

SYS843

A.4 Les algorithmes d'apprentissage profond.

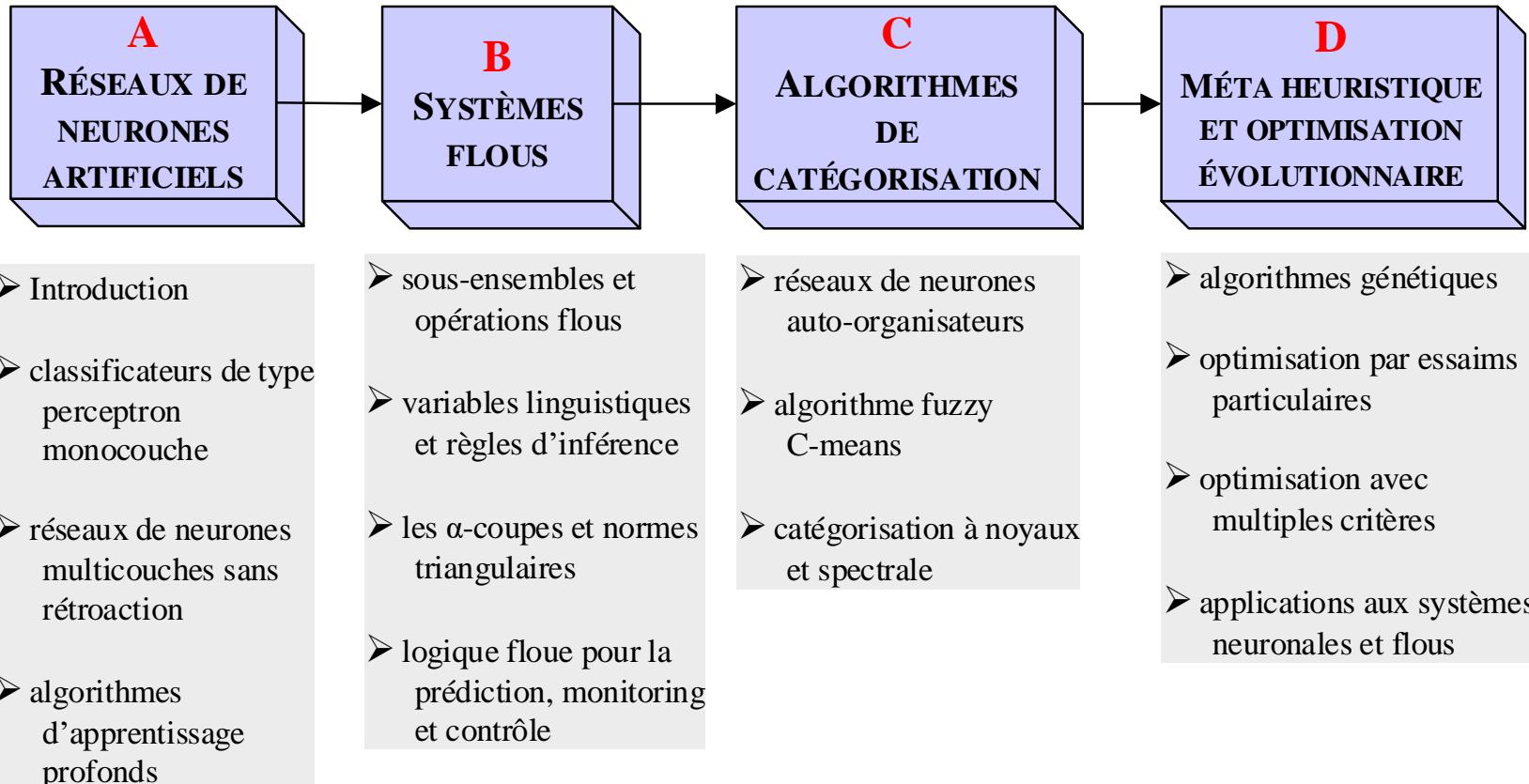
Marc-André Carbonneau

Eric Granger

Ismail Ben Ayed

Hiver 2016

CONTENU DU COURS



CONTENU DU COURS

Réseaux de neurones multicouches

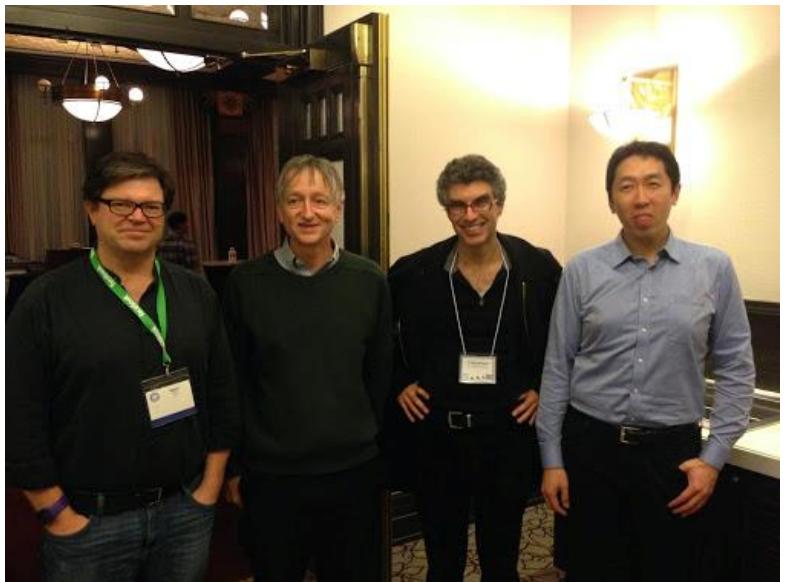
- ▶ Les RNA multicouches sont difficiles à entraîner en utilisant la rétropropagation quand le réseau possède plus de 2 couches cachées.
- ▶ Des techniques particulières ont été proposée pour entraîner des réseau plus complexes. On parle dans ce cas d'apprentissage profond (deep learning).

Introduction

Citations intéressantes

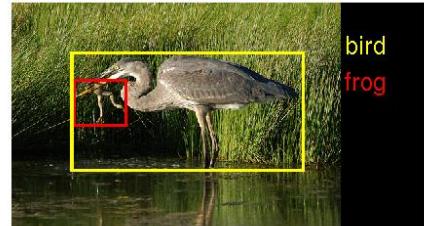
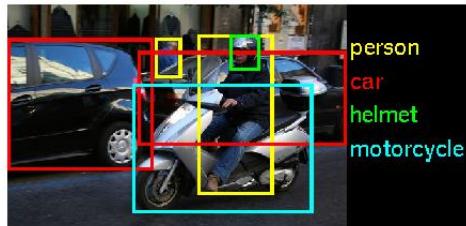
The remarkable thing was that it [Google Brain 2012] had discovered the concept of a cat itself. No one had ever told it what a cat is. That was a milestone in machine learning. That was the milestone that helped inspire many other companies, including Facebook and Baidu and a few others.
(Andrew Ng 2014)

<http://www.forbes.com/sites/roberthof/2014/08/28/interview-inside-google-brain-founder-andrew-nings-plans-to-transform-baidu/>



Classification: La base de données ImageNet

- ▶ Comprend 1,3 million d'image à haute résolution trouvé sur le web.
- ▶ 1000 classes d'objets à reconnaître.
- ▶ En 2010, le meilleur système obtenais 47% d'erreur pour son premier choix et 25% pour ses 5 premiers choix.
- ▶ En 2013, le concours consistait à localiser des objets dans les images.
 - Il y avait 200 classes d'objets
 - Plus de 400 000 images
 - Remporté par le GoogLeNet et un ensemble de RNA à convolution
 - 6.67% d'erreur

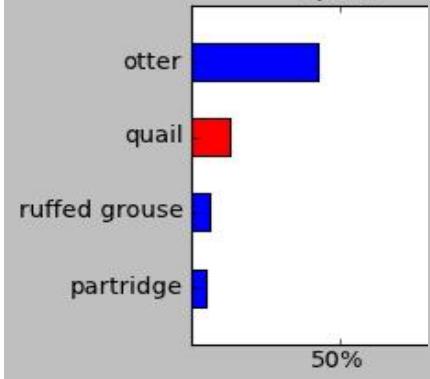


Tirée du cours de “[Neural Networks for Machine Learning](#)” Geoffrey Hinton (<https://class.coursera.org>) et de <http://image-net.org/challenges/LSVRC/2013/>

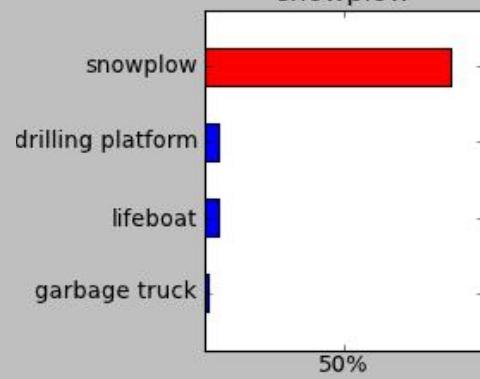
Classification: La base de données ImageNet



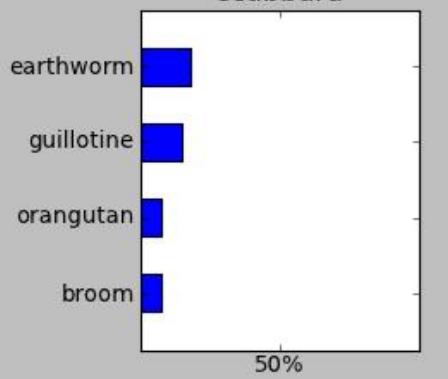
quail



snowplow



scabbard



Tirée du cours de "[Neural Networks for Machine Learning](#)" Geoffrey Hinton (<https://class.coursera.org/>)

Classification: La base de données ImageNet



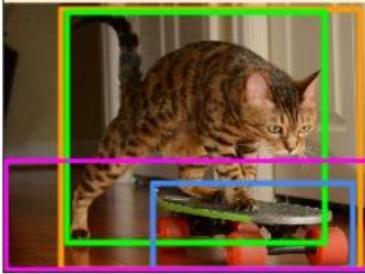
Tirée du cours de “[Neural Networks for Machine Learning](#)” Geoffrey Hinton
(<https://class.coursera.org/>)

Deep Dreams!



<http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html>

Au-delà de la reconnaissance d'images

| Classification | Captioning | Dense Captioning | | | | |
|--|---|---|--------------------|----------------------------|-------------------------|-------------------------|
|  |  Cat |  A cat riding a skateboard | | | | |
| | | <table border="1"> <tr><td>Orange spotted cat</td></tr> <tr><td>Skateboard with red wheels</td></tr> <tr><td>Cat riding a skateboard</td></tr> <tr><td>Brown hardwood flooring</td></tr> </table> | Orange spotted cat | Skateboard with red wheels | Cat riding a skateboard | Brown hardwood flooring |
| Orange spotted cat | | | | | | |
| Skateboard with red wheels | | | | | | |
| Cat riding a skateboard | | | | | | |
| Brown hardwood flooring | | | | | | |

[PDF] Connecting Images and Natural Language, [Andrej Karpathy](#), PhD Thesis, 2016



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.

<http://cs.stanford.edu/people/karpathy/>

Au-delà de la reconnaissance d'images

Cell sensitive to position in line:

```
The sole importance of the crossing of the Berezina lies in the fact
that it plainly and indubitably proved the fallacy of all the plans for
cutting off the enemy's retreat and the soundness of the only possible
line of action--the one Kutuzov and the general mass of the army
demanded--namely, simply to follow the enemy up. The French crowd fled
at a continually increasing speed and all its energy was directed to
reaching its goal. It fled like a wounded animal and it was impossible
to block its path. This was shown not so much by the arrangements it
made for crossing as by what took place at the bridges. When the bridges
broke down, unarmed soldiers, people from Moscow and women with children
who were with the French transport, all--carried on by vis inertiae--.
pressed forward into boats and into the ice-covered water and did not,
surrender.
```

