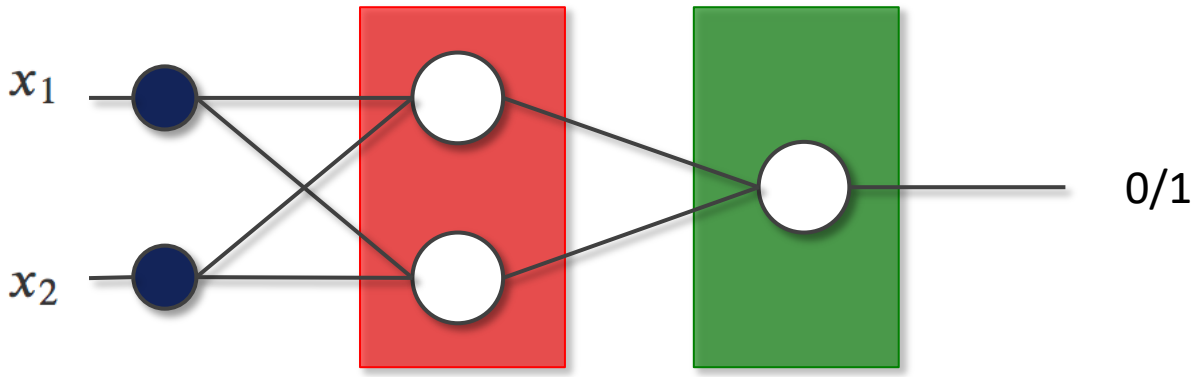
- Autograd : package python pour la différentiation automatique (DA)

- Tutorial:
  - https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html

- Exercice : 02_autograd.ipynb

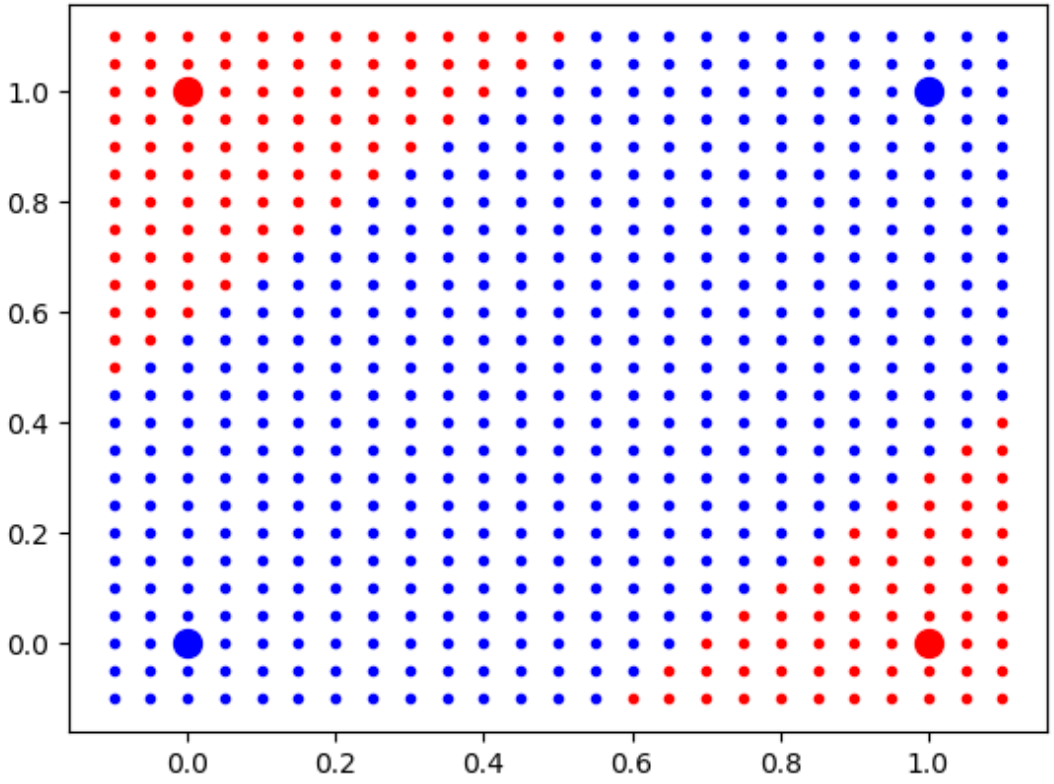
# Exercice

- Le problème du Xor
  - Couche cachées : 2 neurones (fonction d'activation : sigmoide)
  - Une couche de sortie avec un neurone (linéaire)



# Exercice

- Compléter le notebook : 03_XOR_lightning.ipynb

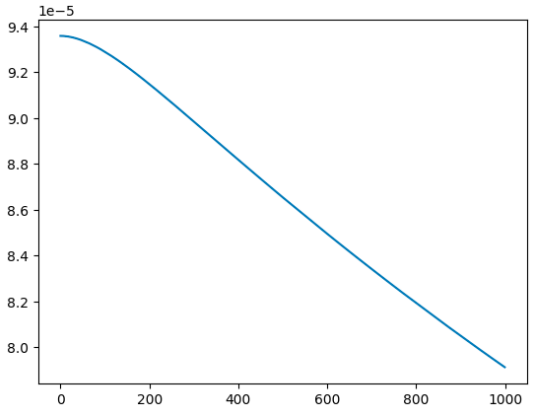


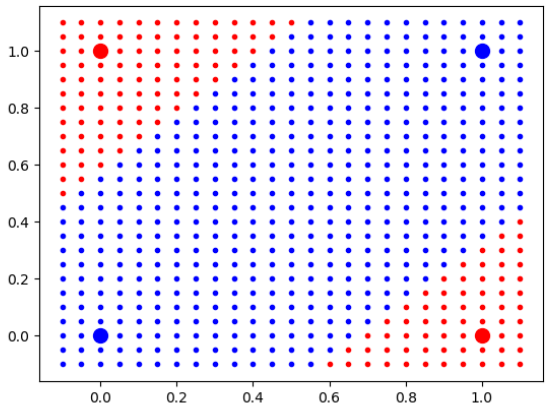

Pour vous aider : https://lightning.ai/docs/pytorch/stable/

# Exercice

- En utilisant lightning, la boucle d'apprentissage est encapsulée dans la fonction fit()

- Compléter le notebook : 03_XOR_pytorch.ipynb
  - La boucle d'apprentissage devra être explicite
  - Enregistrer les valeurs successives de la fonction de coût et les afficher




Pour vous aider : https://lightning.ai/docs/pytorch/stable/