

RÉSEAUX CNN : DIVISION DU TRAVAIL, INVARIANCE DE LA TRANSLATION ET PERCEPTION

PHI 6385

Séance 9

Jonathan Simon

PROGRAMME

- 1) Obstacles au progrès avec les perceptrons multicouches
- 2) Réseaux neuronaux convolutionnels : Qu'est-ce que c'est ?
- 2) Réseaux neuronaux convolutionnels : Perçoivent-ils ?

RÉSEAUX NEURONAUX
CONVOLUTIONNELS : QU'EST-CE
QUE C'EST

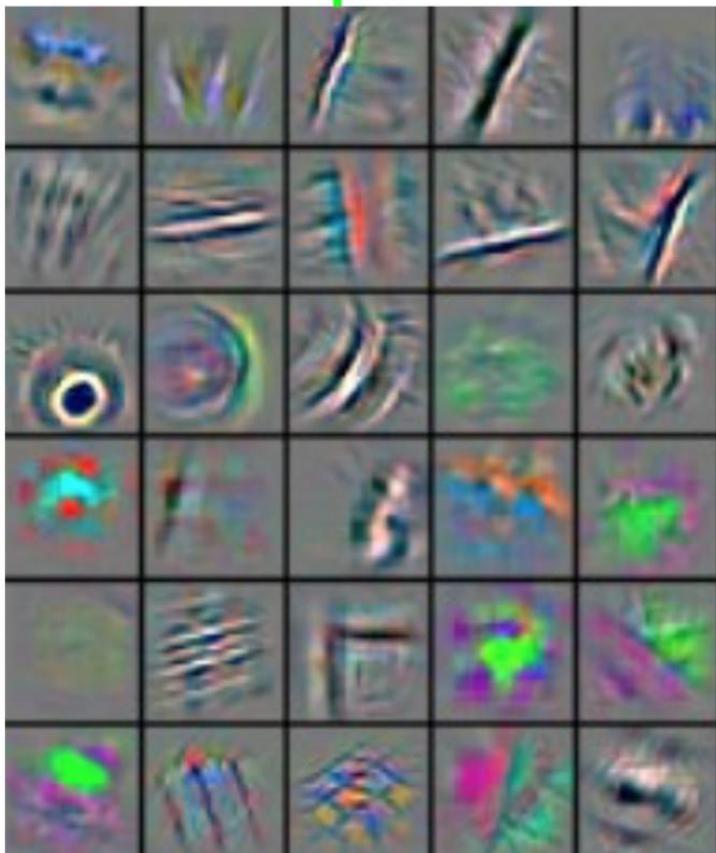
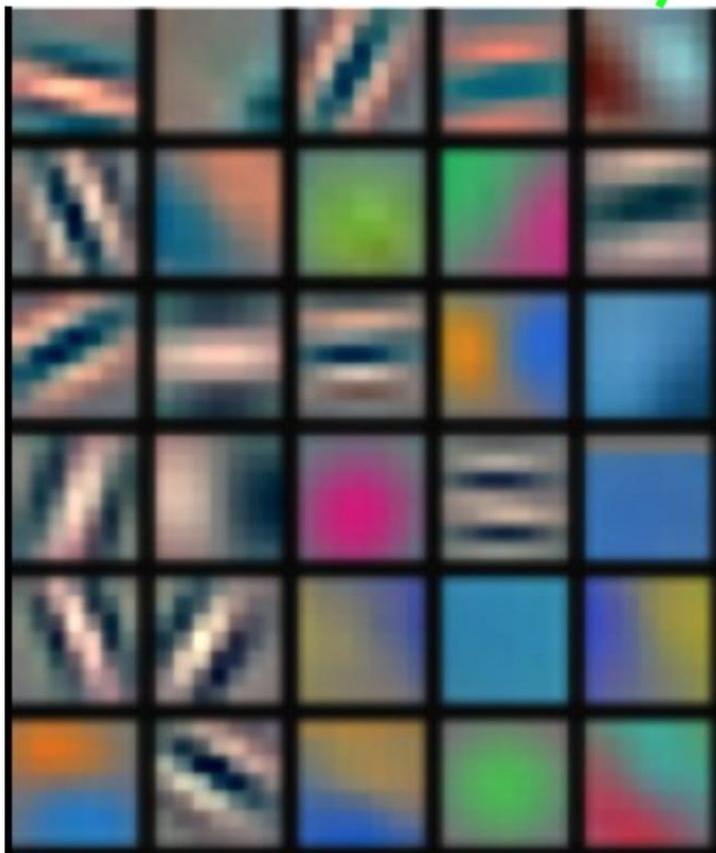