Cell that turns on inside quotes:

```
"You mean to imply that I have nothing to eat out of.... On the
contrary, I can supply you with everything even if you want to give
dinner parties," warmly replied Chichagov, who tried by every word he
spoke to prove his own rectitude and therefore imagined Kutuzov to be
animated by the same desire.
```

```
Kutuzov, shrugging his shoulders, replied with his subtle penetrating
smile: "I meant merely to say what I said."
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!!(current->notifier)(current->notifier_data)) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```

<http://cs.stanford.edu/people/karpathy/>

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

Motivations – Deep Learning

- ▶ **Les cerveaux animaux et humains ont une architecture profonde (deep).**
- ▶ **Les humains organisent leurs idées de manière hiérarchique.**
- ▶ **Permet la représentation de données complexes.**

CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement
- 5) Type de couches cachées
- 6) Les réseaux à convolution

Les auto-encodeurs

Aussi appelés autoassociators et Diabolo networks

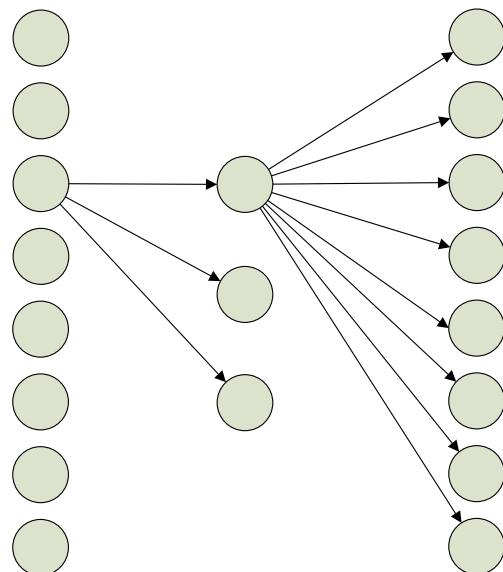
Les auto-encodeurs

Pour entraîner un auto-encodeur, on lui présente toutes les formes de la base de données et on entraîne le réseau à l'aide de la rétropropagation ou un autre algorithme d'apprentissage supervisé.

Il y a autant de neurones d'entrée que de neurones de sortie.

La réponse désirée pour l'entraînement est la même forme d'entrée:

$$d = x$$



Les auto-encodeurs

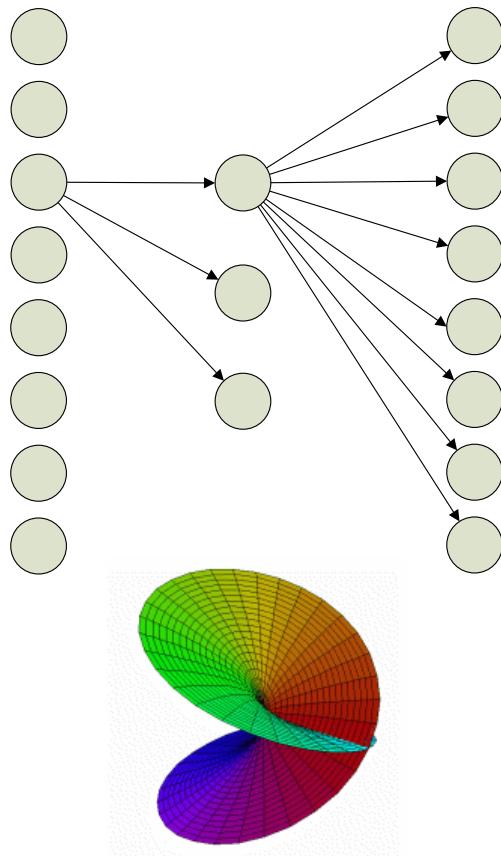
Utilisés pour compresser de l'information.

- Moins de neurones sur la couche cachée que sur la couche d'entrée.
- Similaire à une PCA*.

Utilisés pour extraire les caractéristiques (composante) d'une base de données.

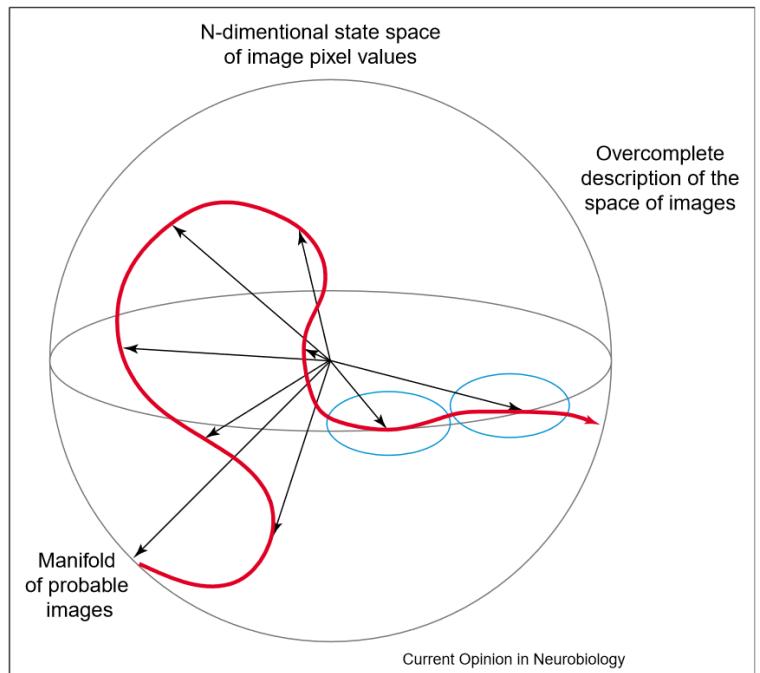
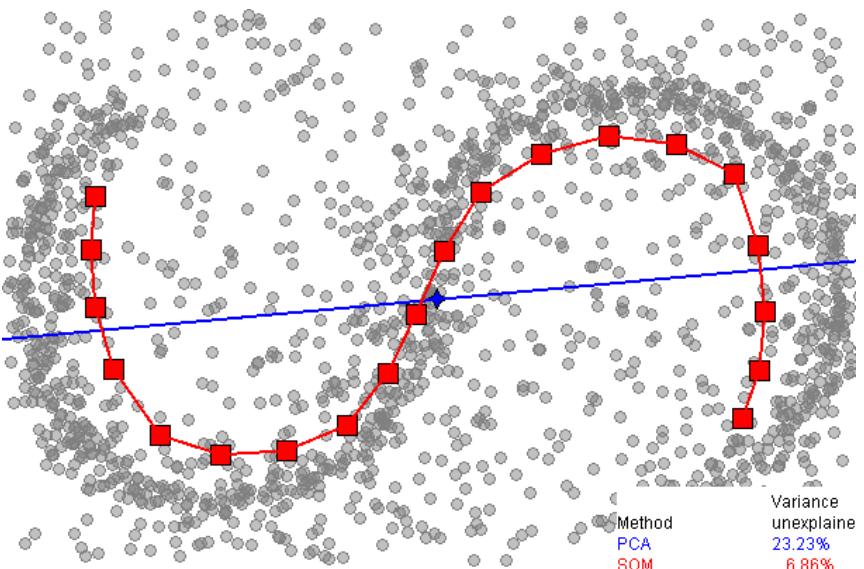
- Plus de neurones sur la couche cachée que sur la couche d'entrée.
- Découvre la structure naturelle des données.

Quand des réseaux multicouches sont utilisés avec des fonction d'activation non-linéaires, ils permettent d'encoder l'information dans un espace à géométrie complexe souvent appelé manifold.



Manifolds

Informellement : un système d'axes non linéaire où résident les données d'un problème.



Objectif d'optimisation dans les réseaux de neurones

On a une base de données d'entraînement

$$X = \{(x_1, d_1), (x_2, d_2), \dots, (x_N, d_N)\}$$

On veut minimiser l'erreur quadratique moyenne (MSE) on a que :

$$J(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \|d_i - y_i\|^2$$

La variable y_i est la sortie du réseau de neurones quand x_i est à l'entrée.

La matrice \mathbf{W} contient les poids synaptiques du réseau.

Objectif d'optimisation dans les réseaux de neurones

Pour éviter que les poids synaptiques ne deviennent trop grands et donc prévenir le sur-apprentissage, on ajoute un terme de régularisation :

$$J(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \|d_i - y_i\|^2 + \lambda \|\mathbf{W}\|_F^2$$

La matrice \mathbf{W} contient les poids synaptiques w du réseau.

λ est un paramètre de compromis entre les deux objectifs d'optimisation.

La norme de Frobenius est donnée par :

$$\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m w_{ij}^2}$$

Sparsity

- ▶ En utilisant un encodage sparse, il est possible de découvrir une structure dans les données. En d'autres mots, trouver les éléments de base permettant de décrire le contenu de la base de donnée.
- ▶ En français, sparse pourrait être traduit par : épars, clairsemé, peu abondant.
- ▶ Une matrice est dite sparse si la plupart de ses éléments sont nuls.
- ▶ Par sparse, on veut dire que peu d'éléments sont utilisés pour encoder un signal.
- ▶ Dans le cas d'un auto-encodeur neuronique, ça signifie que très peu de neurones sont activés en même temps.

Objectif d'optimisation dans les réseaux de neurones

Pour obtenir un encodage sparse, on doit ajouter un terme à notre objectif d'optimisation :

$$J(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \|d_i - y_i\|^2 + \lambda \|\mathbf{W}\|_F^2 + \sum_{k \in L} KL(p || \hat{p}_k)$$

Ce terme sert à s'assurer, qu'en moyenne, seule une faible proportion p de neurones sur les couches cachées (L) sont activés.

