

Apprentissage profond

Apprentissage et reconnaissance – GIF-4101 / GIF-7005
Professeur : Christian Gagné



Semaine 13 : 30 novembre 2016

Historique des réseaux de neurones

- 1957 : proposition du perceptron par Frank Rosenblatt
- 1967 : démonstration par Marvin Minsky que le perceptron est incapable de traiter des données non linéairement séparables, désintérêt pour les approches neuronales
- 1986 : Rumelhart, Hinton et Williams démontrent l'utilisation de la rétropropagation des gradients pour l'entraînement du perceptron multicouche
- 1995-2005 : développement des SVM, perte d'intérêt pour les réseaux de neurones
- 2006 : premières architectures profondes de réseaux de neurones
- 2012 : résultats en reconnaissance d'objets (Toronto, ImageNet) et de la parole (Microsoft) démontre le potentiel de technologie disruptive de l'apprentissage profond
- 2014 : explosion d'investissements privés en apprentissage automatique, en particulier en apprentissage profond

Émergence des réseaux profonds

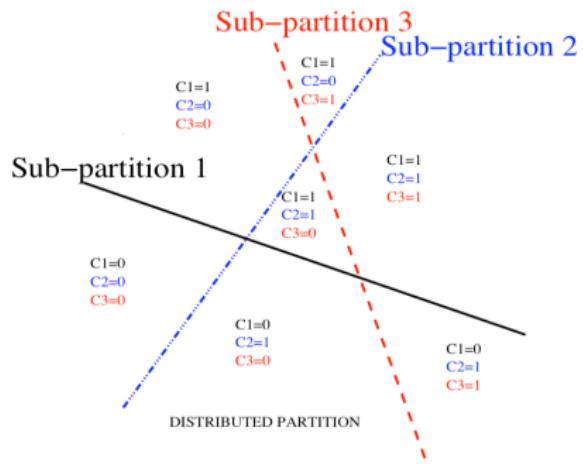
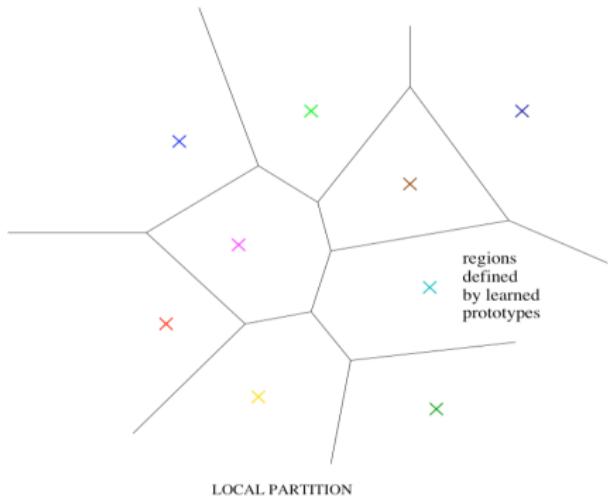
- Conditions ayant permis l'émergence des réseaux profonds :
 - ➊ Disponibilités de très grands jeux de données (*big data*)
 - ➋ Disponibilité d'une capacité de calcul faramineuse (GPU)
 - ➌ Nouveaux modèles d'apprentissage très flexibles, avec des *a priori* permettant de bien gérer la malédiction de la dimensionnalité

Composition de modèles

- « Compositionnalité » de modèle est nécessaire en apprentissage automatique
 - ▶ Comme le langage, il faut composer des éléments pour définir un langage donnant un sens à des notions complexes
- Exploiter la compositionnalité permet un gain exponentiel en puissance de représentation
 - ▶ Représentations distribuées, apprentissage de caractéristiques
 - ▶ Architectures profondes : plusieurs niveaux d'apprentissage de représentations
- Composition de modèles est utile pour décrire notre monde efficacement

Représentation locale vs distribuée

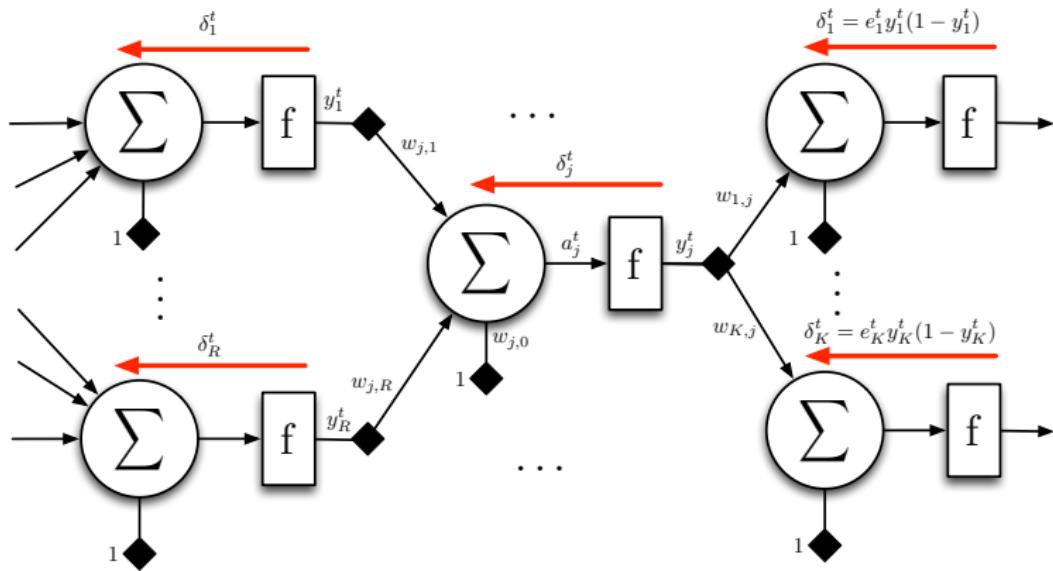
- Ensemble de discriminants distribués (non mutuellement exclusifs) est exponentiellement plus efficace sur le plan statistique que des représentations locales (k -plus proches voisins, clustering)



Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

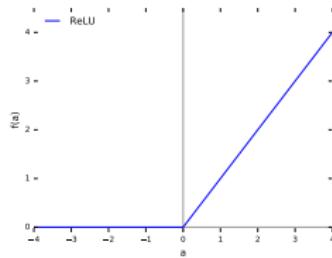
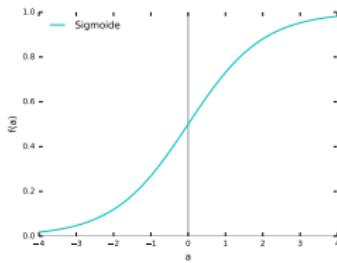
Problème de la dilution du gradient

- L'entraînement de perceptron multicouche de plus de deux couches cachées avec rétropropagation ne fonctionne pas bien
 - ▶ Neurones saturés, avec gradient est très faible
 - ▶ Dilution du gradient (*vanishing gradient*) de couche en couche