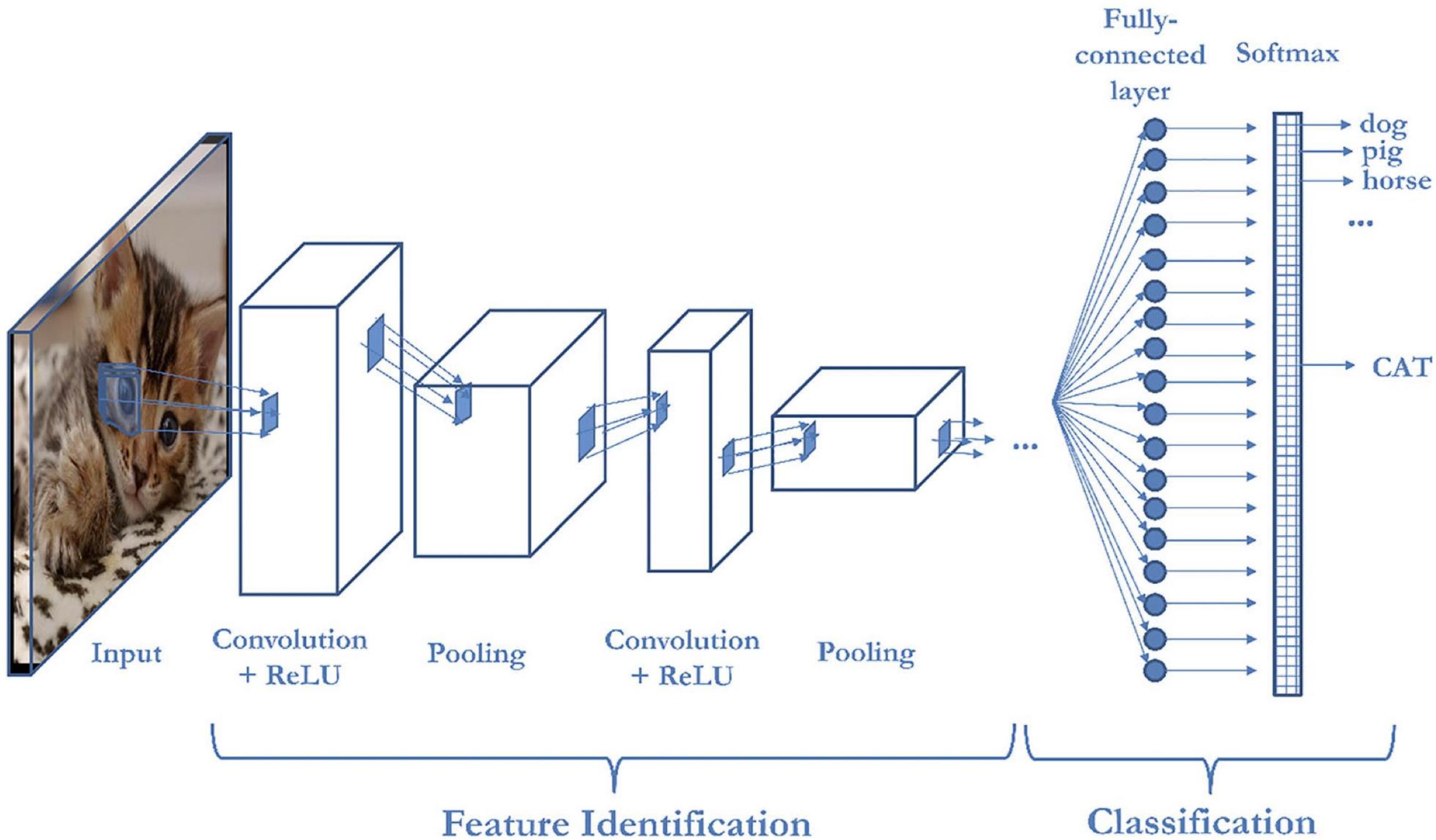
Low-Level
Feature

Mid-Level
Feature

High-Level
Feature

Trainable
Classifier





LES RESEAUX CNN

- 1) En général, que peuvent faire les réseaux neuronaux profonds que les réseaux peu profonds ne peuvent pas faire ?
- 2) Caractéristiques spécifiques des réseaux neuronaux convolutionnels

LES RESEAUX CNN

- 1) En général, que peuvent faire les réseaux neuronaux profonds que les réseaux peu profonds ne peuvent pas faire ?

LES RESEAUX CNN

- Réseaux plus anciens : peu profonds, uniformes, entièrement connectés.
- Réseaux plus récents : profonds (plus de couches), fonctions d'activation hétérogènes, connectés de manière éparses, méthodes pour éviter l'overfitting.