Objectif d'optimisation dans les réseaux de neurones

$$J(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \|d_i - y_i\|^2 + \lambda \|\mathbf{W}\|_F^2 + \sum_{k \in L} KL(p || \hat{p}_k)$$

\hat{p}_k est une approximation de la proportion d'activation du neurone k pour toute la base d'apprentissage :

$$\hat{p}_k = \frac{1}{N} \sum_{i=1}^N y_i^{(k)}$$

La mesure de divergence de Kullback-Leibler (KL) est donnée par :

$$KL(p || \hat{p}_k) = p \log \frac{p}{\hat{p}_k} + (1 - p) \log \frac{1 - p}{1 - \hat{p}_k}$$

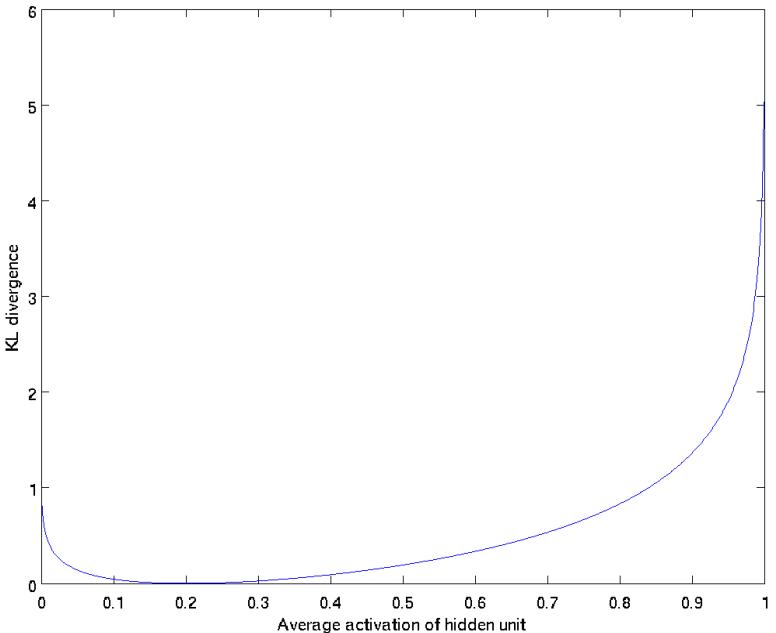
Divergence de Kullback-Leibler (KL)

Supposons qu'on veut que les neurones des couches cachées soient actif 20% du temps:

$$p = 0.2$$

Le graphique montre le terme de pénalité de *sparsity* pour un neurone donné.

$$KL(p||\hat{p}_k) = p \log \frac{p}{\hat{p}_k} + (1 - p) \log \frac{1 - p}{1 - \hat{p}_k}$$



Source: <http://deeplearning.stanford.edu/wiki/index.php>

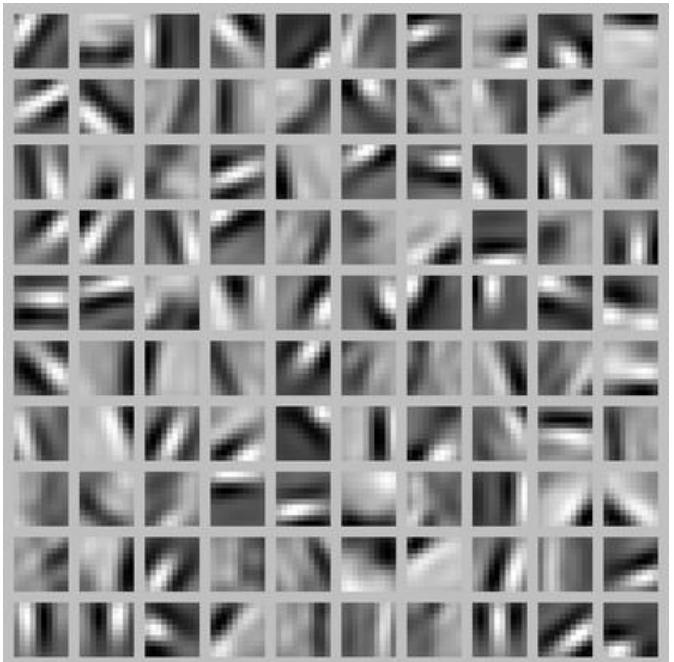
Exemple – apprentissage de caractéristiques

Si l'on prend des imagettes extraites d'images naturelles et qu'on apprend avec un auto-encodeur sparse, on obtient les bases représentées à droite.

Ce sont les images qui provoquent une activation maximale pour chacun des 100 neurones.

On obtient des détecteurs d'arrêtes (edge detectors).

Il s'agit du genre de détecteurs que l'on retrouve dans les niveaux inférieurs du cortex visuel.

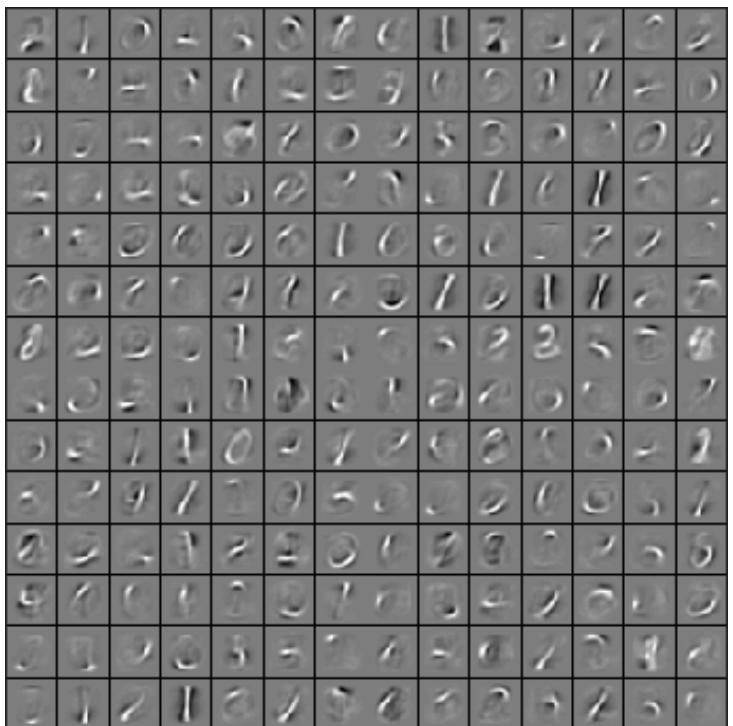


Source: <http://deeplearning.stanford.edu/wiki/index.php>

Exemple – apprentissage de caractéristiques

Ici, on a apprend les bases en utilisant des imagettes de chiffres manuscrits :

0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9



Self-Taught Learning

CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement
- 5) Type de couches cachées
- 6) Les réseaux à convolution

Plus de données = meilleures performances

“Sometimes it's not who has the best algorithm that wins; it's who has the most data.”

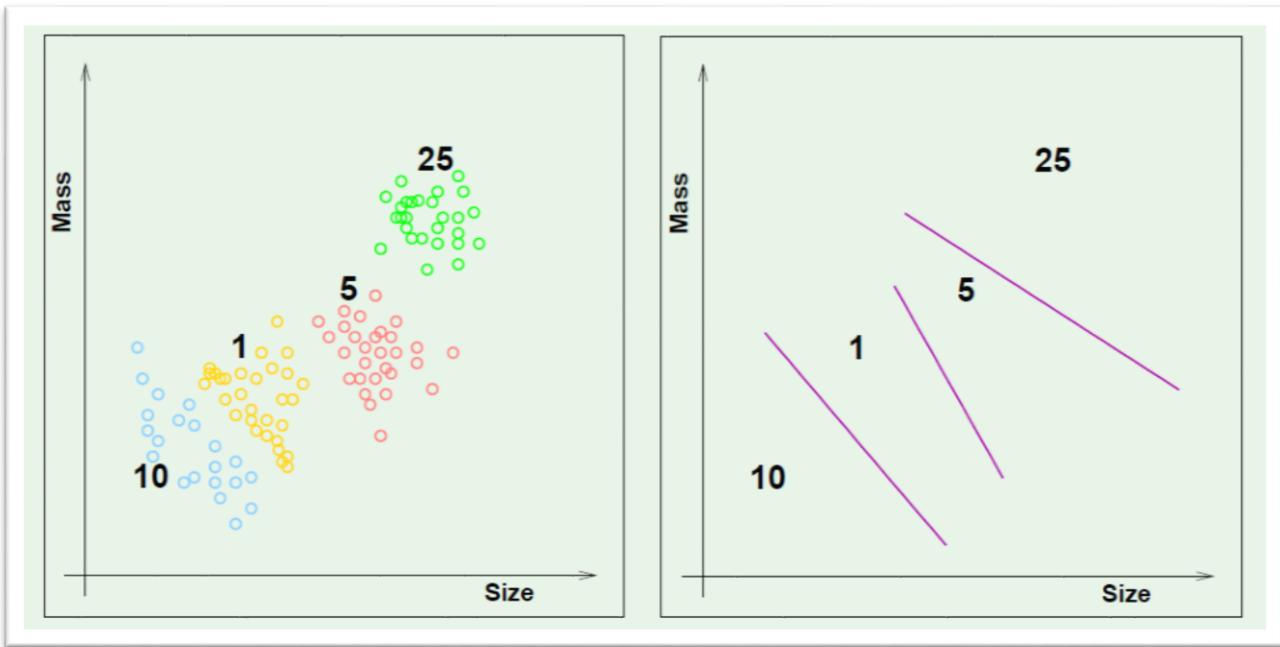
http://deeplearning.stanford.edu/wiki/index.php/Self-Taught_Learning

- ▶ **Dans plusieurs applications, obtenir des données est relativement facile.**
- ▶ **Par contre, obtenir des étiquettes est généralement coûteux en temps et en ressources.**
- ▶ **Serait-il possible d'utiliser des données non-étiquetées pour l'apprentissage?**

Types d'apprentissage

- ▶ **Supervisé**
- ▶ **Non-supervisé**
- ▶ **Semi-supervisé**
- ▶ **Faiblement-supervisé**
- ▶ **Par renforcement**

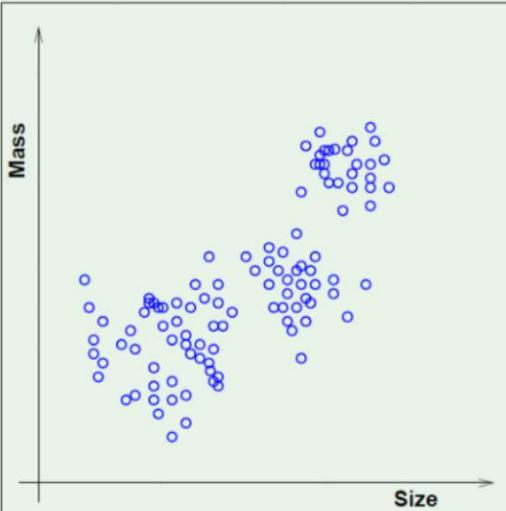
Apprentissage Supervisé



Source image: Caltech's Machine Learning Course - CS 156 <https://www.youtube.com/watch?v=mbyG85GZ0PI>

Apprentissage Non-Supervisé

Instead of
(input, correct output),
we get
(input, ?)



Source image: Caltech's Machine Learning Course - CS 156 <https://www.youtube.com/watch?v=mbyG85GZ0PI>

Apprentissage de caractéristiques non-supervisé

- ▶ Si on utilise une très grande quantité de données, il est possible d'apprendre des caractéristiques (features) et de les utiliser pour des tâche de reconnaissance.
- ▶ Dans beaucoup d'applications, cette approche surpassé les approches reposant sur des caractéristiques créées par un ingénieur.

Exemple :

- ▶ Pour reconnaître des objets dans des images, on peut extraire les arrêtes, des histogrammes de couleur, le spectre fréquentiel, des ondelettes, etc.
- ▶ Alternativement, on peut utiliser un auto-encodeur tel que vu à la section précédente.

Self-Taught Learning

Objectif : Utiliser une grande quantité de données non-étiquetées pour améliorer les performances des algorithmes d'apprentissage.

1. On apprend des caractéristiques de manière non-supervisée en utilisant une très grande quantité de données.
 - **On obtient donc une bonne représentation des données.**
2. On utilise ensuite les données étiquetées, moins nombreuses, pour l'apprentissage final en mode supervisé.

Self-Taught Learning - Exemple

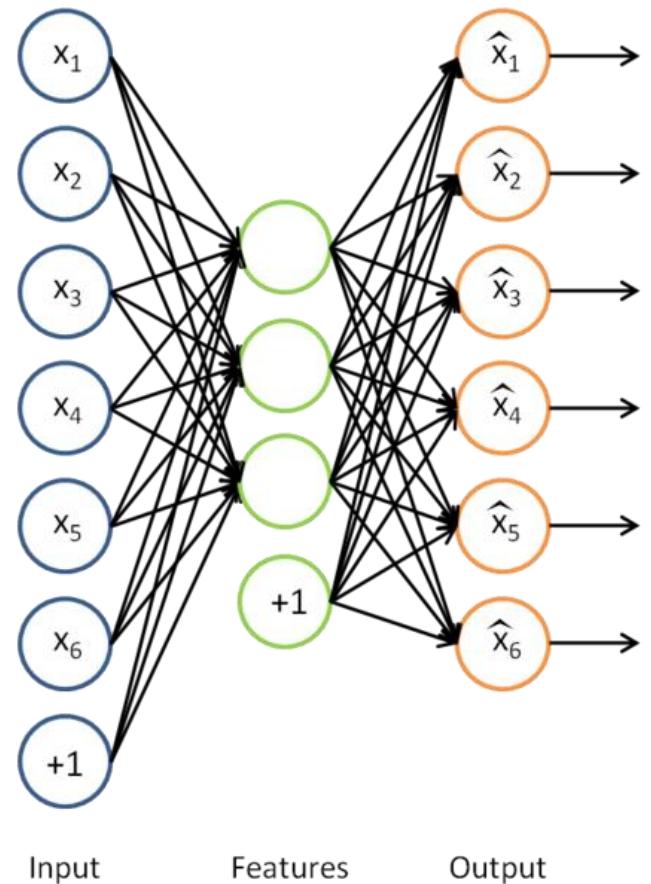
1. On apprend des caractéristiques de manière non-supervisée en utilisant une très grande quantité de données.

- On obtient donc une bonne représentation des données.

On utilise un auto-encodeur.