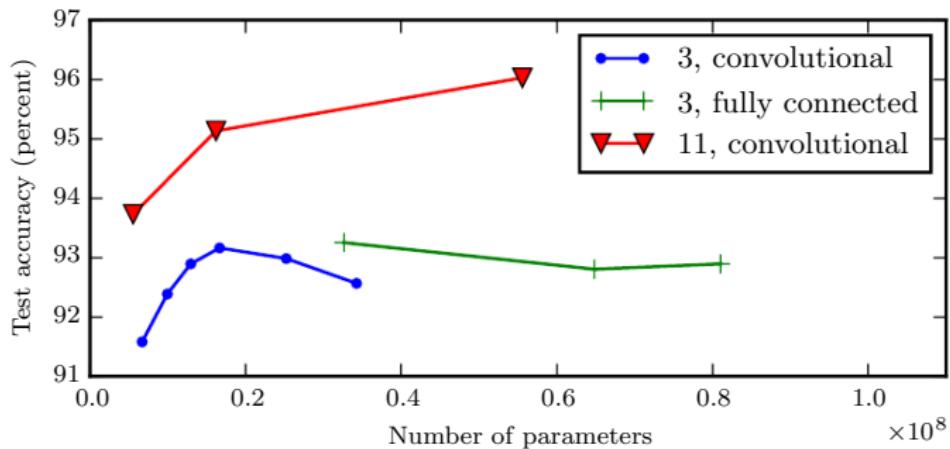
Fonctions de transfert

- Fonction sigmoïde
 - ▶ Interprétation probabiliste
 - ▶ Approximation d'une fonction *step* (binaire)
 - ▶ Problème de saturation sur le gradient
- Fonctions de transfert doivent inclure des non-linéarités
- Fonction ReLU (*Rectified Linear Unit*),
 $f_{\text{ReLU}}(a) = \max(0, a)$
 - ▶ Modèle simple de fonction de transfert avec non-linéarité
 - ▶ Composition de ReLU permet de l'approximation linéaire par morceaux
 - ▶ Motivation biologique de réseaux profonds avec ReLU (*leaky integrate-and-fire model*)
 - ▶ Apprentissage de réseaux profonds avec ReLU possible sans pré-entraînement non supervisé



Profondeur des réseaux

- Les réseaux profonds, lorsque bien entraînés, apprennent mieux que les réseaux obèses (*fat networks*)
 - ▶ Capacité des réseaux croît linéairement selon la largeur d'une couche, exponentiellement selon la profondeur du réseau

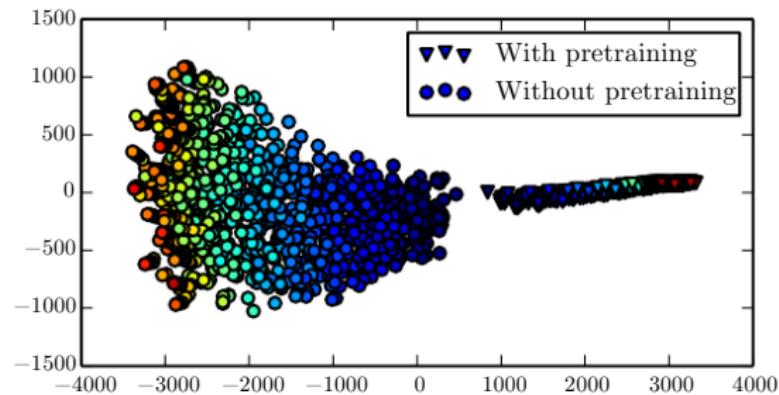


Tiré de I. Goodfellow, Y. Bengio et A. Courville, Deep Learning, MIT Press, 2016.

- ▶ Réseau obèse fait du surapprentissage à 20M de poids, réseau profond fonctionne bien avec 60M de poids

Pré-entraînement non supervisé

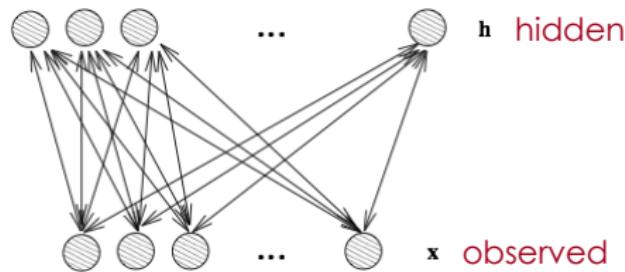
- Réseaux profonds avant 2011 : pré-entraînement non supervisé nécessaire
 - ▶ Initialisation aléatoire de réseaux profonds génère une grande variété de solutions sous-optimales (minima locaux)
 - ▶ Pré-entraînement non supervisé permet de démarrer la rétropropagation dans une « bonne configuration » (bassin d'attraction)



Tiré de I. Goodfellow, Y. Bengio et A. Courville, Deep Learning, MIT Press, 2016.

Machine de Boltzmann restreinte

- Machine de Boltzmann restreinte (RBM) : modèle génératif de réseau de neurones
 - ▶ Peut apprendre des distributions sur les données d'entrées
 - ▶ Couche de neurones visibles (v) et de neurones cachées (h)



Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

- h et v sont binaires, modèle permet de calculer $P(v,h)$, $P(v)$, $P(v|h)$, $P(h|v)$
- Utilisé pour apprendre *deep belief network*, avec apprentissage non supervisé par couche de RBM, suivi d'un raffinement par rétropropagation des erreurs (supervisé)

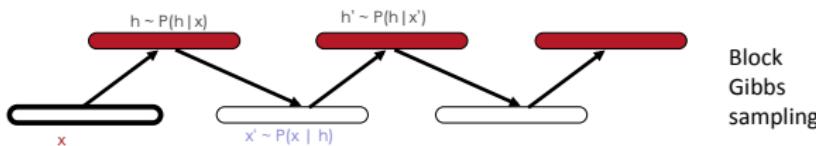
Machine de Boltzmann restreinte

- Fonction d'énergie des RBM

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

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp[-E(\mathbf{v}, \mathbf{h})]$$

- Z est une fonction de partition, permet de normaliser les valeurs pour que les probabilités somment à 1
 - ▶ Calcul de Z , et donc $P(\mathbf{v}, \mathbf{h})$, est intractable ($Z = \sum_{\forall \mathbf{v}, \mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h})]$)
 - ▶ Solution : échantillonnage de Gibbs (Monte Carlo)



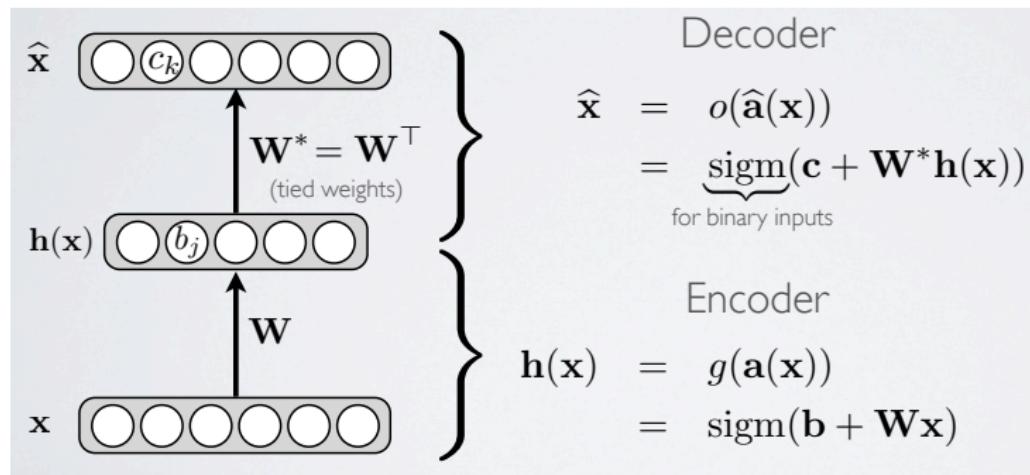
Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

- Computationnellement très lourd

- ▶ RBM très peu utilisé de nos jours pour réseaux profonds

Autoencodeurs

- Autoencodeur : modèle permettant de faire une compression de l'entrée (encodeur) et une décompression de celle-ci (décodeur)
 - ▶ Objectif : compresser tout en gardant l'erreur $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ faible
 - ▶ Poids du décodeur liés aux poids de l'encodeur (habituellement, transposé)

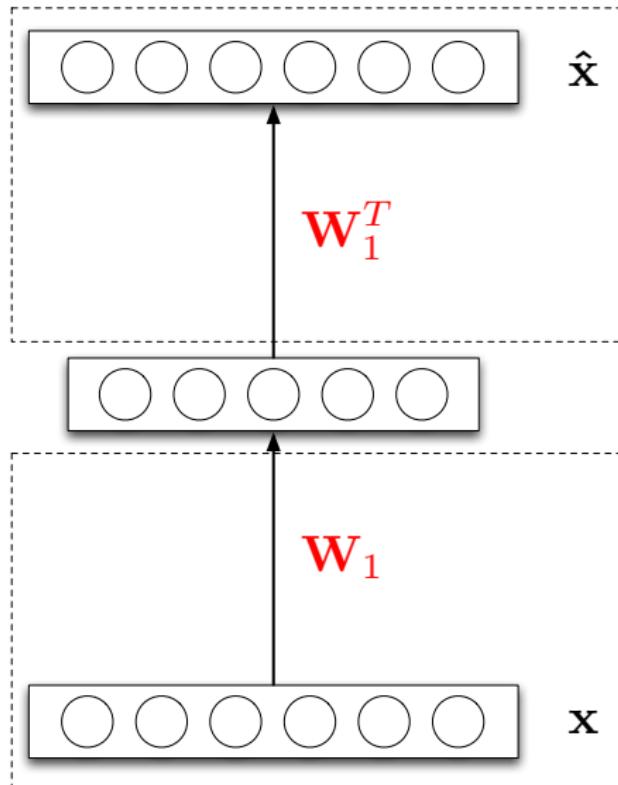


Tiré du cours IFT-725, Réseaux neuronaux, Hugo Larochelle (Université de Sherbrooke), présentation 6.3, automne 2014.