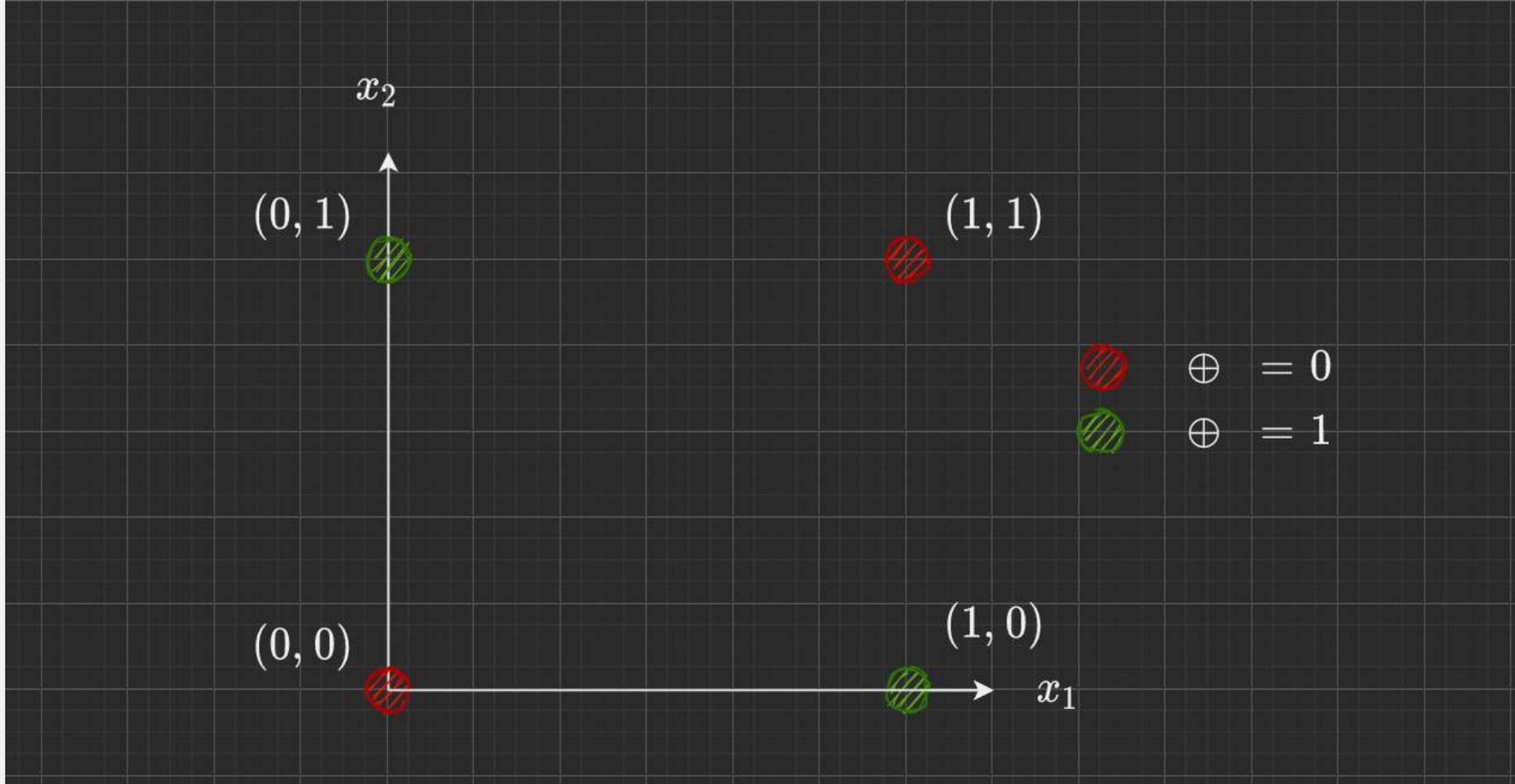
LES RESEAUX CNN

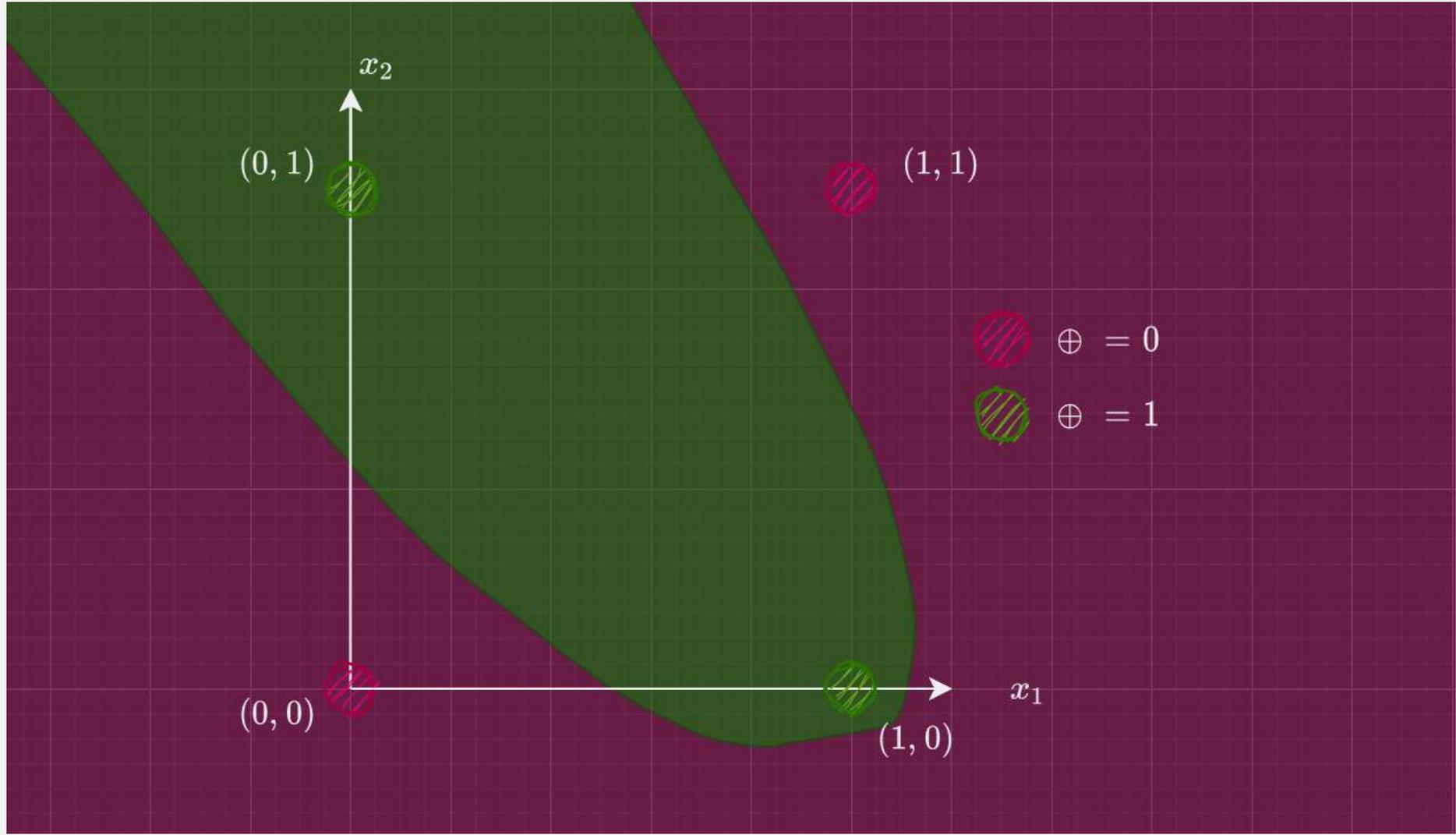
- Les avantages de la profondeur :
- 1) non-linéarité (avoir au moins une couche cachée permet une approximation générale de la fonction : avec une seule couche de perceptron, pas de place pour les courbes).
- 2) division (et réutilisation) du travail : l'analogie de la chaîne de montage

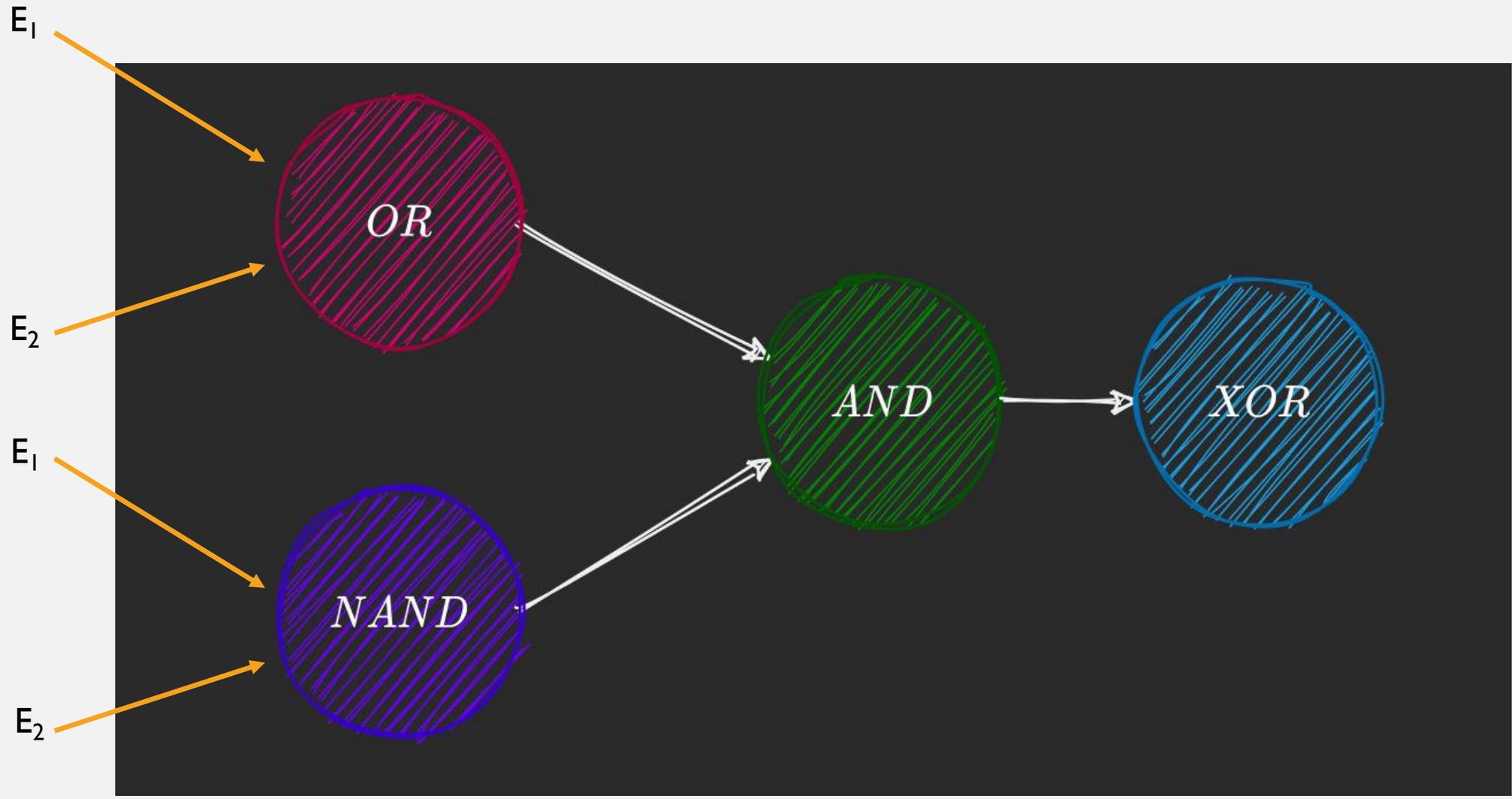
LES RESEAUX CNN

- Non-linéarité : exemple célèbre : XOR (la fonction de vérité « *ou exclusive* » - vrai si exactement un des deux disjonctifs est vrai)

