Deep Learning[1]

École Centrale de Marseille
38 Rue Juliot Curie
13013 Marseille

Laura FRANKE
Pauline DAME
Étienne GAUTIER
Guilherme MEIRELLES BODIN DE MORAES

[1] "Deep learning : Nature : Nature Research." 28 Mai. 2015, http://www.nature.com/nature/journal/v521/n7553/abs/nature14539.html.

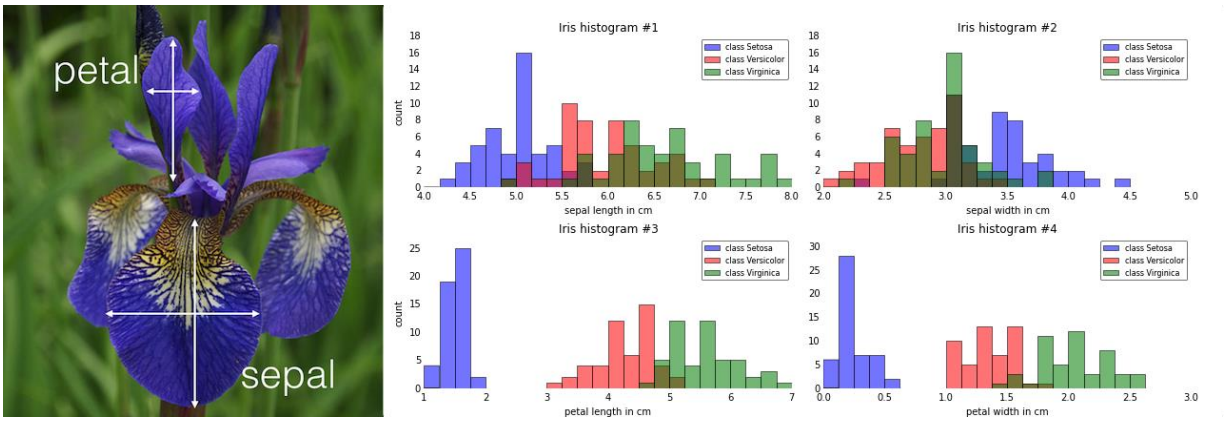
Introduction

Dans notre société, le *Deep Learning* (DL) et le *Machine Learning* (ML) trouvent de plus en plus de champs d'application. Avant toute autre chose, il est nécessaire de distinguer le DL du ML. En effet, en pratique le ML correspond à l'implémentation des différents algorithmes d'*Intelligence Artificielle* (IA) dans le but de créer une machine capable de prendre des décisions par elle-même. Les capacités des méthodes de ML conventionnelles à traiter des données brutes ont été longtemps limitées de par la complexité et l'expertise nécessaire à leur mise en œuvre. L'apprentissage de représentations (ou *Representation learning*) est une sous-méthode du ML qui permet d'extraire des caractéristiques utiles ou des représentations à partir des données brutes. Selon LeCun, Bengio, Hinton des méthodes de DL sont des méthodes de *Representation learning* avec plusieurs niveaux de représentation. Ils sont générés avec des modules non-linéaire qui transforment des données brutes dans une représentation abstraite. Dans les niveaux hauts les aspects représentés sont ceux qui sont importants pour distinguer l'image ou l'extrait de musique. Dans tous les cas le point plus important est que ces features important pour la distinction ne sont pas donnés par une personne, mais découverts et appris par la machine.

Dans notre rapport, on décrit les différentes méthodes de DL et leurs champs d'application.

Supervised Learning

La forme plus commune de ML est *supervised learning* (fr.: apprentissage supervisé). Généralement, on donne des par exemple des images avec des chats a une machine et cette machine qui va générer un pourcentage pour chaque résultat possibles (chat, chien, humain, gâteaux etc.). Les haute percentages correspondent alors au résultat correct, pour arriver à ce résultat la machine doit 'réviser'. Pour ça, on essaie de minimiser la distance, aussi appelé erreur, entre le résultat obtenu et les résultats désirés. Après, la machine règle ses paramètres internes, aussi appelés *weights* (fr.: poids) pour la réduire.