Entraînement d'autoencodeurs

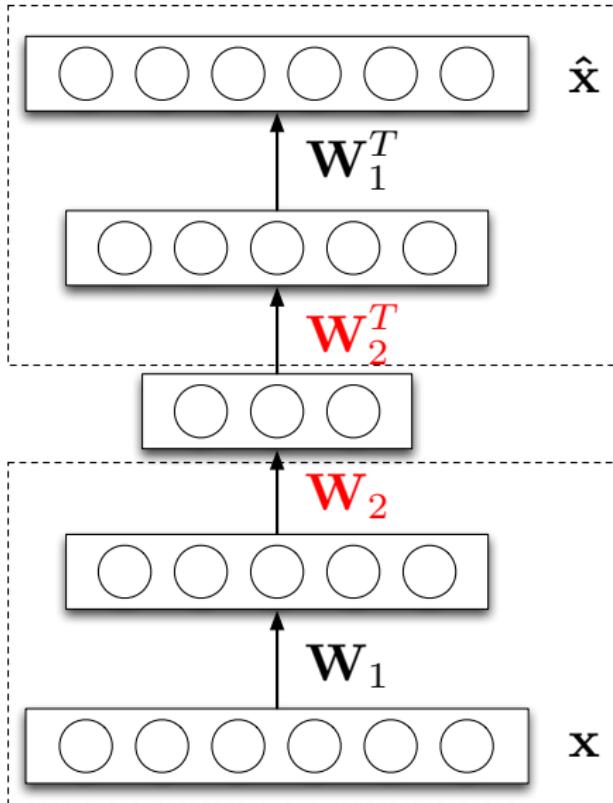
- Autoencodeur entraîné de façon non supervisée, pour apprendre représentation
 - ▶ Encodeur utilisé pour extraire une représentation compacte
- Entraînement vorace, une couche à la fois
 - ▶ Entraînement de la couche la plus externe
 - ▶ Ajout d'une nouvelle couche, qui est entraînée individuellement, couche externe étant fixée, et ainsi de suite
- Fonction de transfert non linéaire entre les couches
 - ▶ Nécessaire, sinon plusieurs couches non linéaires pourraient se simplifier en une seule couche
 - ▶ Apprentissage des poids par descente du gradient (mais utilisation d'une ACP serait possible)
- Couche de sortie ajoutée à l'encodeur, avec entraînement supervisé
 - ▶ Entraînement complet de la couche de sortie par rétropropagation
 - ▶ Ajustement des poids de l'encodeur par rétropropagation (*fine-tuning*)

Exemple d'entraînement d'un autoencodeur



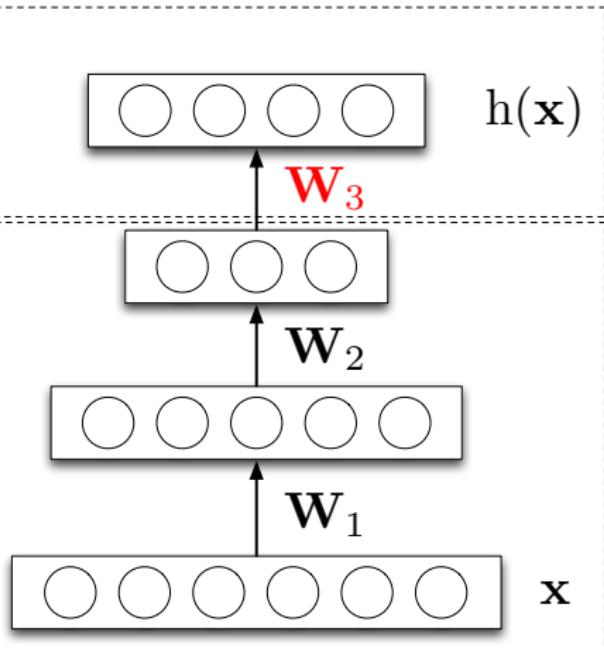
- Entraînement non supervisé du poids \mathbf{W}_1 , poids \mathbf{W}_1^T lié
- Minimize erreur $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$
- Représentation intermédiaire dans valeurs centrales

Exemple d'entraînement d'un autoencodeur



- Ajout de deux nouvelles couches (une dans encodeur et une dans décodeur)
- Entraînement non supervisé du poids \mathbf{W}_2 , poids \mathbf{W}_1 fixés
- Minimize toujours erreur $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$
- Nouvelle représentation intermédiaire
- Peut être répété ainsi sur plusieurs couches

Exemple d'entraînement d'un autoencodeur

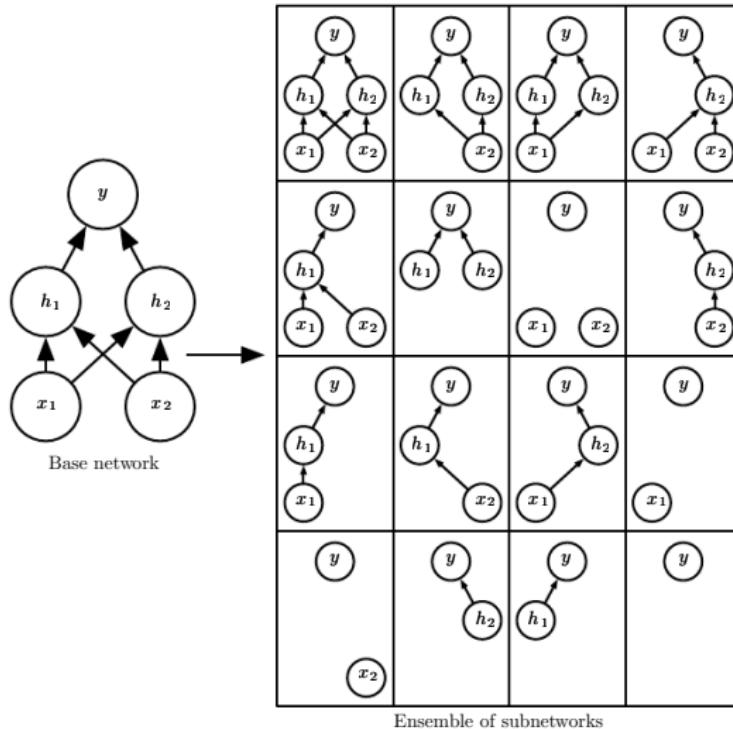


- Retrait de la partie décodeur du réseau
- Ajout d'une couche de sortie, avec autant de sorties que de classe
- Entraînement **supervisé** de W_3 par rétropropagation
- Poids W_1 et W_2 souvent également ajustés finement par rétropropagation (*fine-tuning*)