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0





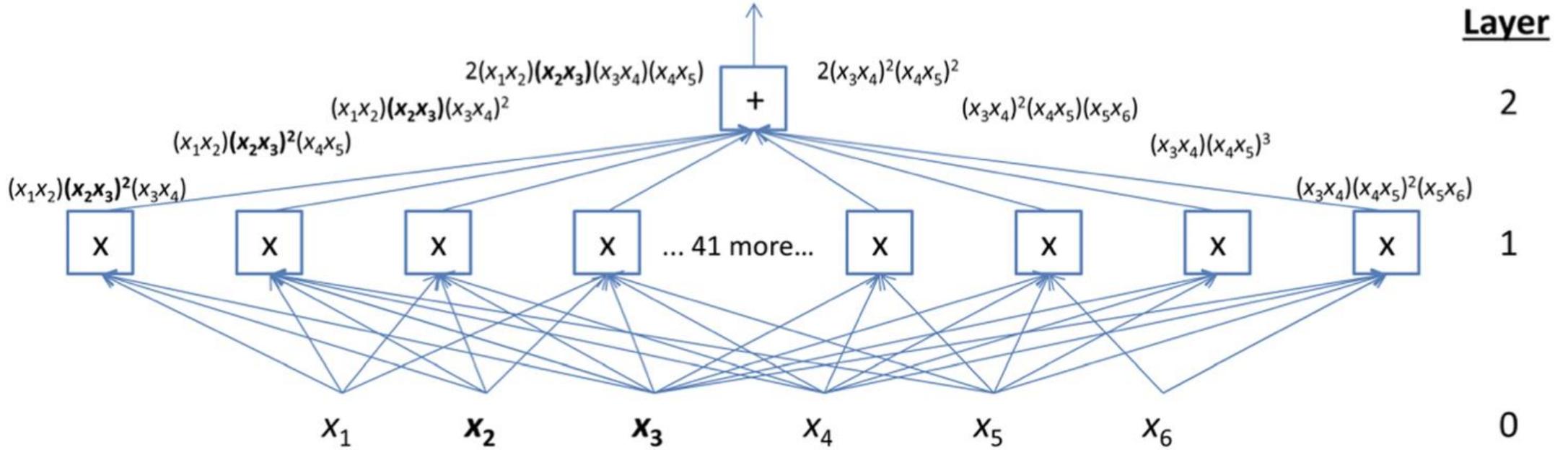


LES RESEAUX CNN

- 2) division (et réutilisation) du travail : l'analogie de la chaîne de montage

Shallow Architecture

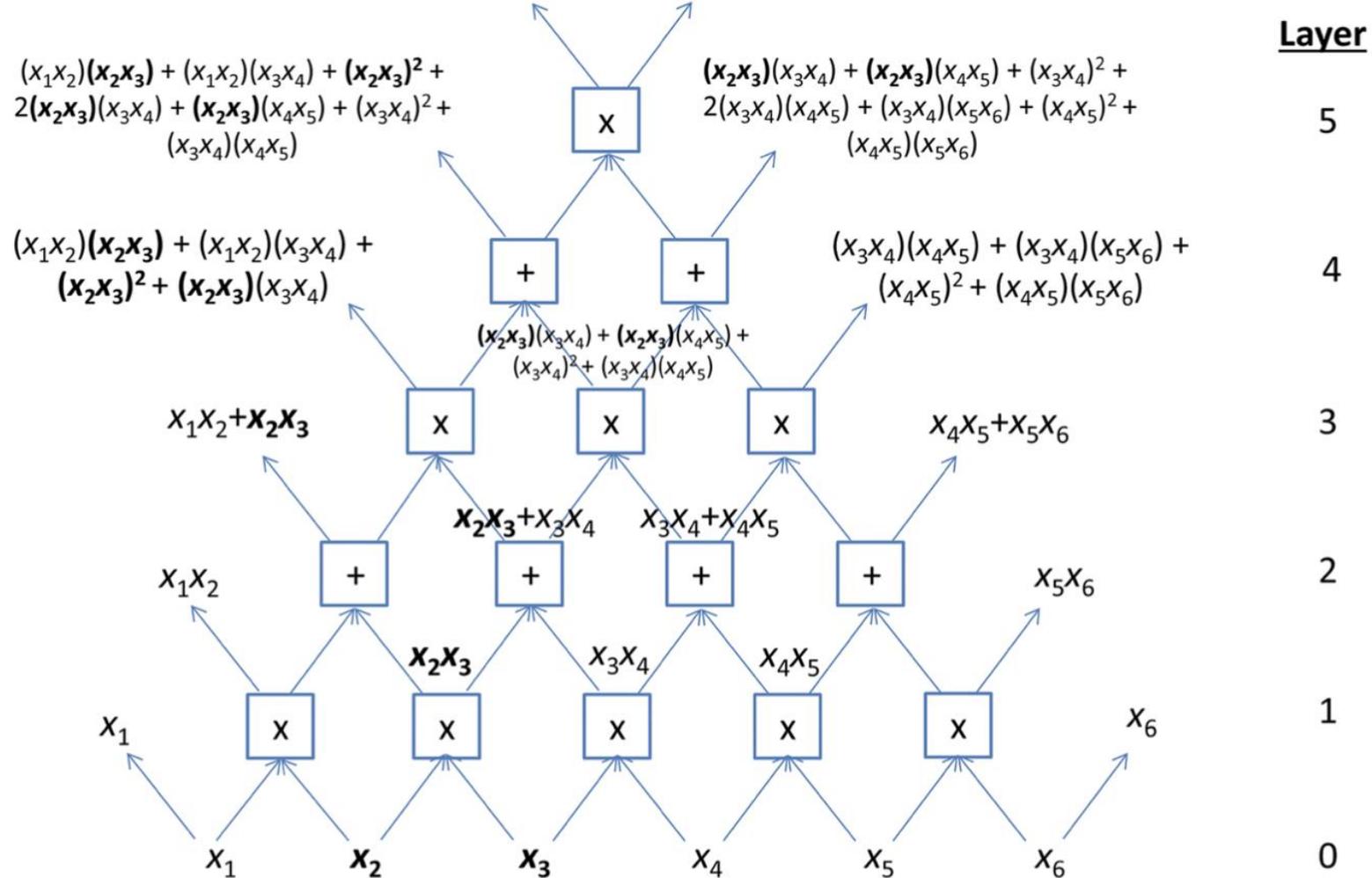
$$\begin{aligned}
 & (x_1x_2)(x_2x_3)^2(x_3x_4) + (x_1x_2)(x_2x_3)^2(x_4x_5) + (x_1x_2)(x_2x_3)(x_3x_4)^2 + 2(x_1x_2)(x_2x_3)(x_3x_4)(x_4x_5) + (x_1x_2)(x_2x_3)(x_3x_4)(x_5x_6) + (x_1x_2)(x_2x_3)(x_4x_5)^2 + (x_1x_2)(x_2x_3)(x_4x_5)(x_5x_6) + \\
 & (x_1x_2)(x_2x_3)(x_3x_4)^2 + (x_1x_2)(x_3x_4)(x_2x_3)(x_4x_5) + (x_1x_2)(x_3x_4)^3 + 2(x_1x_2)(x_3x_4)^2(x_4x_5) + (x_1x_2)(x_3x_4)^2(x_5x_6) + (x_1x_2)(x_3x_4)(x_4x_5)^2 + (x_1x_2)(x_3x_4)(x_4x_5)(x_5x_6) + \\
 & (x_2x_3)^3(x_3x_4) + (x_2x_3)^3(x_4x_5) + (x_2x_3)^2(x_3x_4)^2 + 2(x_2x_3)^2(x_3x_4)(x_4x_5) + (x_2x_3)^2(x_3x_4)(x_5x_6) + (x_2x_3)^2(x_4x_5)^2 + (x_2x_3)^2(x_4x_5)(x_5x_6) + \\
 & 2(x_2x_3)^2(x_3x_4)^2 + 2(x_2x_3)^2(x_3x_4)(x_4x_5) + 2(x_2x_3)(x_3x_4)^3 + 4(x_2x_3)(x_3x_4)^2(x_4x_5) + 2(x_2x_3)(x_3x_4)^2(x_5x_6) + 2(x_2x_3)(x_3x_4)(x_4x_5)^2 + 2(x_2x_3)(x_3x_4)(x_4x_5)(x_5x_6) + \\
 & (x_2x_3)^2(x_4x_5)(x_3x_4) + (x_2x_3)^2(x_4x_5)(x_4x_5)^2 + (x_2x_3)(x_4x_5)(x_3x_4)^2 + 2(x_2x_3)(x_4x_5)^2(x_3x_4) + (x_2x_3)(x_4x_5)(x_3x_4)(x_5x_6) + (x_2x_3)(x_4x_5)^3 + (x_2x_3)(x_4x_5)^2(x_5x_6) + \\
 & (x_3x_4)^3(x_2x_3) + (x_3x_4)^2(x_2x_3)(x_4x_5) + (x_3x_4)^4 + 2(x_3x_4)^3(x_4x_5) + (x_3x_4)^3(x_5x_6) + (x_3x_4)^2(x_4x_5)^2 + (x_3x_4)^2(x_4x_5)(x_5x_6) + \\
 & (x_3x_4)^2(x_4x_5)(x_2x_3) + (x_3x_4)(x_4x_5)^2(x_2x_3) + (x_4x_5)(x_3x_4)^3 + 2(x_3x_4)^2(x_4x_5)^2 + (x_3x_4)^2(x_4x_5)(x_5x_6) + (x_3x_4)(x_4x_5)^3 + (x_3x_4)(x_4x_5)^2(x_5x_6)
 \end{aligned}$$