Si $x = \{x_1, \dots, x_6\}$ et $\hat{x} = \{\hat{x}_1, \dots, \hat{x}_6\}$

- On veut que $\hat{x} = x$



Source: http://deeplearning.stanford.edu/wiki/index.php/Self-Taught_Learning

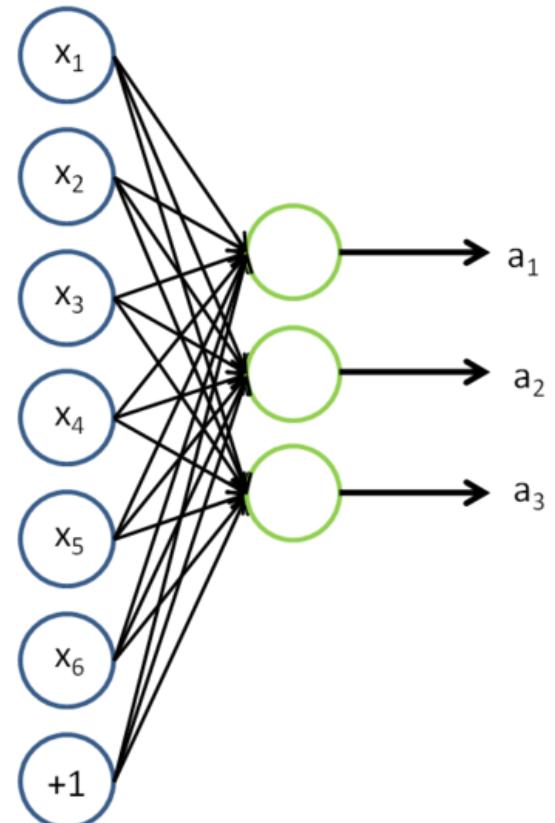
Self-Taught Learning - Exemple

Une fois l'apprentissage terminé, on retire la couche de sortie de l'auto-encodeur.

La sortie des neurones de la couche cachées deviennent des caractéristiques décrivant les données.

Pour la représentation finale des données on peut utiliser $f = \{a_1, \dots, a_3\}$ directement ou concaténer x et a pour une représentation plus riche :

$$f = \{x_1, \dots, x_6, a_1, \dots, a_3\}.$$

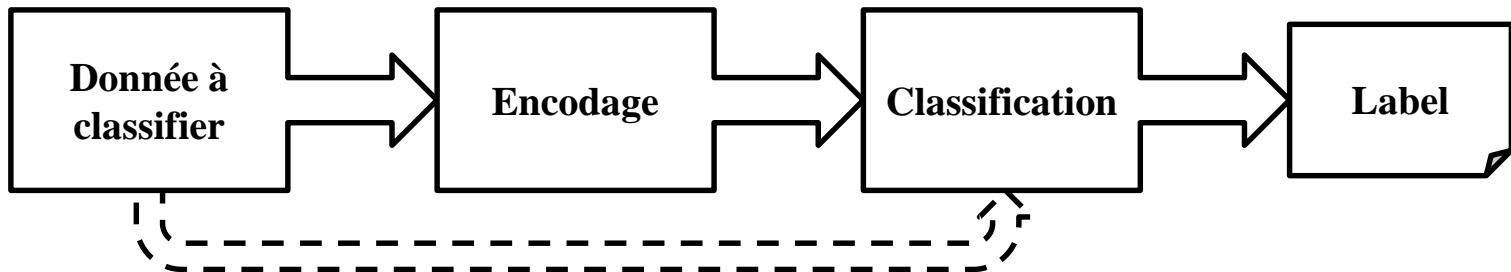


Input Features

Source:
http://deeplearning.stanford.edu/wiki/index.php/Self-Taught_Learning

Self-Taught Learning - Exemple

- ▶ L'étape finale consiste à entraîner un classificateur en utilisant les données étiquetés et leur nouvelle représentation donnée par f .
- ▶ Ce classificateur peut-être de n'importe quel type : SVM, régression logistique, MLP, etc.



Apprentissage Semi-Supervisé vs. Self-Taught Learning

Dans les problèmes semi-supervisés :

- ▶ On possède une base de données dont seulement une portion est étiquetée.
- ▶ Cette situation n'est pas très courante.
- ▶ Formellement, on suppose que les données sans étiquette proviennent exactement de la même distribution que celles étiquetées.

Dans un problème Self-Taught Learning :

- ▶ On suppose que les données sont du même type sans toutefois provenir de même distribution.

Les réseaux *deep-learning*

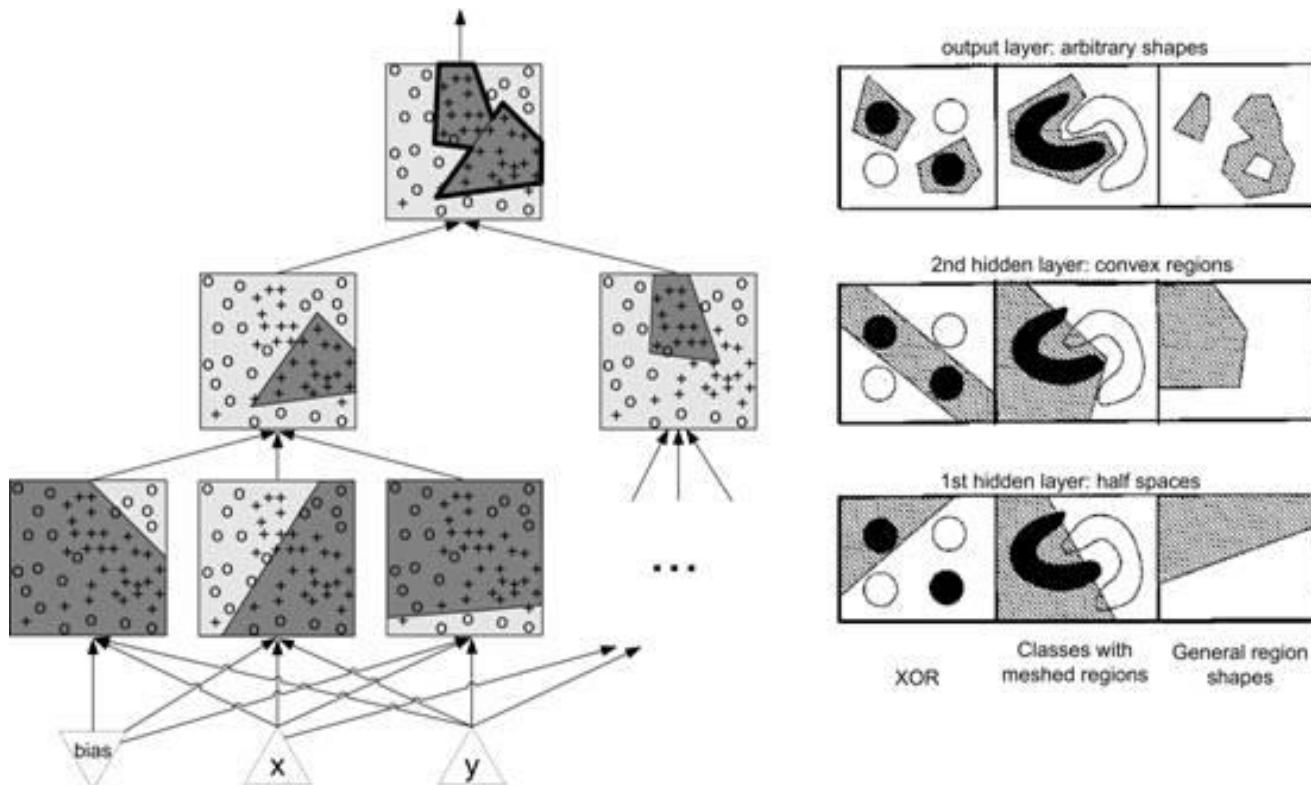
CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement
- 5) Type de couches cachées
- 6) Les réseaux à convolution

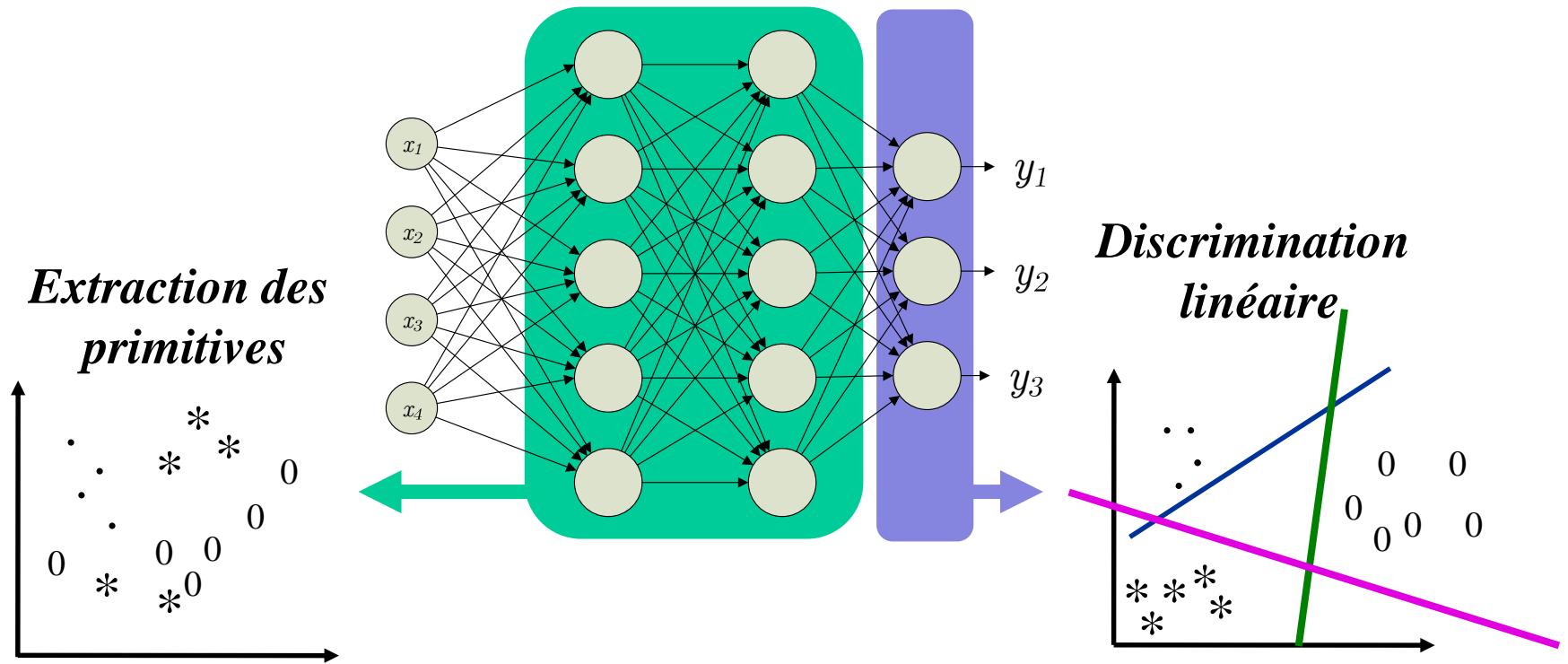
Pourquoi le MLP?

On peut obtenir des frontières de décisions plus complexes et disjointes.



Pourquoi le MLP?

Le système peut trouver lui-même ses propres *features*.



Problèmes avec les MLP

Si on a plus de 2 couches cachées :

- ▶ Pendant la rétropropagation, les gradients s'approchent du zéro-machine (ou de ∞). Autrement dit, les gradients deviennent tellement petits ou grands qu'ils ne peuvent plus être représentés par un ordinateur.
 - ▶ Il y a beaucoup de paramètres (poids synaptiques) à apprendre.
 - On a besoin de beaucoup de données.
 - On a besoin d'une grande puissance de calcul.
 - ▶ Beaucoup de minimum locaux dans la fonction d'optimisation.
 - La descente de gradient n'est plus aussi efficace.
-

Solutions

- ▶ Stratégies d'entraînement
 - Pre-entraînement
 - Pénalité de sparsity
 - Restricted Boltzmann machines
 - Drop out
 - ▶ Génération de grande bases de données
 - ▶ Self-Taught Learning
 - ▶ Utilisation des cartes graphiques (GPU) pour le calcul
-

MLP = Deep learning?

- ▶ **Oui... ...et non...**
 - ▶ **Dans un réseau deep learning :**
 - Il y a beaucoup de couches cachées (>3).
 - Il n'y a pas d'extraction de caractéristiques.
 - Chaque couche encode un niveau d'abstraction différent.
-

Nouvelle idée?

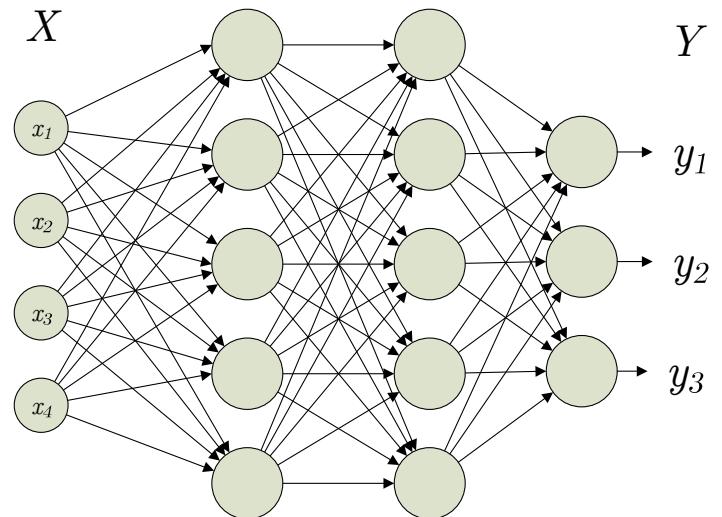
- ▶ On attribue l'invention du deep-learning à la Canadian Mafia:
 - Yann Lecun
 - Geoffrey Hinton
 - Yoshua Bengio
- ▶ Par contre, Alexey Grigorevich Ivakhnenko
 - 1965 publie le premier réseaux deep learning fonctionnel.
 - 1971 publie un réseau à 8 couches.
- ▶ Plusieurs auteurs critiquent la Canadian Mafia qui s'attribue les mérites de l'invention des concepts du deep learning
 - <http://people.idsia.ch/~juergen/deep-learning-conspiracy.html>

Avantages du Deep Learning

- ▶ Offre une représentation plus compacte.
- ▶ Permet une décomposition des données en niveaux d'abstraction.
- ▶ Ne nécessite pas de design de caractéristiques.
- ▶ Permet d'approximer des fonctions complexes (expressive power).

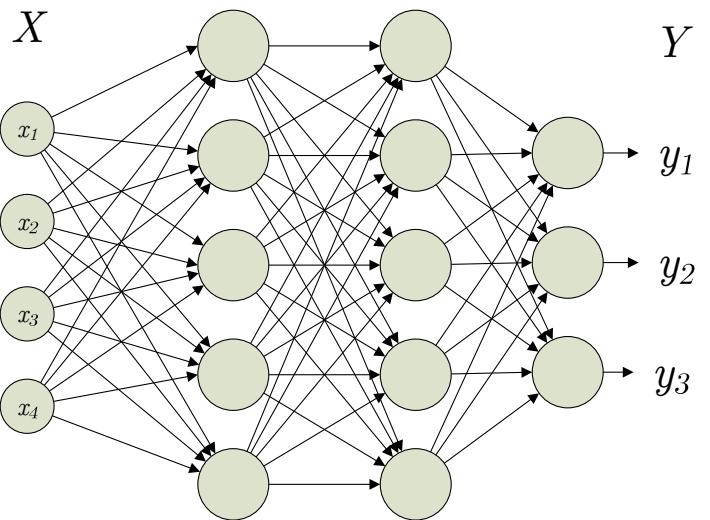
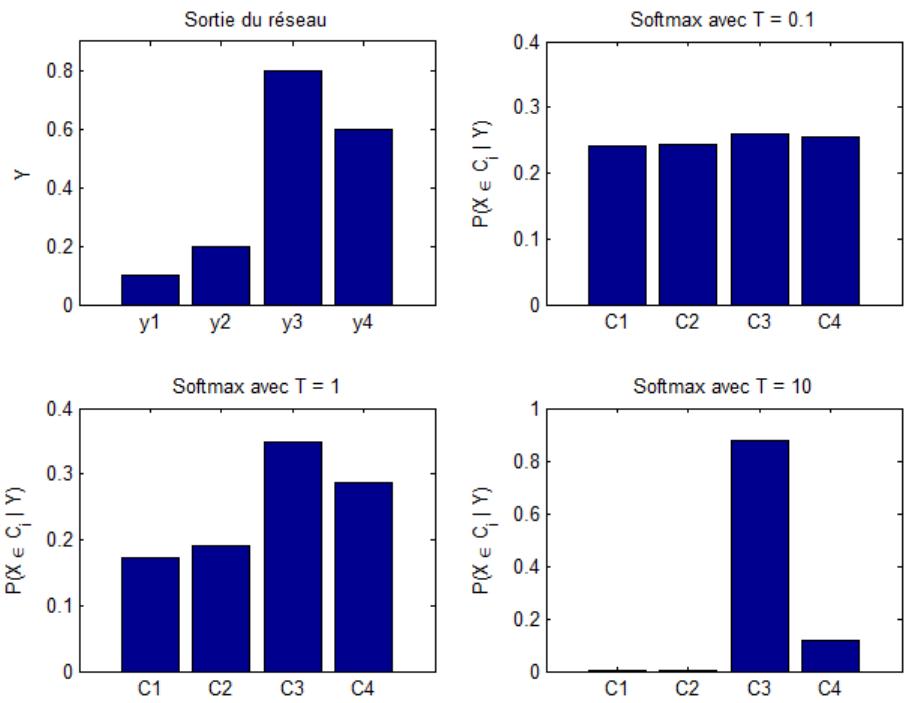
Softmax – classification multi-classe

- ▶ Pour réaliser la classification dans un problème à plus de deux classes, on utilise plusieurs neurones de sortie.
- ▶ On combine les sorties à l'aide de la fonction softmax.
- ▶ La fonction softmax approxime la probabilité que l'exemple X appartienne à la classe C_i en se basant sur toutes les valeurs de sortie Y du réseau.