Dropout

- Dropout : méthode d'entraînement par désactivation aléatoire des neurones
 - ▶ Typiquement, moitié des neurones des couches cachées (80 % des entrées) sont activés à la présentation de chaque donnée durant l'entraînement
 - ▶ Masques aléatoires pour sélectionner neurones actifs, un différent à chaque présentation
- Effectue une régularisation du réseau
 - ▶ Force l'apprentissage d'une représentation distribuée dans l'ensemble du réseau
 - ▶ Rend difficile l'émergence de « neurones grand-mère »
 - ▶ S'est avéré très efficace pour améliorer les performances des réseaux profonds
- Évaluation de nouvelles données en test par moyennage sur plusieurs masques de sélection
 - ▶ Analogie avec méthodes par ensemble (vu la semaine prochaine), en particulier bagging

Dropout



Tiré de *I. Goodfellow, Y. Bengio et A. Courville, Deep Learning, MIT Press, 2016.*

Batch normalization

- Modification d'un poids par rétropropagation basé sur gradient local
 - ▶ Poids des couches précédentes et suivantes eux aussi modifiés !
- *Batch normalization* : normaliser activation des neurones entre toutes les données d'un mini-lot (*mini-batch*)
 - ▶ Mini-lot : petit sous-ensemble d'instances de données de l'ensemble d'entraînement (typiquement quelques centaines)
- Activation des neurones \mathbf{H} normalisées selon

$$\mathbf{H}' = \frac{\mathbf{H} - \mu}{\sigma}, \quad \mu = \frac{1}{m} \sum_i \mathbf{H}_{i,:}, \quad \sigma = \sqrt{\epsilon + \sum_i (\mathbf{H} - \mu)_i^2}$$

- ▶ \mathbf{H} : activation des neurones (ligne) d'une couche pour les données du minilot (colonne)
- ▶ ϵ : petite valeur (typiquement 10^{-8}) pour éviter division par zéro lorsque variance nulle

Traitement de texte

- Comment donner des documents (séquence de chaînes de caractères) à un réseau de neurones (vecteur de réels de taille fixe) ?
- Modèle *Bag-of-Words* (BoW)
 - ▶ Identifier dictionnaire de mots les plus fréquents / intéressants
 - ▶ Calculer la fréquence de chaque mot dans le document (vecteur d'entiers de taille fixe)
- Ne tient pas compte de l'ordre
 - ▶ Modèles avec N-gram mesure fréquence de groupes de mots adjacents
 - ▶ Skip-gram : mots connexes peuvent ne pas être adjacents
- Tenir compte de la séquence pour encoder le texte
 - ▶ *Continuous BoW* : prédire le mot selon les mots qui précèdent et suivent
 - ▶ *Continuous skip-gram* : prédire les mots qui précèdent et suivent selon le mot d'intérêt

Continuous BoW

Input layer

$$x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \\ \vdots \\ x_V$$

Hidden layer

$$h_1 \\ h_2 \\ h_i \\ h_N$$

$$\mathbf{W}_{V \times N} = \{w_{ki}\}$$

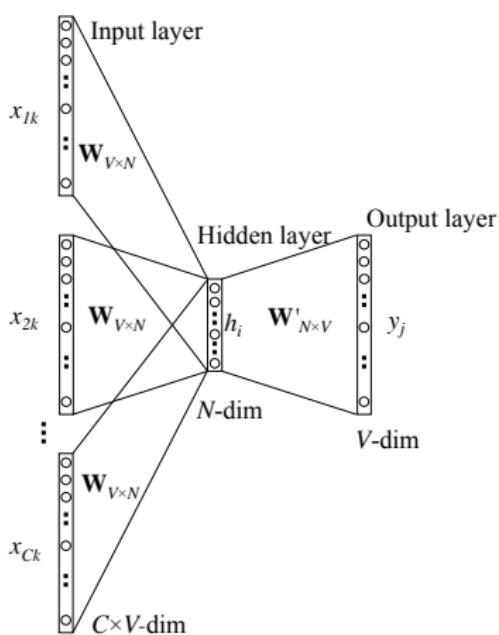
Output layer

$$y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_j \\ \vdots \\ y_V$$

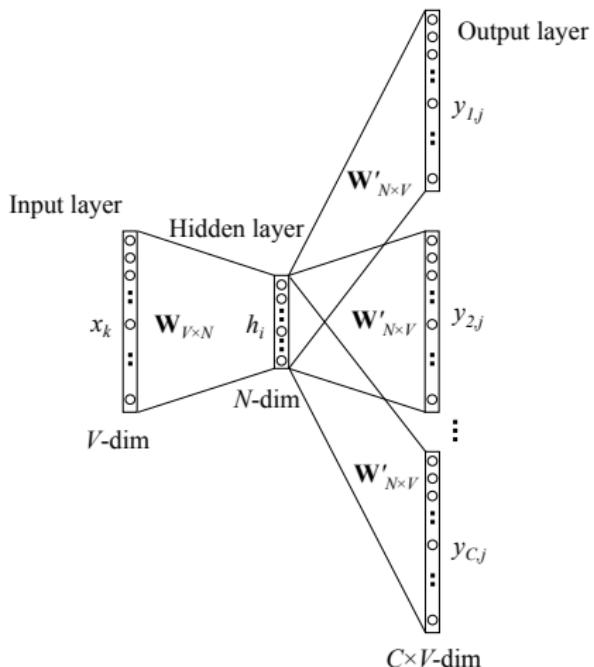
$$\mathbf{W}'_{N \times V} = \{w'_{ij}\}$$

Tiré de X. Rong, word2vec Parameter Learning Explained, arXiv:1411.2738v4, 2016.

Représentation neuronale de mots



Continuous BoW



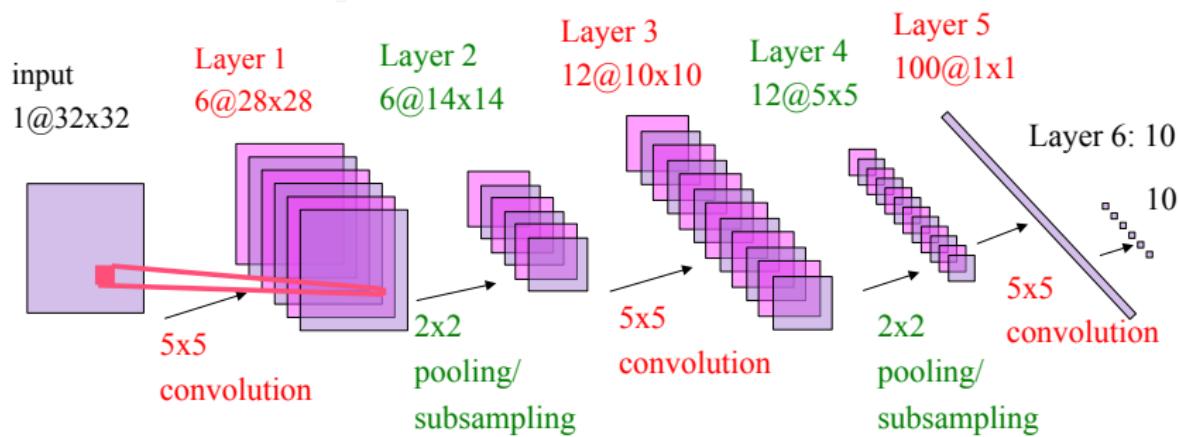
Continuous skip-gram

Tiré de X. Rong, *word2vec Parameter Learning Explained*, arXiv:1411.2738v4, 2016.

