



# C4. Introduction aux réseaux de neurones profonds

**Advanced Machine Learning (MLA)**

Kévin Bailly

[kevin.bailly@sorbonne-universite.fr](mailto: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 Tensorflow

# 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)

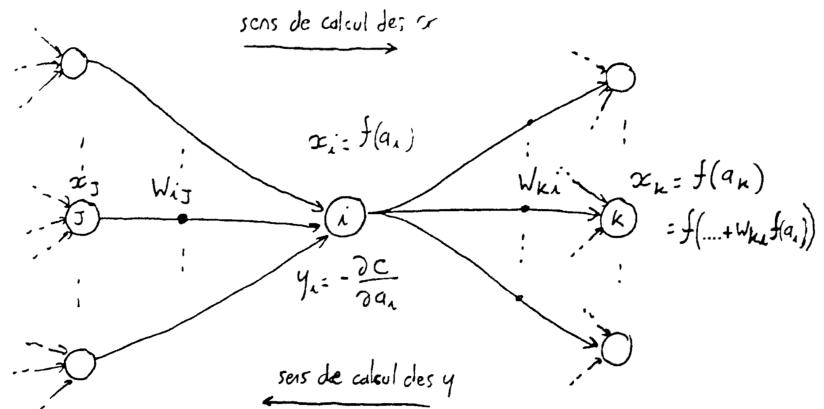
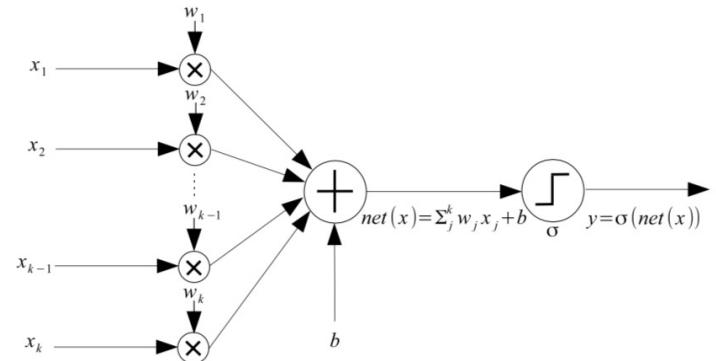


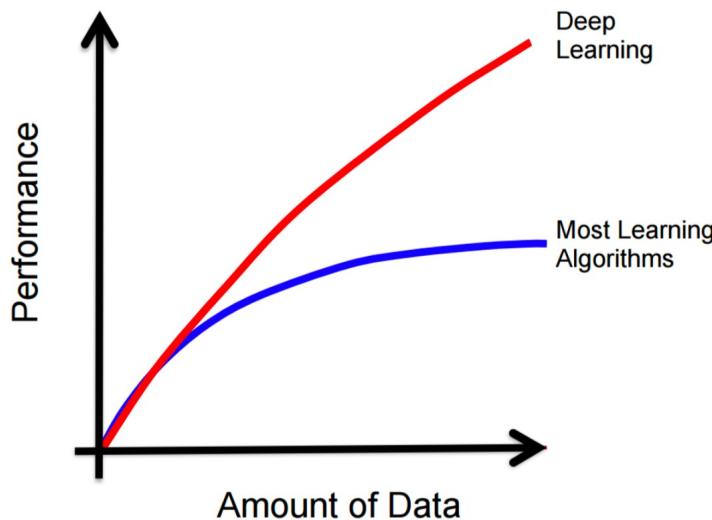
Figure 3. 4: Chaîne des dépendances dans un réseau multicouche

[LeCun, PhD 87]

# Les facteurs de la révolution

# Les facteurs de la révolution

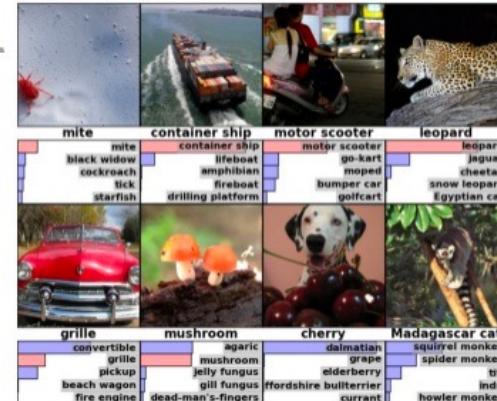
- Plus de données (labélisées)



## ImageNet Challenge

IMAGENET

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



4

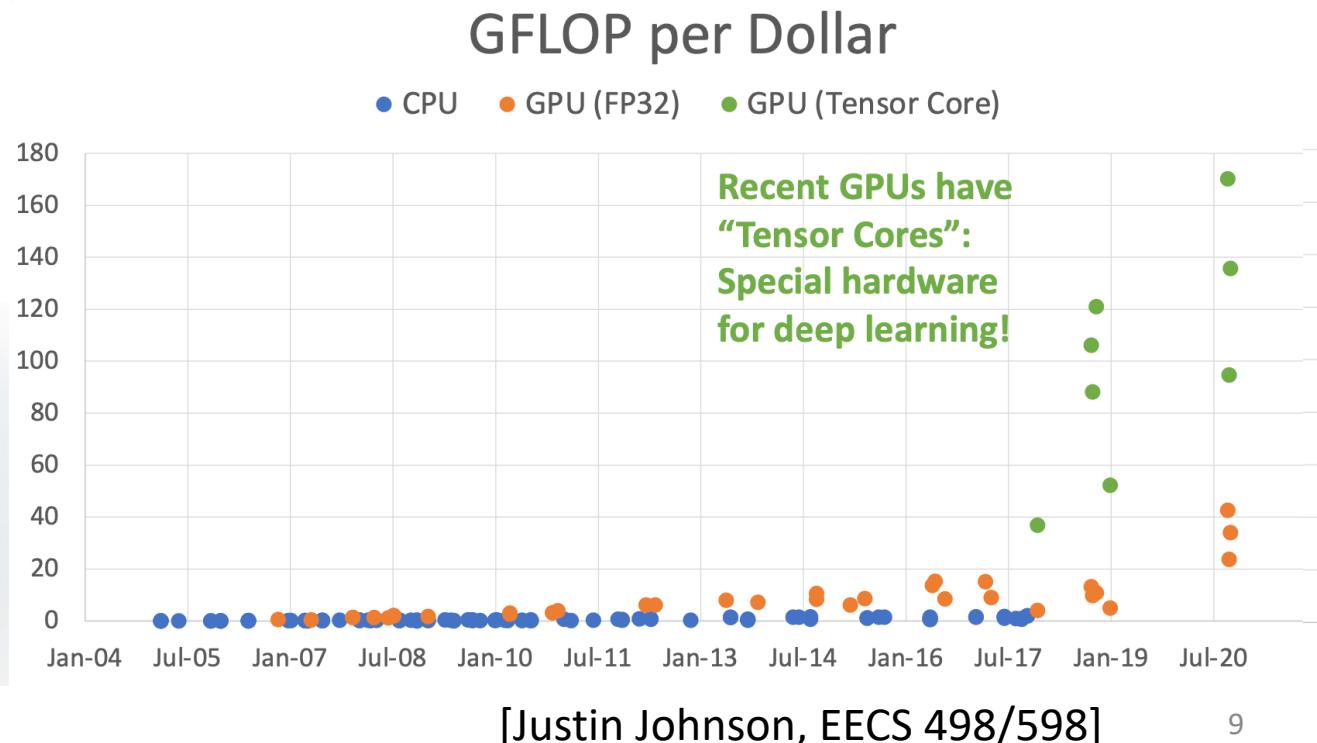
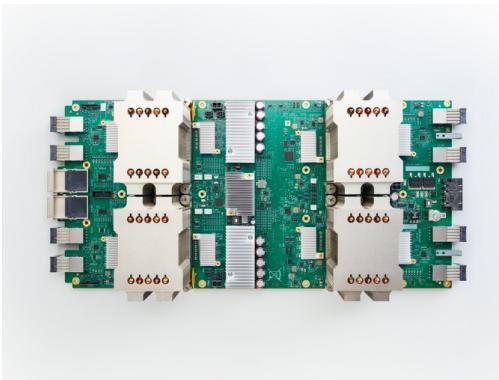
kaggle

crowdAI



# Les facteurs de la révolution

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



# 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

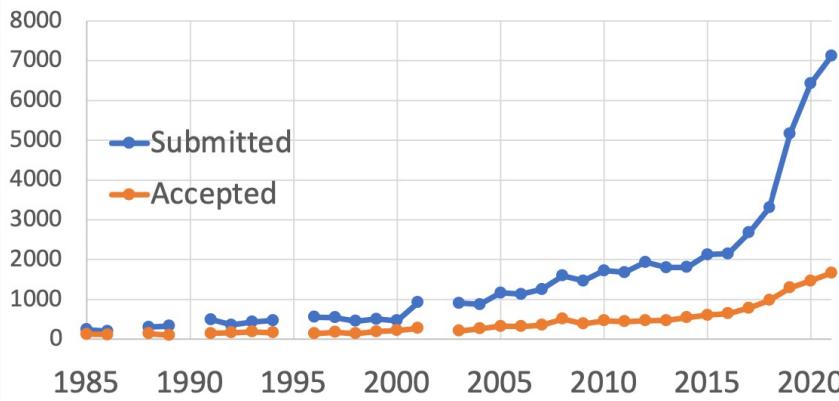
- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)
- Des outils performants, flexibles et open-sources
- Nouveaux algorithmes
  - Fonctions d'activations
  - Initialisation des paramètres
  - Méthodes d'optimisation (RMSProp, Adam)
  - Nouvelles architectures (CNN, ResNet, Inception)

# Les facteurs de la révolution

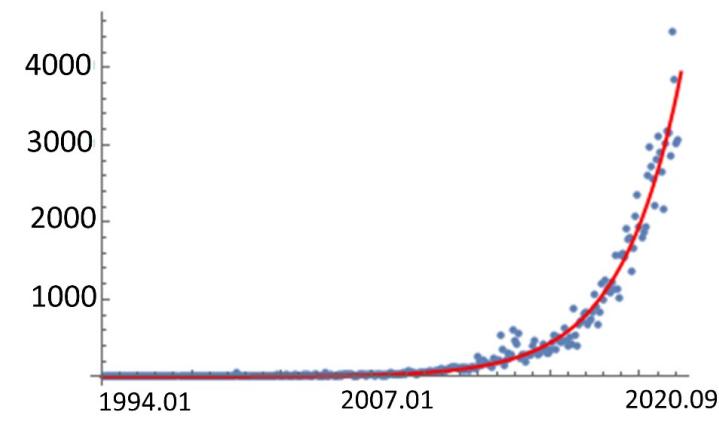
- Plus de données (labélisées)
- Plus de ressources (ex: GPU, TPU)
- Des outils performants et open-sources
- Intensification de la recherche (publique et privée)