- ▶ T est le paramètre de température du réseau et doit être ajusté empiriquement.



$$P(X \in C_i | Y) \simeq \frac{e^{Ty_i}}{\sum_{j=1}^{|Y|} e^{Ty_j}}$$

Softmax – classification multi-classe



$$P(X \in C_i | Y) \leq \frac{e^{Ty_i}}{\sum_{j=1}^{|Y|} e^{Ty_j}}$$

Stratégie d'entraînement

CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement**
- 5) Type de couches cachées
- 6) Les réseaux à convolution

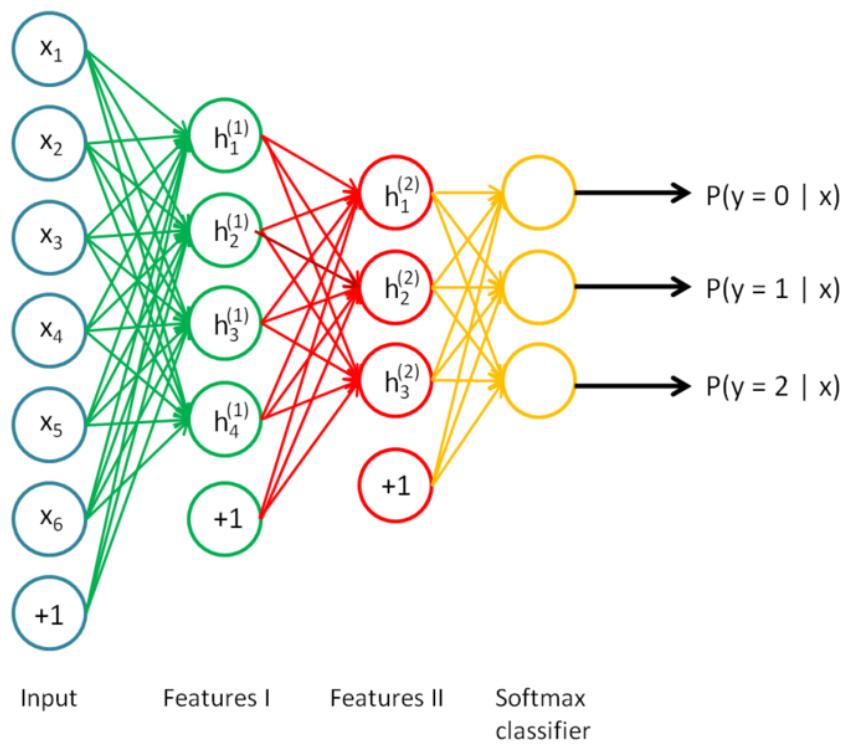
Stratégie d'entraînement

Pré-entraînement (Greedy Layer-wise training) :

1. On entraîne chacune des couches individuellement de manière non-supervisée.
 - Les couches peuvent être des auto-encodeurs ou des restricted Boltzmann machine (RBM)
2. On finalise l'apprentissage avec le système complet de manière supervisée.

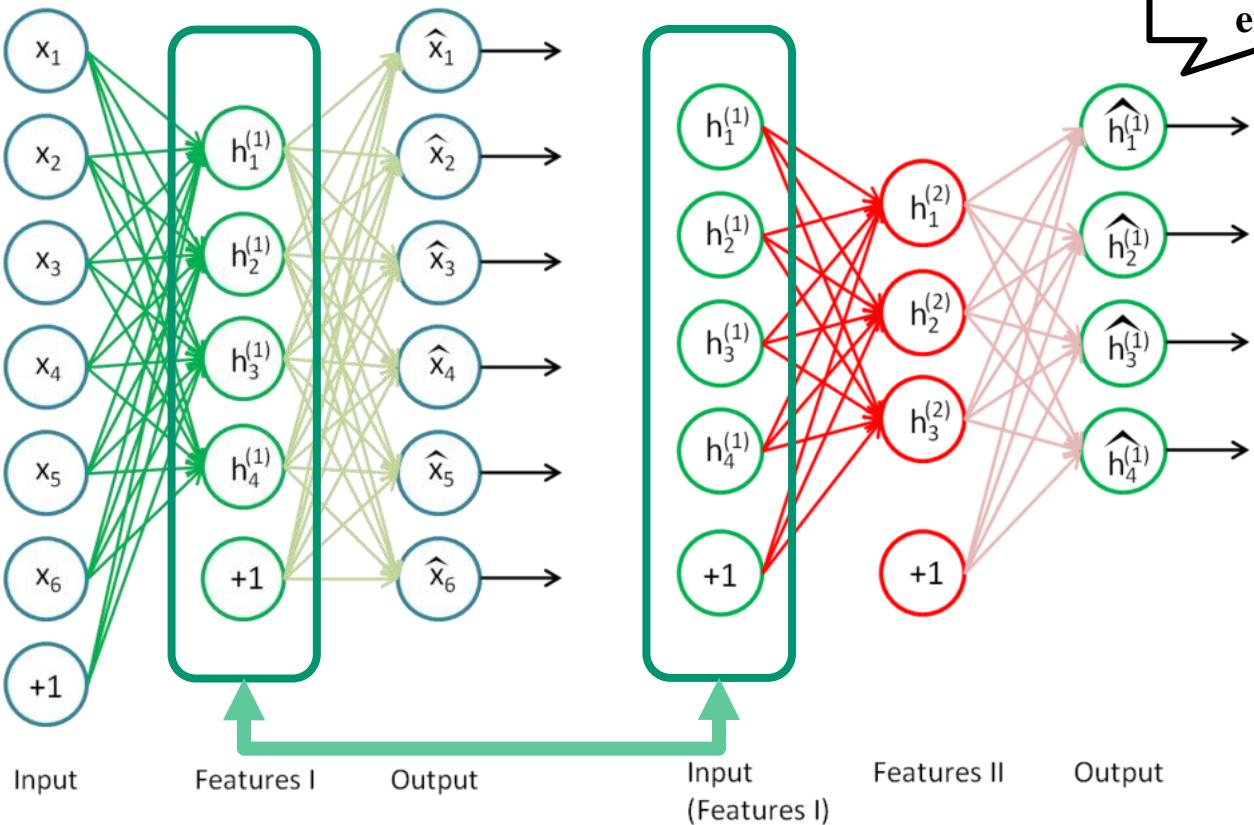
Stratégie d'entraînement - Exemple

On veut entraîner un réseau deep-learning avec 2 couches cachées. Le réseau sert à faire la classification de données $\mathbf{x} = \{x_1, \dots, x_6\}$ pouvant appartenir à 3 classes distinctes.



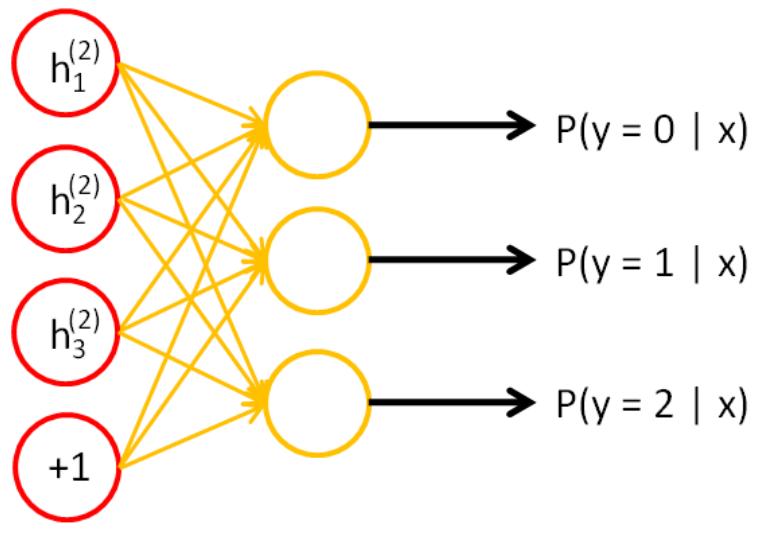
Stratégie d'entraînement - Exemple

Étape 1 :
Entraîner le
premier auto-
encodeur



Stratégie d'entraînement - Exemple

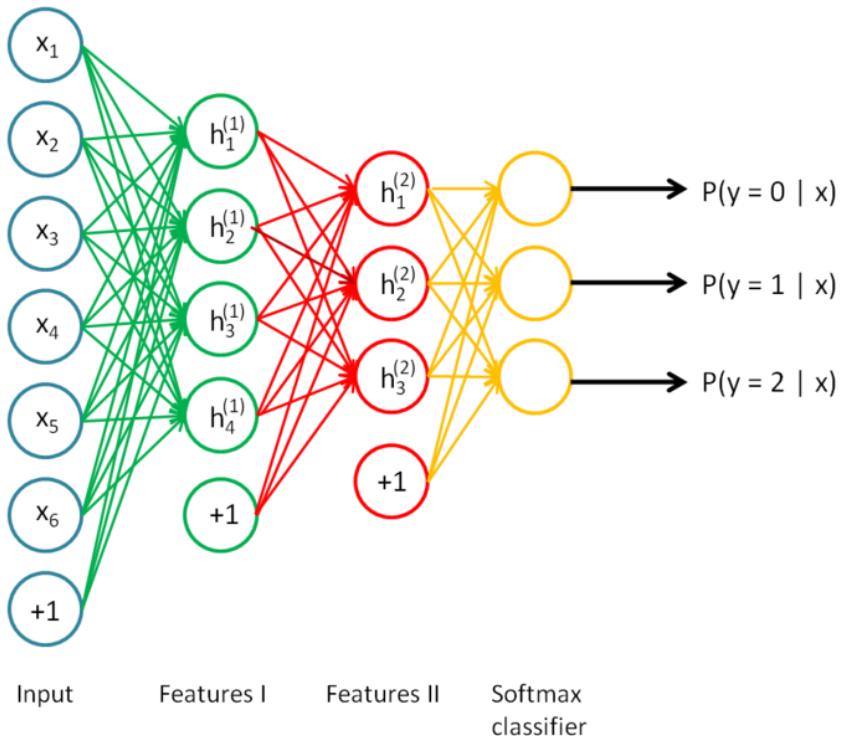
Étape 3 :
Entraîner un classificateur
(softmax, SVM, etc.)



Input
(Features II) Softmax
classifier

Stratégie d'entraînement - Exemple

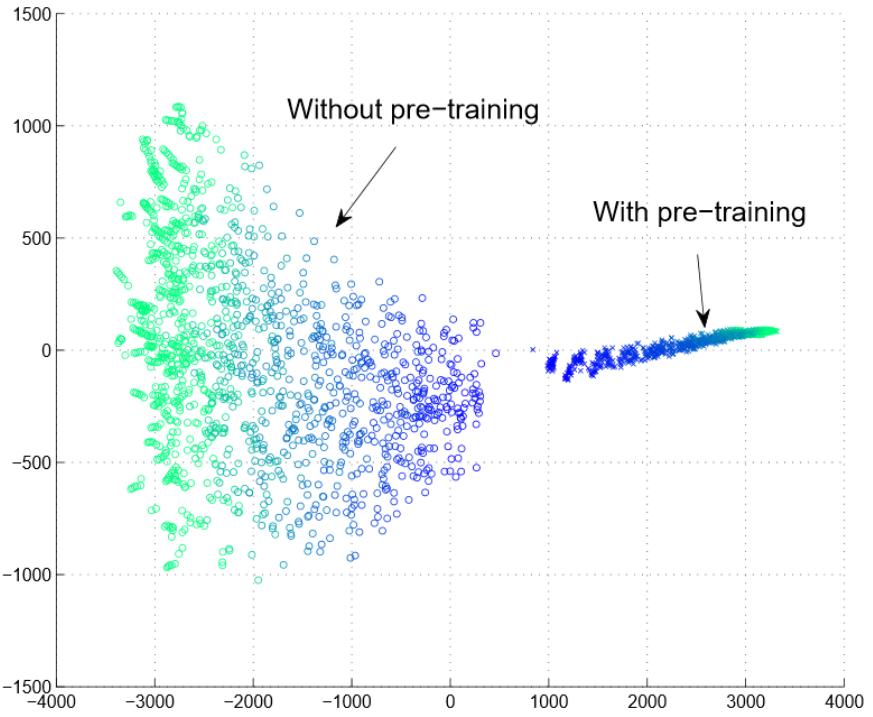
Étape 4 :
Entrainer de manière
supervisée tout le réseau
(fine tuning)



Effet de l'initialisation avec pré-entraînement

- ▶ Chaque point est un modèle dans l'espace *hypothèse*.
- ▶ La couleur encode le nombre d'époque d'entraînement.
- ▶ En haut : trajectoires sans pré-entraînement.
- ▶ En bas : trajectoires avec pré-entraînement.

- ▶ **Conclusion :** le pré-entraînement initialise la phase d'apprentissage à un meilleur endroit ce qui évite de converger vers l'un des nombreux minimums locaux.

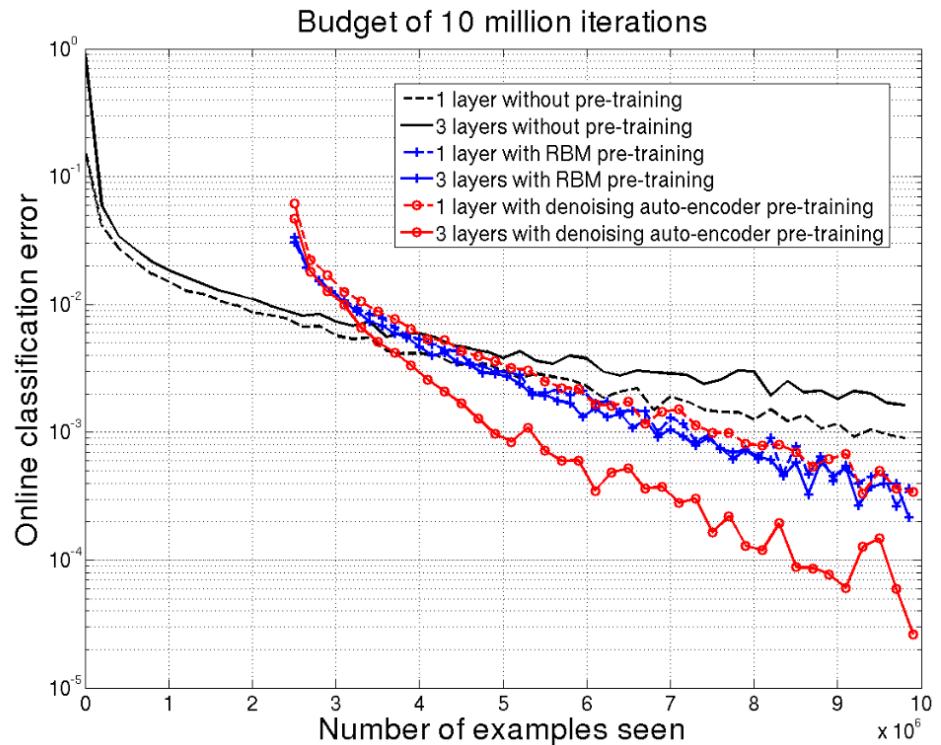


D. Erhan, et al., "Why Does Unsupervised Pre-training Help Deep Learning?", *JMLR*, 2010.

Effet de l'initialisation avec pré-entraînement

Dans cet exemple on utilise la base données de chiffres manuscrits *infinite* MNIST

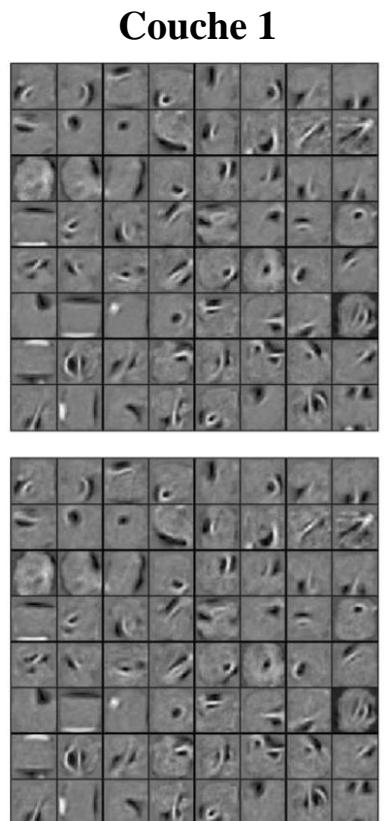
Conclusion : l'utilisation de restricted Boltzmann machine (RBM) et de denoising auto-encoders pré-entraînés donnent de meilleurs résultats de classification.



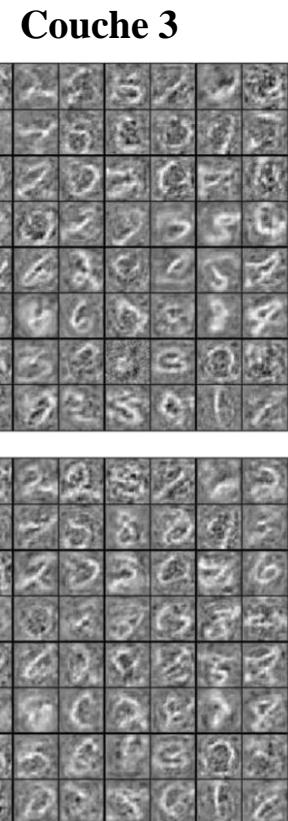
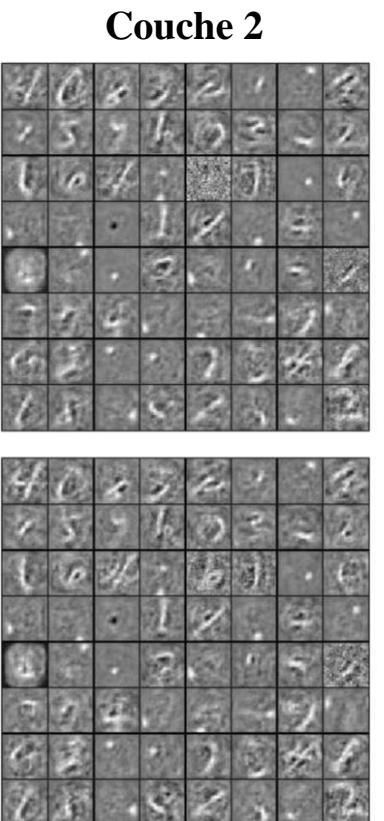
D. Erhan, et al., "Why Does Unsupervised Pre-training Help Deep Learning?", *JMLR*, 2010.

Effets du fine tuning

Après pré-entraînement
(non-supervisé)



Après *fine tuning*
(supervisé)



D. Erhan, et al., "Why Does Unsupervised Pre-training Help Deep Learning?", *JMLR*, 2010.

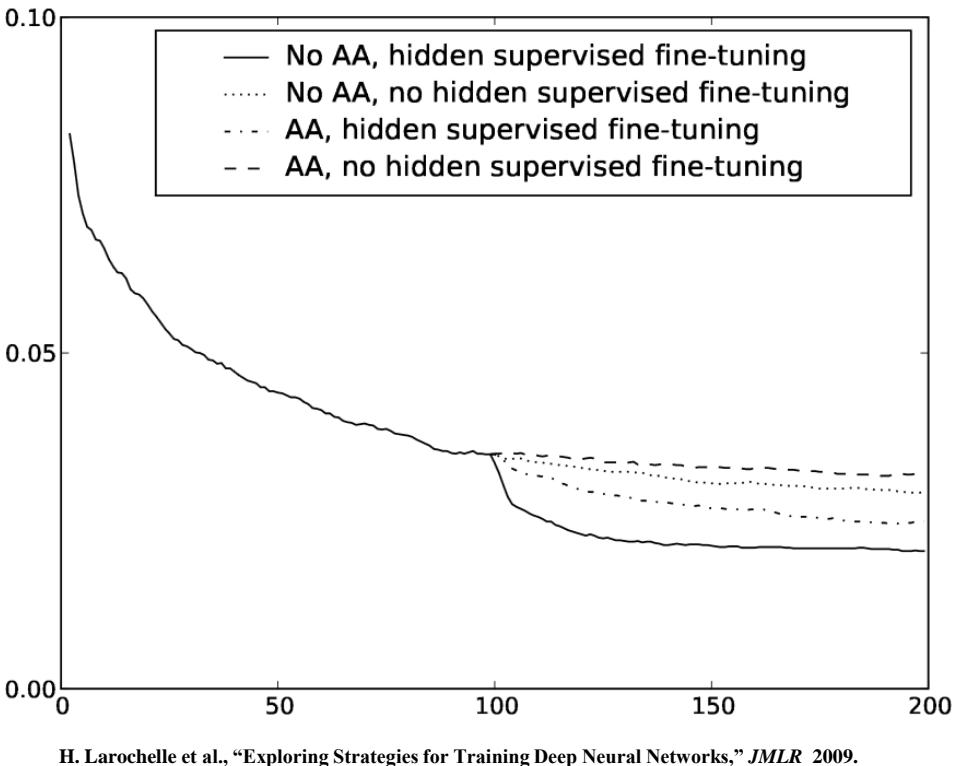
Effets du fine tuning

Au départ le réseau est entraîné couche par couche.

À la 100^e itération :

- ▶ On commence le fine tuning sur la dernière couche ou toutes les couches (hidden supervised fine-tuning).
- ▶ On continue ou non d'entrainer les auto-encodeur (AA/No AA).

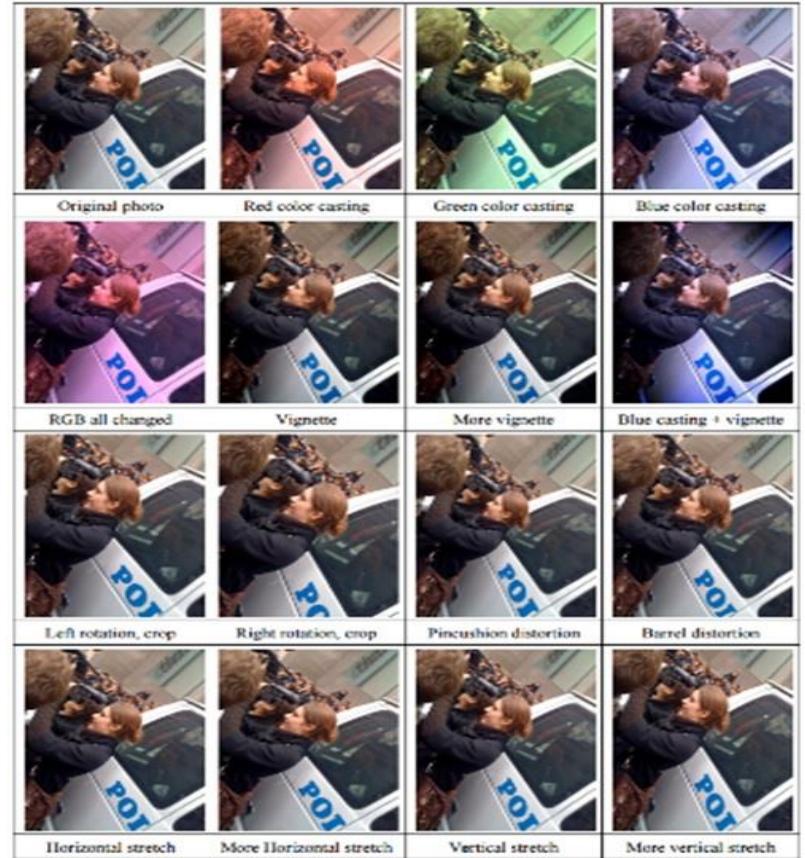
Conclusion : Le fine-tuning améliore les performance. À cette étape, il est préférable de ne faire qu'un apprentissage supervisé.



H. Larochelle et al., "Exploring Strategies for Training Deep Neural Networks," *JMLR* 2009.

Création de données synthétiques

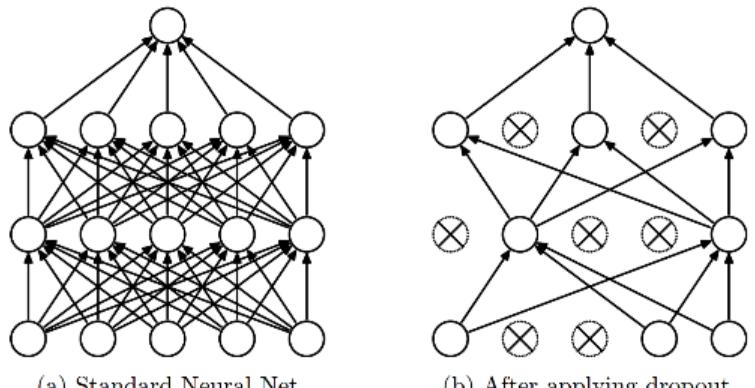
- ▶ On crée de nouveau exemple à partir de la base d'entraînement.
- ▶ Par exemple pour une image on peut:
 - Appliquer une symétrie
 - Recadrer
 - Changer la balance de couleur
 - Appliquer des distorsions géométrique.



Deep Image [[Wu et al. 2015](#)]

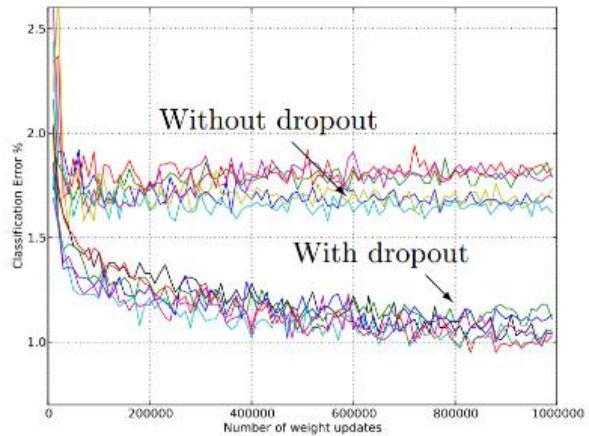
Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)

Dropout



(a) Standard Neural Net

(b) After applying dropout.



| Model | Top-1 (val) | Top-5 (val) | Top-5 (test) |
|--|----------------|----------------|-----------------|
| SVM on Fisher Vectors of Dense SIFT and Color Statistics | - | - | 27.3 |
| Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT | - | - | 26.2 |
| Conv Net + dropout (Krizhevsky et al., 2012) | 40.7 | 18.2 | - |
| Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012) | 38.1 | 16.4 | 16.4 |