Réseau à convolution

- Réseau à convolution : traiter des signaux temporels ou spatiaux
 - ▶ Signal temporel : son et parole
 - ▶ Signal spatial : image
- Couche de convolution : filtres convolués sur données temporelles/spatiales
 - ▶ Données peuvent être valeurs d'entrée du réseau ou sorties de couches précédentes
 - ▶ Convolution sur chaque canal (plusieurs canaux possibles)
 - ▶ Apprentissage des filtres par rétropropagation
- Couche de *pooling* : sélection de valeurs (maximum d'une fenêtre)
 - ▶ Permet de réduire taille des valeurs, sinon explosion de la taille du modèle en vue !
- Neurones pleinement connectés en sortie pour prise de décision

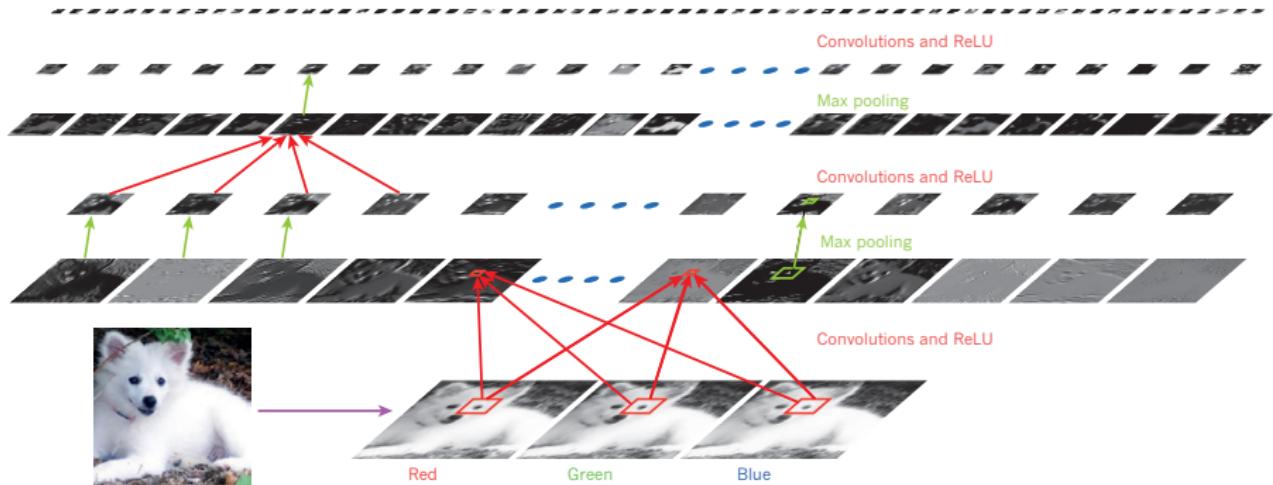
Réseau à convolution



Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

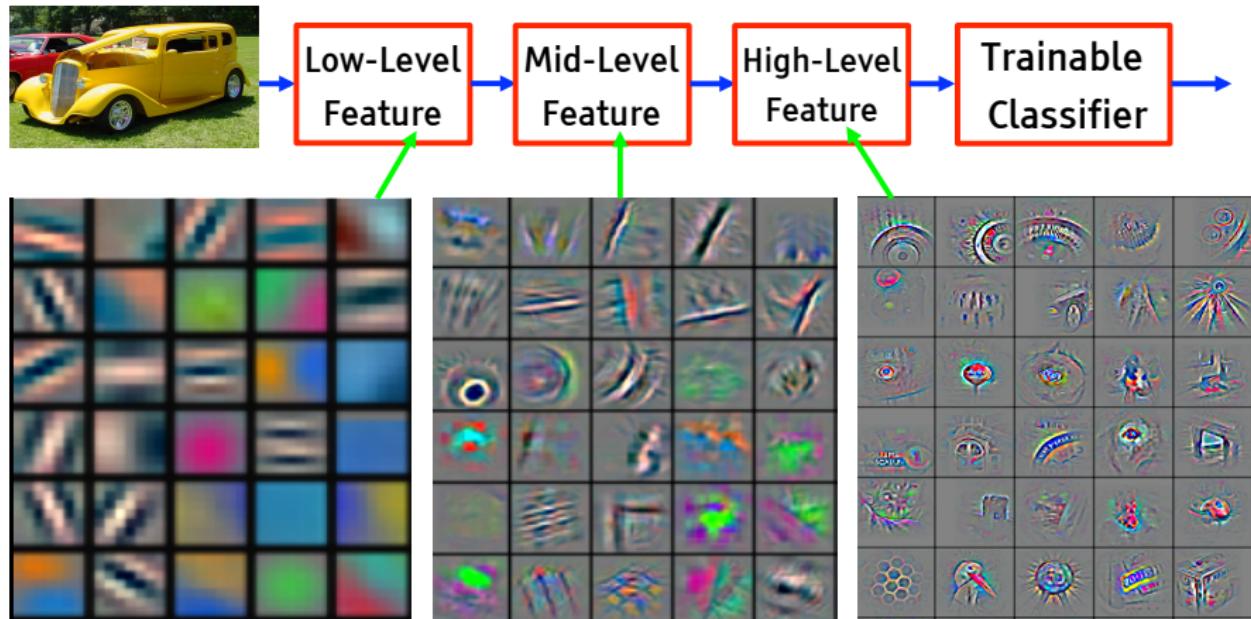
Réseau à convolution

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



Tiré de Y. LeCun, Y. Bengio et G. Hinton, Deep Learning, Nature, vol. 521, 28 mai 2015.

Composition de filtres



Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

Reconnaissance d'objets

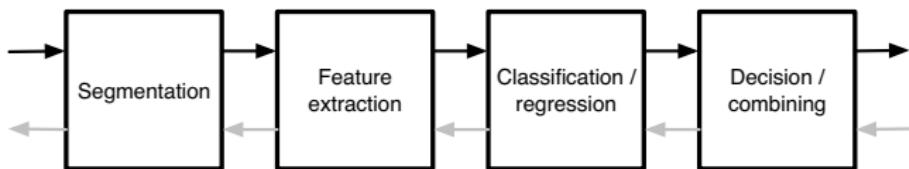
- Compétition ImageNet : reconnaître les objets d'images (1000 classes), en donnant la bonne classe dans un top 5

2012 Teams	%error	2013 Teams	%error	2014 Teams	%error
Supervision (Toronto)	15.3	Clarifai (NYU spinoff)	11.7	GoogLeNet	6.6
ISI (Tokyo)	26.1	NUS (singapore)	12.9	VGG (Oxford)	7.3
VGG (Oxford)	26.9	Zeiler-Fergus (NYU)	13.5	MSRA	8.0
XRCE/INRIA	27.0	A. Howard	13.5	A. Howard	8.1
UvA (Amsterdam)	29.6	OverFeat (NYU)	14.1	DeeperVision	9.5
INRIA/LEAR	33.4	UvA (Amsterdam)	14.2	NUS-BST	9.7
		Adobe	15.2	TTIC-ECP	10.2
		VGG (Oxford)	15.2	XYZ	11.2
		VGG (Oxford)	23.0	UvA	12.1

Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015.

Apprentissage de représentations

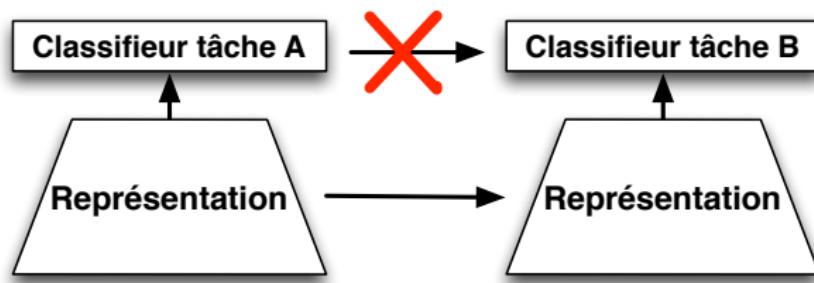
- Pipeline classique de la reconnaissance des formes



- Dans le passé, chaque module conçu indépendamment
- Apprentissage profond permet l'apprentissage de représentations
 - ▶ Apprentissage de tous les modules simultanément
 - ▶ Possibilité de récupérer les représentations (segmentation, extraction de caractéristiques) et les utiliser avec d'autres modules de classement et de prise de décision