(b)

Deep Architecture

$$\begin{aligned}
 &(x_1x_2)(x_2x_3)^2(x_3x_4) + (x_1x_2)(x_2x_3)^2(x_4x_5) + (x_1x_2)(x_2x_3)(x_3x_4)^2 + 2(x_1x_2)(x_2x_3)(x_3x_4)(x_4x_5) + (x_1x_2)(x_2x_3)(x_3x_4)(x_5x_6) + (x_1x_2)(x_2x_3)(x_4x_5)^2 + (x_1x_2)(x_2x_3)(x_4x_5)(x_5x_6) + \\
 &(x_1x_2)(x_2x_3)(x_3x_4)^2 + (x_1x_2)(x_3x_4)(x_2x_3)(x_4x_5) + (x_1x_2)(x_3x_4)^3 + 2(x_1x_2)(x_3x_4)^2(x_4x_5) + (x_1x_2)(x_3x_4)^2(x_5x_6) + (x_1x_2)(x_3x_4)(x_4x_5)^2 + (x_1x_2)(x_3x_4)(x_4x_5)(x_5x_6) + \\
 &(x_2x_3)^3(x_3x_4) + (x_2x_3)^3(x_4x_5) + (x_2x_3)^2(x_3x_4)^2 + 2(x_2x_3)^2(x_3x_4)(x_4x_5) + (x_2x_3)^2(x_3x_4)(x_5x_6) + (x_2x_3)^2(x_4x_5)^2 + (x_2x_3)^2(x_4x_5)(x_5x_6) + \\
 &2(x_2x_3)^2(x_3x_4)^2 + 2(x_2x_3)^2(x_3x_4)(x_4x_5) + 2(x_2x_3)(x_3x_4)^3 + 4(x_2x_3)(x_3x_4)^2(x_4x_5) + 2(x_2x_3)(x_3x_4)^2(x_5x_6) + 2(x_2x_3)(x_3x_4)(x_4x_5)^2 + 2(x_2x_3)(x_3x_4)(x_4x_5)(x_5x_6) + \\
 &(x_2x_3)^2(x_4x_5)(x_3x_4) + (x_2x_3)^2(x_4x_5)(x_4x_5)^2 + (x_2x_3)(x_4x_5)(x_3x_4)^2 + 2(x_2x_3)(x_4x_5)^2(x_3x_4) + (x_2x_3)(x_4x_5)(x_3x_4)(x_5x_6) + (x_2x_3)(x_4x_5)^3 + (x_2x_3)(x_4x_5)^2(x_5x_6) + \\
 &(x_3x_4)^3(x_2x_3) + (x_3x_4)^2(x_2x_3)(x_4x_5) + (x_3x_4)^4 + 2(x_3x_4)^3(x_4x_5) + (x_3x_4)^3(x_5x_6) + (x_3x_4)^2(x_4x_5)^2 + (x_3x_4)^2(x_4x_5)(x_5x_6) + \\
 &(x_3x_4)^2(x_4x_5)(x_2x_3) + (x_3x_4)(x_4x_5)^2(x_2x_3) + (x_4x_5)(x_3x_4)^3 + 2(x_3x_4)^2(x_4x_5)^2 + (x_3x_4)^2(x_4x_5)(x_5x_6) + (x_3x_4)(x_4x_5)^3 + (x_3x_4)(x_4x_5)^2(x_5x_6)
 \end{aligned}$$



(a)

LES RESEAUX CNN

- Hétérogénéité : convolution, rectification, pooling

LES RESEAUX CNN

- Convolution : amplifie une certaine caractéristique (par exemple, un bord) dans une petite zone
- Rectification : définit un seuil pour la présence ou l'absence de cette caractéristique dans cette zone
- Pooling: Regroupe les rectificateurs de régions proches ou qui se chevauchent, par exemple, dit "oui" si l'un des rectificateurs dit "oui".

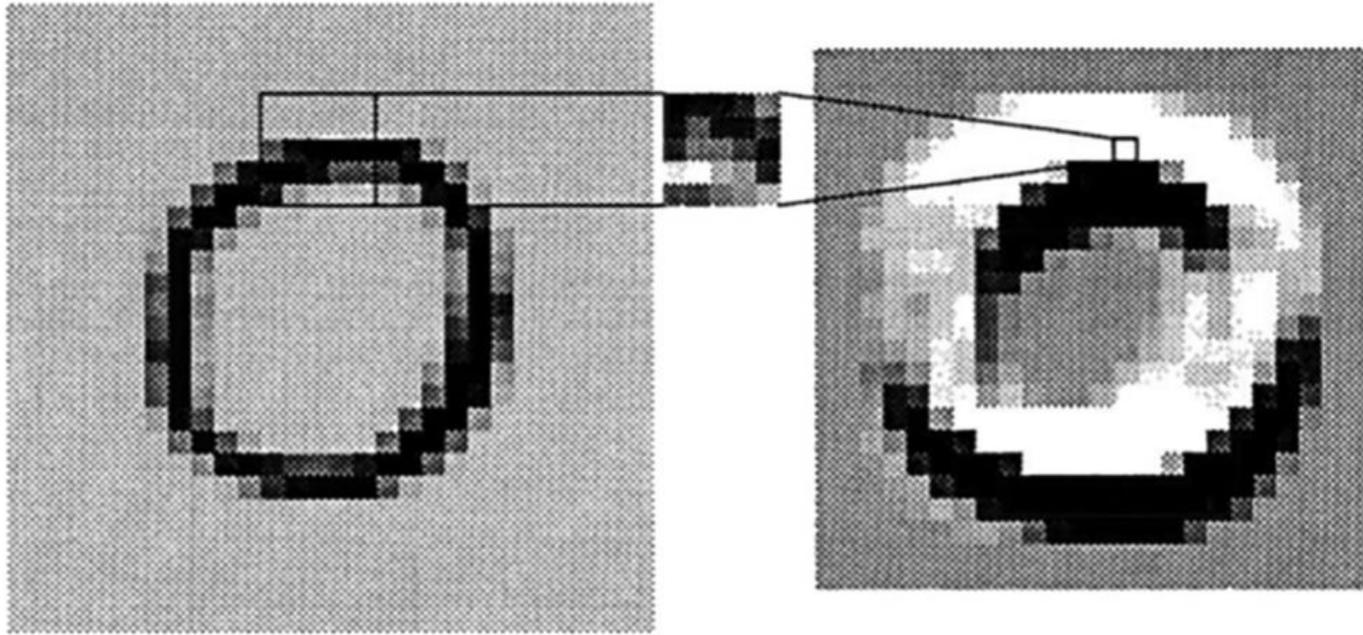


FIGURE 3 An example of the visual output of a single convolutional kernel on real handwritten digit data in a trained convolutional network (reproduced from LeCun et al., 1990, 399). This (learned) kernel (in center) detects something like curves at the top or bottom of a digit (with source data on the left and transformed output on the right—often called a “feature map”)