Exemple pour des iris. Dépendant des différentes tailles, l'algorithme trouve le type d'iris[2].

Pour trouver le meilleur ajustement, l'algorithme d'apprentissage calcule un vecteur gradient pour indiquer les changements d'erreur avec le changement des *weights*. Le but est de trouver

le minimum où l'erreur de sortie est en moyenne le plus petite. Une méthode suivant utilisé est le *Stochastic Gradient Descent* (SGD, fr.: algorithme du gradient stochastique). À plusieurs reprises, l'algorithme prend plusieurs vecteurs d'entrées, calculer les résultats, erreurs et gradient et modifie les *weights*. Ca c'est fait jusqu'à la moyenne de la fonction objective atteigne un minimum. Après cet entraînement, la performance est mesurée sur des différents dispositifs de test. Ce processus est un exemple pour un classifieur linéaire.

Un problème général est que les méthodes linéaires séparent des régions plat par une hyperplan. Pour les opérations plus complexes, p.e. La reconnaissance des visages, les variables d'entrées comme la position de visage pourraient changer, mais l'algorithmes doit toujours détecter des caractéristiques fines.

Backpropagation to train multilayer architectures

Les méthodes de rétropropagation ont été maîtrisés dans les années 1980 et leur objectif est de simplifier computationnellement les calculs pour trouver le vecteur de poids qui minimise l'erreur de sortie de l'algorithme. Le grand avantage c'est qu'en séparant les calculs par couches on peut toujours essayer de faire des calculs plus simples. L'ensemble de tous ces calculs simples peut devenir un outil très puissant de calcul.

L'idée de la rétropropagation pour trouver ces points optimums est de calculer le gradient des fonctions par rapports aux poids de chaque couche, c'est-à-dire, décrire l'importance de chaque couche pour l'optimisation et la minimiser. Ce calcul est en fait une application de la loi de dérivation en chaîne, le point de vue clé est que la dérivée de la fonction par rapport à l'entrée d'une couche peut être calculée en travaillant en arrière à partir du gradient par rapport à la sortie de cette même couche. L'équation de rétropropagation peut être appliquée à plusieurs fois pour propager des gradients à travers tous les couches. Une fois ces gradients calculés, il est simple de calculer les gradients par rapport aux poids de chaque module ainsi permettant trouver nos points optimums.

Convolutional Neural Networks

Les réseaux convolutionnels *(ConvNets)* sont un outil désigné pour traiter des masses de données qui sont en forme de plusieurs vecteurs, par exemple, une image est un ensemble de pixels qui peut être exprimé en trois vecteurs *(position horizontale et verticale du pixel et intensité en gris du pixel)* on peut aussi l'exprimer avec plus de vecteurs, si on veut la couleur en RGB on doit ajouter l'intensité en rouge, vert et bleu.

On peut repérer 4 idées clés pour les *ConvNets:* connections locales, poids partagés, pooling et l'utilisation de plusieurs couches. Organisant les couches en *convolutional* et en *pooling*.

Dans les couches de convolution le but est de reconnaître des motifs/patrons locaux et dans les couches de *pooling* le but est de réunir tous les motifs/patrons locaux dans une seule image

pour essayer de donner une signification sémantique, par exemple, réunir le visage du chien utilisant les motifs locaux (yeux, bouche, nez et oreilles).