Table 6: Results on the ILSVRC-2012 validation/test set.

Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)

Type de couches cachées

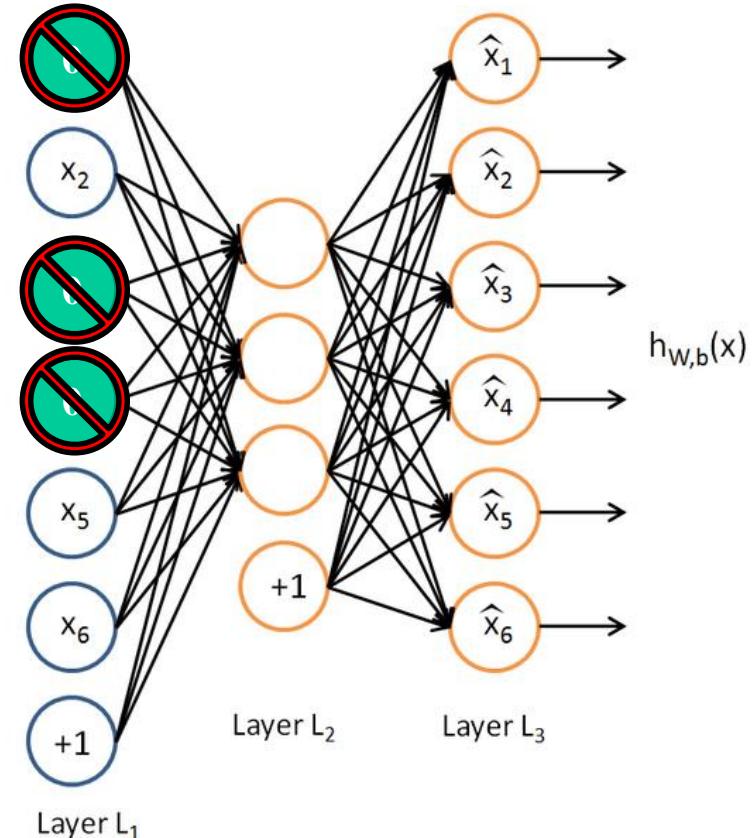
CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement
- 5) Type de couches cachées**
- 6) Les réseaux à convolution

Denoising Auto-Encoder

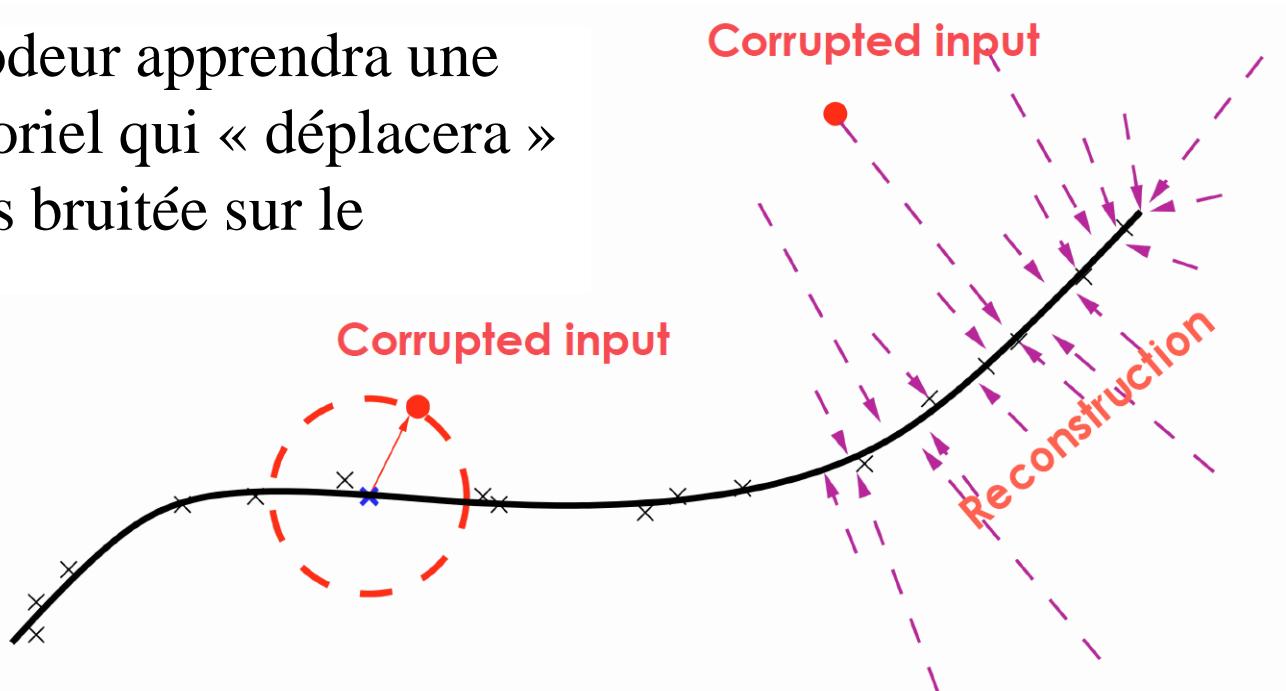
- ▶ L'auto-encodeur tel que vu jusqu'ici ne donne pas des résultats optimaux.
- ▶ En pratique on utilisera plutôt un auto-encodeur débruiteur (denoising).
- ▶ La structure est la même que pour l'auto-encodeur.
- ▶ À l'entraînement, la réponse désirée est toujours la forme d'entrée ($d = x$)
- ▶ Par contre, avant de présenter x au réseau certaines champs du vecteur seront forcés à **0**.



Source : http://deeplearning.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity

Denoising Auto-Encoder

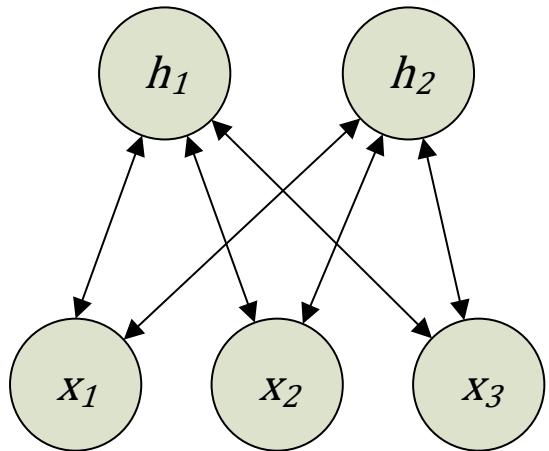
- ▶ La ligne noire est le manifold contenant les données du problème.
- ▶ L'auto-encodeur apprendra une champ vectoriel qui « déplacera » les instances bruitée sur le manifold.



P. Vincent, et al., “Extracting and Composing Robust Features with Denoising Autoencoders,” *ICML*, 2008

Restricted Boltzmann Machines (RBM)

- ▶ Le type de couche le plus communément utilisé dans les deep architectures.
- ▶ Avantage: on peut aussi créer des exemples à partir du modèle.
- ▶ Modèle basé sur une fonction d'énergie.
- ▶ Les connexions sont bi-directionnelles (*Boltzmann Machines*)
- ▶ Le modèle possède deux couches, une observable et l'autre cachée.
- ▶ Les neurones d'une même couche ne sont pas reliés entre eux (*restricted*)



Neurones stochastiques binaires

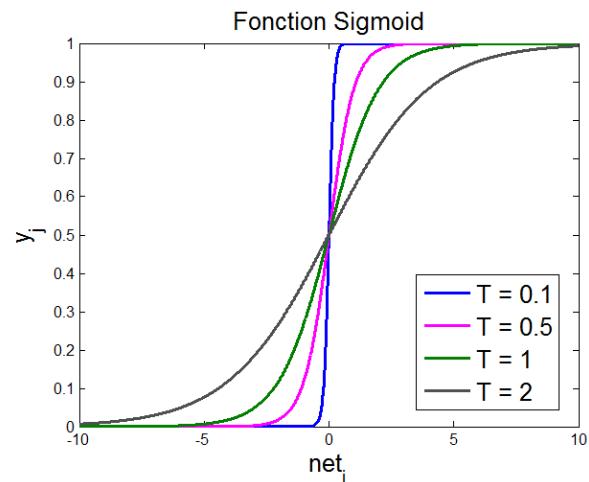
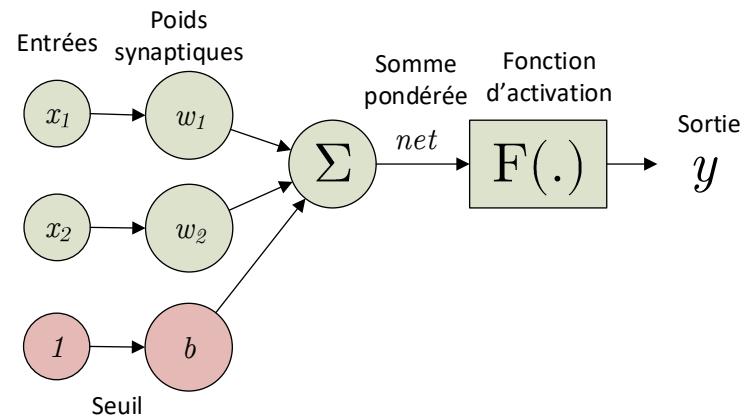
Les neurones sont activés ($y = 1$) selon une probabilité :

$$P(y = 1) = P(y) = \frac{1}{1 + e^{-net/T}}$$

Où

$$net = b + \sum_i x_i w_i$$

Et T est un parameter contrôlant la pente généralement fixé à 1.



Neurones stochastiques binaires

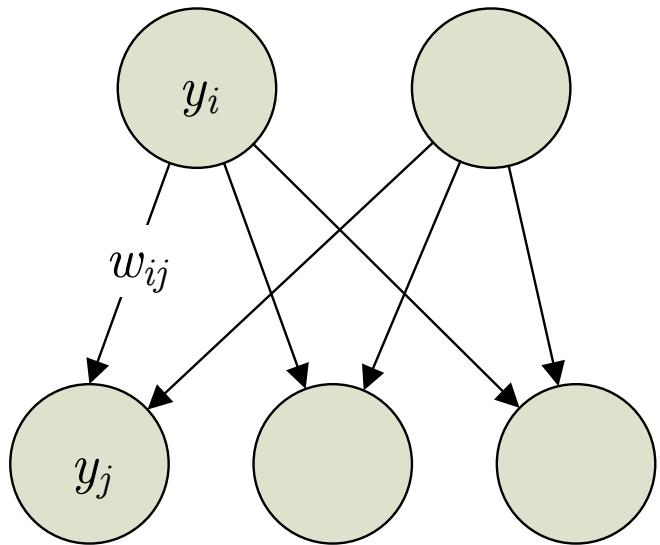
Règle d'apprentissage

Les neurones sont activés ($y = 1$)
selon une probabilité :

$$P(y_j = 1) = P(y_j) = \frac{1}{1 + e^{-\text{net}_j}}$$

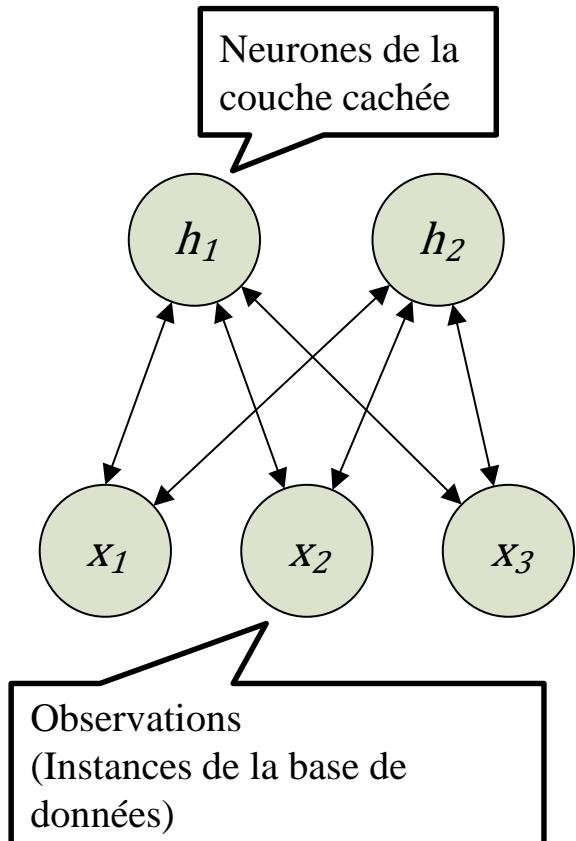
Quand les neurones sont activés en
même temps, on renforce la
connexion (règle de Hebb) :

$$\Delta w_{ij} = \eta y_i (y_j - P(y_j))$$



Restricted Boltzmann Machines

- ▶ Modèle basé sur une fonction d'énergie.
- ▶ Le connexions sont bi-directionnelles (*Boltzmann Machines*)
- ▶ Le modèle possède deux couches, une observable et l'autre cachée.
- ▶ Les neurones d'une même couche ne sont pas reliés entre eux (*restricted*)



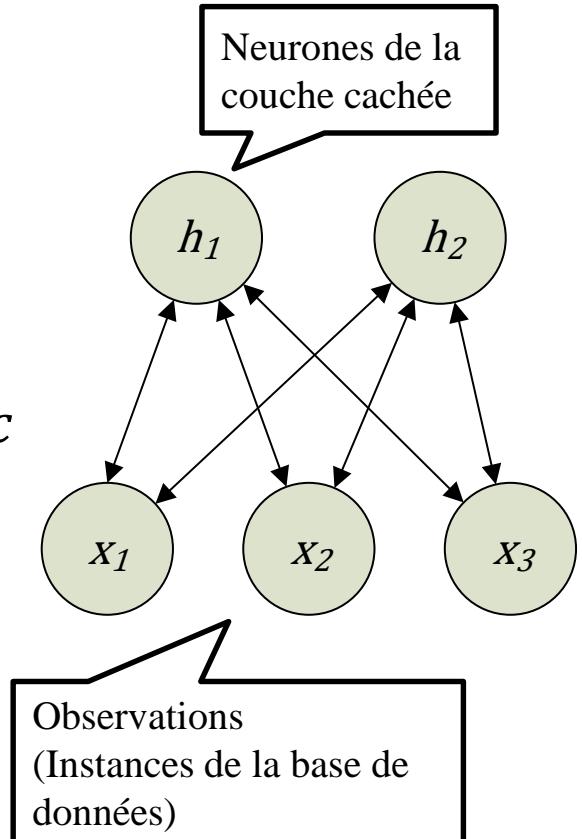
Restricted Boltzmann Machines

La fonction d'énergie pour une configuration est donnée par :

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{b}^T \mathbf{x} - \mathbf{c}^T \mathbf{h} - \mathbf{h}^T \mathbf{W} \mathbf{x}$$

Où W est matrice de poids synaptiques et b et c sont les vecteurs de *bias*.

$$-\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial w_{ij}} = h_i x_j$$



Restricted Boltzmann Machines

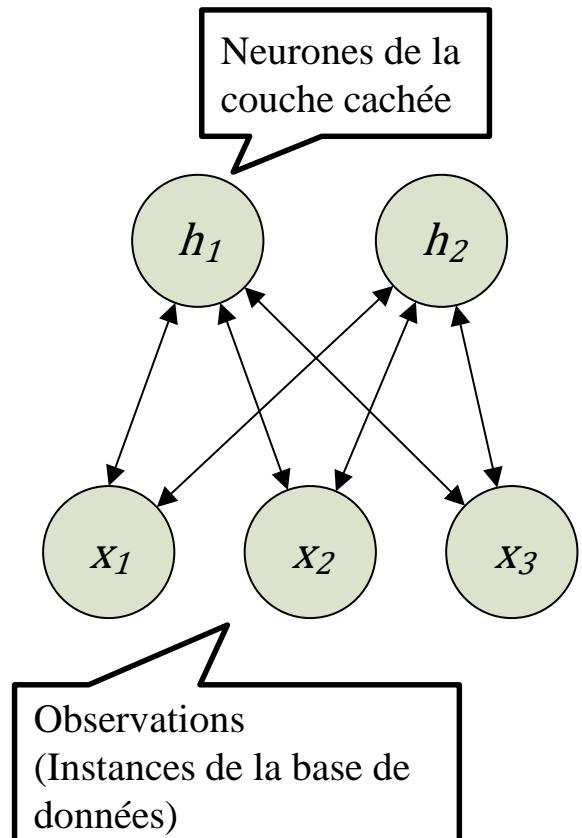
- D'énergie à probabilités:

$$P(\mathbf{x}, \mathbf{h}) \propto e^{-E(\mathbf{x}, \mathbf{h})}$$

Parce que :

$$P(\mathbf{x}, \mathbf{h}) = \frac{e^{-E(\mathbf{x}, \mathbf{h})}}{\sum_{\mathbf{x}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}}$$

Fonction de partition
difficile à calculer



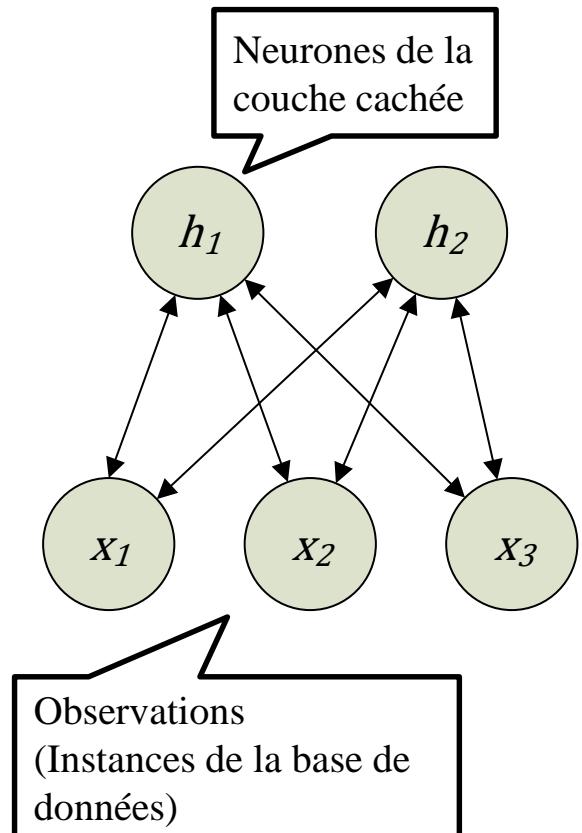
Restricted Boltzmann Machines

- ▶ Les probabilités pour une configuration \mathbf{x}, \mathbf{h} :

$$P(\mathbf{x}, \mathbf{h}) = \frac{e^{-E(\mathbf{x}, \mathbf{h})}}{\sum_{\mathbf{x}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}}$$

- ▶ Si on intègre par rapport à \mathbf{h} (marginaliser):

$$P(\mathbf{x}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}}{\sum_{\mathbf{x}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}}$$



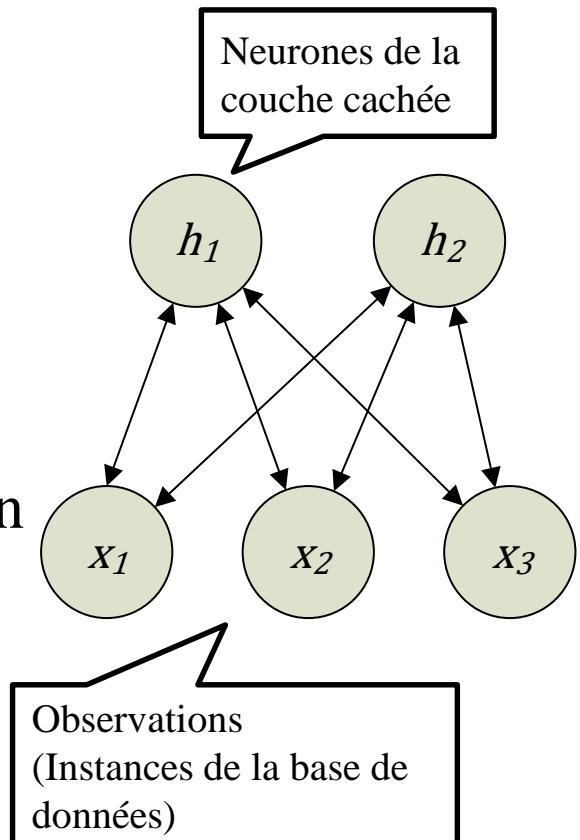
Restricted Boltzmann Machines

Pour une instance donnée, la dérivée de la probabilité logarithmique par rapport au poids synaptique (w_{ij}) est donnée par:

$$\frac{\partial \log p(x)}{\partial w_{ij}} = (h_i x_j)_{data} - (h_i x_j)_{model}$$

Comme on veut augmenter les probabilité que le model produise un donnée valable, on fait un ascension de gradient et donc :

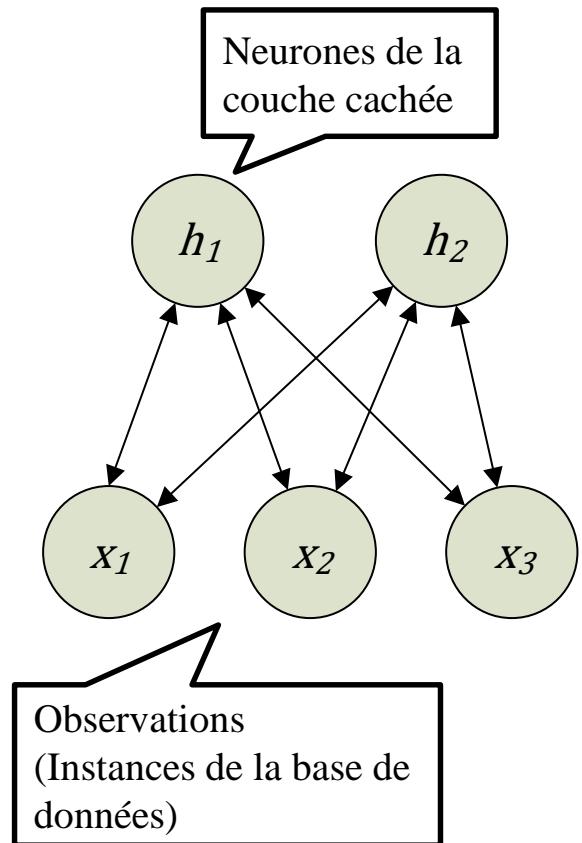
$$\Delta w_{ij} = \eta \left\{ (h_i x_j)_{data} - (h_i x_j)_{model} \right\}$$



Restricted Boltzmann Machines

Observations
(Instances de la base de données)

$$\Delta w_{ij} = \eta \left\{ (h_i x_j)_{data} - (h_i x_j)_{model} \right\}$$



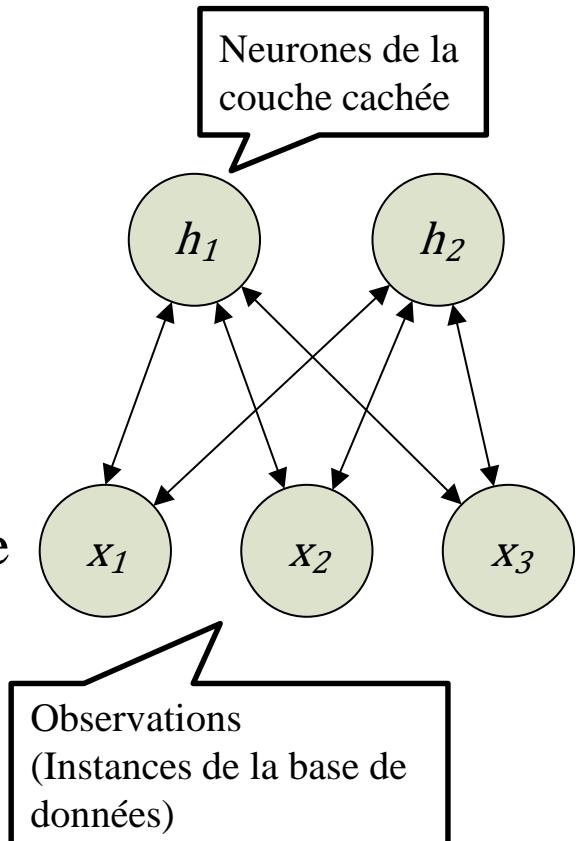
Restricted Boltzmann Machines

- ▶ À cause de la structure du modèle, si on fixe x , les neurones cachés (h) sont indépendant et vice-versa :

$$P(\mathbf{h}|\mathbf{x}) = \prod_i P(h_i|\mathbf{x})$$

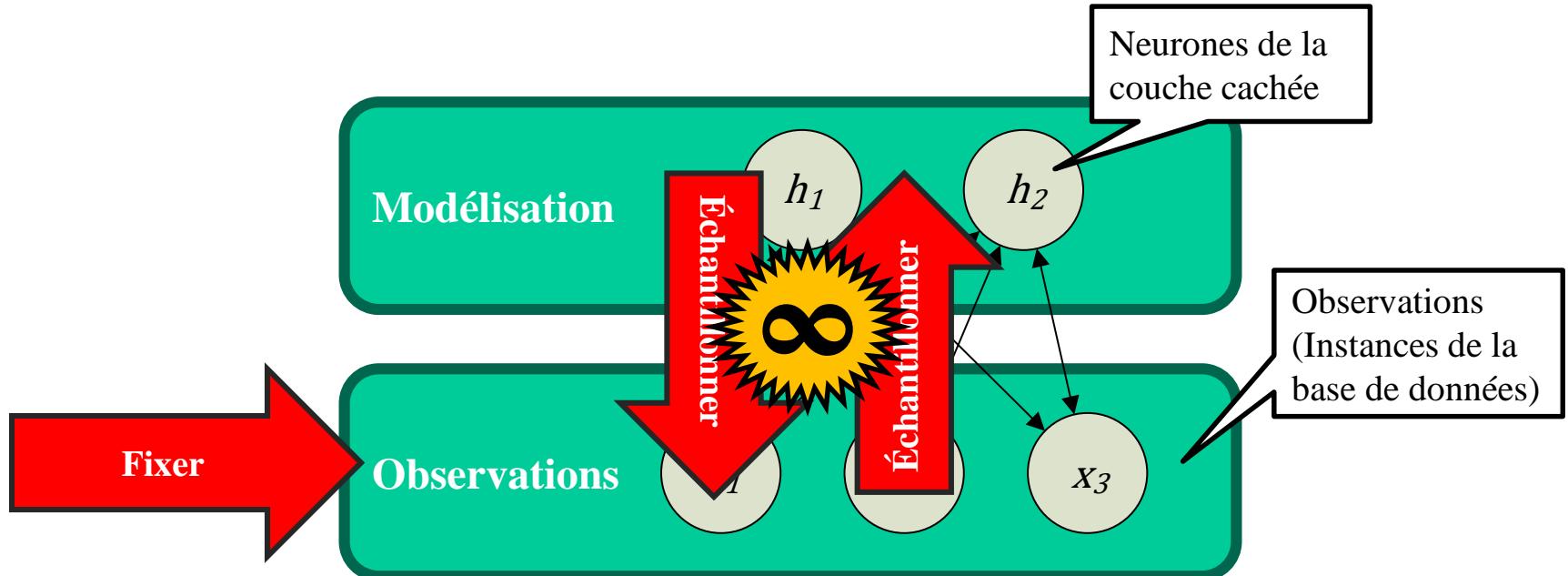
$$P(\mathbf{x}|\mathbf{h}) = \prod_i P(x_i|\mathbf{h})$$

- ▶ En échantillonnant et en fixant \mathbf{h} et \mathbf{x} à tour de rôle on peut obtenir un échantillonnage de Gibbs.
- ▶ Il est donc possible de « créer » des observations à partir de ce que le modèle a appris.

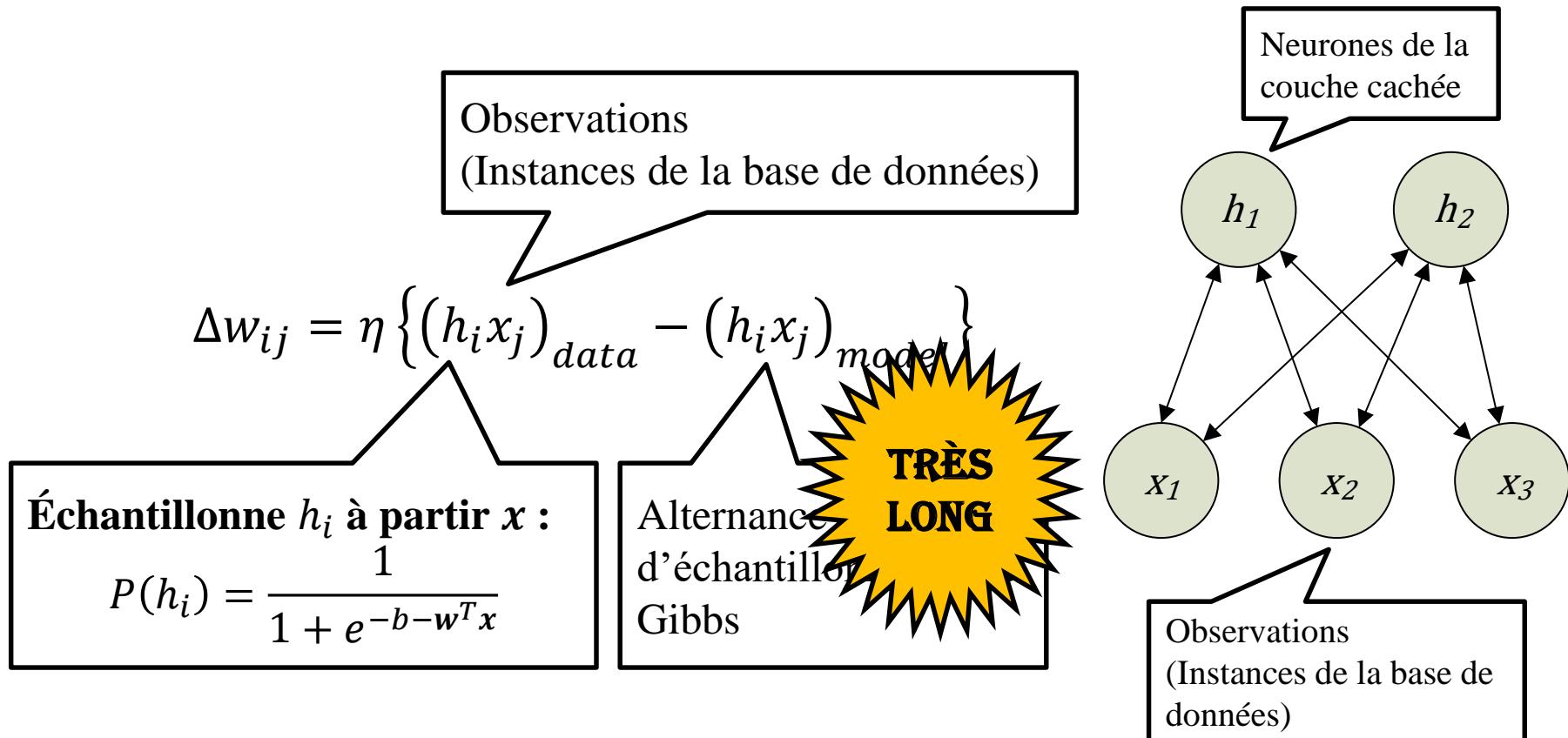


Restricted Boltzmann Machines

- ▶ Comment voir ce que le modèle a appris?
- ▶ Solutions: échantillonner alternativement d'une couche à l'autre. (Échantillonnage de Gibbs)

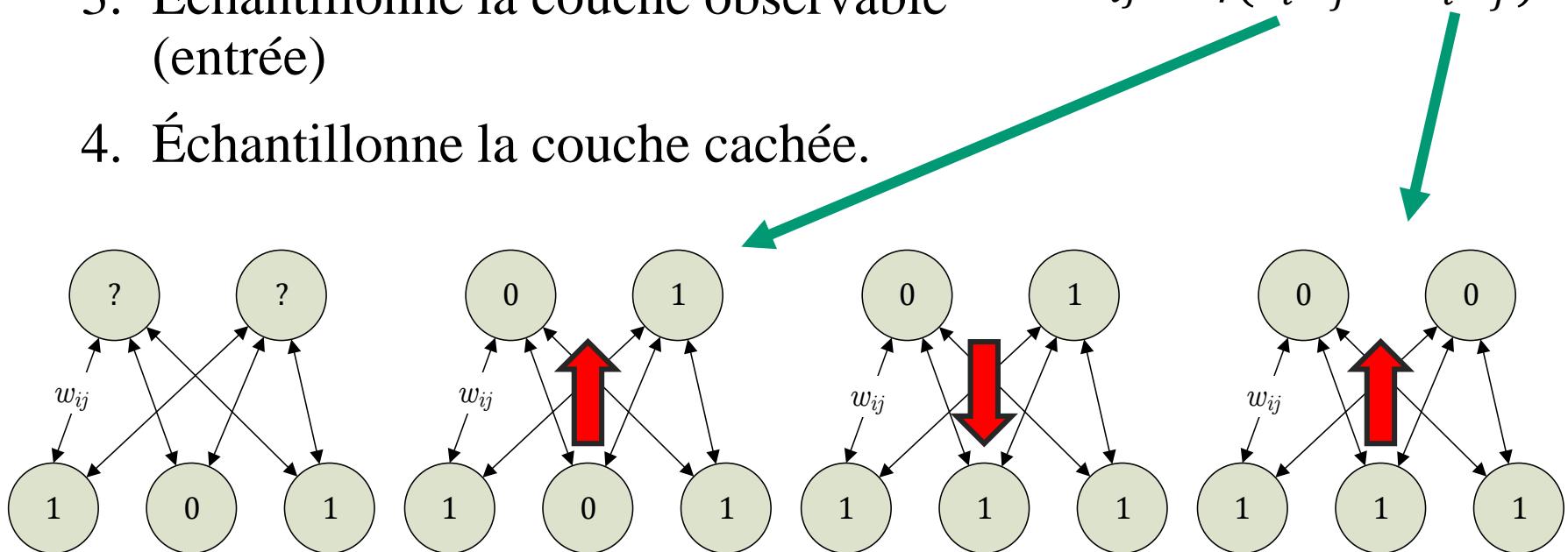


Restricted Boltzmann Machines



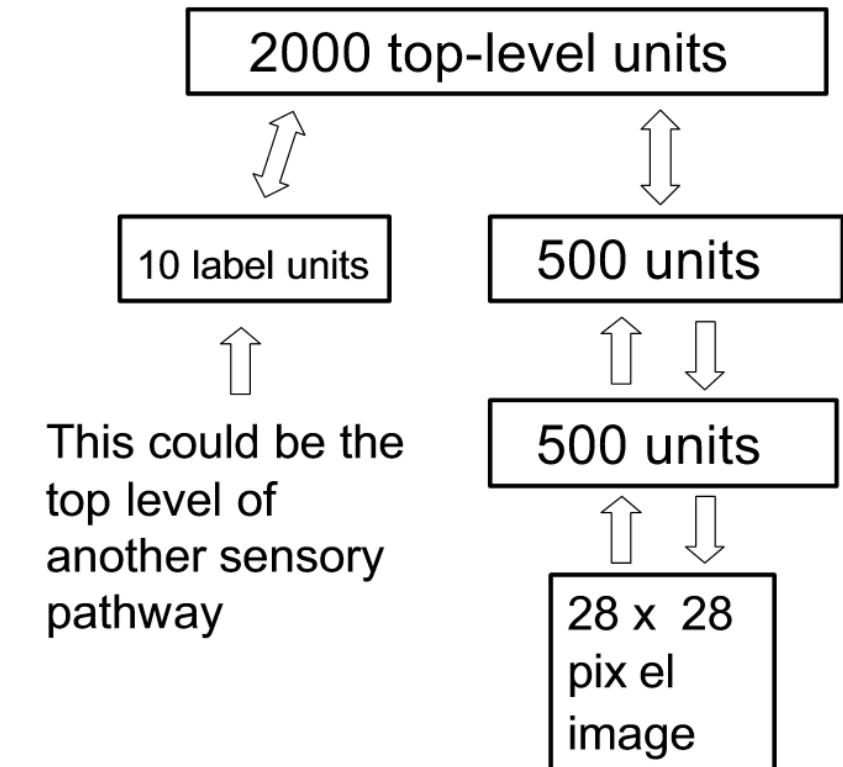
1. Présente un forme à l'entrée.
2. Échantillonne la couche cachée.
3. Échantillonne la couche observable (entrée)
4. Échantillonne la couche cachée.

$$\Delta w_{ij} = \eta(x_i^0 h_j^0 - x_i^1 h_j^1)$$



Modèle de Hinton pour la reconnaissance de chiffre manuscrits

- ▶ Ce modèle permet de faire de la reconnaissance.
- ▶ Il permet aussi de produire des chiffres qu'il n'a jamais vu auparavant.
- ▶ Visualisation disponible :
<http://www.cs.toronto.edu/~intor/adi/index.htm>



Hinton et al., A fast learning algorithm for deep belief nets, Neural Computation (2006)

Les réseaux à convolution

CONTENU DU COURS

A.4 Les algorithmes d'apprentissage profond.

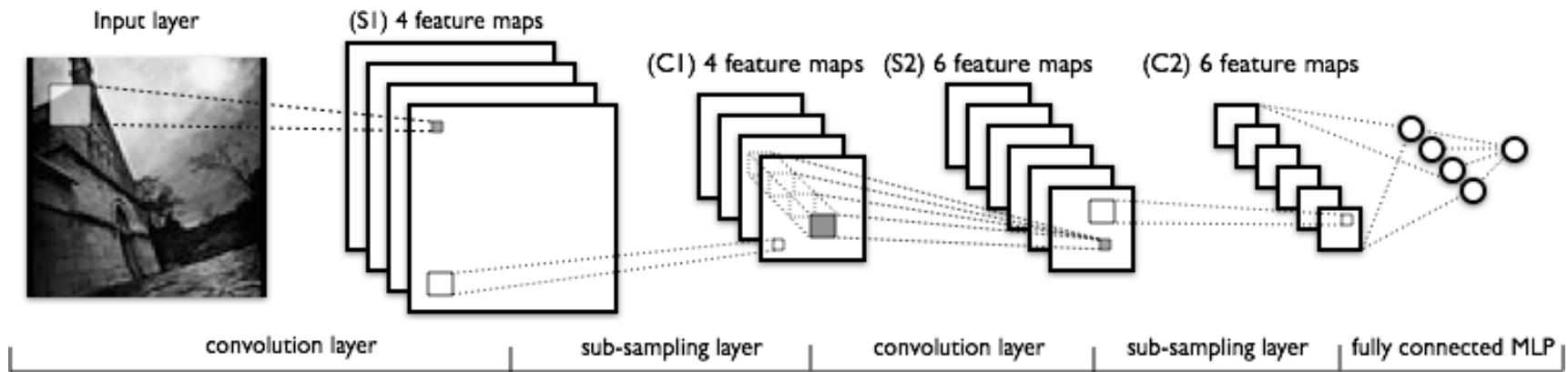
- 1) Les auto-encodeurs
- 2) Self-Taught Learning
- 3) Les réseaux *deep-learning*
- 4) Stratégie d'entraînement
- 5) Type de couches cachées
- 6) Les réseaux à convolution**

Les réseaux à convolution (CNN)

- ▶ Semblables aux réseaux vus précédemment.
- ▶ Développés pour la classification des images.
- ▶ Architecture connectée localement.
- ▶ Utilise des convolutions.
- ▶ Un étage de *pooling* est incorporé dans l'architecture.
- ▶ Une couche typique :

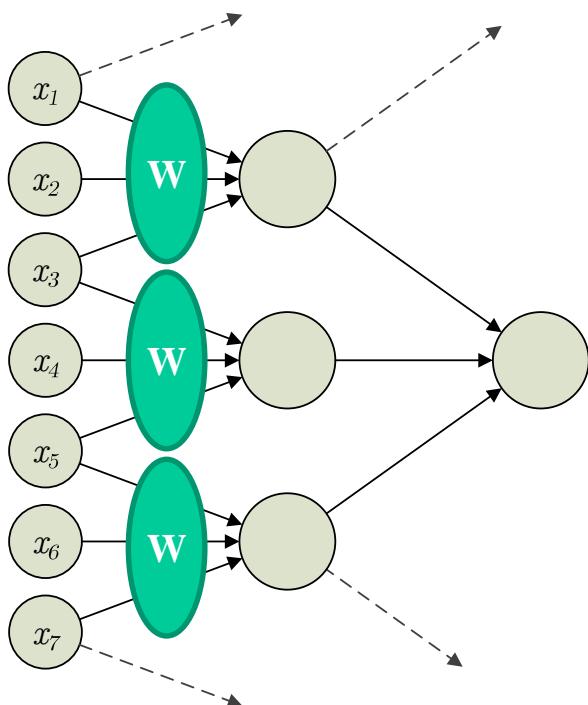


Exemple d'architecture



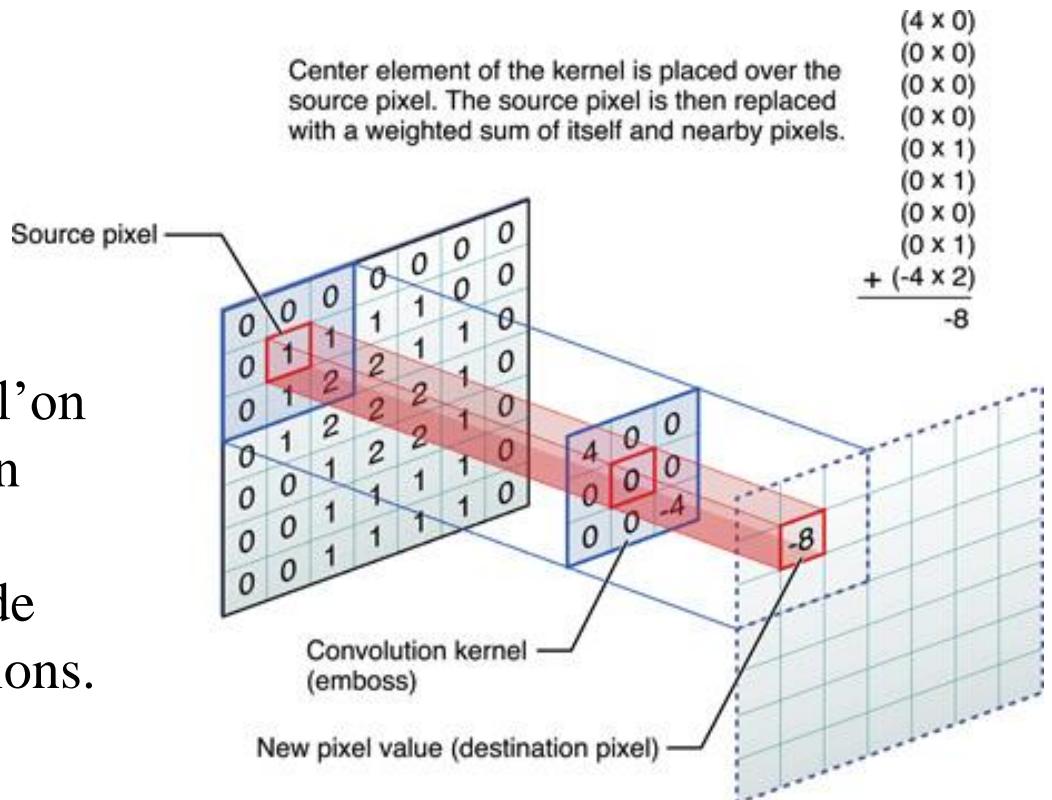
Réseaux à connexions locales

- ▶ Pour des applications où les données d'entrée sont de grande dimensionnalité (images), les réseaux deeps comportent beaucoup (trop...) de poids synaptiques.
- ▶ On peut réduire le nombre de poids à apprendre en utilisant une réseau connecté localement.
- ▶ Encore mieux on peut utiliser les mêmes poids pour connexions différentes.



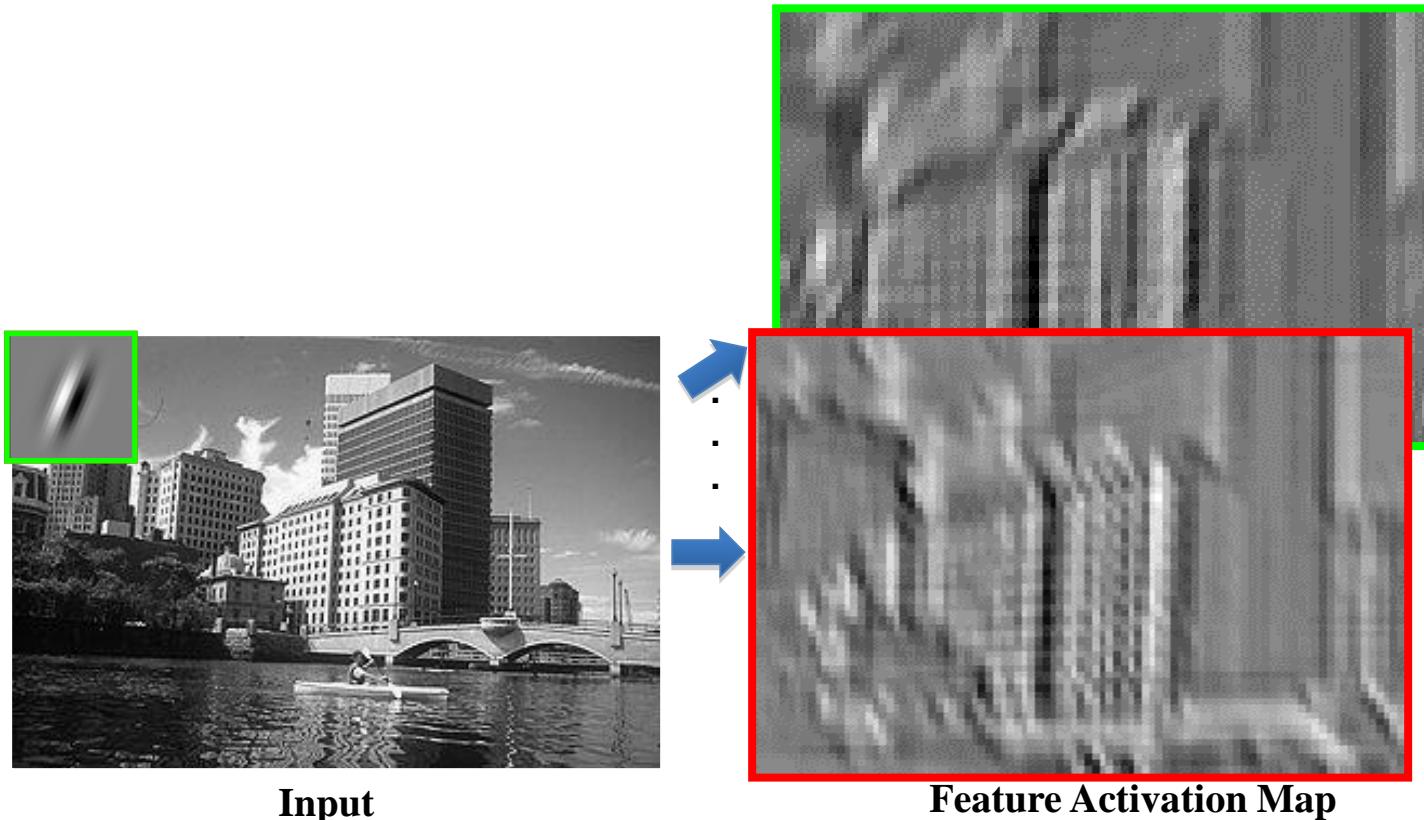
Qu'est-ce qu'une convolution?

- Une somme pondérée que l'on applique successivement en déplaçant une fonction.
- Correspond à une mesure de similarité entre deux fonctions.



<https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>

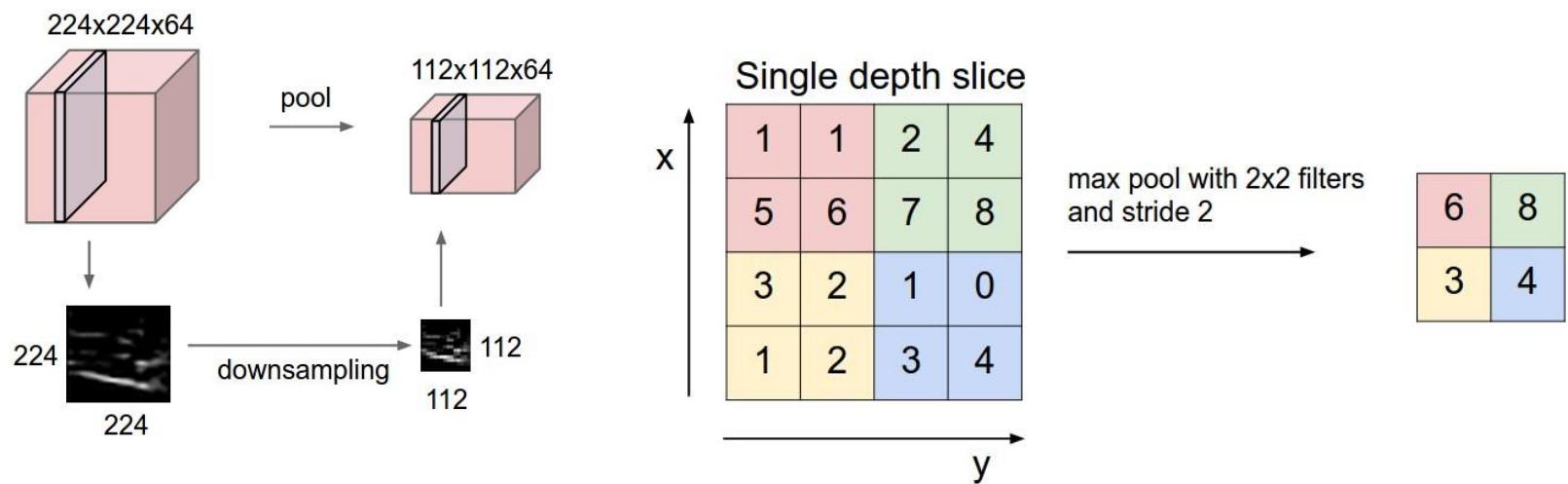
Qu'est-ce qu'une convolution?



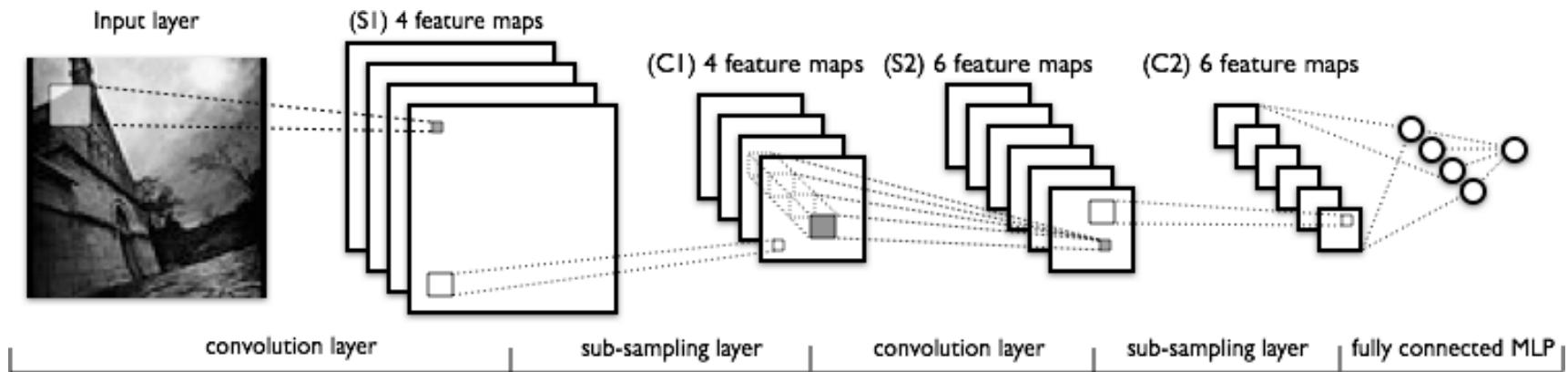
Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)

Pooling

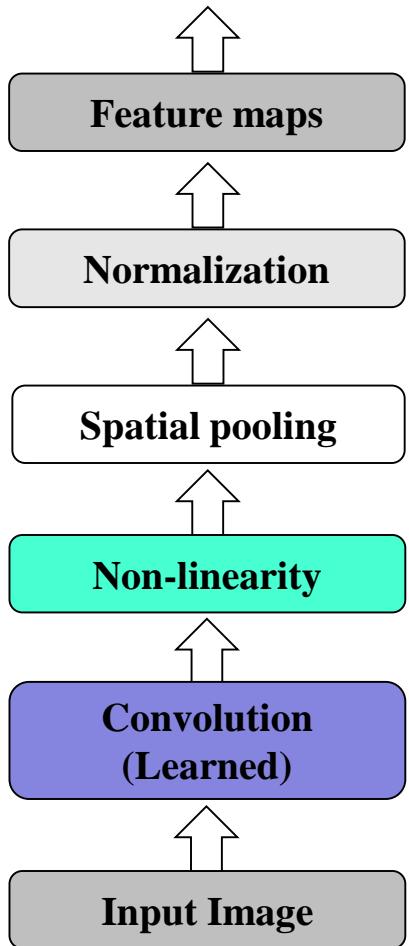
- ▶ Sous-échantillonnage souvent en utilisant la fonction max.
- ▶ Sert à limiter la quantité d'information à traiter par le réseau.
- ▶ Offre une certaine invariance à la position et l'échelle.



Exemple d'architecture

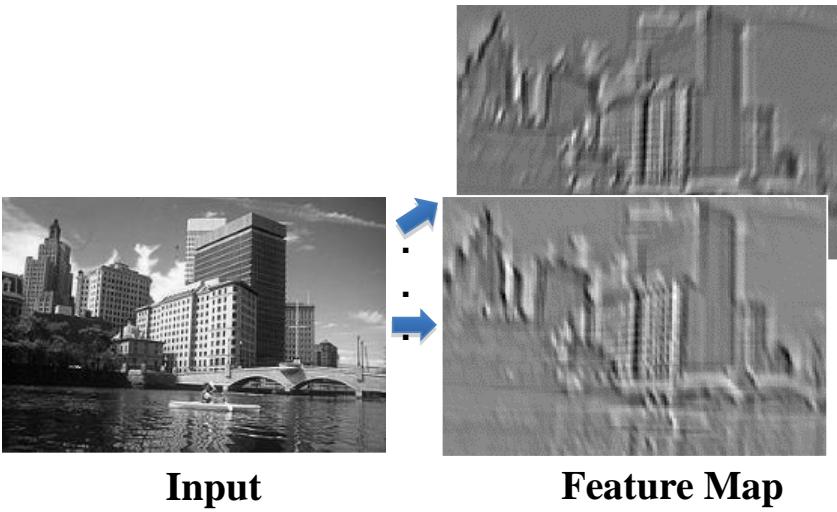
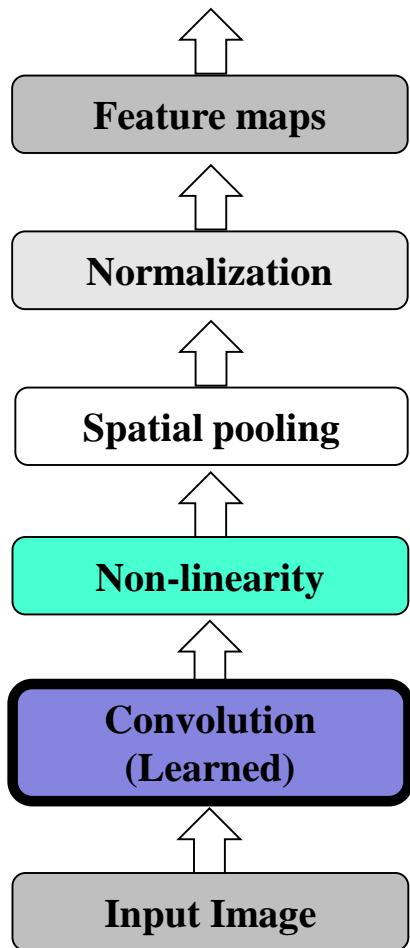


Visualisation d'un CNN



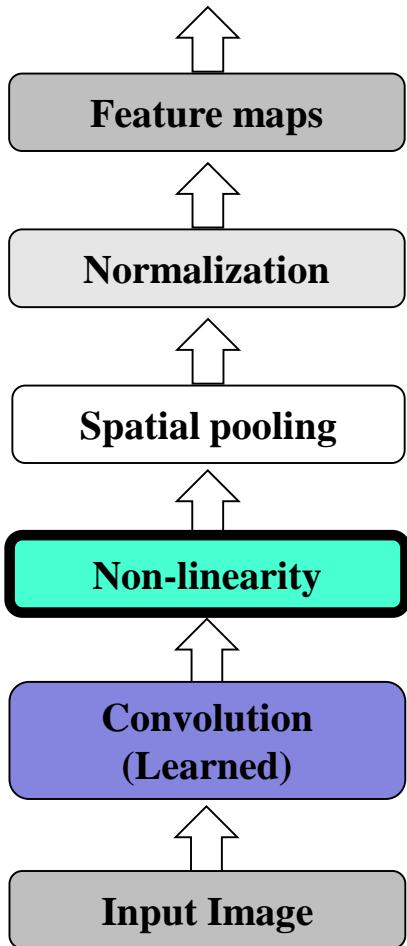
Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)
slide credit: S. Lazebnik

Visualisation d'un CNN

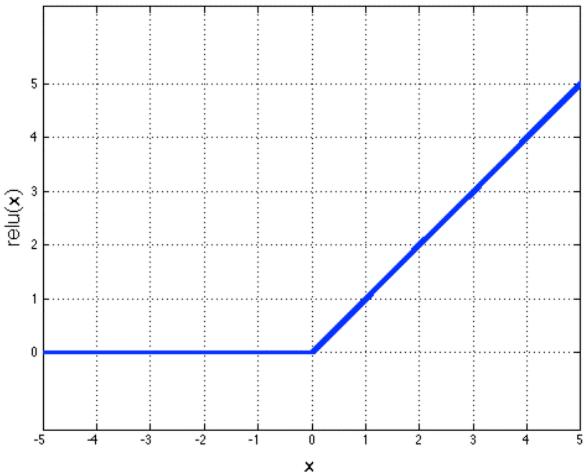


Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)
slide credit: S. Lazebnik

Visualisation d'un CNN

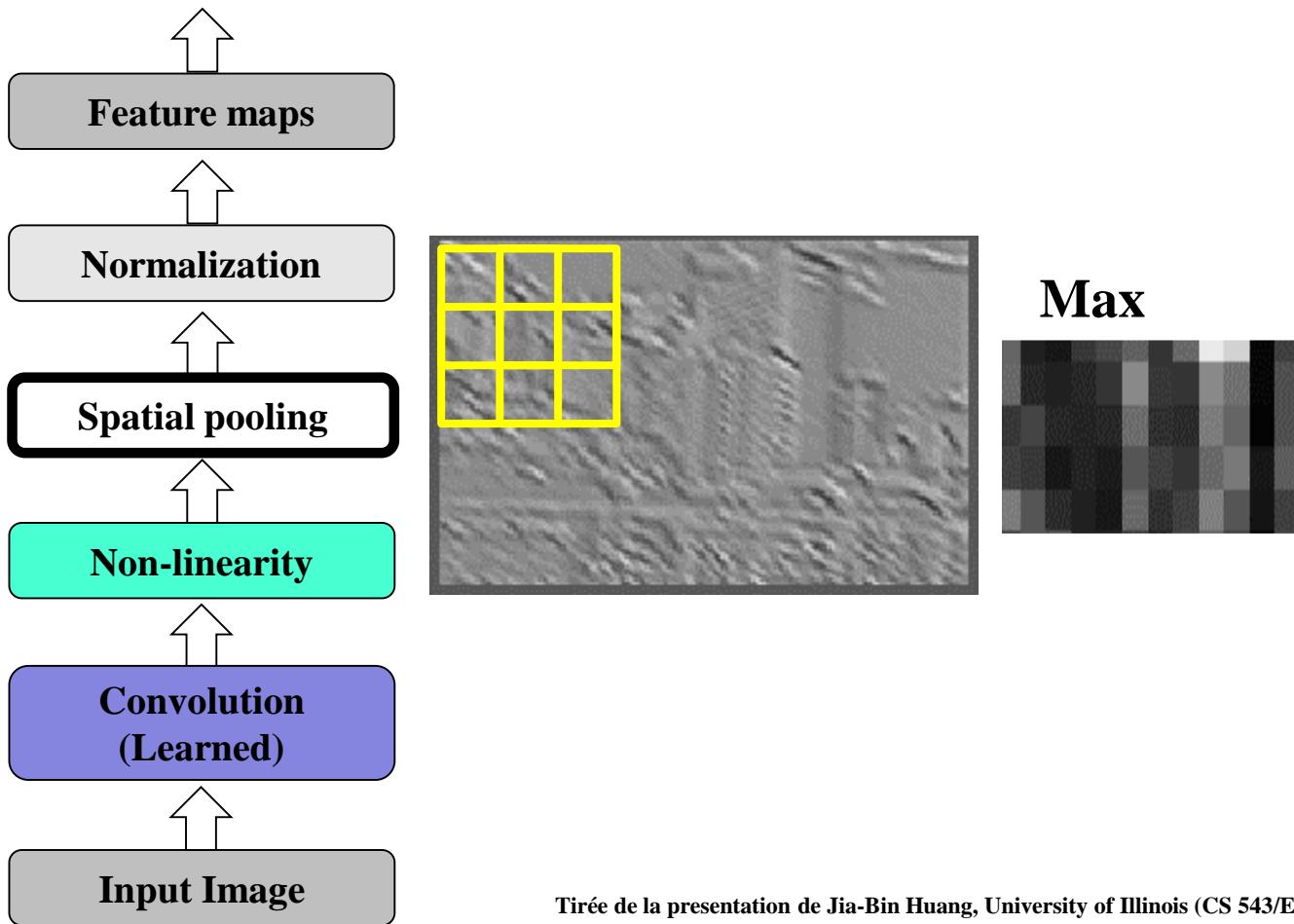


Rectified Linear Unit (ReLU)

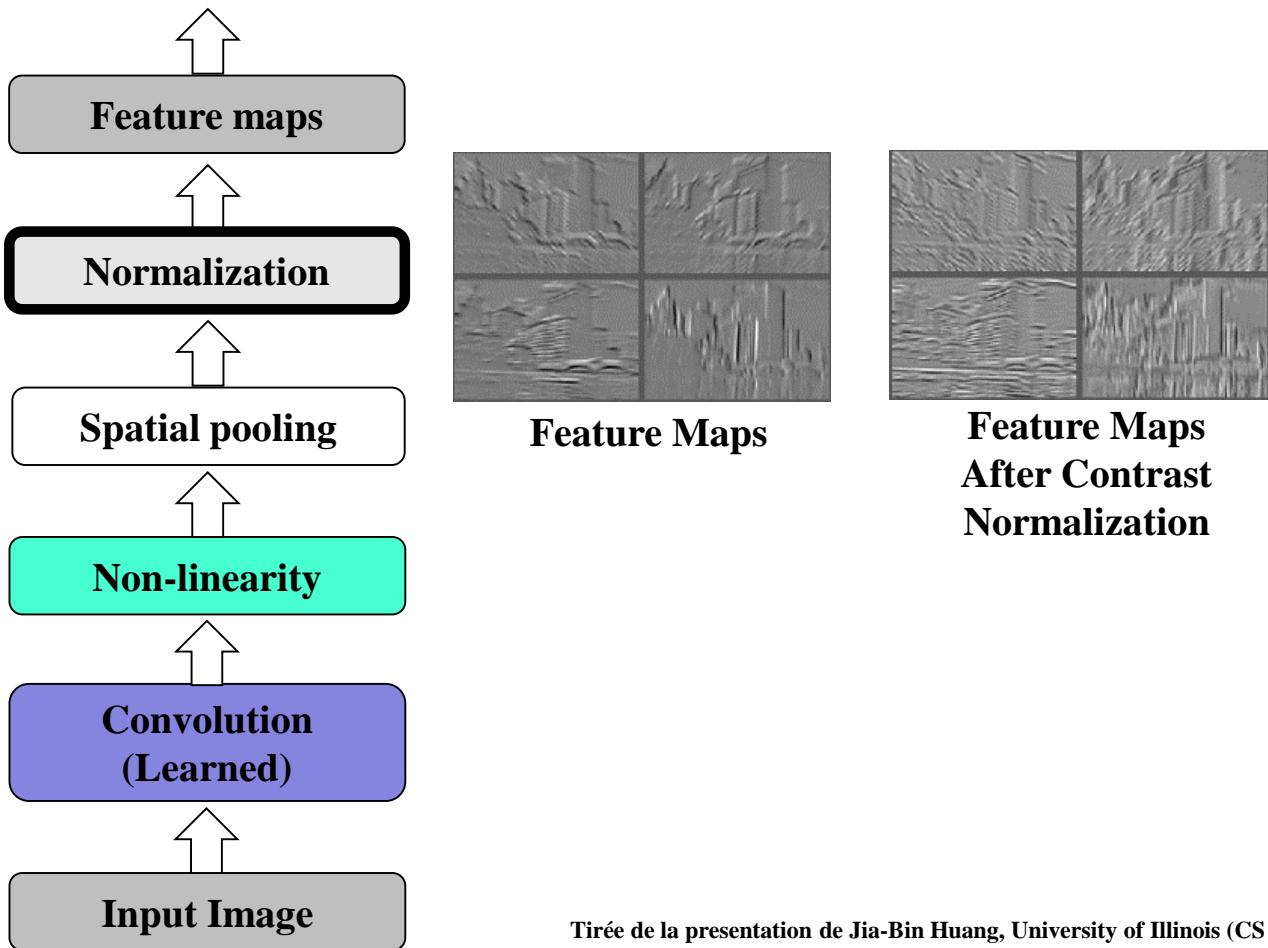


Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)