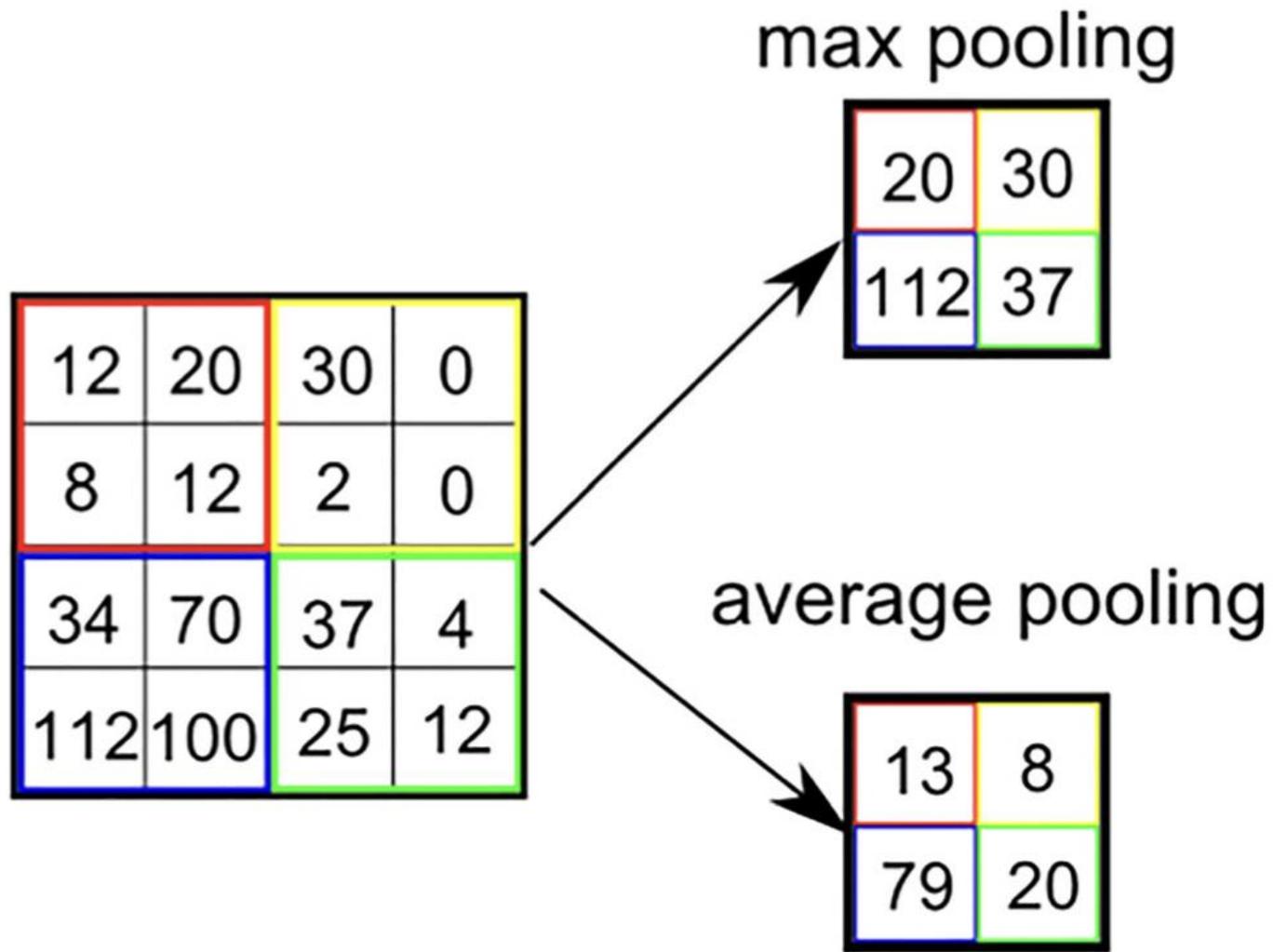


FIGURE 4 A comparison of max pooling with average pooling for downsampling across activation received from the same receptive fields (reproduced from Singhal, 2017)

LES RESEAUX CNN

- Outils pour lutter contre l'overfitting :
régularisation - comment éviter "d'enseigner au test".
- De nombreuses approches ici - perturber un peu au hasard, d'autres façons d'exiger que le mappage soit aussi simple que possible...

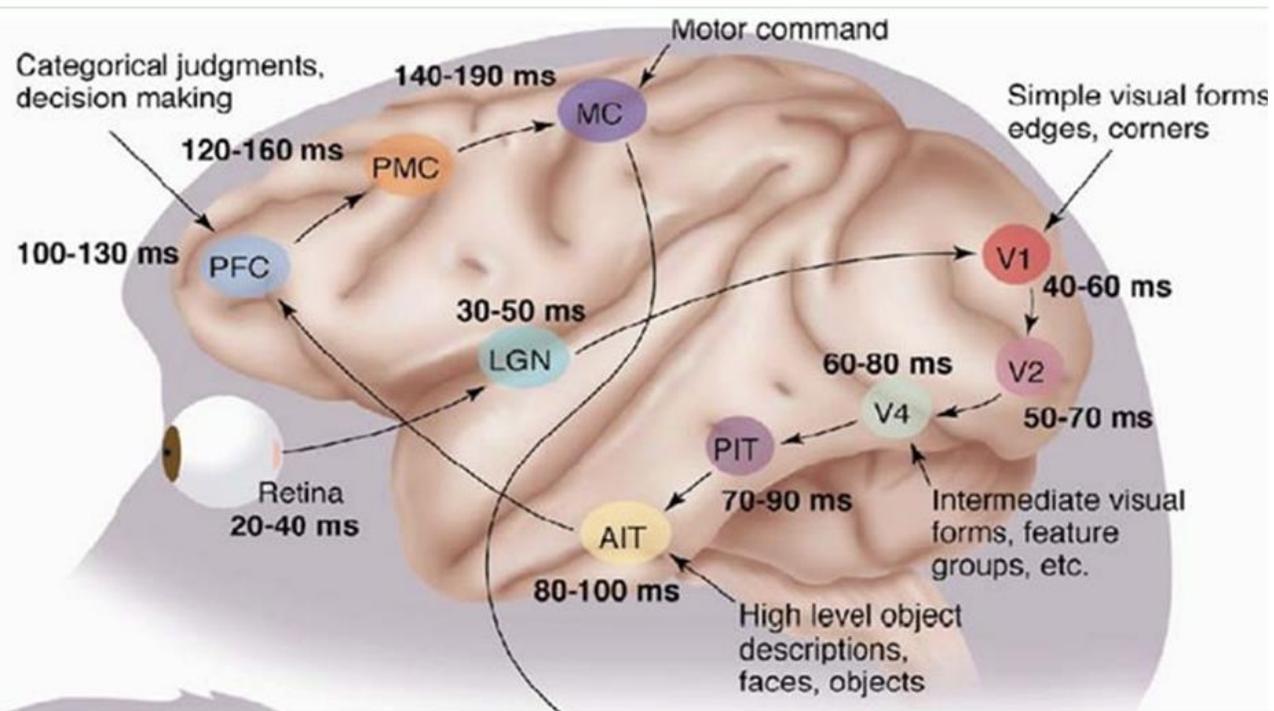
LES CNN : PERÇOIVENT-ILS ?