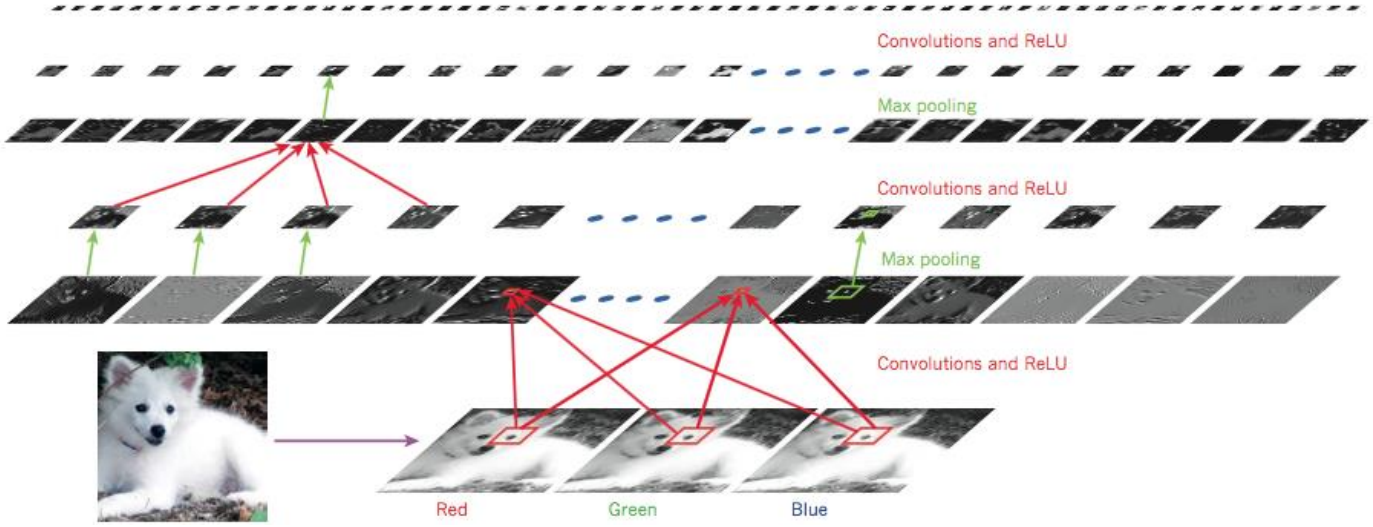


Image 2: Exemple de ConvNet pour des images.[2]

Image understanding with deep convolutional networks

Depuis les années 2000, les ConvNets (réseaux convolutionnels) sont utilisés dans la reconnaissance d'image, avec des applications pour la conduite autonome, etc. Mais c'est en 2012 qu'ils ont atteint un nouveau stade en classifiant un set d'images vers un millier de classes avec un taux d'erreur deux fois inférieur à la compétition. Ces progrès ont été permis par l'utilisations de GPUs, des ReLu (unité rectifié linéaires) et d'apprentissage sur des images générées par ordinateur qui forment des petites variations autour des images réelles. Ces résultats font des CNN l'approche majoritaire pour les algorithmes de détection et de classification.

Les CNN récents combinent une dizaine de couches de ReLu et des centaines de millions de poids et des milliards de connections. La phase d'apprentissage de ces méthodes de ML prennent aujourd'hui seulement quelques heures contre des semaines il y a seulement 2 ans. Des produits utilisant ces techniques sont déjà sur le marché et de nombreuses autres applications devraient voir le jour dans des systèmes embarqués grâce à des innovation sur des puces optimisés pour le machine learning.

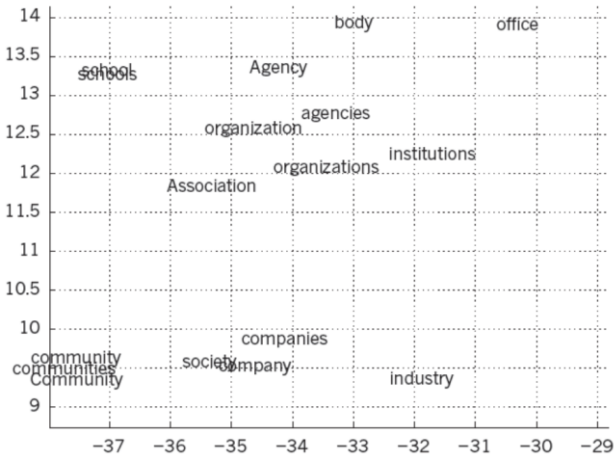
Distributed representation and language processing

Les réseaux de neurones ont deux avantages importants sur les techniques classiques qui n'utilisent pas "distributed representation". Le premier est d'être capable de généralisation au-

[2] "Deep learning : Nature : Nature Research." 28 Mai. 2015, http://www.nature.com/nature/journal/v521/n7553/abs/nature14539.html.

delà des cas vus lors de la phase d'apprentissage. Le deuxième est l'utilisation de nombreuses couches qui offre de nouvelles possibilités.

Les couches cachées d'un réseau de neurones apprennent à représenter les données d'entrées de telles sortes que leur traitement soit le plus simple possible. Dans le cas du traitement du langage naturel dans le but de déterminer le mot suivant dans une phrase, le mot courant ainsi que tous les mots précédents sont convertis en un vecteur donc chaque composante vaut soit 0 soit 1. Dans la première couche, à chaque mot est associé un vecteur unique. Au fur et à mesure du traitement qui prend en compte les mots précédents, ce vecteur est modifié jusqu'à ce les différents vecteurs du réseau représentent la probabilité de chaque mot du dictionnaire d'apparaître après le mot courant. Les caractéristiques sémantiques du texte sont donc prises en compte par le réseau de neurones sous la forme de règles sans pourtant avoir eu besoin d'être explicitées à aucun moment. Ainsi deux mots proches vont avoir des vecteurs assez semblables. Ce type de problèmes ont vu deux conceptions de la représentation s'affronter. La conception classique, basée sur la logique et non sur les réseaux de neurones, consistait à associer des fréquences à un mot ou à une courte série de mots et ne pouvait pas généraliser au-delà du vocabulaire qui lui était déjà connu. De plus, ce traitement devenait rapidement lourd dans le cas d'un grand vocabulaire.



Projection en 2D d'une représentation distribuée

Recurrent Neural Network

Les RNN sont particulièrement adaptés aux données séquentielles tels que le texte ou la voix. Ils traitent l'entrée pas à pas (mot à mot pour un texte) et font évoluer un vecteur d'état qui représente indirectement les entrées précédentes.

Ces systèmes sont entraînés avec une technique de rétropropagation du gradient, le problème qui se pose est d'éviter que ce gradient ne finisse par augmenter ou diminuer trop fortement au fil des étapes. Une fois ces limitations dépassées, ces systèmes obtiennent de très bonnes performances pour deviner la prochaine lettre ou le prochain mot d'une phrase. Ils peuvent aussi réaliser des tâches plus complexes: après avoir lu une phrase en anglais, le vecteur d'état représente le sens de la phrase (grâce à la représentation distribuée); et un 'décodeur' en français peut à partir de ce vecteur construire mot à mot une phrase en français au sens le plus proche possible. Cette méthode de traduction donne de très bon résultat qui remettent en question l'utilité des règles d'inférence (système expert qui a une connaissance de la grammaire de la langue) utilisées jusqu'alors.

Un des principaux défauts des RNN est leur capacité de mémorisation très limitée sur le long terme. Une des pistes explorées pour corriger cela est de d'introduire une couche

supplémentaire de LSTM (mémoire à long court-terme) qui a pour seul but de garder en mémoire une entrée pendant plusieurs itérations et dans certain cas de la remplacer par l'entrée.

Une autre piste envisagée est la machine de Turing neuronal (NTM) qui consiste à interagir (écrire et lire) avec une mémoire. Les NTM ont accumulé quelques succès sur la compréhension de textes simples, ou la réalisation d'un algorithme de tri seulement basé sur l'exemple.

Conclusion

Si les méthodes d'apprentissage non supervisées ont permis de ranimer l'intérêt autour du deep learning, elles ont rapidement été éclipsées par les méthodes supervisées. Cependant, elles sont primordiales dans la mesure où la plupart des systèmes réels sont non supervisés.

En ce qui concerne les RNN et les réseaux convolutionnels, leur application dans le domaine des systèmes de vision et du traitement du langage naturel obtiennent d'excellents résultats dont les applications seront sans nul doute très importantes dans les années à venir. Les avancées les plus importantes se situent désormais du côté des systèmes mêlant l'apprentissage de représentations et les raisonnements complexes afin de se substituer aux traitements basés sur l'utilisation de règles et de la symbolique.