slide credit: S. Lazebnik

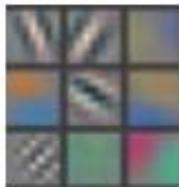
Visualisation d'un CNN



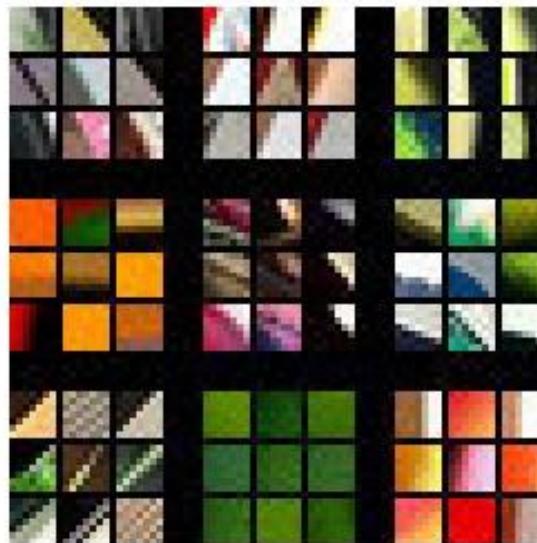
Visualisation d'un CNN



Couche 1

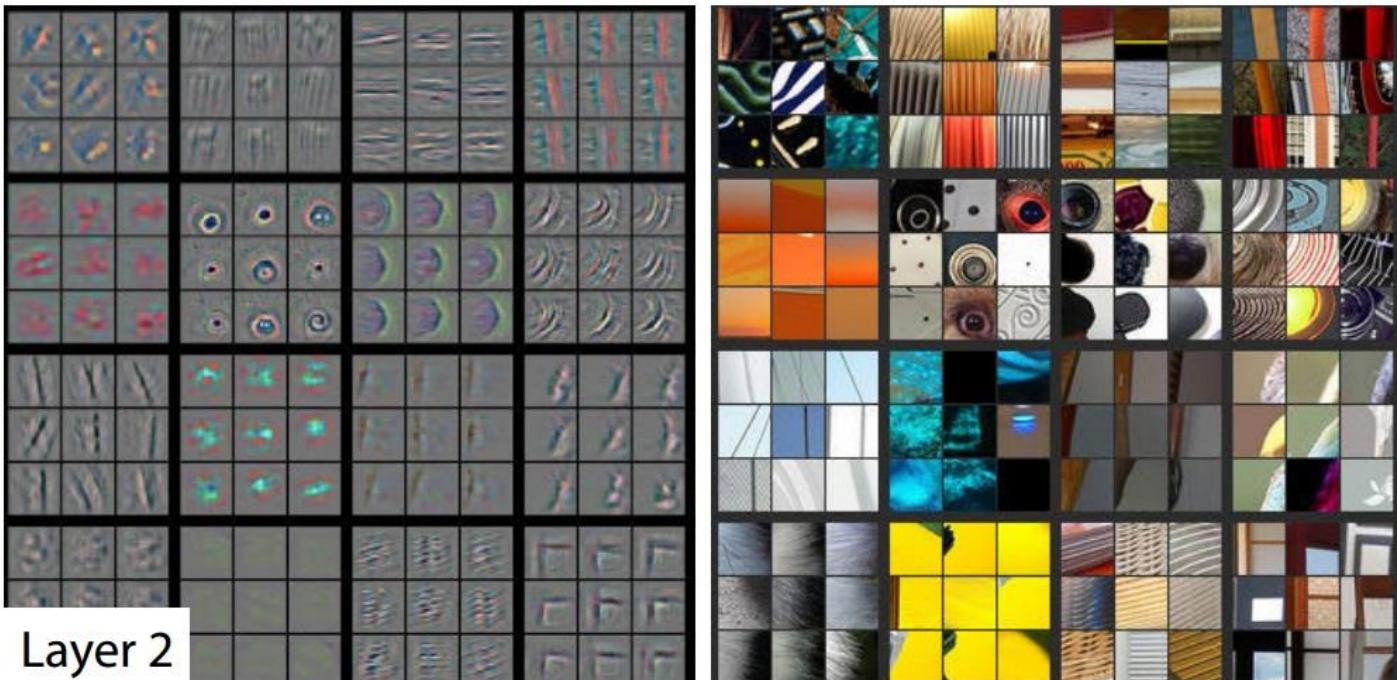


Layer 1

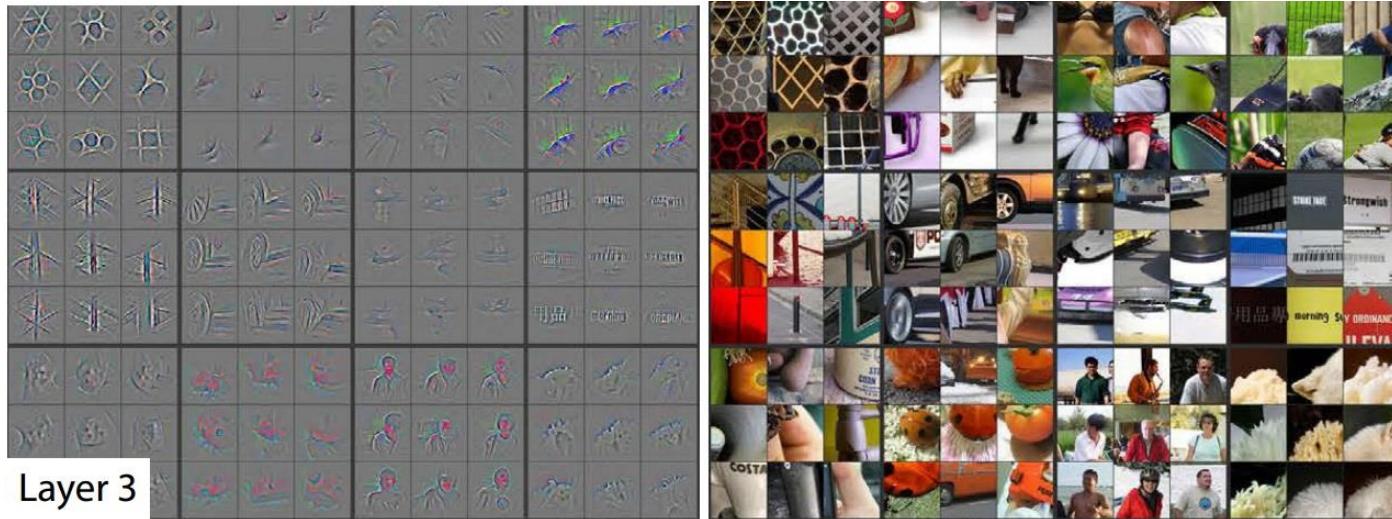


Visualizing and Understanding Convolutional Networks
[\[Zeiler and Fergus, ECCV 2014\]](#)

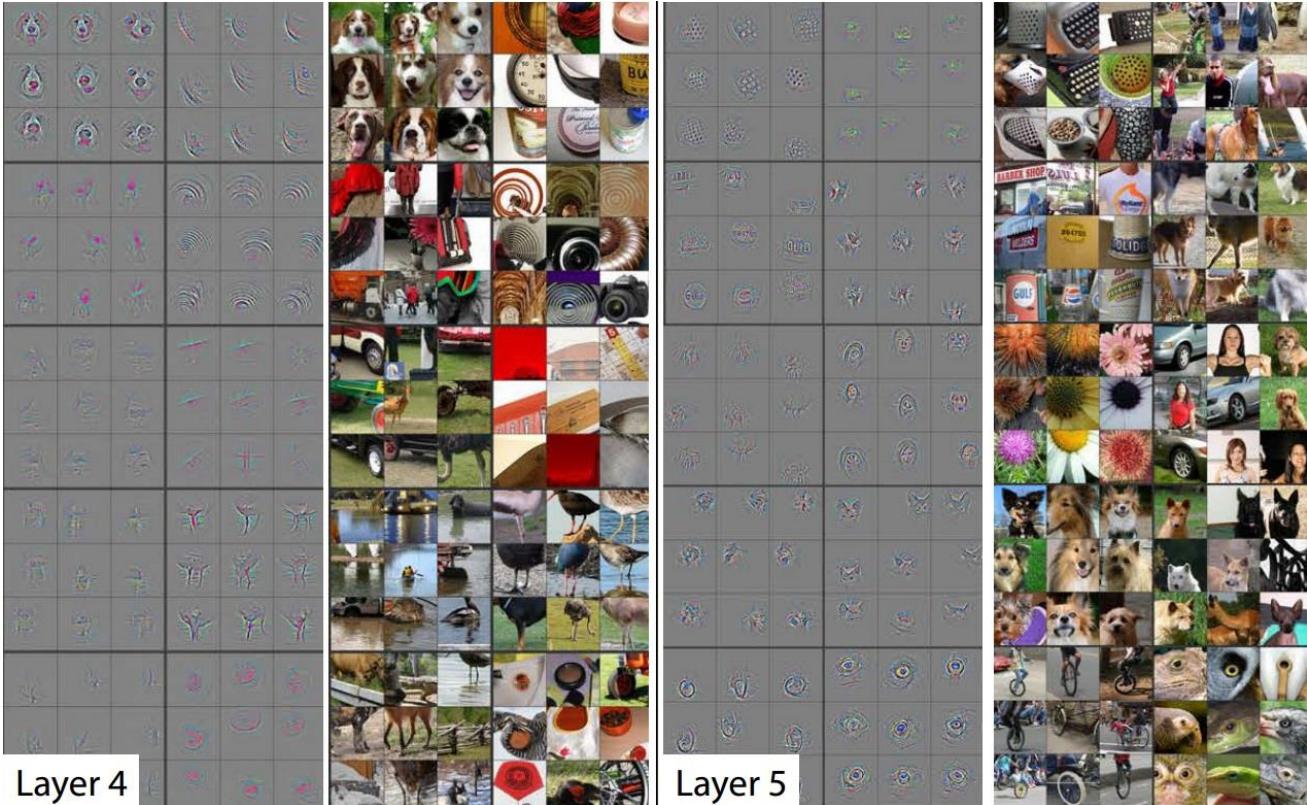
Couche 2



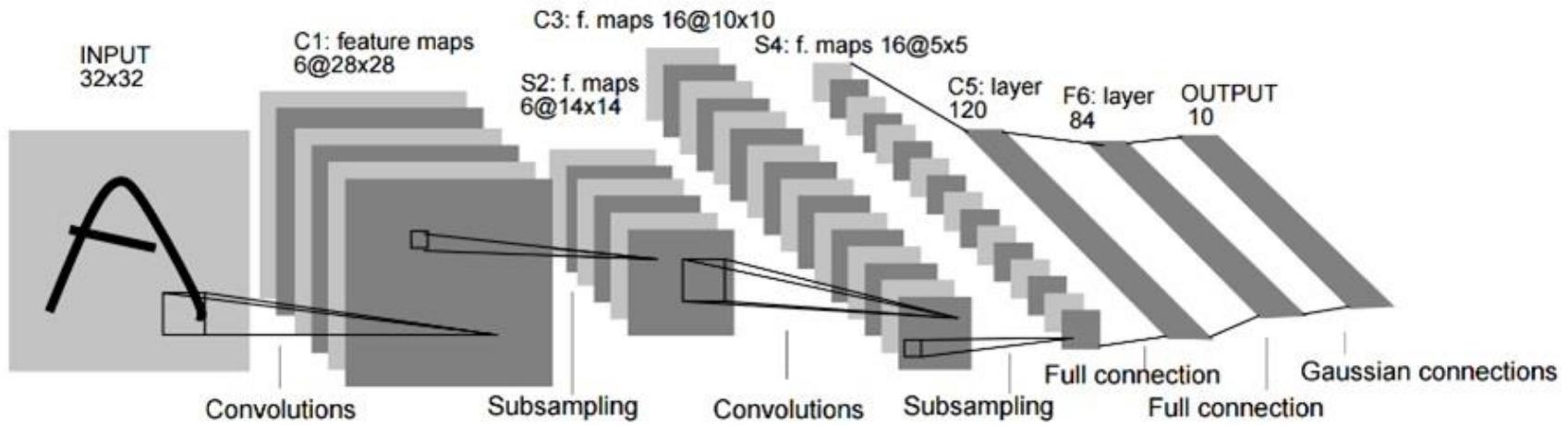
Couche 3



Couche 4 et 5

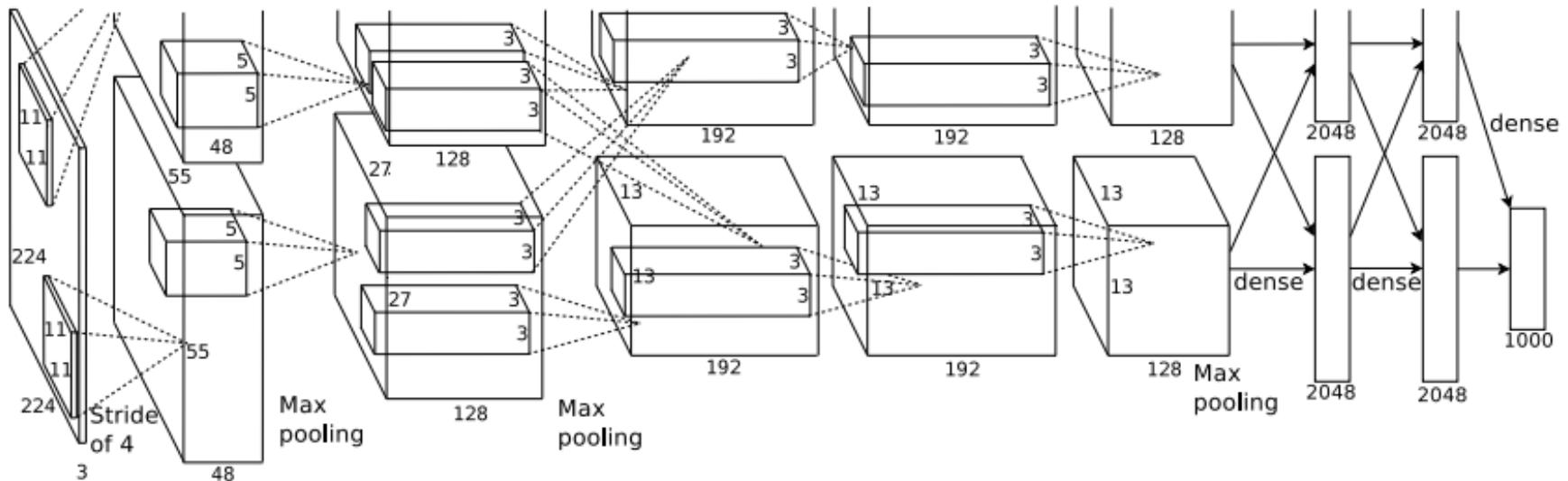


LeNet [LeCun et al. 1998]

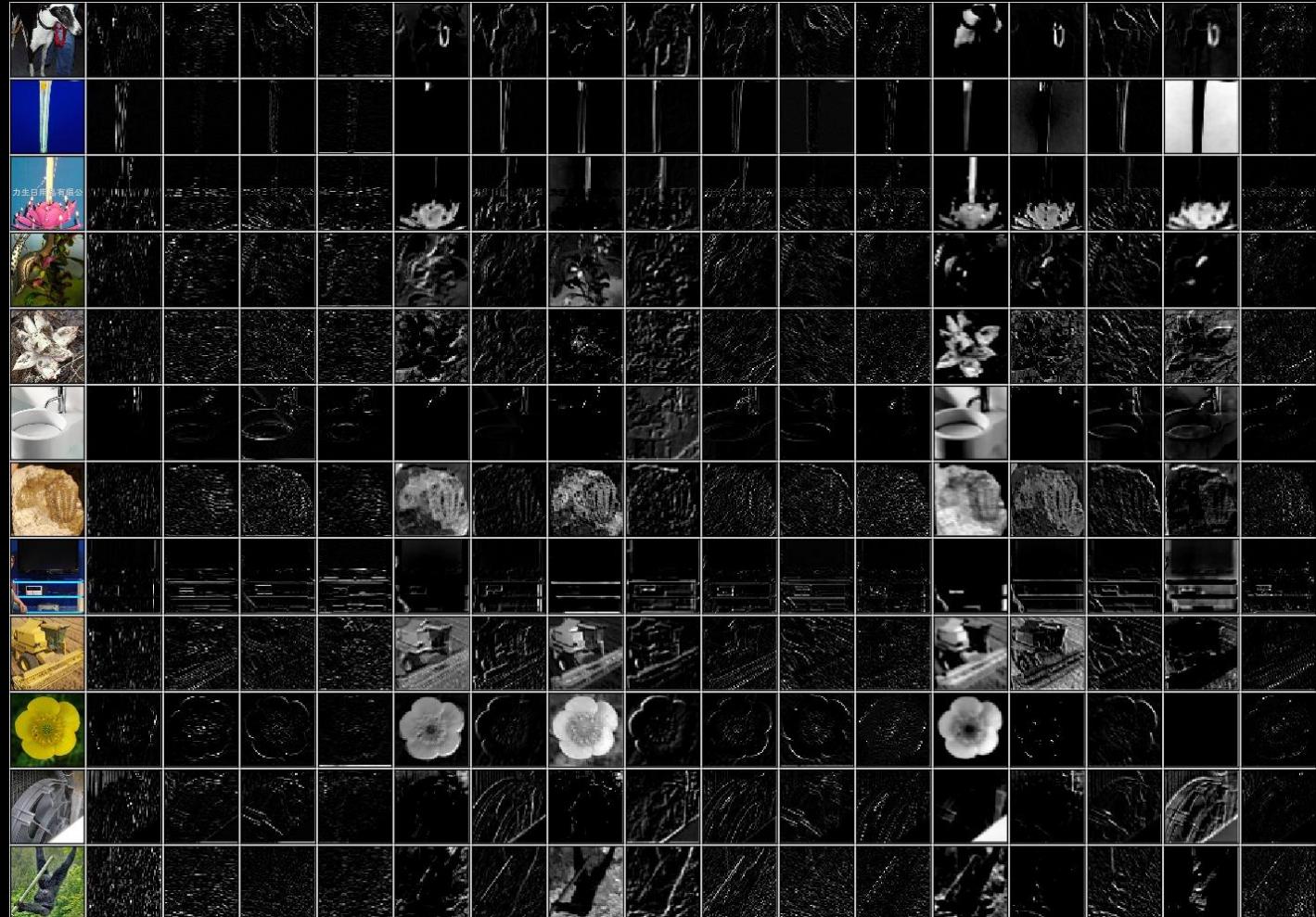


Alex net

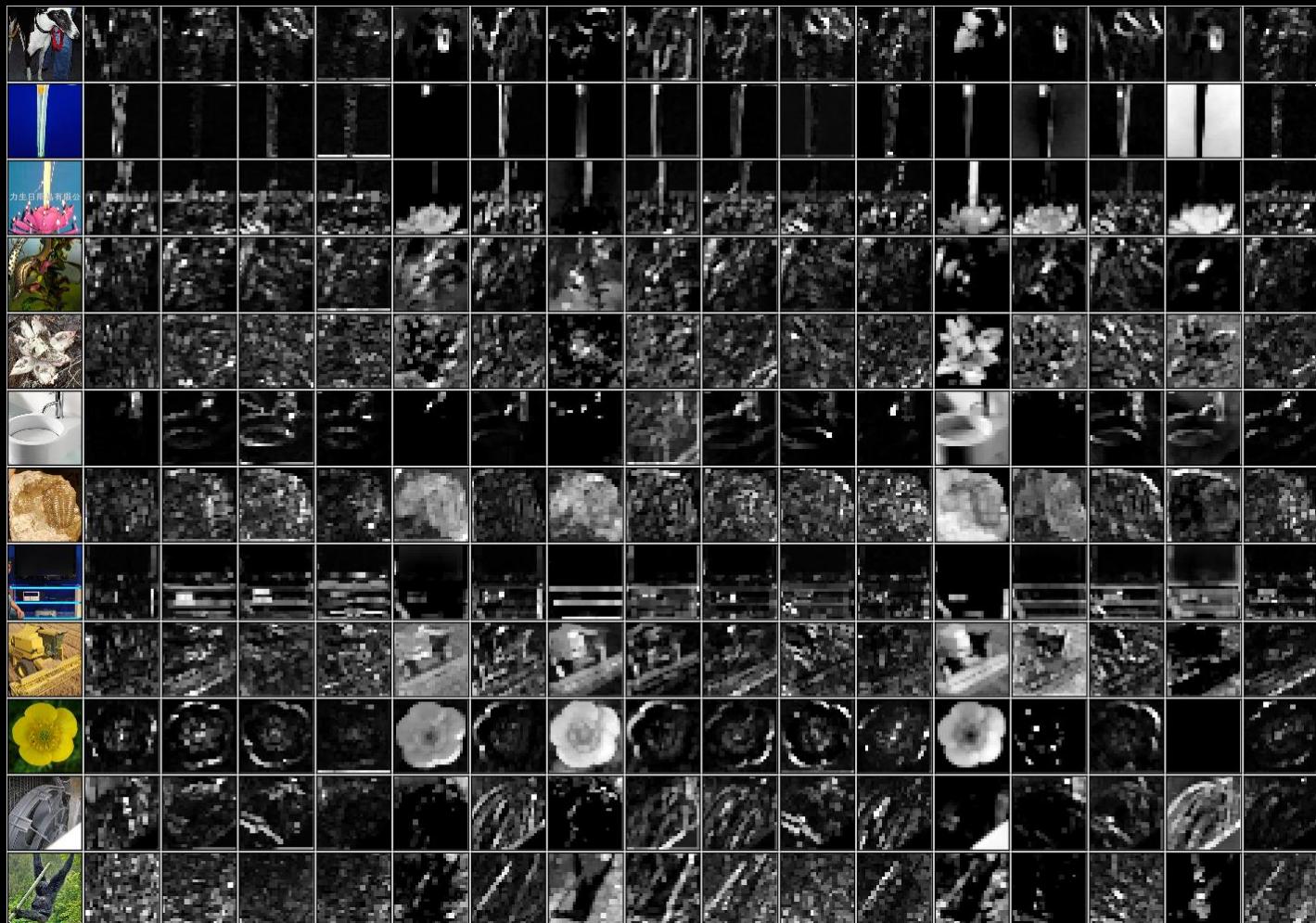
- Gagnant de la compétition ImageNet 2012



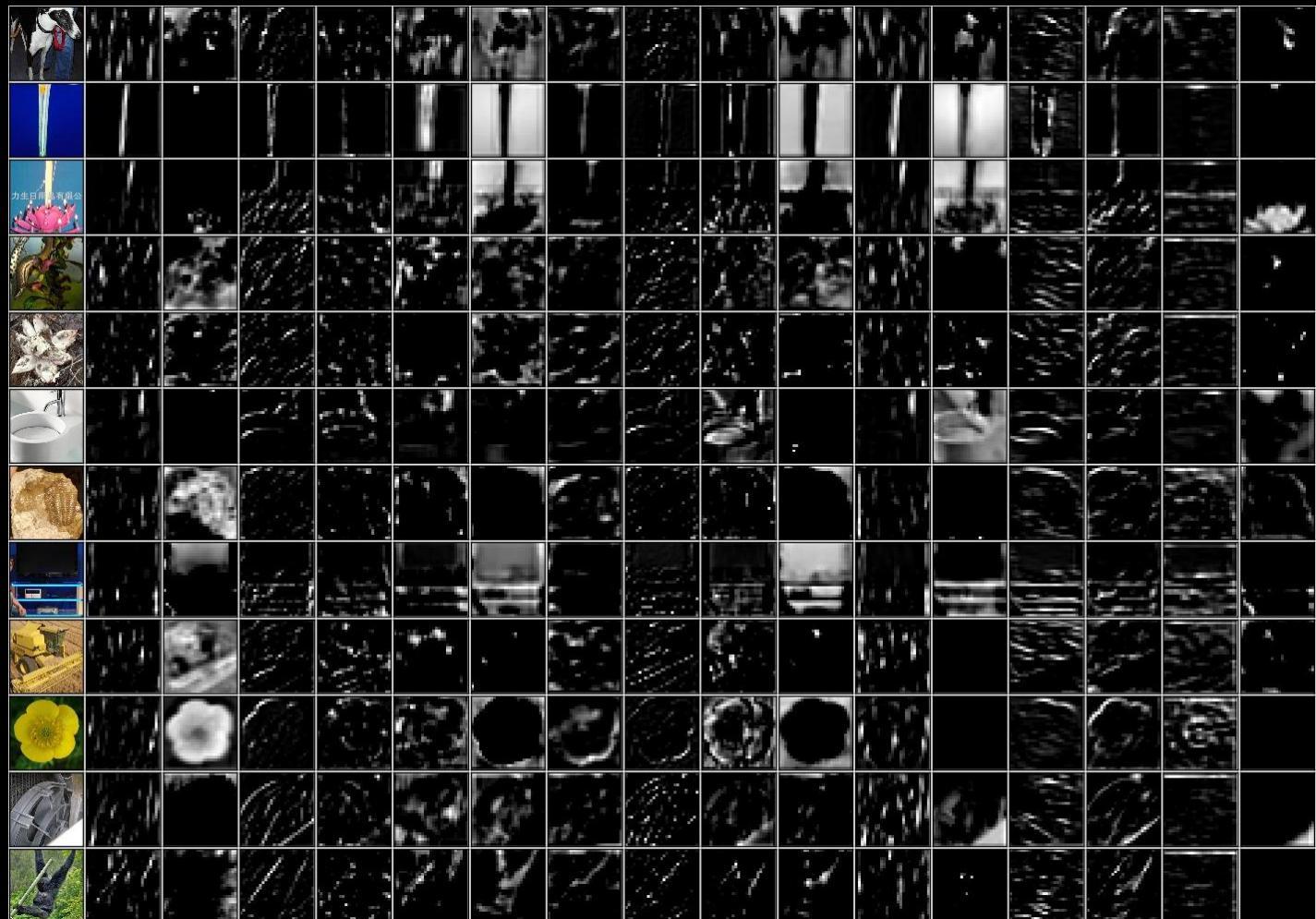
A. Krizhevsky, et al., "ImageNet Classification with Deep Convolutional Neural Networks," NIPS, 2012.



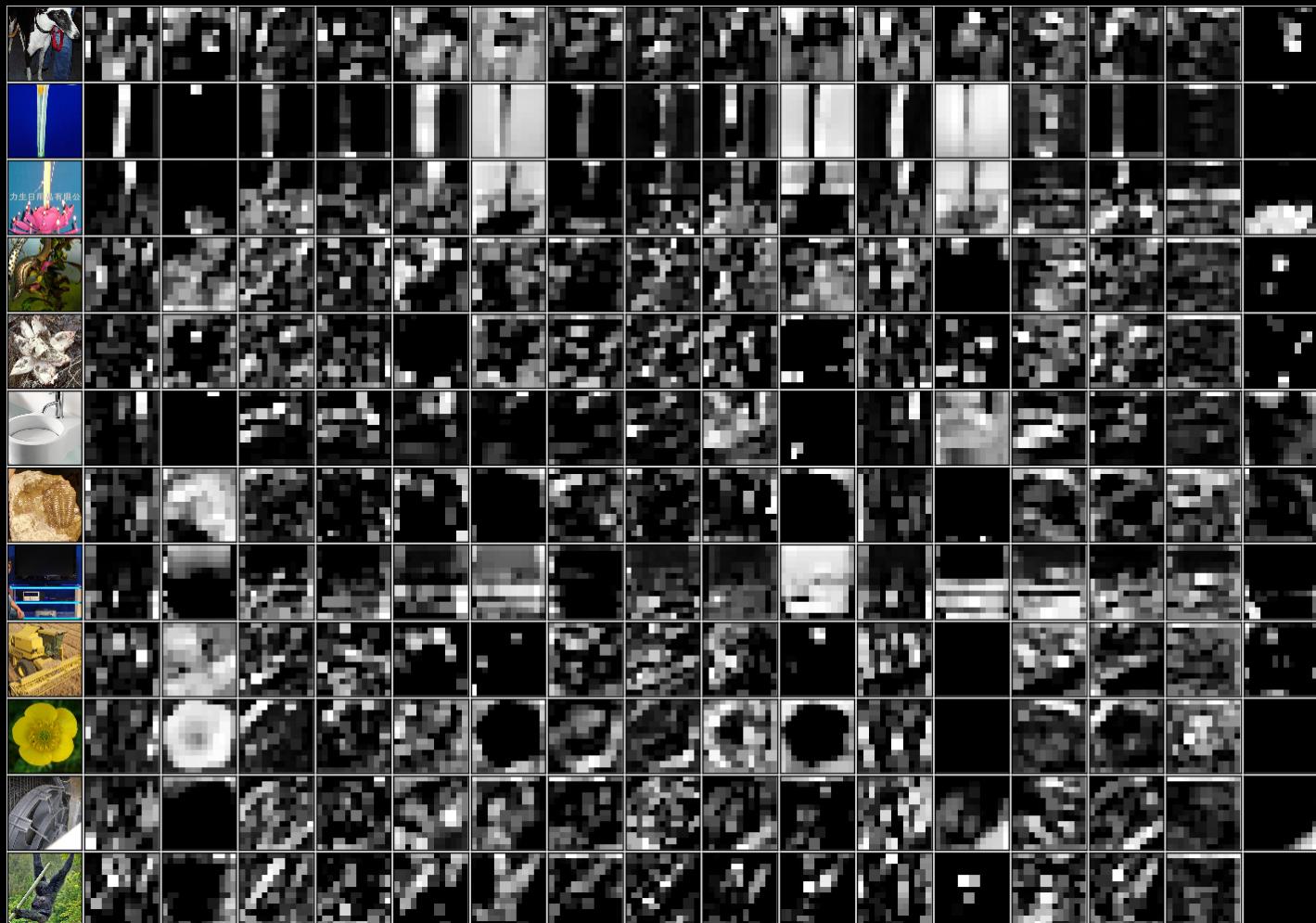
data -> **conv1** -> pool1 -> conv2 -> pool2 -> conv3 -> conv4 -> conv5 -> pool3



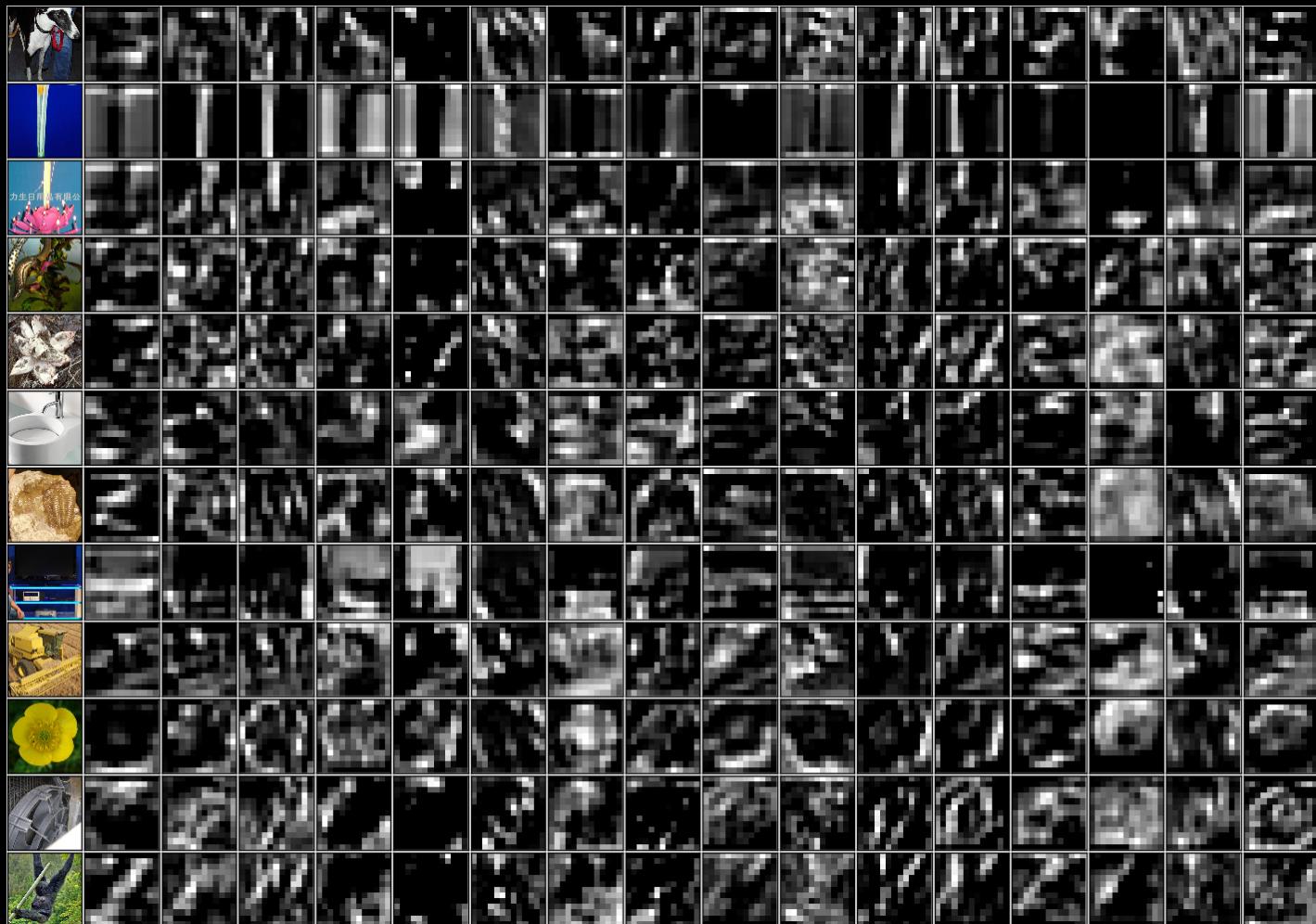
data -> conv1 -> **pool1** -> conv2 -> pool2 -> conv3 -> conv4 -> conv5 -> pool3



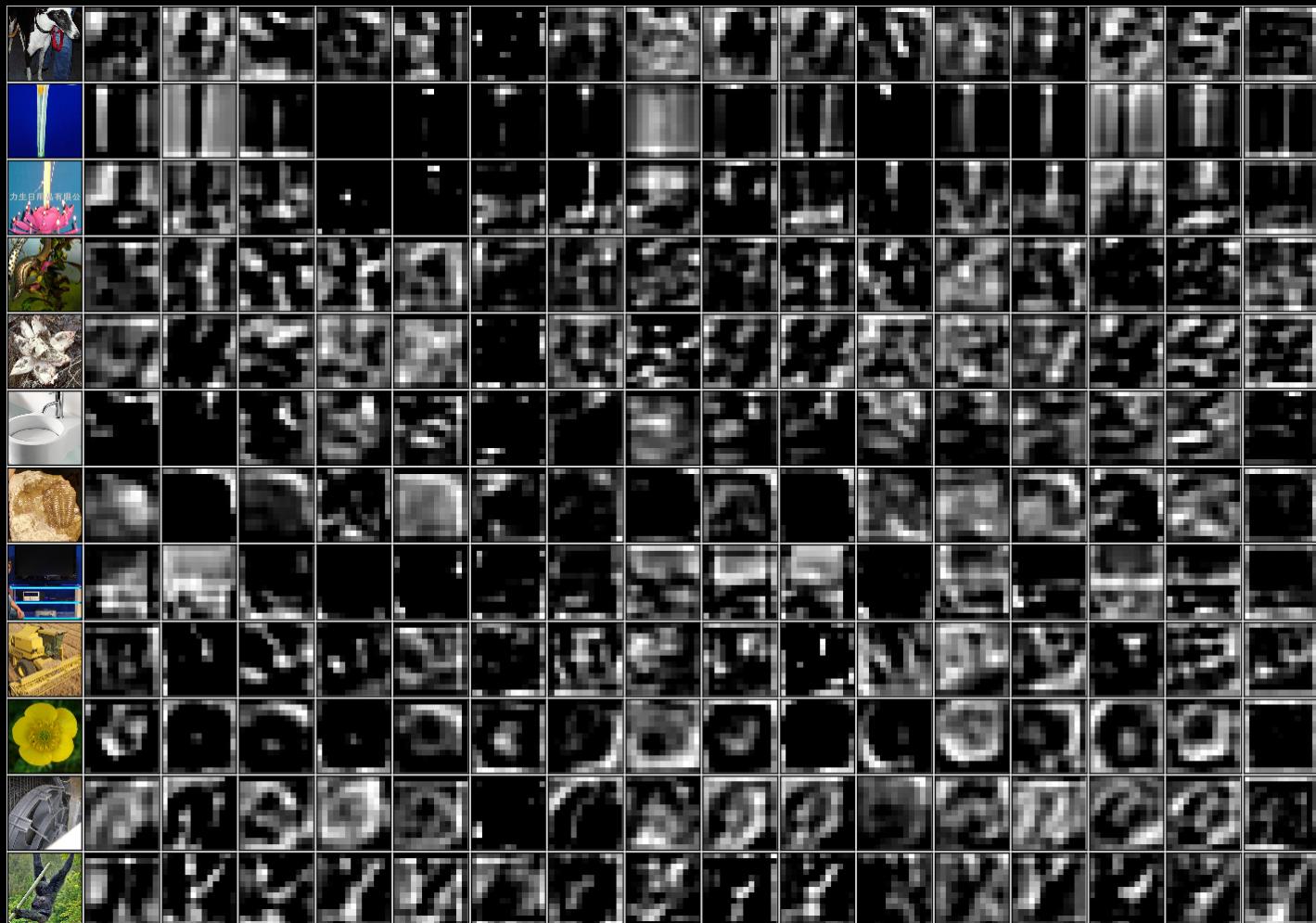
data -> conv1 -> pool1 -> **conv2** -> pool2 -> conv3 -> conv4 -> conv5 -> pool3



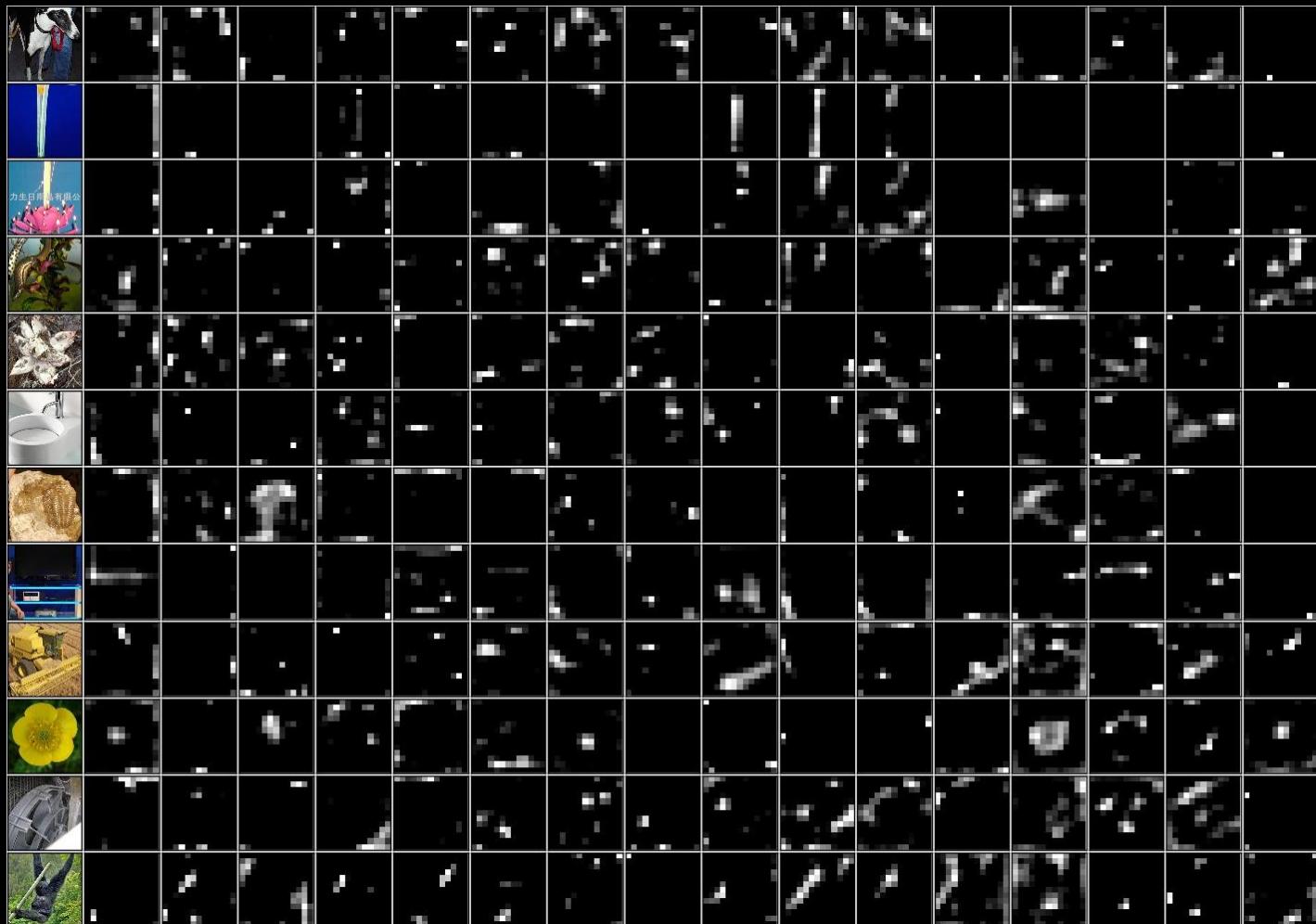
data -> conv1 -> pool1 -> conv2 -> **pool2** -> conv3 -> conv4 -> conv5 -> pool3



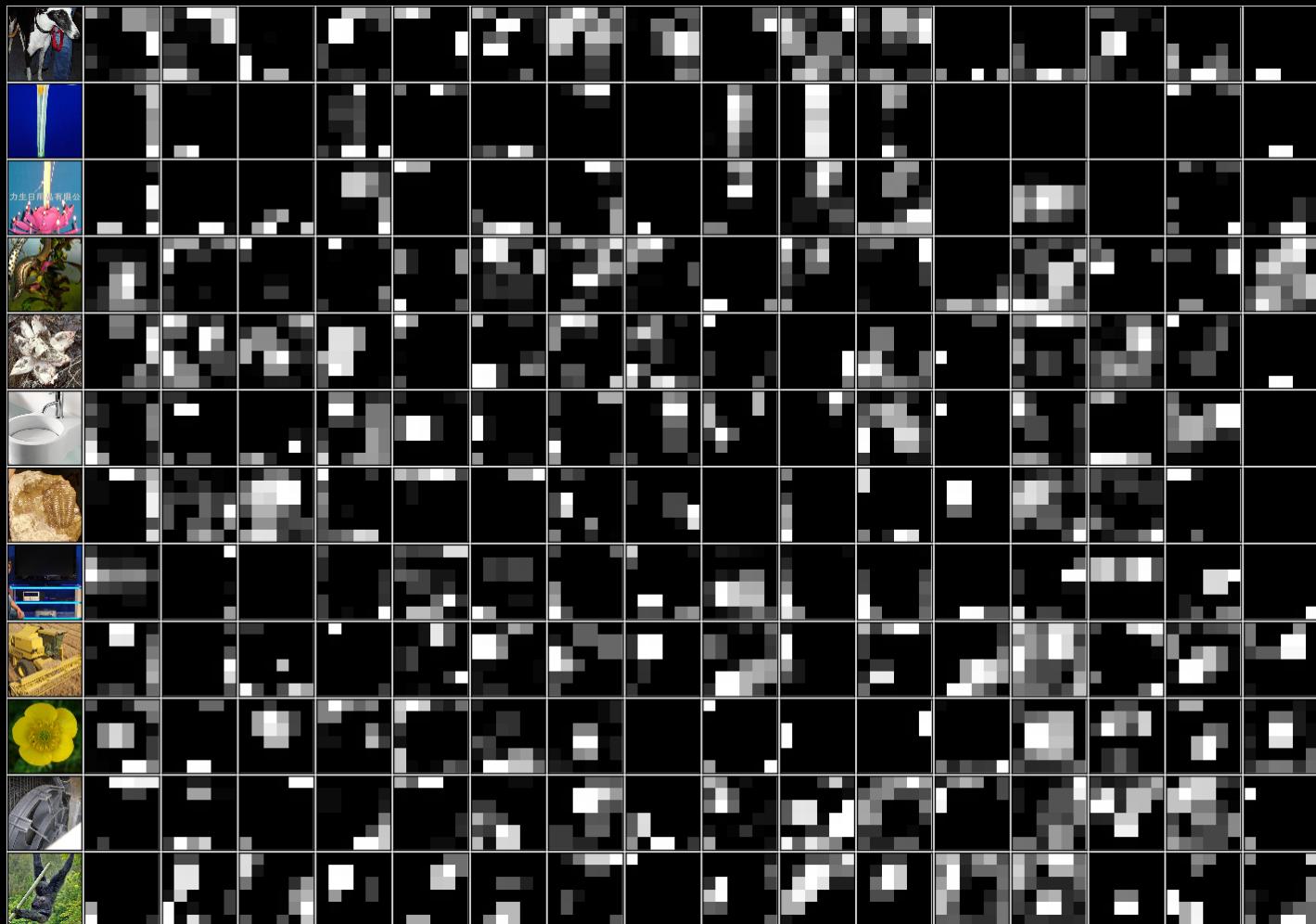
data -> conv1 -> pool1 -> conv2 -> pool2 -> **conv3** -> conv4 -> conv5 -> pool3



data -> conv1 -> pool1 -> conv2 -> pool2 -> conv3 -> **conv4** -> conv5 -> pool3



data -> conv1 -> pool1 -> conv2 -> pool2 -> conv3 -> conv4 -> **conv5** -> pool3



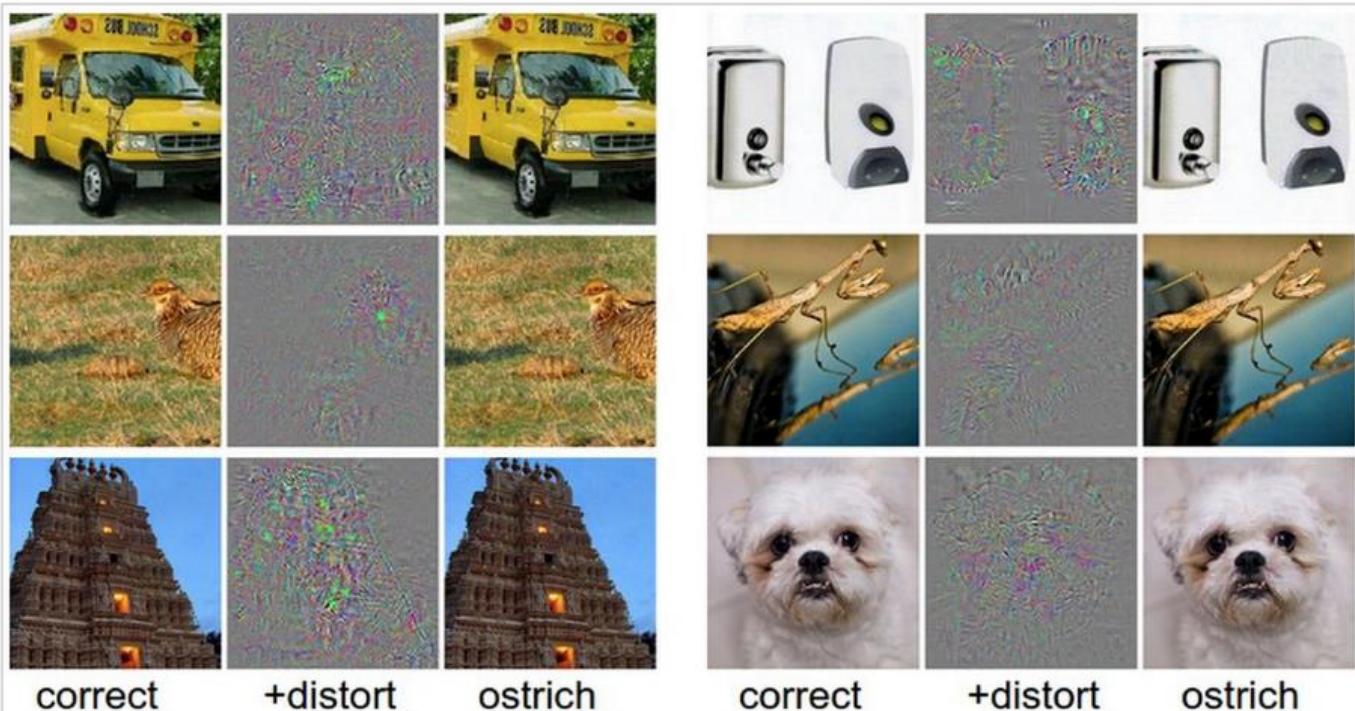
data -> conv1 -> pool1 -> conv2 -> pool2 -> conv3 -> conv4 -> conv5 -> **pool3**

Classification: *Labrador retriever*



pool3 -> ... -> **output**

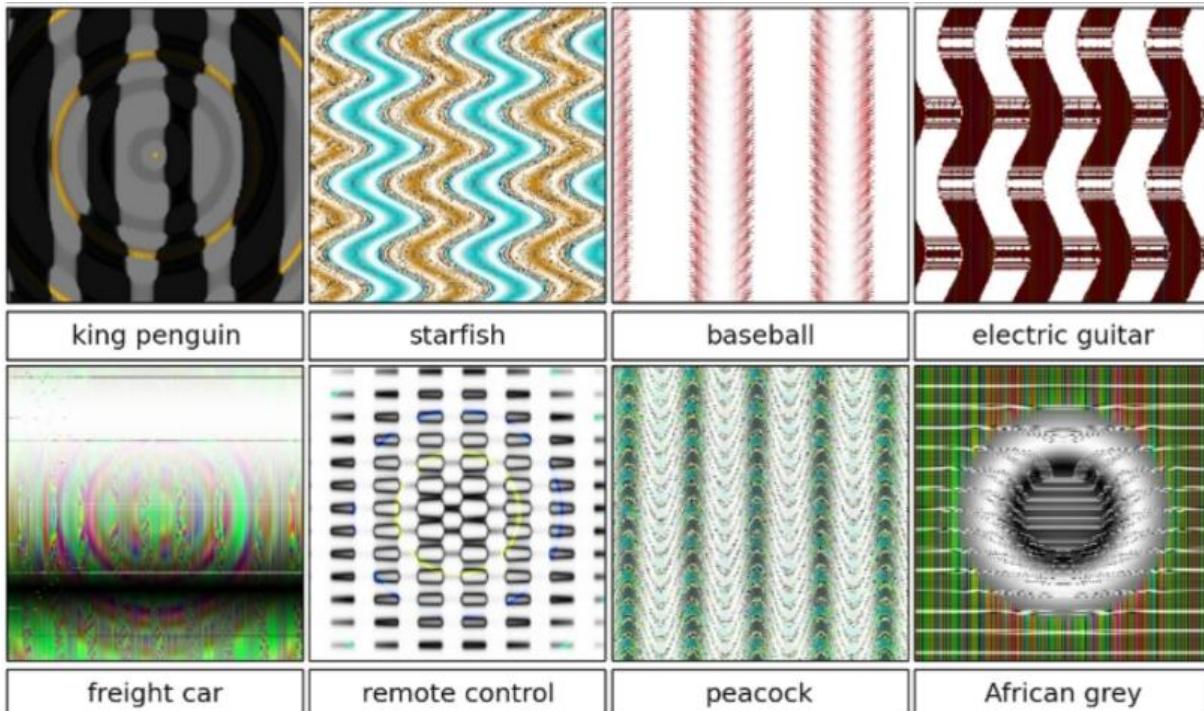
Comment confondre le CNN



Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).

Intriguing properties of neural networks [[Szegedy ICLR 2014](#)]

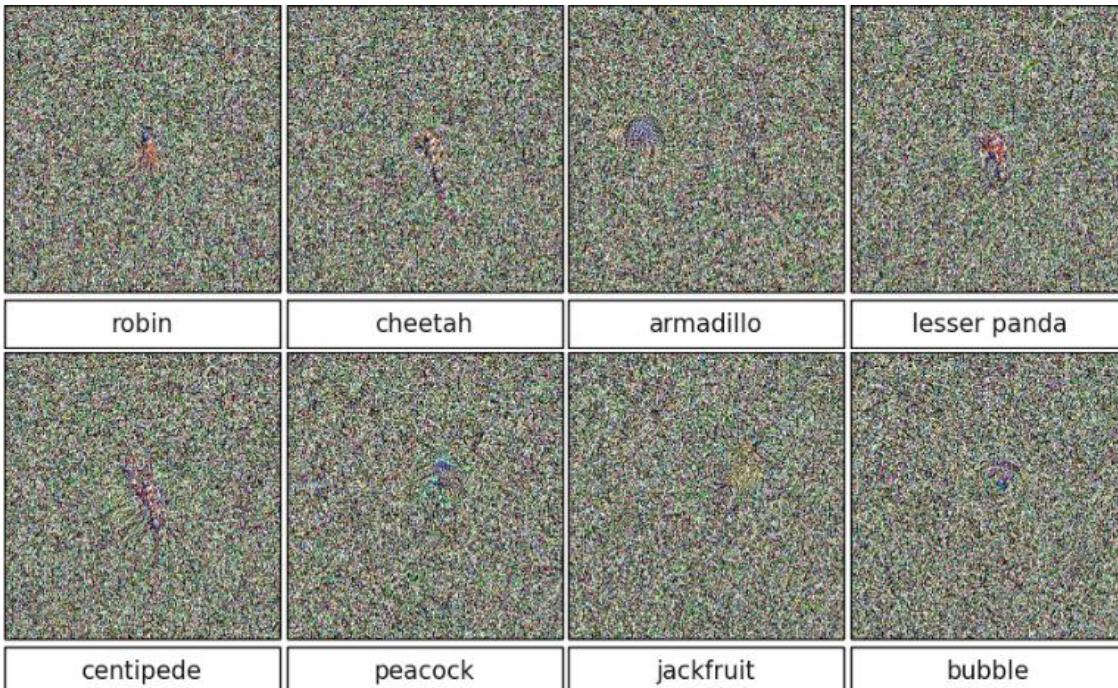
Autres erreurs



Deep Neural Networks are Easily Fooled: High Confidence Predictions for
Unrecognizable Images [[Nguyen et al. CVPR 2015](#)]

Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)

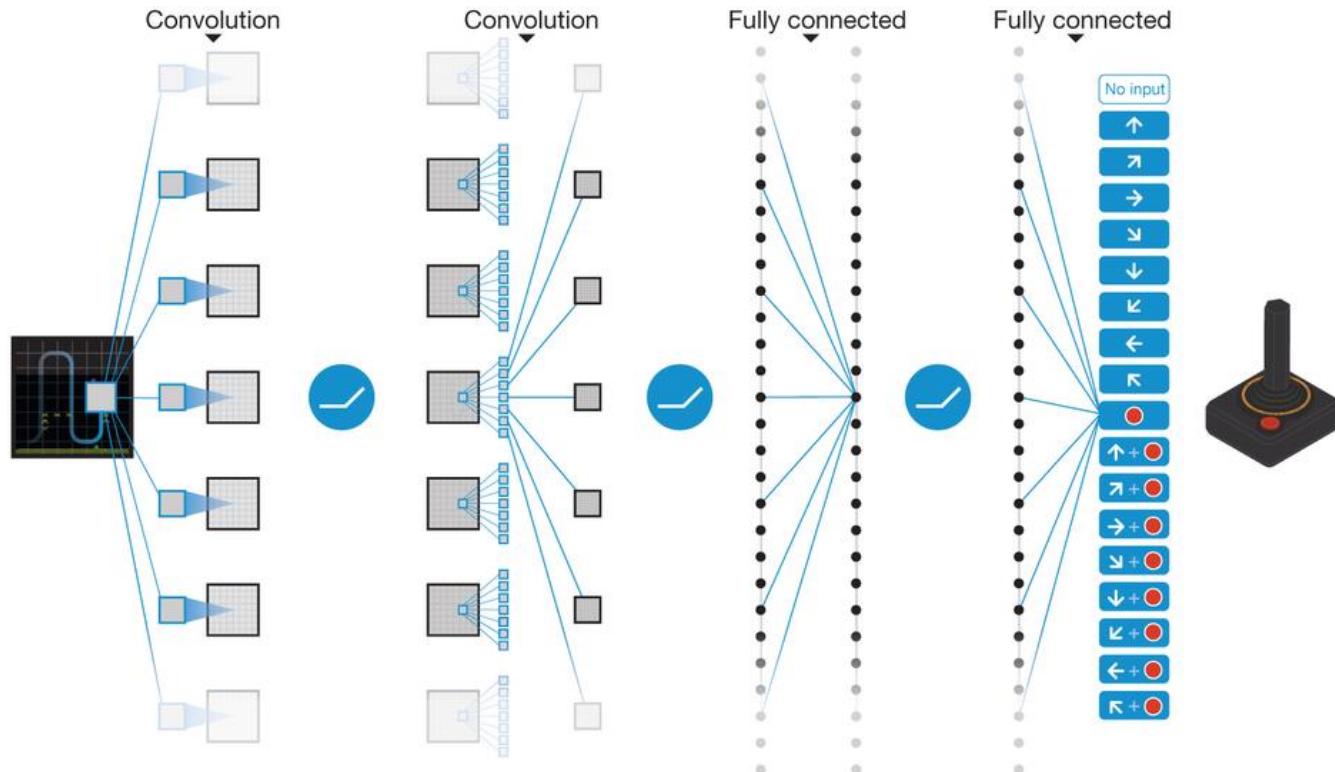
Autres erreurs



Deep Neural Networks are Easily Fooled: High Confidence Predictions for
Unrecognizable Images [[Nguyen et al. CVPR 2015](#)]

Tirée de la présentation de Jia-Bin Huang, University of Illinois (CS 543/ECE 549)

Jouer à un jeu vidéo avec un CNN



Ressources

Ressources Internet

UFLDL Tutorial

http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial

Nando de Freitas - Deep Learning at Oxford

<https://www.youtube.com/watch?v=dV80NAIEins&list=PLE6Wd9FR--EfW8dtjAuPoTuPcqmOV53Fu>

Conférences

http://videolectures.net/jul09_hinton_deeplearn/

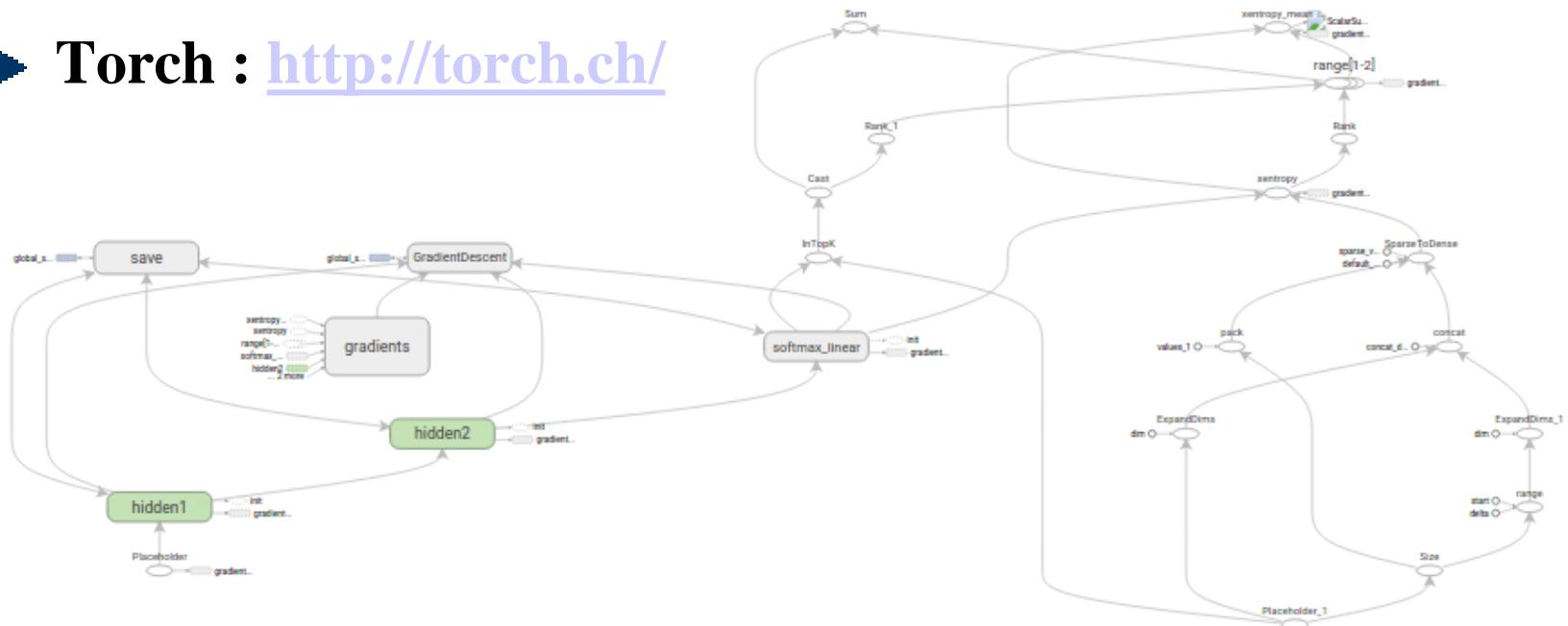
http://videolectures.net/icml09_bengio_lecun_tldar/

Démo Hinton

<http://www.cs.toronto.edu/~hinton/adi/index.htm>

Outils – Deep Learning

- ▶ **TensorFlow de Google :** <https://www.tensorflow.org/>
- ▶ **Caffe :** <http://caffe.berkeleyvision.org/>
- ▶ **Theano :** <http://deeplearning.net/software/theano/>
- ▶ **Torch :** <http://torch.ch/>



Torch

- ▶ Created/Used by NYU, Facebook, Google DeepMind
- ▶ *De rigueur* for deep learning research
- ▶ Its language is *Lua*, NOT Python
- ▶ Lua's syntax is somewhat Pythonic. Check it out.
- ▶ Torch's main strengths are its features, which is why I mention it though here we are at PyData.
- ▶ See <http://bit.ly/1KzuFhd> for a closer look.

Caffe

- ▶ **Created/Used by Berkeley, Google**
- ▶ **Best tool to get started with:**
 - Lots of pre-trained reference models
 - Lots of standard deep learning datasets
- ▶ **Easy to configure networks with config files.**
- ▶ See <http://bit.ly/1Db2bHT> to get started.

Theano

- ▶ **Created/Used by University of Montreal**
- ▶ **Very flexible, very sophisticated:**
 - Lower level interface allows for lots of customization
 - Lots of libraries being built ON TOP of Theano, e.g.:
 - Keras, PyLearn2, Lasagne, etc.
- ▶ **Pythonic API, and very well documented.**
- ▶ See <http://bit.ly/1KBsMAv> to get started.