LES RESEAUX CNN

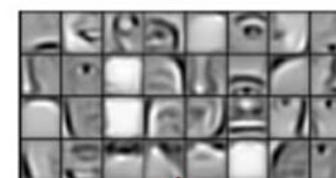
- Structurellement, il y a beaucoup de points communs avec le système visuel humain.

ConvNets VS Visual Neuroscience

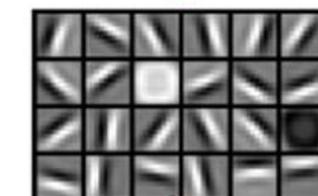
- Lower-level \rightarrow higher-level
- LGN-V1-V2-V4-IT ventral pathway (Hubel, 1962, ref. 43; Felleman, 1991, ref. 44)
- Time-delay neural networks (ref. 45-48)
- Document reading, object detection, ... (ref. 49-52)



object models



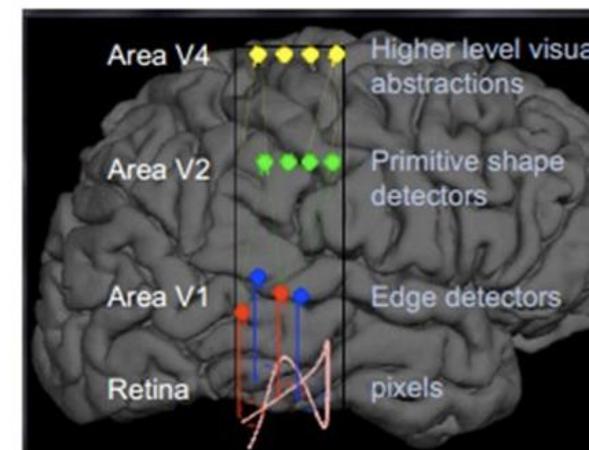
object parts
(combination
of edges)



edges



pixels



LES RESEAUX CNN

- Bien sûr, d'autres éléments peuvent être nécessaires pour la conscience (un espace de travail global, etc.) mais considère des points de vue comme celui de Ned Block, selon lequel les mécanismes d'ordre supérieur ne sont pas nécessaires !

LES RESEAUX CNN

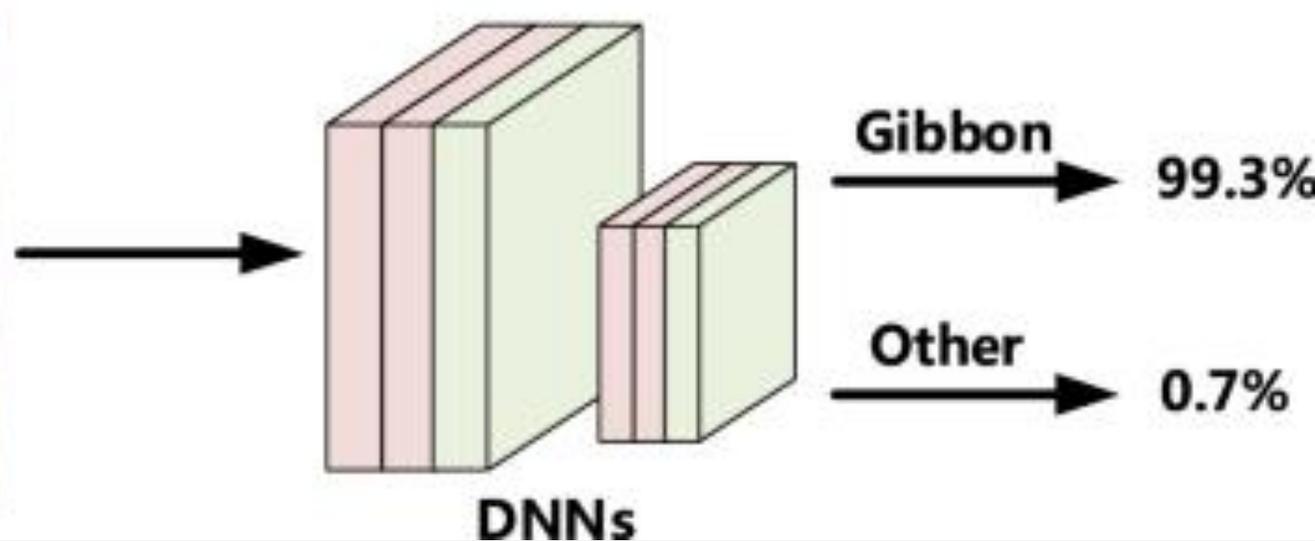
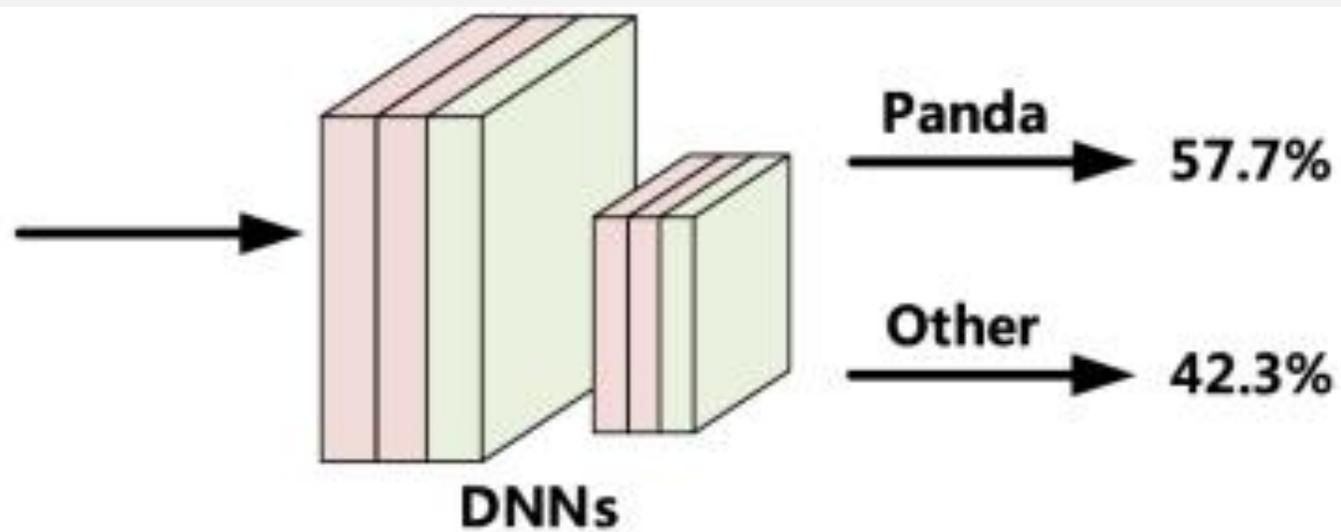
- Une caractéristique très étrange : les exemples adversarielles.
- La plupart ou tous les classificateurs d'images (ou de sons) peuvent être perturbés par un signal spécialement adapté, mais très faible, pour changer n'importe quelle décision en presque n'importe quelle autre !