CVPR Papers



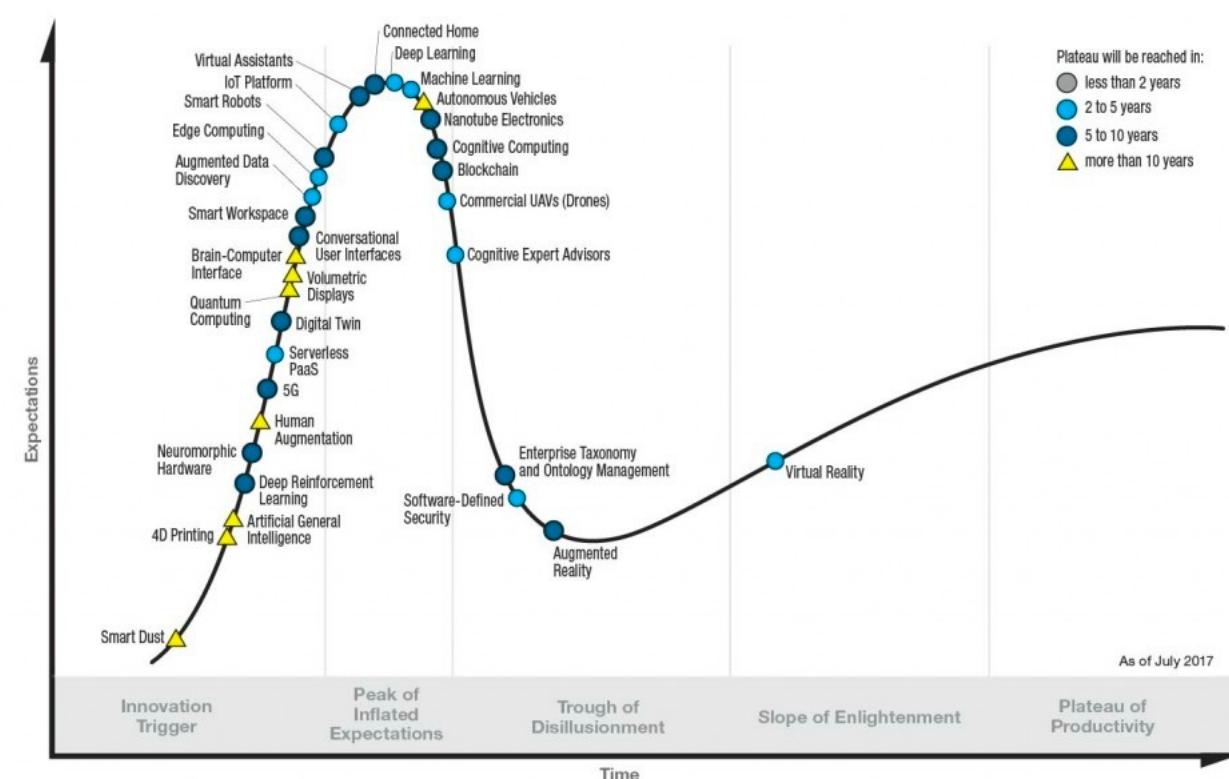
ML+AI arXiv papers per month



Publications at top Computer Vision conference

arXiv papers per month ([source](#))

# Gartner Hype Cycle 2017



[gartner.com/SmarterWithGartner](http://gartner.com/SmarterWithGartner)

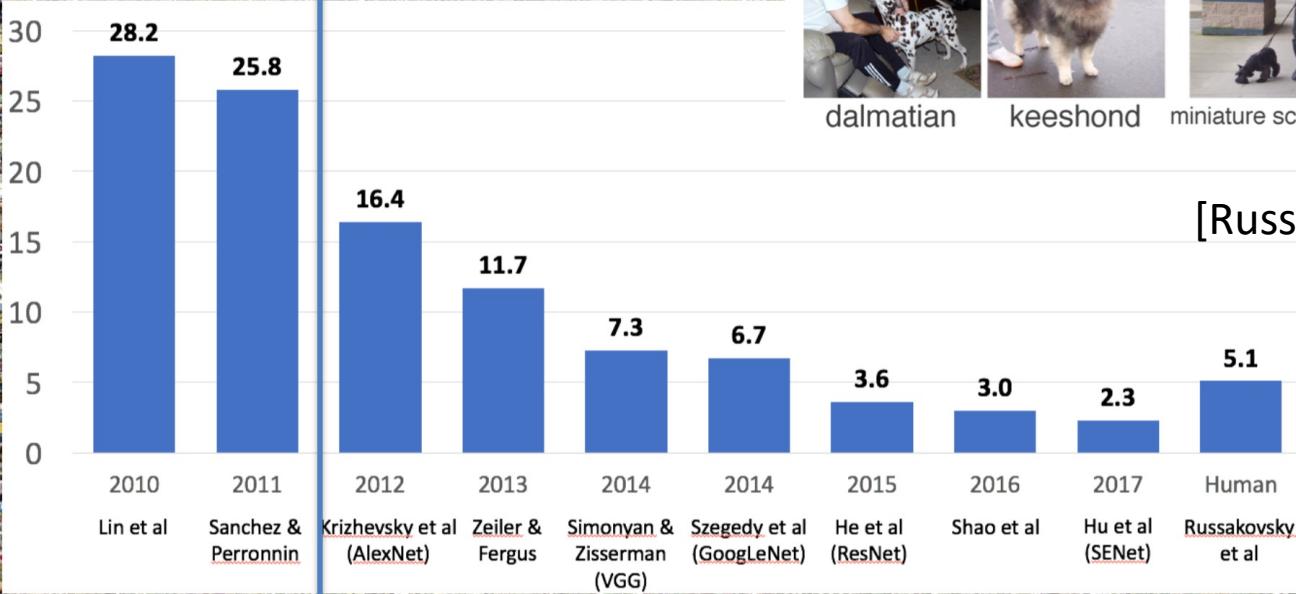
Source: Gartner (July 2017)  
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.

**Gartner**

# Applications

ImageNet Challenge  
1000 classes  
>1M d'images

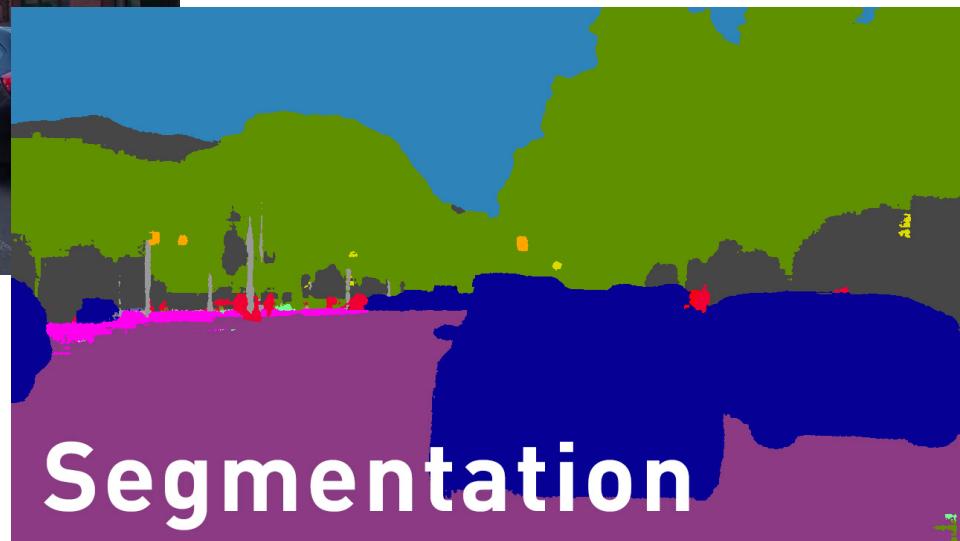
Deep learning



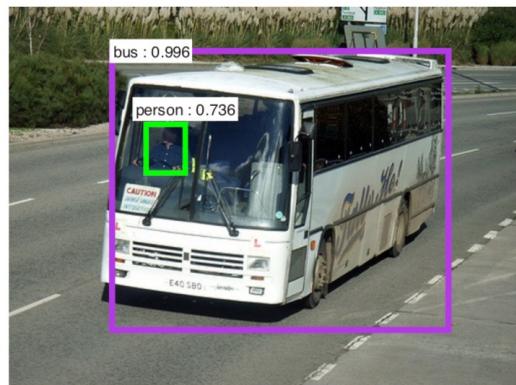
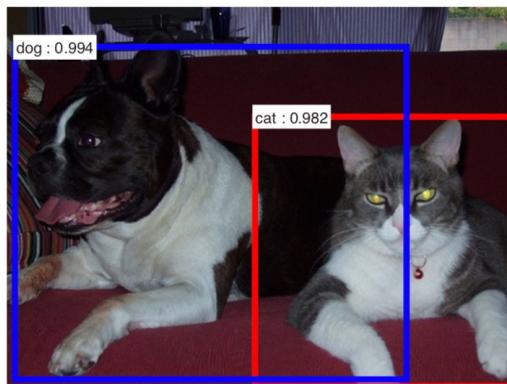
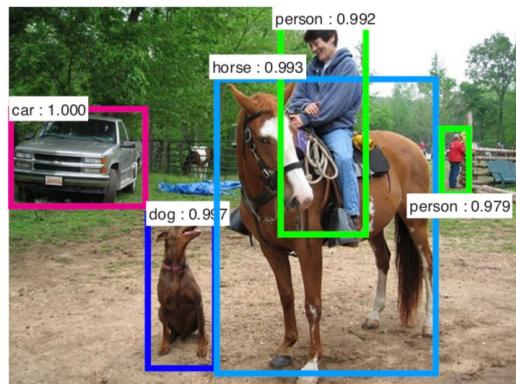
[Russakovsky et al. 2015, IJCV]

# Applications : vision

- Segmentation sémantique



# 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  
bakery  
pastry

art supplies  
grocery  
grocery

Is the display case as full as it could be?

no  
no  
no



How many bikes are there?