Transfert de représentations

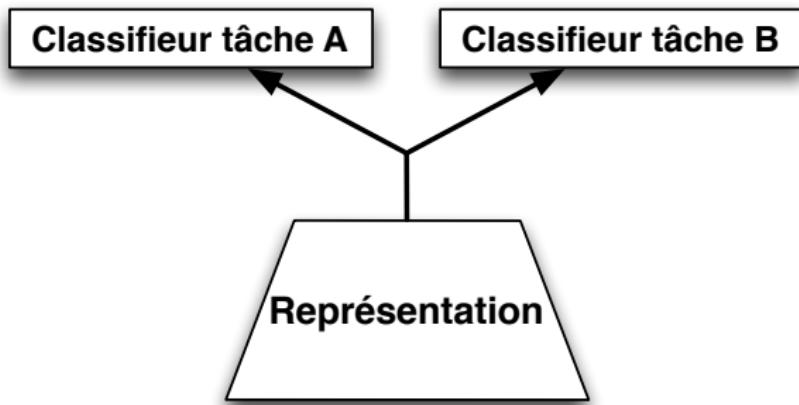
- Apprentissage d'un réseau profond sur tâche A
- Nouvelle tâche B, basée sur données similaires à tâche A
 - ▶ Récupérer représentation de tâche A
 - ▶ Entraîner nouveau classifieur pour tâche B



- ▶ Permet un transfert de représentation (*transfer learning*)

Apprentissage multitâche

- Apprentissage multitâche : apprendre *simultanément* une représentation pour des opérations distinctes
 - ▶ Réseau à deux têtes, une pour chaque tâche



- ▶ Rétropropagation provient d'une tête à la fois
- ▶ Mélange des données et des tâches durant l'apprentissage
- ▶ Performe bien à produire des représentations capturant des concepts généraux

Génération d'exemples

- Idée : générer des données d'entrées à partir d'une sortie désirée
 - ▶ Générer donc un modèle de la donnée pouvant produire la sortie selon le réseau de neurones
- Approche : descendre le gradient sur la donnée d'entrée

$$\Delta \mathbf{x} = -\eta \frac{\partial E(\mathbf{x}|\theta)}{\partial \mathbf{x}}$$

- ▶ On va donc générer une nouvelle donnée à partir de la valeur initiale de \mathbf{x} et la sortie désirée \mathbf{r}
 - ▶ Poids du réseau ne changent pas
- Utilisé dans diverses circonstances
 - ▶ Générateur d'images *Deep Dream* de Google

Deep dream



Horizon



Trees



Leaves



Towers & Pagodas



Buildings



Birds & Insects

Tiré de <https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.
Licensed by Google Inc. under a Creative Commons Attribution 4.0 International License.

Données adversariales

- Utiliser génération de données pour déterminer plus petite variation permettant de faire une erreur de classement

$$\mathbf{x} + .007 \times \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y)) = \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$$

\mathbf{x}
 $y = \text{"panda"}$
w/ 57.7%
confidence

“nematode”
w/ 8.2%
confidence

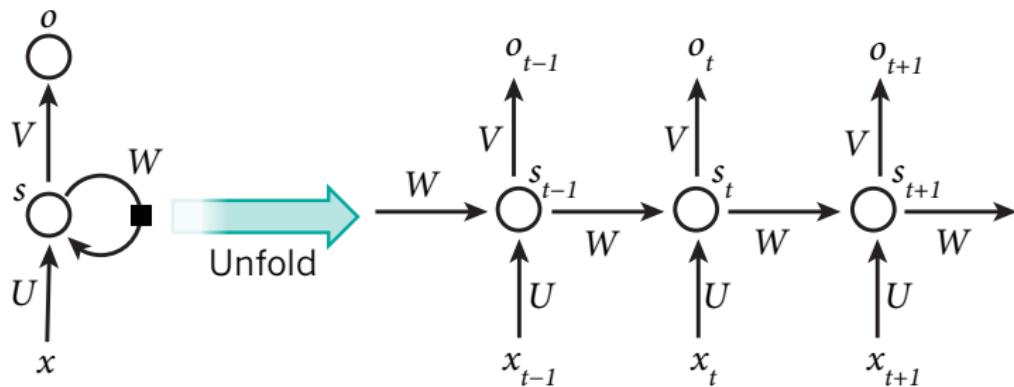
“gibbon”
w/ 99.3 %
confidence

Tiré de *I. Goodfellow, Y. Bengio et A. Courville, Deep Learning, MIT Press, 2016.*

- Causé par l'utilisation de représentation distribuée dans un espace à très haute dimensionnalité
- Illustre une difficulté actuelle avec réseaux profonds, robustesse aux données adversariales doit être améliorée

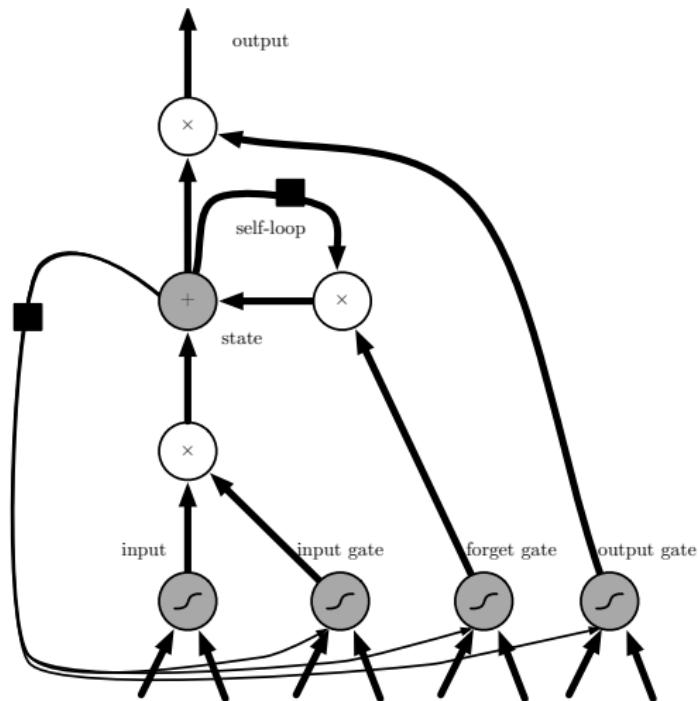
Réseau récurrent

- Réseaux usuels (*feedforward*) : données propagées dans le réseau, indépendant des données suivantes / précédentes
 - ▶ Traitement de données séquentielles important dans nombreux contextes
- Réseaux récurrents : connexions avec valeurs précédentes
 - ▶ Traitement avec algorithmes habituels en déroulant le réseau



Tiré de Y. LeCun, Y. Bengio et G. Hinton, Deep Learning, Nature, vol. 521, 28 mai 2015.

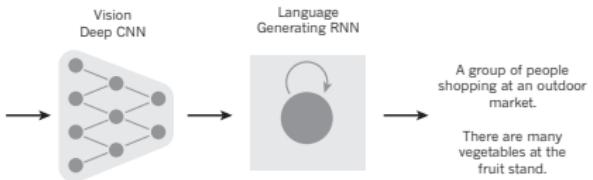
Long Short-Term Memory (LSTM)



Tiré de G. Hinton, Y. Bengio et Y. LeCun,
Deep Learning NIPS'15 Tutorial, 2015.

- Modèle LSTM : ajouter de la mémoire au réseau
- Cellule de mémoire (état), avec quatre neurones
 - ▶ Entrée
 - ▶ Activation de l'entrée
 - ▶ Activation de l'oubli
 - ▶ Activation de la sortie

Génération de légendes



A woman is throwing a **frisbee** in a park.



A **dog** is standing on a hardwood floor.



A **stop** sign is on a road with a mountain in the background



A little **girl** sitting on a bed with a teddy bear.



A group of **people** sitting on a boat in the water.



A **giraffe** standing in a forest with **trees** in the background.

Tiré de Y. LeCun, Y. Bengio et G. Hinton, *Deep Learning*, Nature, vol. 521, 28 mai 2015.