+



=





x

“panda”

57.7% confidence

+ .007 ×



$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

=

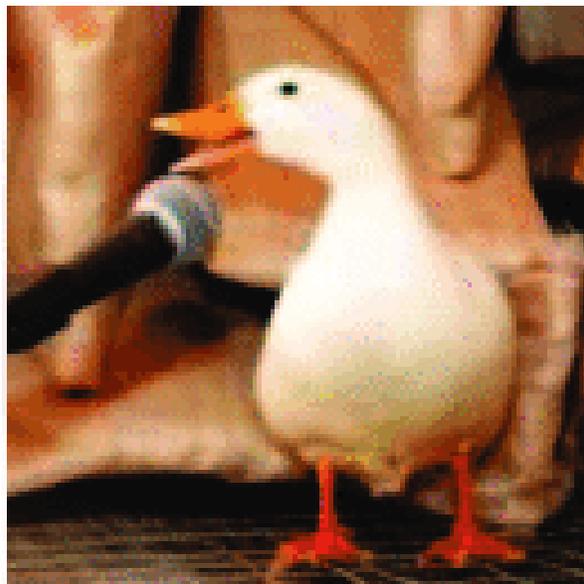


$x +$

$\epsilon \text{sign}(\nabla_x J(\theta, x, y))$

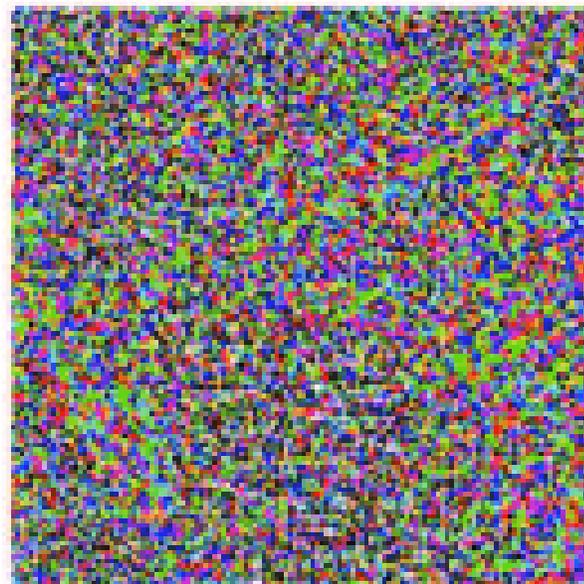
“gibbon”

99.3 % confidence



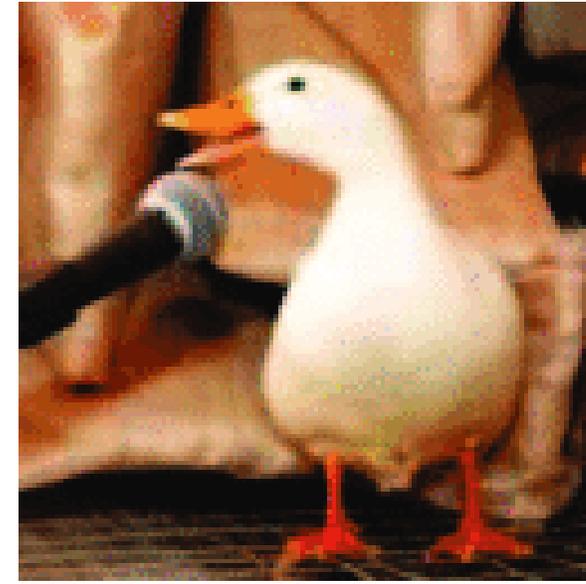
‘Duck’

+

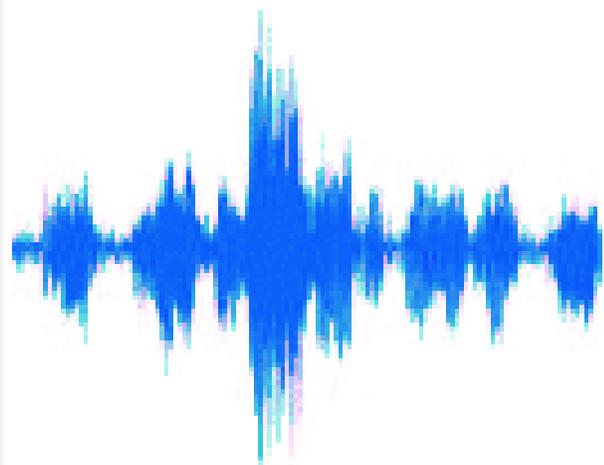


$\times 0.07$

=



‘Horse’



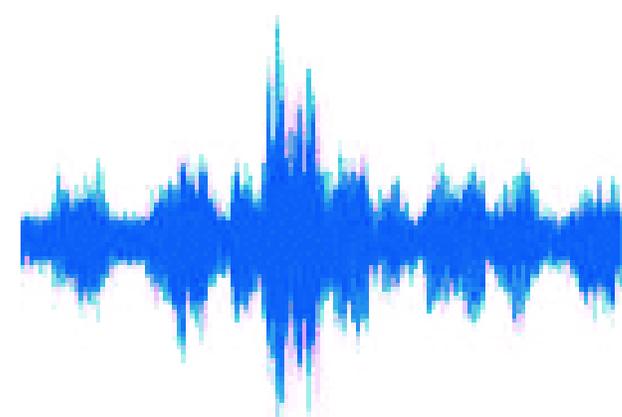
‘How are you?’

+



$\times 0.01$

=



‘Open the door’

LES RESEAUX CNN

- On peut se demander ce qu'il faut dire à ce sujet.
On peut dire que nous sommes sensibles à des choses similaires :





LES RESEAUX CNN

- Exemples audio également (Yanni-laurel)
- <https://www.nytimes.com/interactive/2018/05/16/upshot/audio-clip-yanny-laurel-debate.html>

LES RESEAUX CNN

- En gros, le point est qu'il existe des chemins détournés surprenants dans l'espace de qualité...
- Mais la question est de savoir si celles que nous avons sont profondément différentes de celles que les ANNs ont

LES RESEAUX CNN

- S'agit-il d'un artefact de sur-apprentissage (même si nous appliquons ici le principe de localité) - ou apprennent-ils simplement des concepts différents des nôtres ?
- Cela montre-t-il qu'ils travaillent fondamentalement différemment de nous ?
- ou apprennent-ils simplement des concepts différents des nôtres ?

NATIVISME VS EMPIRISME

LES RESEAUX CNN

nativisme vs empirisme: e grand vieux débat entre Hume et Kant : apprenons-nous simplement de l'expérience ou avons-nous des concepts innés ?

Les CNN montrent une façon dont nous pouvons apprendre à effectuer une tâche très impressionnante – découvrir des abstractions et les utiliser pour la classification d'images - sans aucun concept inné.

LES RESEAUX CNN

- Il faut souligner que le succès des CNN au cours des 15 dernières années est venu après des décennies d'efforts sur différentes approches dans lesquelles les règles pour savoir comment déduire des arrangements de bords et de contrastes, à des caractéristiques plus complexes (GOFAL / systèmes experts / Fodor) - mais ces approches ont toutes échoué.

LES RESEAUX CNN

- C'est pourquoi je suis d'accord avec Buckner pour dire que les réseaux profonds soulèvent de nouvelles questions philosophiques : ils suggèrent qu'une conception plus "empirique" pour des êtres tels que nous est au moins viable, alors qu'auparavant elle était mise en doute

LES RESEAUX CNN

- Cependant, il n'est pas vrai que les CNN soient des ardoises vierges : leurs hypothèses sur la localité sont spécifiques et contingentes : nous pouvons dire qu'il existe des tendances innées intégrées dans leur architecture

LES RESEAUX CNN

Nous pouvons donc apprendre en les étudiant, et peut-être avoir une idée plus précise de ce que le nativisme pourrait signifier (si le connexionnisme plutôt que le classicisme est correct).

LES RESEAUX CNN

Nous pouvons également apprendre en observant les limites qui subsistent, même dans les systèmes qui fonctionnent très bien :

a) l'avidité des données

b) leur incapacité à gérer les relations logiques (par opposition aux relations probabilistes)...