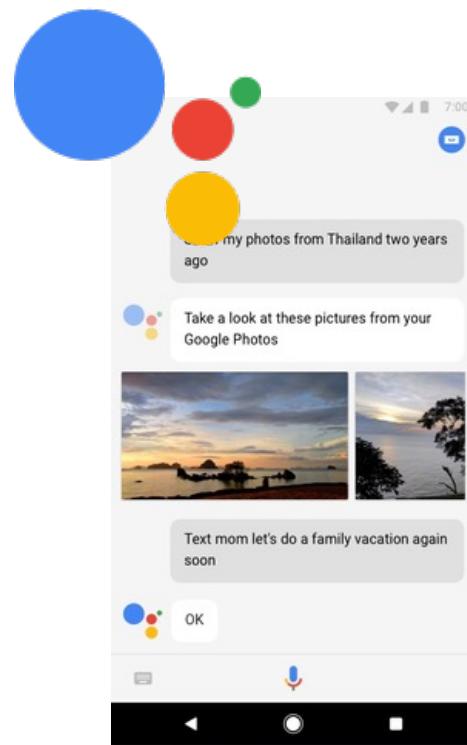
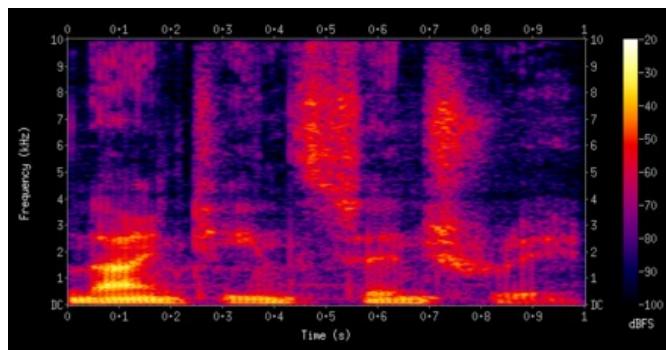
2	3
2	4
2	12

What number is the bus?

48	4
48	46
48	number 6

# Applications

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



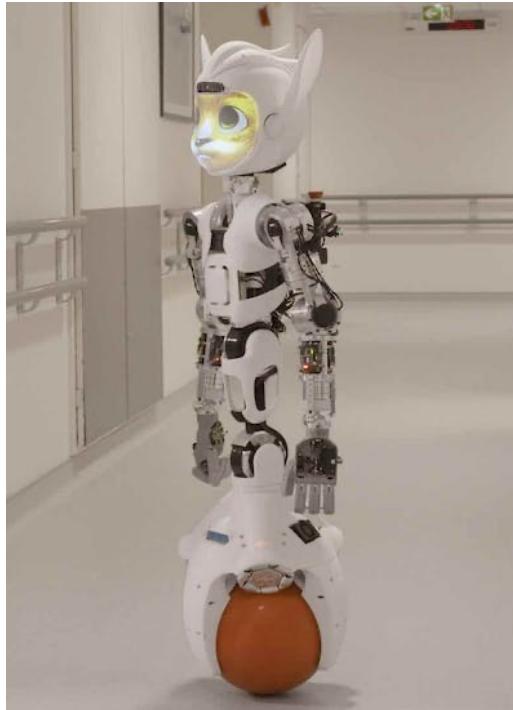
amazon echo



Images: Wikipédia

# Applications

- Robotique
- Véhicules intelligents/autonomes

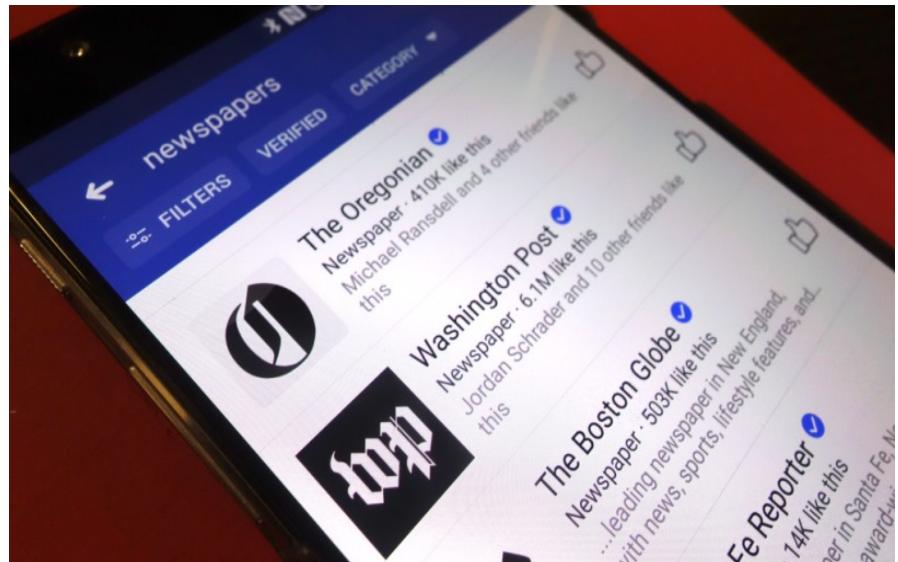
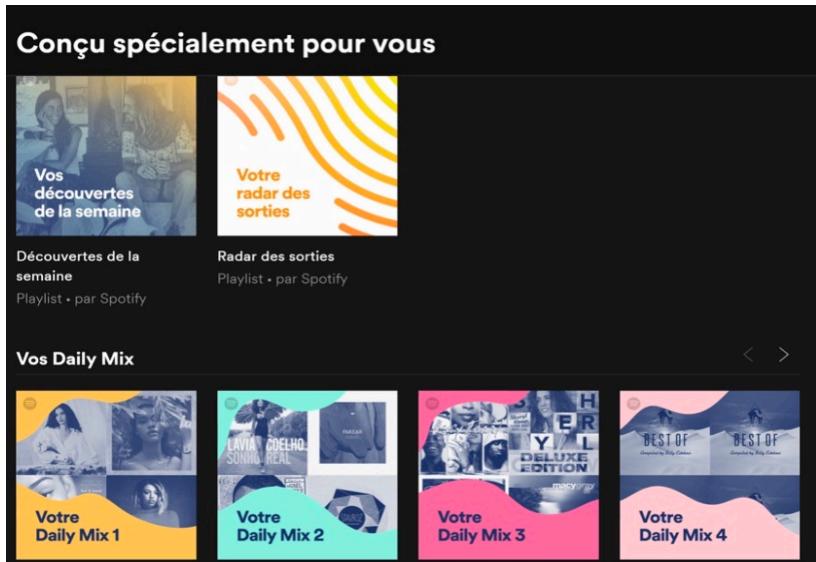


[ <https://enchanteds.tools/robot> ]

[<https://waymo.com>] 20

# Applications

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



# Applications

DeepL

Traducteur Linguee DeepL Pro Blog Infos

Traduire français (langue identifiée) Traduire en anglais

Le cours d'apprentissage profond The deep learning course

Autres traductions :  
The profound learning course

Traduire le document

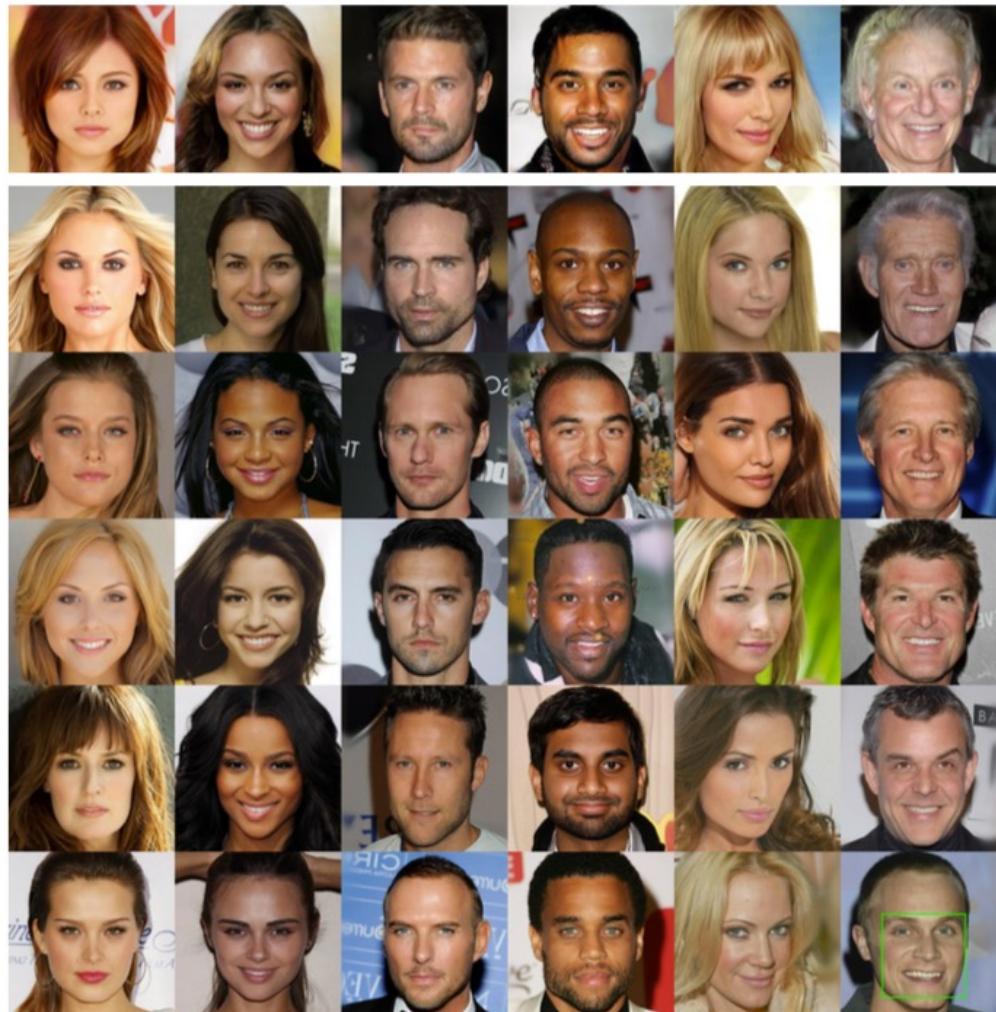
DEEP LEARNING  
Ian Goodfellow, Yoshua Bengio,  
and Aaron Courville

L'APPRENTISSAGE PROFOND  
Ian Goodfellow / Yoshua Bengio / Aaron Courville

MASSOT ÉDITIONS Quantmetry

22

# Applications



Génération automatique  
d'images

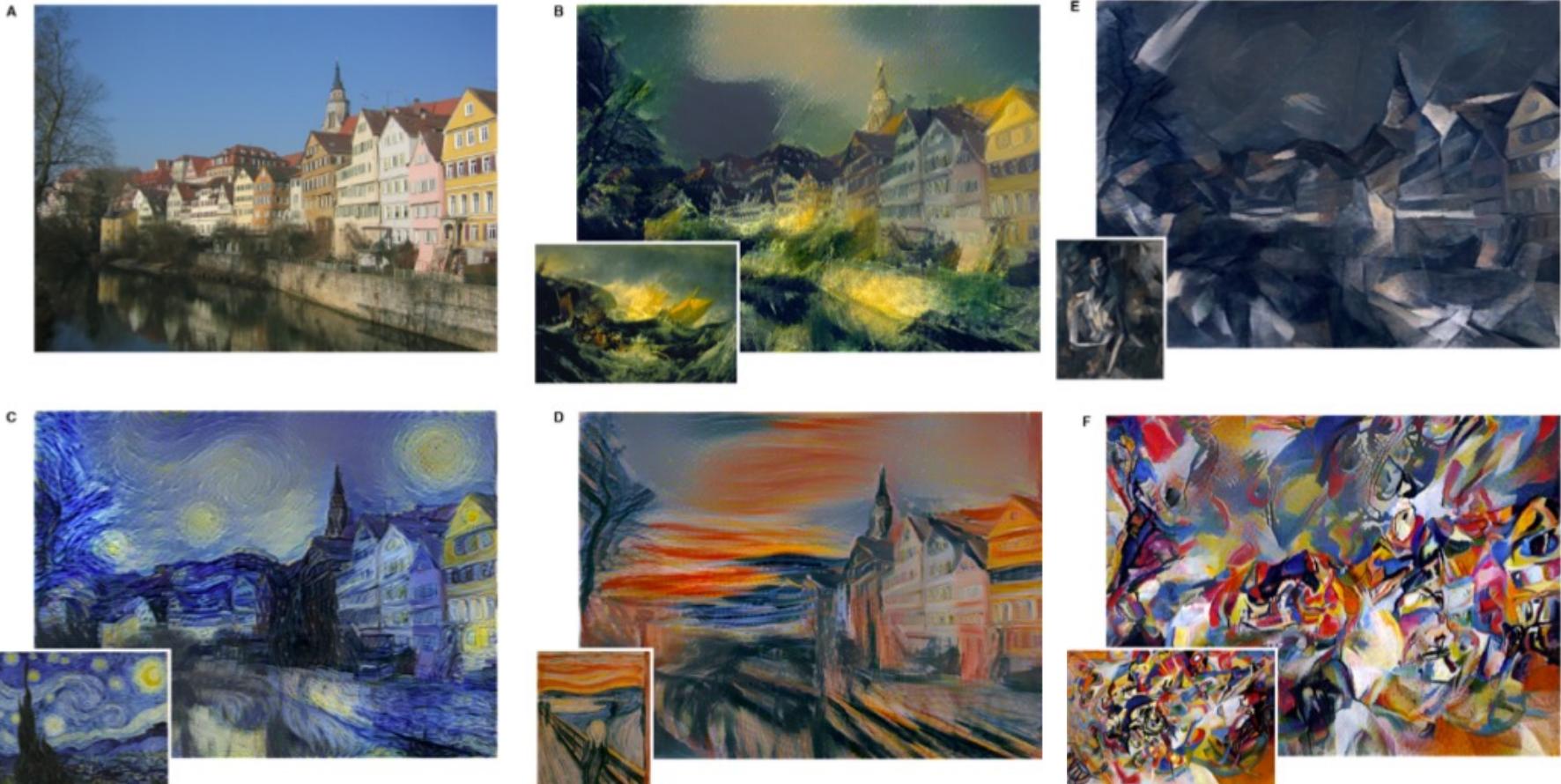
[Karras et al. ICLR 2018]

[https://research.nvidia.com/publication/2017-10\\_Progressive-Growing-of](https://research.nvidia.com/publication/2017-10_Progressive-Growing-of)

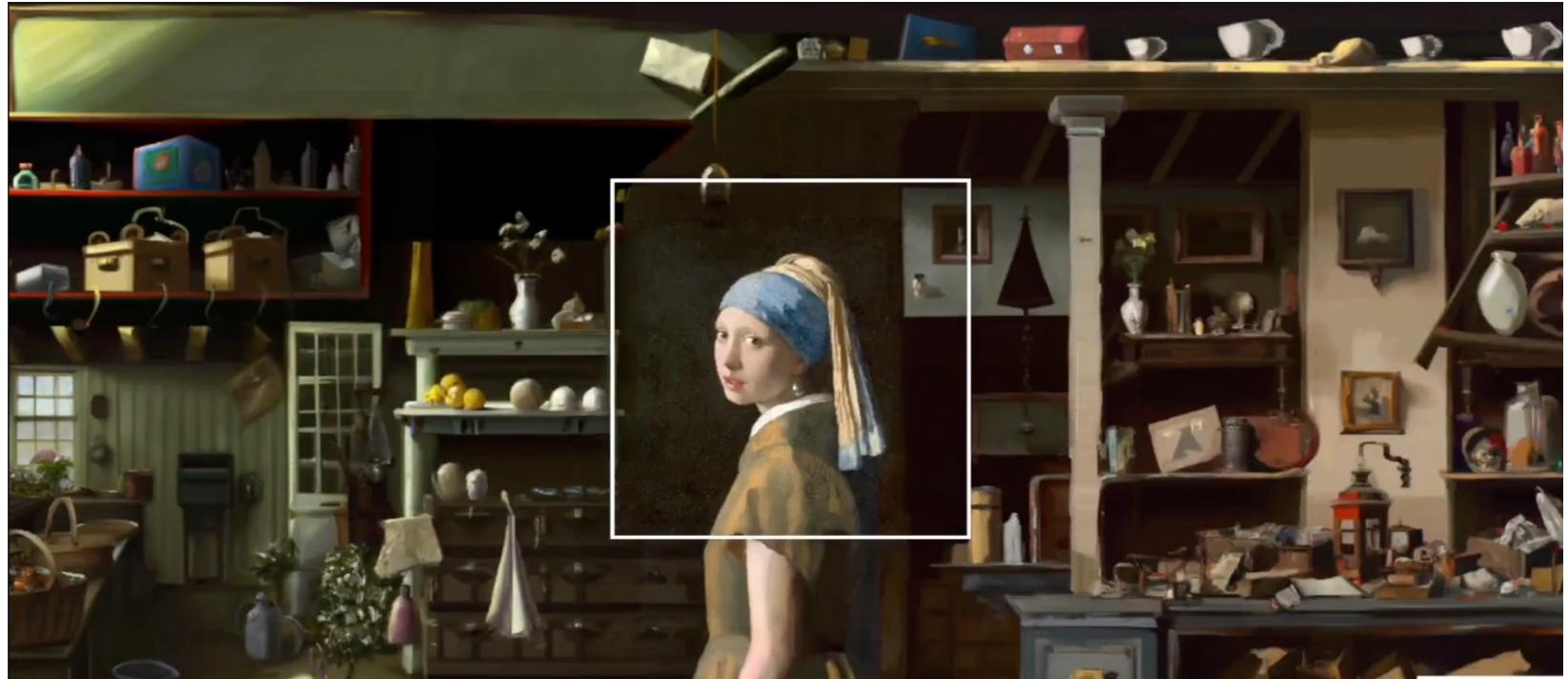
Figure 10: Top: Our CELEBA-HQ results. Next five rows: Nearest neighbors found from the training data, based on feature-space distance. We used activations from five VGG layers, as suggested by Chen & Koltun (2017). Only the crop highlighted in bottom right image was used for comparison in order to exclude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

# Applications

- Transfert de style



# Applications



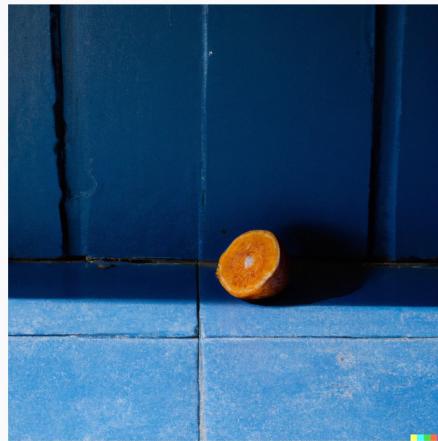
<https://openai.com/dall-e-2/>

# Applications

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

A blue orange sliced in half laying on a blue floor in front of a blue wall

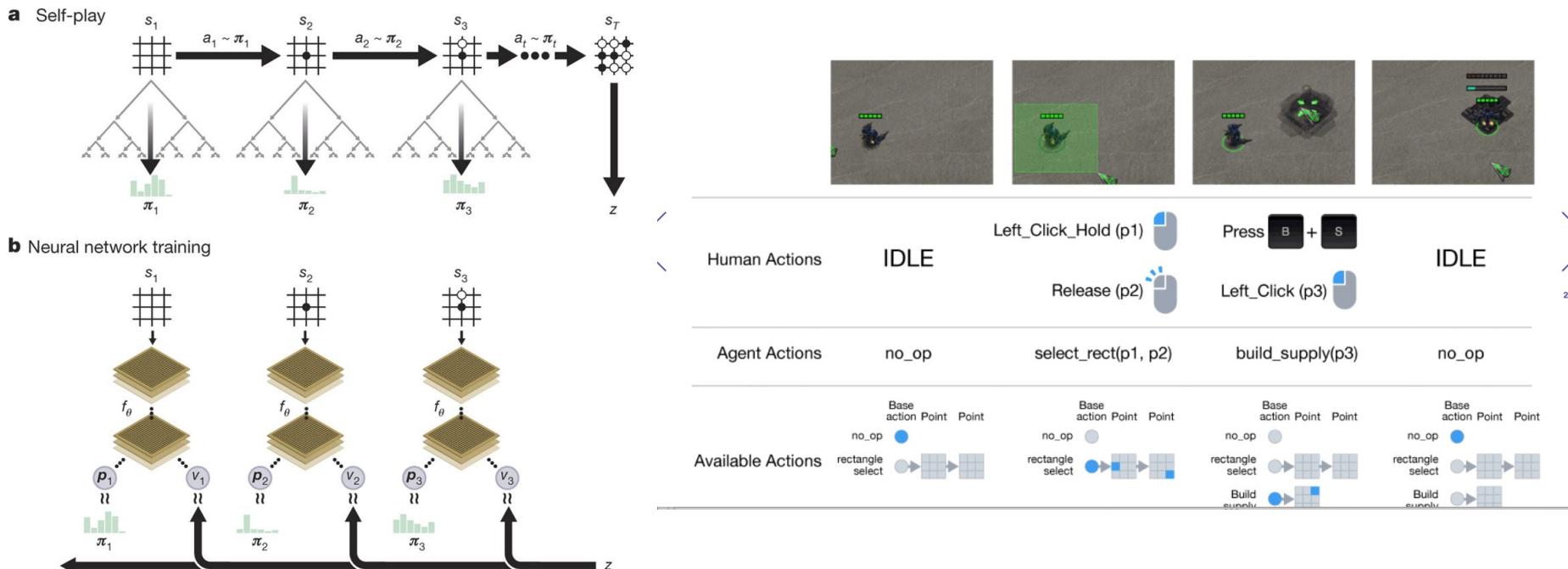
Generate



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

# Applications

- Jeux (jeu de Go, Starcraft, Poker...)

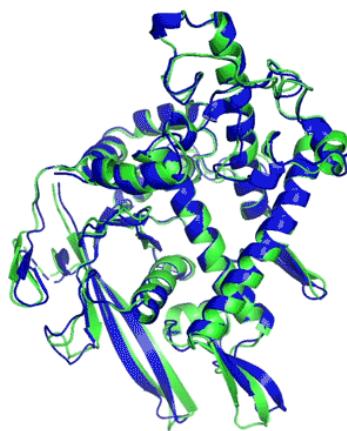


[AlphaGo Nature 2017]

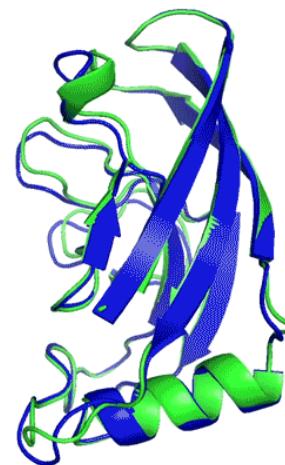
[ <https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/> ]

# Applications

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



T1037 / 6vr4  
90.7 GDT  
(RNA polymerase domain)



T1049 / 6y4f  
93.3 GDT  
(adhesin tip)

● Experimental result  
● Computational prediction

# Applications

- Github Copilot : génération automatique de code

The screenshot shows two code editors in Visual Studio Code. The left editor contains `runtime.go` with the following content:

```
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9 // Get average runtime of successful runs in seconds
10 func averageRuntimeInSeconds|
```

The right editor contains `comment_marker.ts` with the following content:

```
1 interface CommentMarker {
2     start: string;
3     end: string;
4 }
5
6 const markers: { [language:string]: CommentMarker } = {
7     javascript: { start: '/*', end: '*' },
8     python:      { start: '#', end: '' },
9 |
10
11
12
13
14
15
16
```

# Applications

The screenshot shows a web browser window with the URL [chat.openai.com/chat](https://chat.openai.com/chat). The page displays a conversation between a user (KB) and ChatGPT. The user asks for Python code to welcome students with Korean comments. ChatGPT responds with the following code:

```
# 이 프로그램은 새로운 학생들을 환영하는 메시지를 출력합니다.  
print("안녕하세요! 새로운 학생 여러분, 환영합니다.")  
print("이 과정을 수강하면서 새로운 기술과 지식을 얻을 수 있을 것입니다.")  
print("우리 함께 공부해봅시다!")
```

Below this, the user asks if the code can be translated into French. ChatGPT responds with the following French code:

```
#Ce programme affiche un message de bienvenue pour les nouveaux étudiants.  
print("Bonjour ! Chers nouveaux étudiants, bienvenue.")  
print("Vous pourrez acquérir de nouvelles compétences et connaissances en suivant ce cours.")  
print("A一同へんおなまこへん")
```

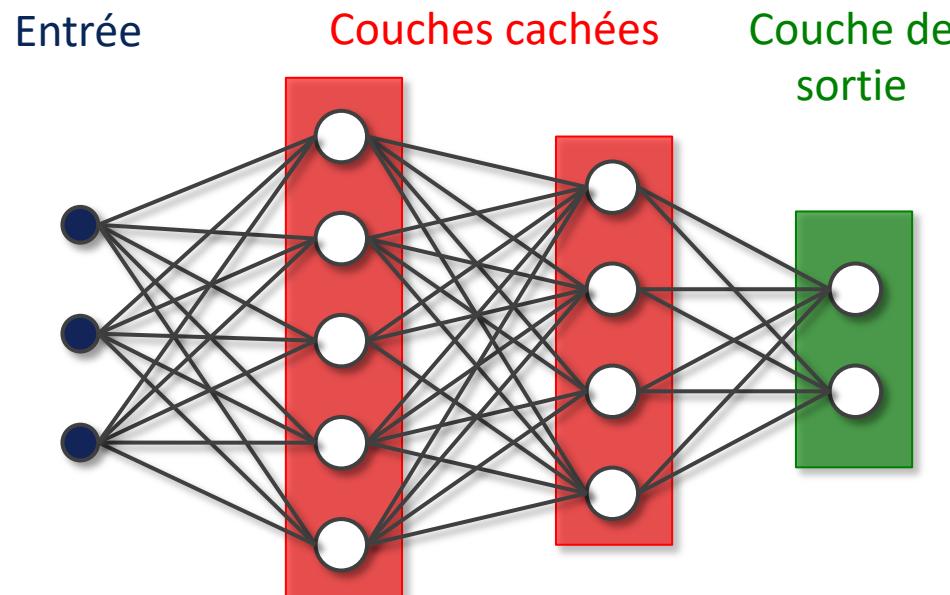
The sidebar on the left includes a 'New chat' button and links for 'Confirm clear conversations', 'Dark mode', 'OpenAI Discord', 'Updates & FAQ', and 'Log out'.

# Plan

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

# Qu'est-ce que l'apprentissage profond

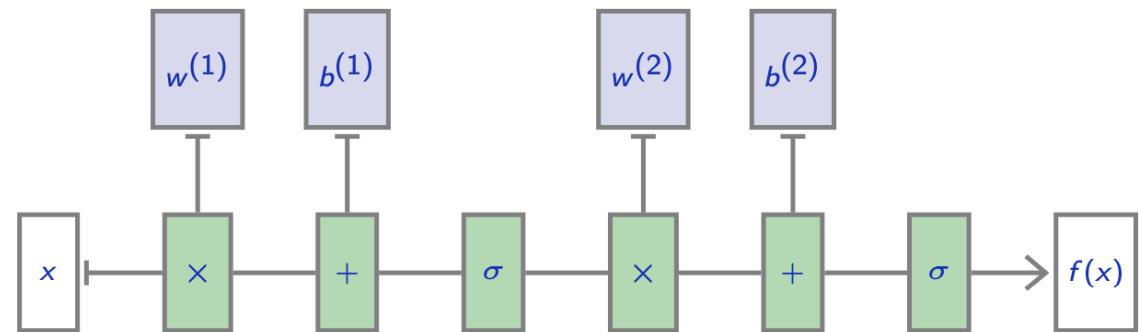
- Principe général : augmenter le nombre de couches dans le réseau
  - Projections **non linéaires** successives
  - Apprentissage conjoint de l'espace de **représentation** et de la **fonction de décision**



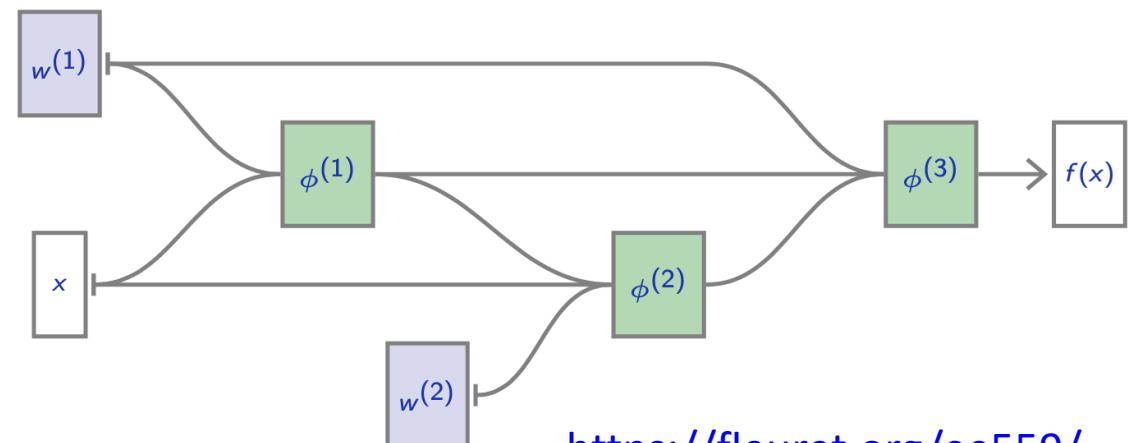
# Qu'est-ce que l'apprentissage profond

- Graphe Orienté Acyclique (DAG)

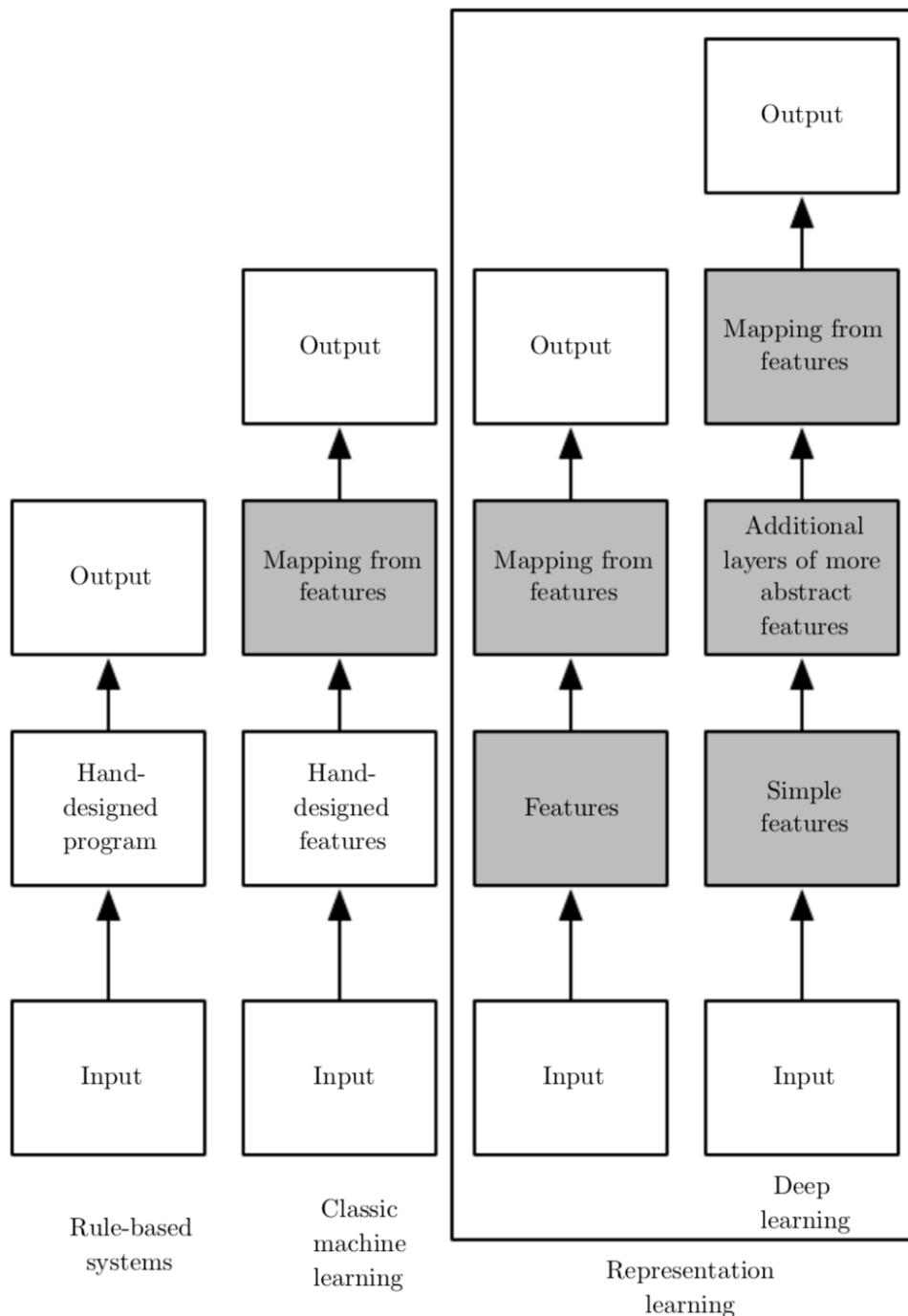
MLP Classique



DAG

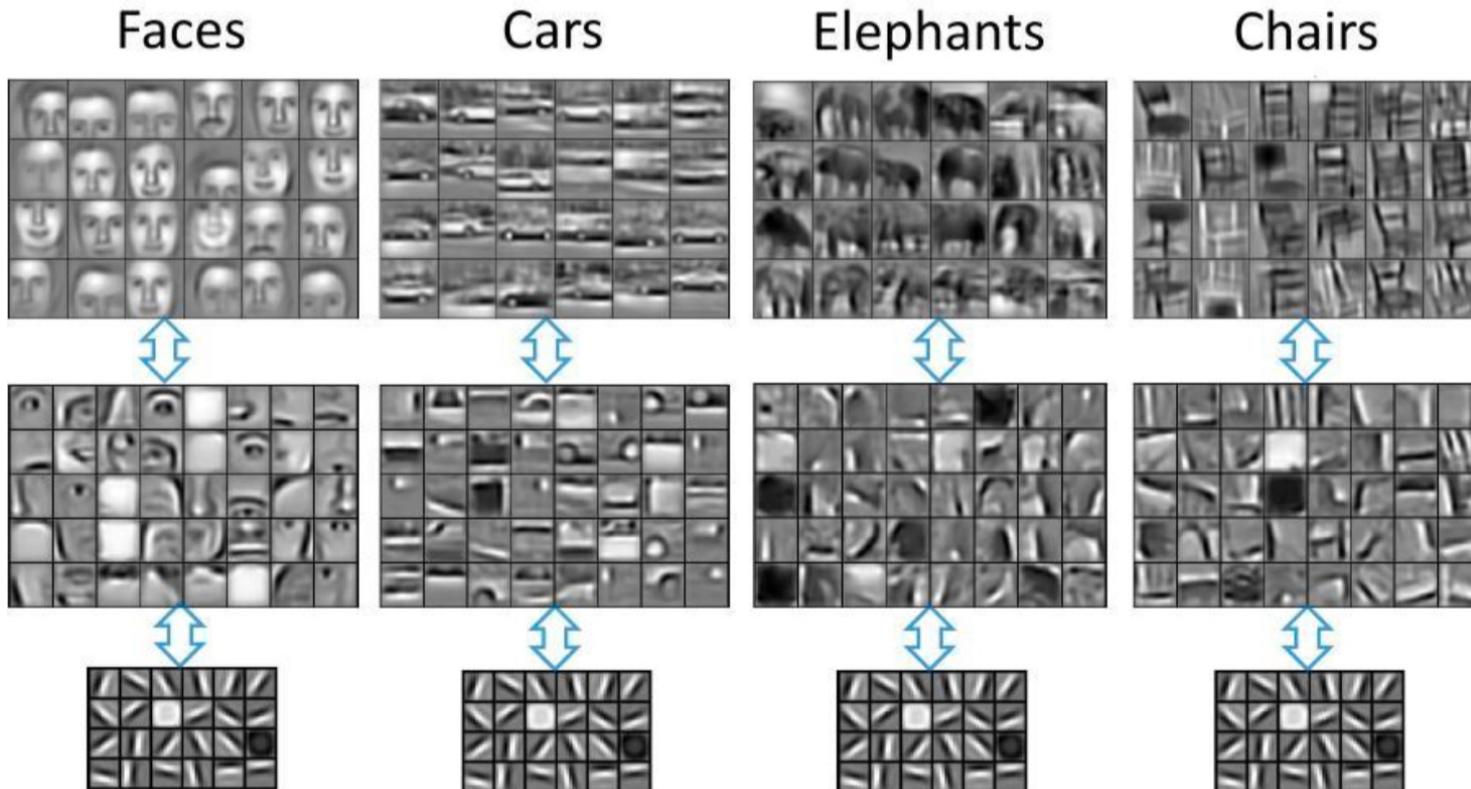


# L'apprentissage profond



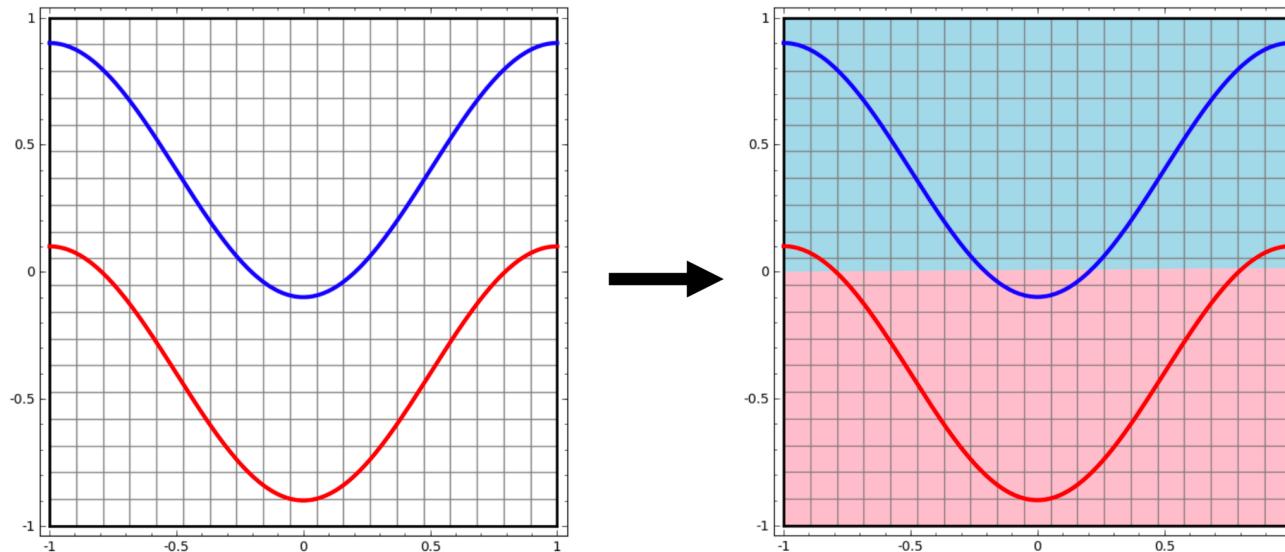
[Goodfellow et al 2016]

# Représentation hiérarchique

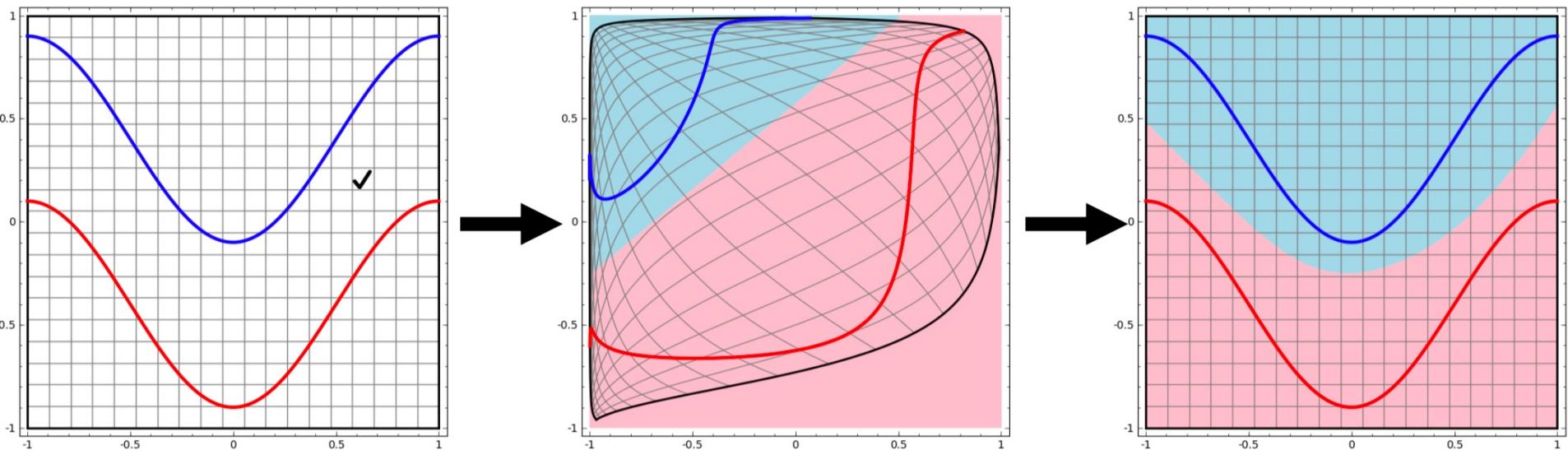


[Lee et al. ICML 2009]

# Importance de la représentation



# Importance de la représentation



# Les modèles linéaires sont limités

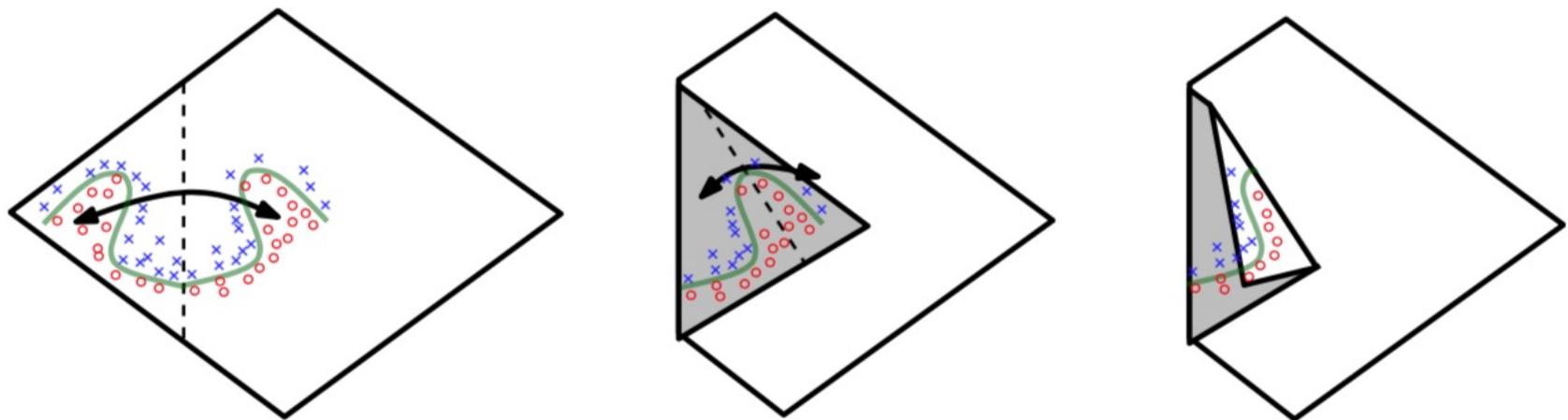
- Avantages des modèles linéaires
  - Stabilité
  - Facilité de calcul
  - Gradients constants
- Mais représentation limitée.
  - Empiler des modèles linéaires ne marche pas, pourquoi ?

# 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

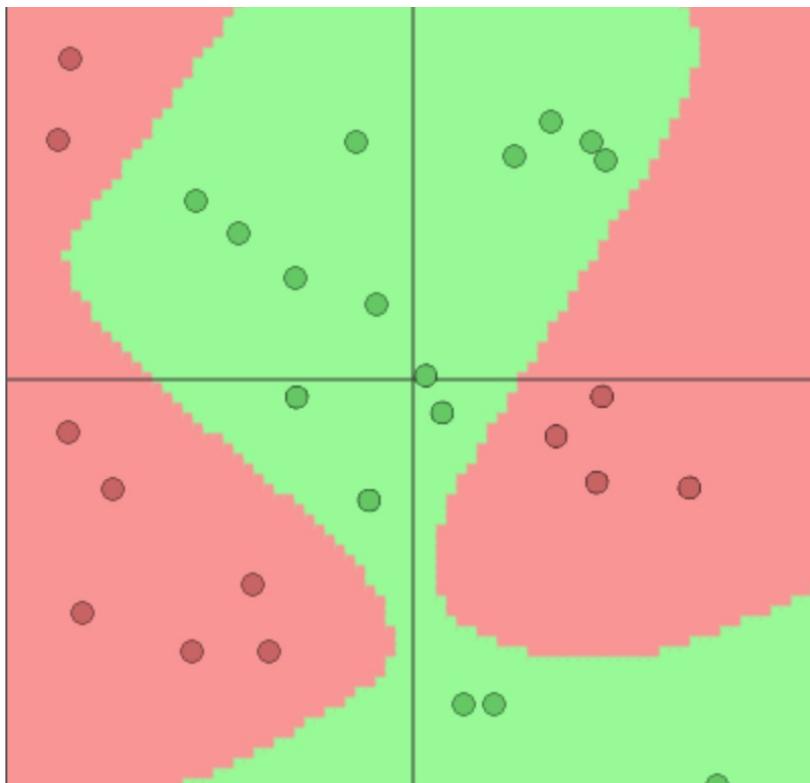
# Pourquoi des réseaux profonds ?

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



[Montufar et al. 2014]

# Demonstration interactive



simple data

circle data

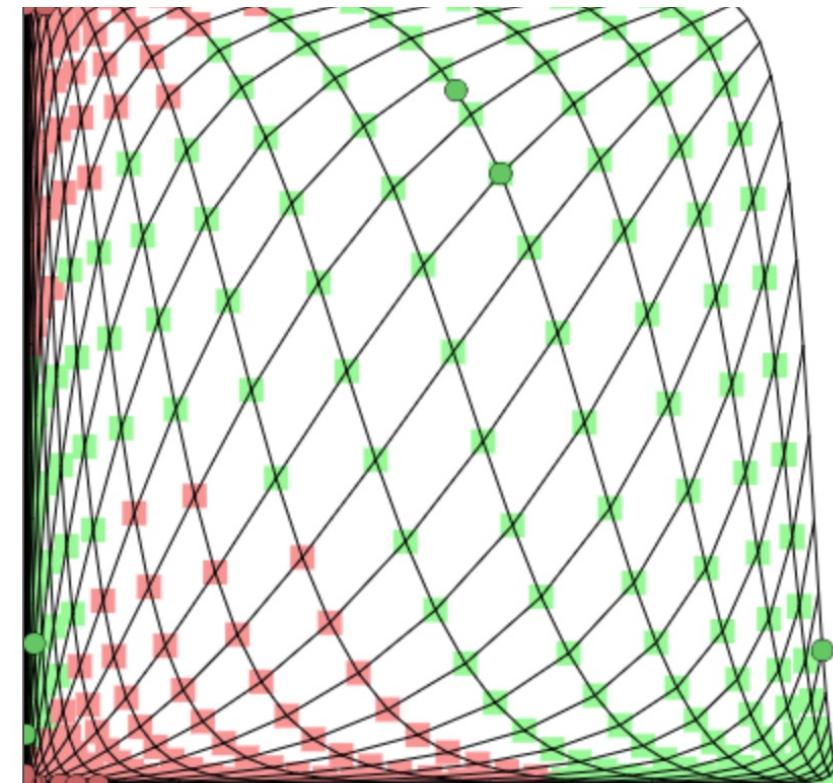
spiral data

random data

Controls:

**CLICK:** Add red data point

**SHIFT+CLICK:** Add green data point



fc(6)

tanh(6)

fc(2)

tanh(2)

fc(2)

cycle through visualized neurons at selected layer (if more than 2)

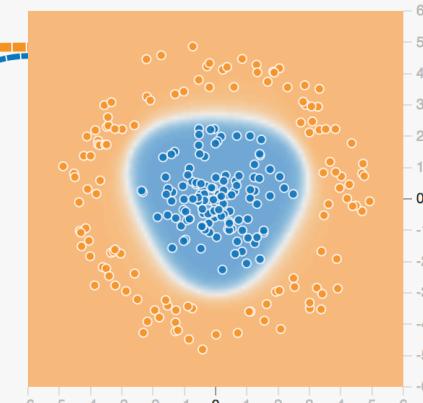
# Demonstration interactive

Epoch 000,237 Learning rate 0.03 Activation Tanh Regularization None Regularization rate 0 Problem type Classification

**DATA**  
Which dataset do you want to use?  
  
Ratio of training to test data: 50%  
Noise: 0  
Batch size: 10

**FEATURES**  
Which properties do you want to feed in?  
 $x_1$ ,  $x_2$ ,  $x_1^2$ ,  $x_2^2$ ,  $x_1x_2$ ,  $\sin(x_1)$ ,  $\sin(x_2)$

**2 HIDDEN LAYERS**  
+ -  
4 neurons  
2 neurons  
The outputs are mixed with varying **weights**, shown by the thickness of the lines.  
This is the output from one **neuron**. Hover to see it larger.

**OUTPUT**  
Test loss 0.003  
Training loss 0.001  
  
Colors shows data, neuron and weight values.  
 Show test data    Discretize output

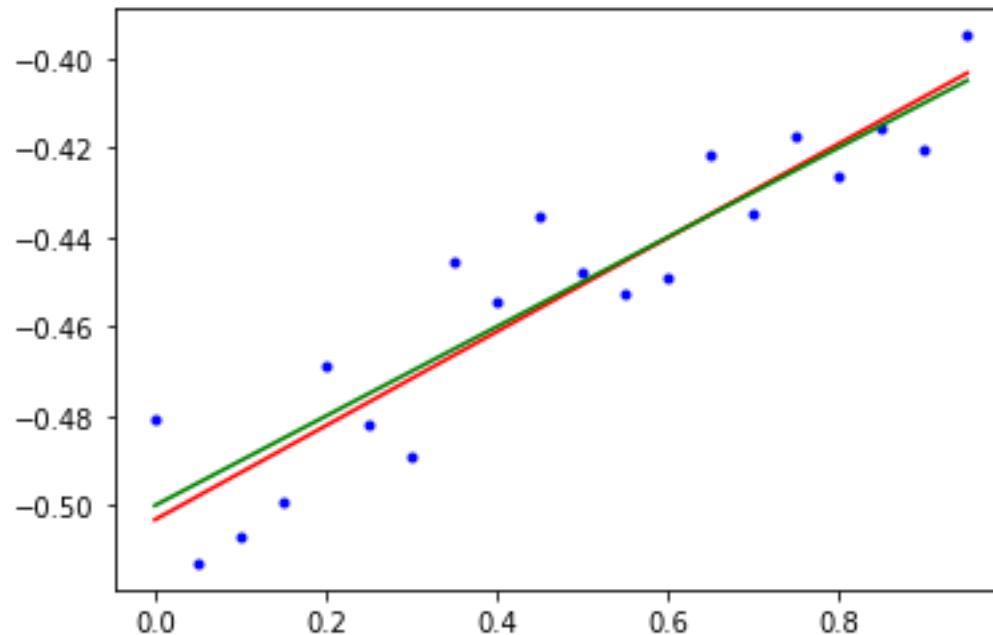
- Introduction (historique et applications)
- Qu'est-ce que l'apprentissage profond
- Comment créer un réseau de neurones avec TensorFlow

# Environnement



# Premier réseau en Tensorflow

- Notebook : **01\_regression\_lineaire.ipynb**



# Calcul du gradient

- Méthodes de calcul du gradient

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

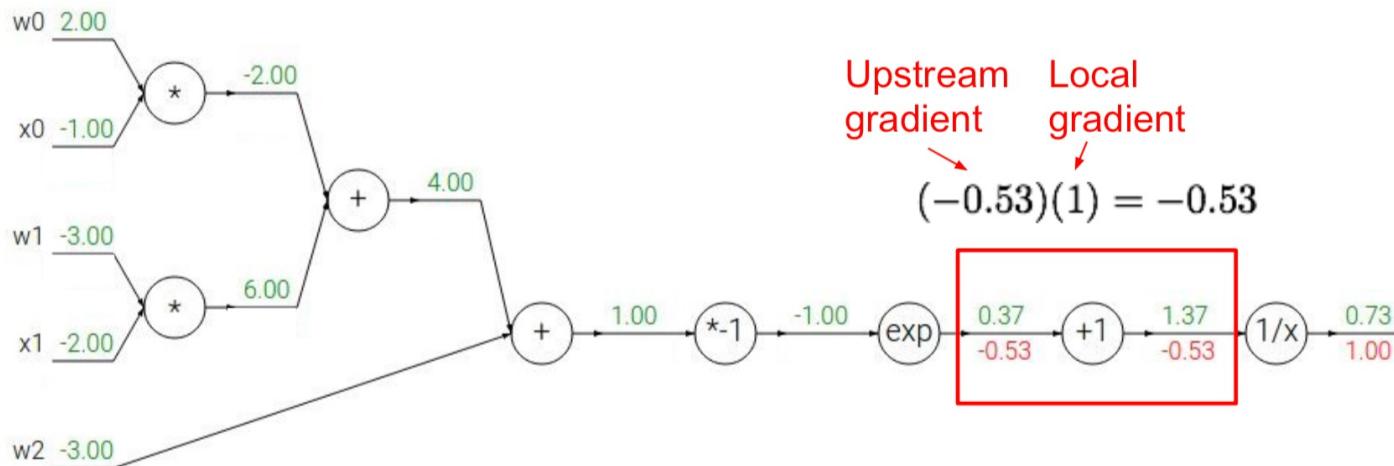
- Numérique
  - Lent, approximation, implémentation facile
- Analytique
  - Rapide, exact, risque d'erreurs
- Différentiation automatique [1]
  - Rapide, exact, implémentation facile
  - Par exemple : AutoGrad

[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

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



$$f(x) = e^x$$

$\rightarrow$

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

$\rightarrow$

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$\rightarrow$

$$\frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x$$

$\rightarrow$

$$\frac{df}{dx} = 1$$

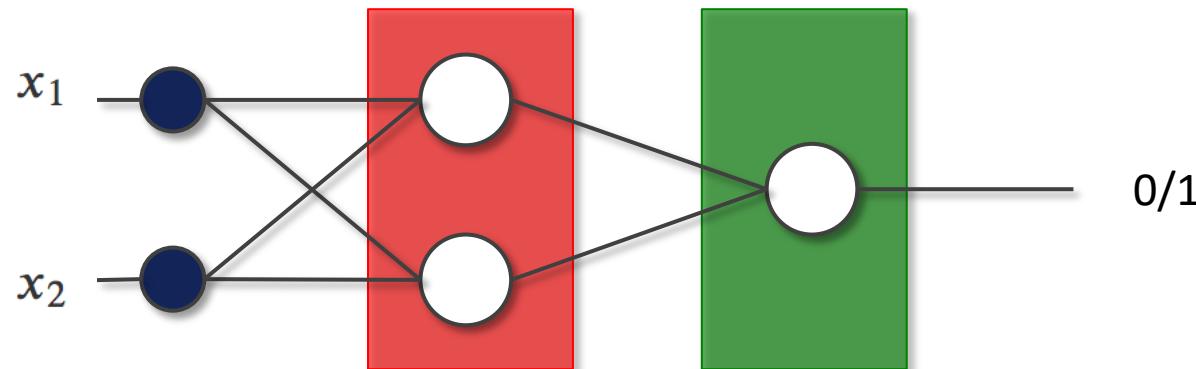
Tiré de: [http://cs231n.stanford.edu/slides/2018/cs231n\\_2018\\_lecture04.pdf](http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture04.pdf)

# Exercices

- Autograd : package python pour la différentiation automatique (DA)
- Tutorial:
  - [https://pytorch.org/tutorials/beginner/blitz/autograd\\_tutorial.html](https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html)
- Exercice : 02\_autograd.ipynb

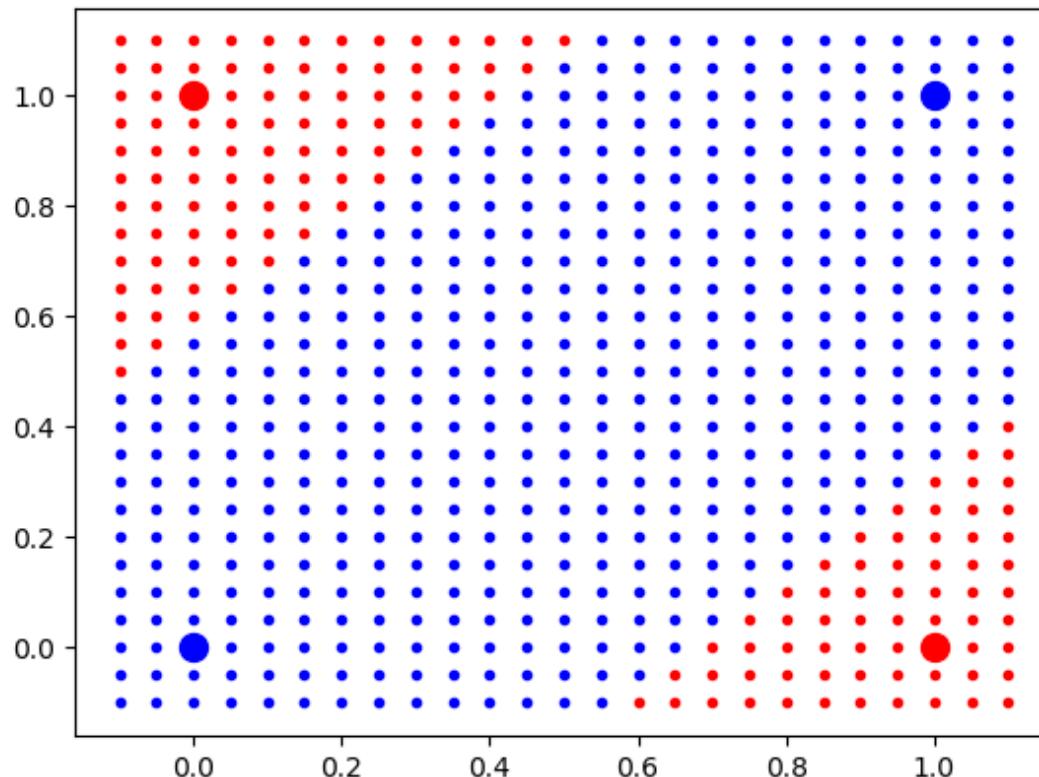
# Exercice

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



# Exercice

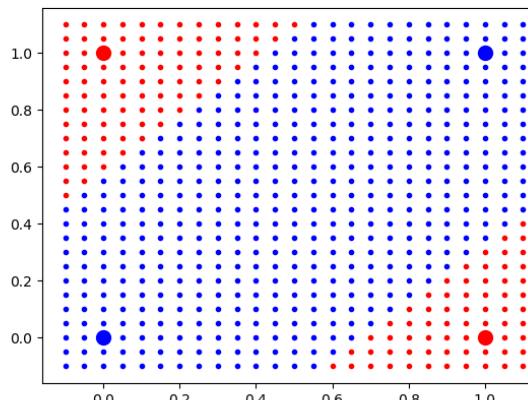
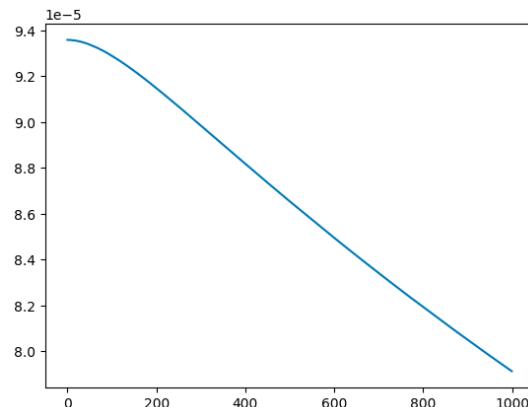
- Compléter le notebook : 03\_XOR\_lightning.ipynb



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/>