Apprentissage par renforcement profond

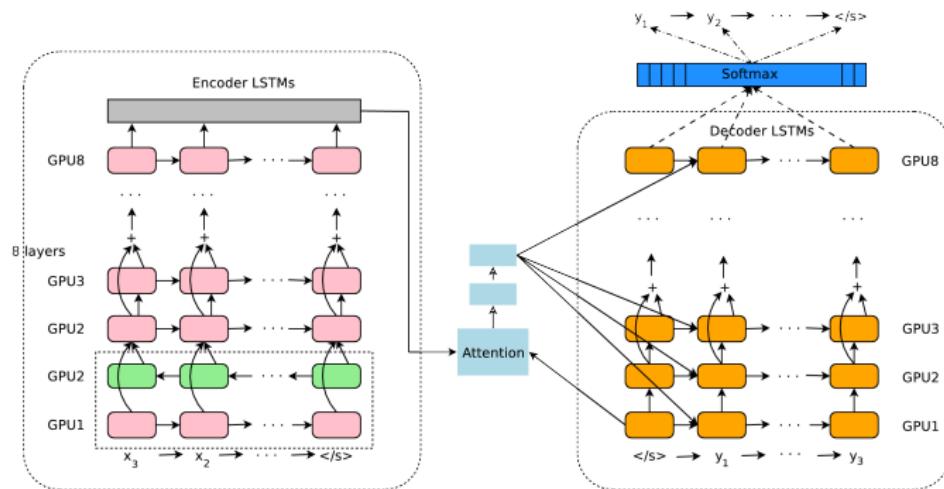
- Apprentissage par renforcement : déterminer les bonnes actions à effectuer selon les conditions actuelles
 - ▶ Guidé par récompense ponctuelle, sans indication précise sur actions décisives
 - ▶ Forme plus élaborée (et plus complexe) d'intelligence que tâches de classement et régression
- Réseaux profonds s'avèrent très prometteurs pour l'apprentissage par renforcement
 - ▶ Simulations massives permettent d'apprendre à effectuer certaines actions précises
- Jeux vidéos (Atari 2600) : Deep Q-learning Network (Deepmind)
 - ▶ Entrée est la capture de l'écran, récompense est le pointage obtenu
 - ▶ 49 jeux différents, performances « surhumaines »
- Jeu de Go : AlphaGo (encore Google Deepmind)
 - ▶ Go : jeu traditionnel asiatique, plus complexe que les échecs
 - ▶ [http://www.theverge.com/2016/3/9/11185030/
google-deepmind-alphago-go-artificial-intelligence-impact](http://www.theverge.com/2016/3/9/11185030/google-deepmind-alphago-go-artificial-intelligence-impact)

Conduite automatisée

- Véhicules autonomes : chamboulement prochain dans la façon de se déplacer
 - ▶ Premiers essais avec capteurs très coûteux (ex. LIDAR longue portée)
 - ▶ Développement avec technologies plus abordables (caméra vidéo, RADAR, Sonar)
- Grand potentiel de l'apprentissage profond pour conduite autonome
 - ▶ Détection d'objets et de piéton
 - ▶ Lecture de la signalisation (panneaux et lumières)
 - ▶ Interprétation de la conduite d'autres véhicules
 - ▶ Contrôle de la conduite
- Prédictabilité de l'apprentissage automatique, en particulier apprentissage profond, reste un problème
- Approche de Tesla pour la conduite autonome
 - ▶ Équiper tous les véhicules fabriqués de capteurs sophistiqués et d'un lien réseau
 - ▶ Collecter information sur la conduite par tous ces véhicules
 - ▶ Automatiser graduellement la conduite par un apprentissage sur ces données à mesure que la qualité des modèles appris augmente

Traduction automatisée

- *Google's Neural Machine Translation System* : nouvelle mouture de Google Translate basée sur des réseaux profonds
 - ▶ Amélioration de 60 % des performances relativement à la version précédente
 - ▶ En cours de déploiement sur les systèmes de Google



Tiré de Wu et al., *Google's Neural Machine Translation System : Bridging the Gap between Human and Machine Translation*, arXiv:1609.08144v1, 2016.

Exemples de traduction avec GNMT (1/3)

Table 11: Some example translations from PBMT [15], our GNMT system (the "NMT before RL", Table 9), and Human. Source and target sentences (human translations) are from the public benchmark WMT En→Fr (newstest2014) data set. The right-hand column shows the human ratings on a scale of 0 (complete nonsense) to 6 (perfect translation). We disagree with some of the human ratings, e.g., the translation "*Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière*" contains grammatical mistakes and changes semantics, and is still scored 6. We present it to illustrate the potential problems of the scoring process.

Source	"The reason Boeing are doing this is to cram more seats in to make their plane more competitive with our products," said Kevin Keniston, head of passenger comfort at Europe's Airbus.	
PBMT	"La raison pour laquelle Boeing sont en train de faire, c'est de concentrer davantage de sièges pour prendre leur avion plus compétitive avec nos produits", a déclaré Kevin M. Keniston, chef du confort des passagers de l'Airbus de l'Europe.	3.0
GNMT	"La raison pour laquelle Boeing fait cela est de créer plus de sièges pour rendre son avion plus compétitif avec nos produits", a déclaré Kevin Keniston, chef du confort des passagers chez Airbus.	6.0
Human	"Boeing fait ça pour pouvoir caser plus de sièges et rendre ses avions plus compétitifs par rapport à nos produits", a déclaré Kevin Keniston, directeur de Confort Passager chez l'avionneur européen Airbus.	6.0

Tiré de Wu et al., *Google's Neural Machine Translation System : Bridging the Gap between Human and Machine Translation*, arXiv:1609.08144v1, 2016.

Exemples de traduction avec GNMT (2/3)

Source	When asked about this, an official of the American administration replied: "The United States is not conducting electronic surveillance aimed at offices of the World Bank and IMF in Washington."	
PBMT	Interrogé à ce sujet, un responsable de l'administration américaine a répondu : "Les Etats-Unis n'est pas effectuer une surveillance électronique destiné aux bureaux de la Banque mondiale et du FMI à Washington".	3.0
GNMT	Interrogé à ce sujet, un fonctionnaire de l'administration américaine a répondu: "Les États-Unis n'effectuent pas de surveillance électronique à l'intention des bureaux de la Banque mondiale et du FMI à Washington".	6.0
Human	Interrogé sur le sujet, un responsable de l'administration américaine a répondu: "les Etats-Unis ne mènent pas de surveillance électronique visant les sièges de la Banque mondiale et du FMI à Washington".	6.0
Source	Martin told CNN that he asked Daley whether his then-boss knew about the potential shuffle.	
PBMT	Martin a déclaré à CNN qu'il a demandé Daley si son patron de l'époque connaissaient le potentiel remaniement ministériel.	2.0
GNMT	Martin a dit à CNN qu'il avait demandé à Daley si son patron d'alors était au courant du remaniement potentiel.	6.0
Human	Martin a dit sur CNN qu'il avait demandé à Daley si son patron d'alors était au courant du remaniement éventuel.	5.0

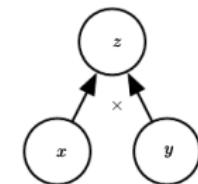
Tiré de Wu et al., *Google's Neural Machine Translation System : Bridging the Gap between Human and Machine Translation*, arXiv:1609.08144v1, 2016.

Exemples de traduction avec GNMT (3/3)

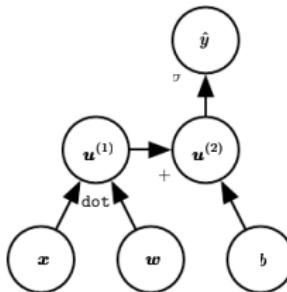
Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0
Source	Analysts believe the country is unlikely to slide back into full-blown conflict, but recent events have unnerved foreign investors and locals.	
PBMT	Les analystes estiment que le pays a peu de chances de retomber dans un conflit total, mais les événements récents ont inquiété les investisseurs étrangers et locaux.	5.0
GNMT	Selon les analystes, il est peu probable que le pays retombe dans un conflit généralisé, mais les événements récents ont attiré des investisseurs étrangers et des habitants locaux.	2.0
Human	Les analystes pensent que le pays ne devrait pas retomber dans un conflit ouvert, mais les récents évènements ont ébranlé les investisseurs étrangers et la population locale.	5.0

Tiré de Wu et al., *Google's Neural Machine Translation System : Bridging the Gap between Human and Machine Translation*, arXiv:1609.08144v1, 2016.

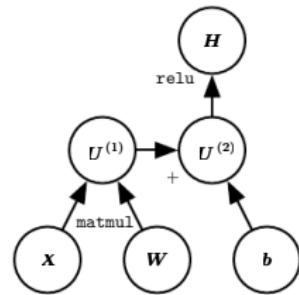
Graphes computationnels



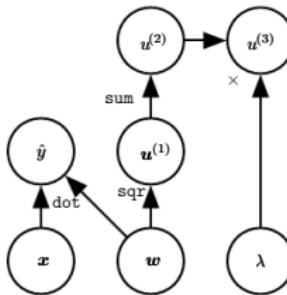
(a)



(b)



(c)

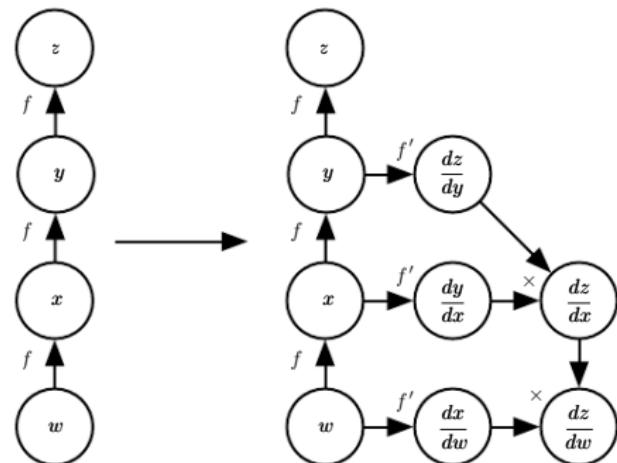


(d)

- Graphe computationnel : représenter les opérations mathématiques d'un réseau dans graphe
- Capture l'ordre et la nature des opérations

Tiré de *I. Goodfellow, Y. Bengio et A. Courville, Deep Learning, MIT Press, 2016.*

Gradient automatique



Tiré de I. Goodfellow, Y. Bengio et A. Courville, *Deep Learning*, MIT Press, 2016.

- Gradient automatique : calculer les gradients **analytiques** sur l'ensemble du réseau automatiquement, via les graphes computationnels
- Permet de définir des topologies complexes et hétérogènes de réseau sans devoir faire les dérivées analytiques manuellement !
- Permet également d'optimiser les traitements sur l'architecture visée (ex. GPU)

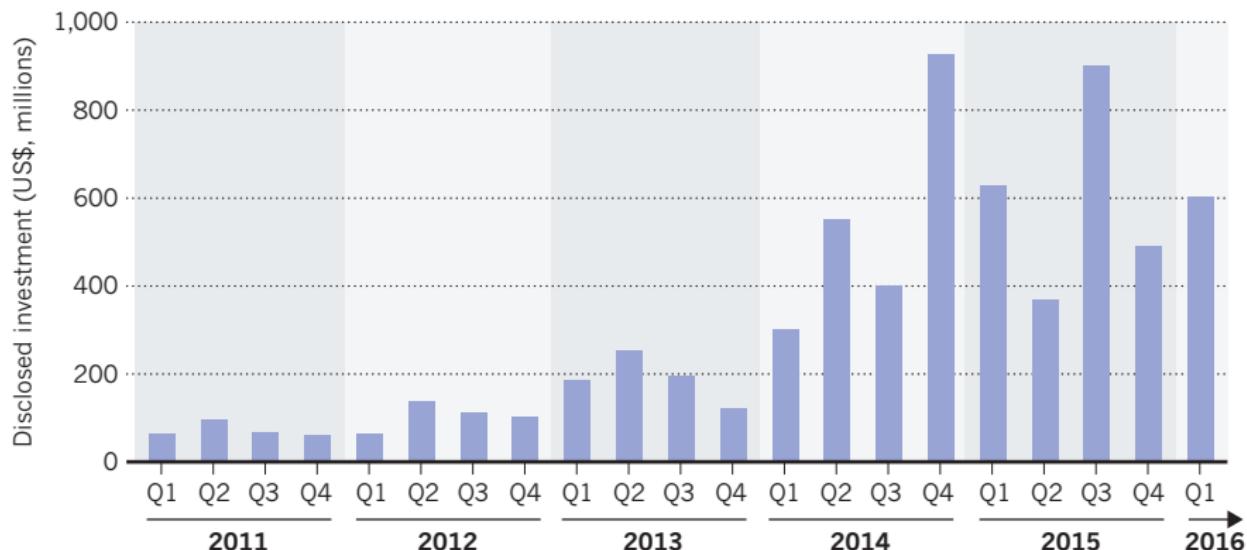
Outils pour l'apprentissage profond

- Theano (Université de Montréal) :
<http://deeplearning.net/software/theano/>
 - ▶ Librairie Python pour calcul matriciel efficace, incluant gradient automatique de graphes computationnel
 - ▶ Optimise traitement par compilation dynamique de code C et exploitation de GPU
 - ▶ Performant mais assez bas niveau, librairies disponibles offrant abstractions de plus haut niveau pour faire de l'apprentissage profond (Lasagne, Keras)
- TensorFlow (Google) : <https://www.tensorflow.org/>
 - ▶ Lancé en novembre 2015, adoption massive par la communauté
 - ▶ Code en C++, avec interface d'utilisation en Python
 - ▶ Entièrement organisé autour de graphes computationnels
- Torch (Collobert et collaborateurs) : <http://torch.ch/>
 - ▶ Programmé en C++, interface en Lua
 - ▶ Plus ancien que Theano et TensorFlow, ajout récent de gradient automatique
 - ▶ Grande versatilité, mais appréciation variable du Lua

Investissement dans l'intelligence artificielle

ON THE RISE

Investment in technologies that use artificial intelligence has climbed in recent years.



Tiré de K. Crawford et R. Calo, *There is a blind spot in AI research*, Nature, vol. 538, 20 octobre 2016.

Montréal et le Québec : nouvelle Silicon Valley de l'IA ?

- L'apprentissage profond a été développé en bonne partie au Canada (Hinton à Toronto, Bengio à Montréal)
 - ▶ Plusieurs des grands joueurs sont maintenant avec les Google, Facebook et cie
 - ▶ Plus grands groupes de recherche en apprentissage profond à Montréal
- Contrecarrer l'exode des cerveaux
 - ▶ Subvention Apogée à Campus Montréal : 98 M\$ sur 7 ans sur l'IA
 - ▶ Incubation d'entreprises dans le domaine (Element AI)
 - ▶ Bureau satellite de Google Brain s'implante à Montréal dans les prochains mois
- Début d'une nouvelle révolution industrielle, par l'automatisation des tâches cognitives ?

Références

-  Yann LeCun, Yoshua Bengio et Geoffrey Hinton. *Deep learning*. Nature, vol. 521, pages 436–444, 2015.
<http://dx.doi.org/10.1038/nature14539>
-  Ian Goodfellow, Yoshua Bengio et Aaron Courville. “Deep Learning”, MIT Press, 2016. <http://www.deeplearningbook.org/>
-  Yoshua Bengio. *Learning deep architectures for AI. Foundations and trends in Machine Learning*, 2, no. 1, pages 1–127, 2009. http://www.iro.umontreal.ca/~bengioy/papers/ftml_book.pdf
-  Geoffrey Hinton, Yoshua Bengio et Yann LeCun, *Deep Learning NIPS'15 Tutorial*, 2015. <http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf>