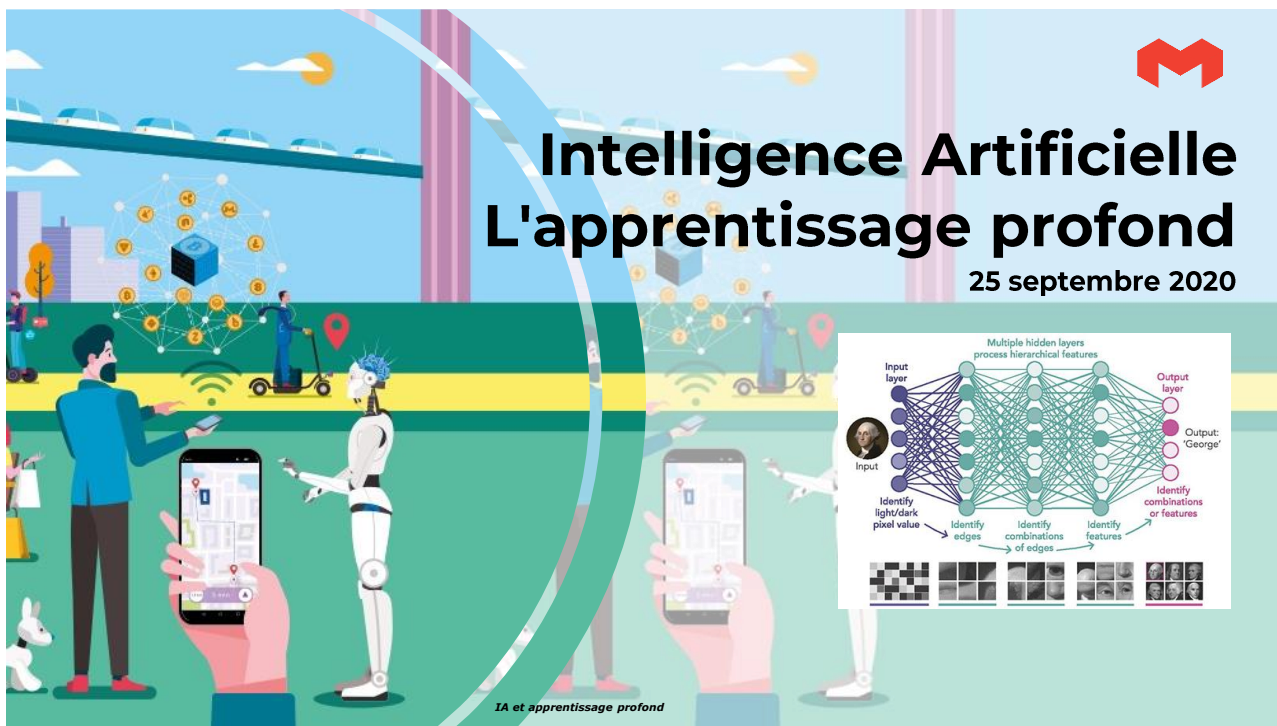




Intelligence Artificielle L'apprentissage profond

25 septembre 2020



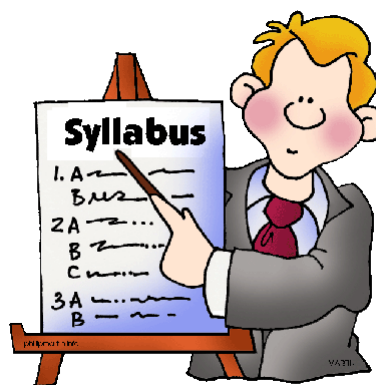
IA et apprentissage profond

L'apprentissage automatique c'est 8,81 G\$ en 2022 (dix fois plus qu'en 2016)... ce n'est pas colossal et il ne faut pas se précipiter ("deep learning" est plus ambitieux avec plus de 18 G\$ en 2023 (*))

Sommaire

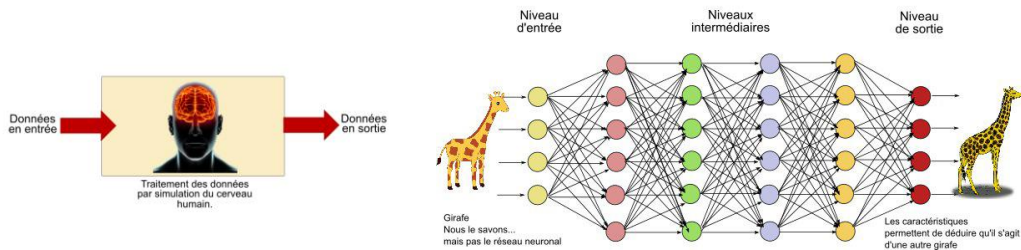
IA et apprentissage profond

- ❖ Les réseaux neuronaux
- ❖ Principes de base d'un réseau neuronal
- ❖ Fonctions de transfert
- ❖ Comprendre la pondération
- ❖ L'apprentissage
- ❖ Les fonctions de transfert
- ❖ Exemples de réseaux neuronaux
- ❖ La mécanique du deep learning
- ❖ Focus sur les réseaux convolutifs
- ❖ Les réseaux de convolution prêts à l'emploi
- ❖ Les API et langages
- ❖ Les applications





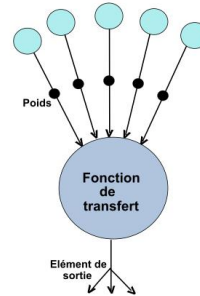
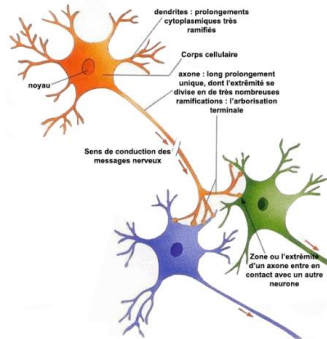
Principes de base d'un réseau neuronal



- Un réseau neuronal est un algorithme qui prend des données en entrée, qu'il traite, pour produire des résultats en sortie
- L'algorithme est fondé sur une simulation (lointaine) du fonctionnement d'un cerveau humain, avec ses neurones, synapses et dendrites
- Il pourra s'agir d'un traitement très simple, avec une seule couche (perceptron) ou plusieurs couches, avec propagation des résultats intermédiaires d'une couche à l'autre, voire de très nombreuses couches (deep learning)
- L'architecture d'un réseau neuronal est faite pour faciliter l'apprentissage, mais peut être très diversifiée, en fonction du type d'apprentissage

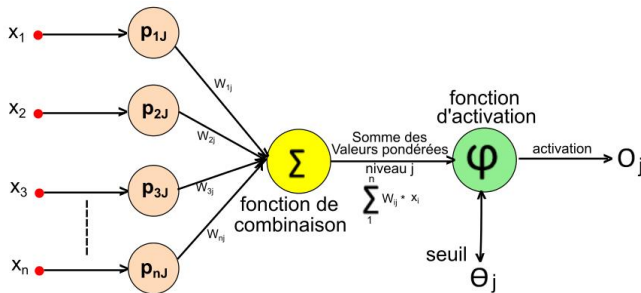
Les réseaux neuronaux

- ❖ Les réseaux neuronaux sont des plates-formes matérielles, sur lesquelles sont implémentées des algorithmes, qui simulent le comportement d'un cerveau humain (neurones et synapses), par un système de couches de processeurs (neurones), reliées par des connexions réseaux (synapses et dendrites).
- ❖ Un neurone biologique reçoit des signaux transmis par d'autres neurones (interaction dendrites-synapses), les stimuli.
- ❖ Le neurone analyse ces signaux en les sommant. Si le résultat obtenu est supérieur au seuil d'activation(excitabilité), il envoie un signal (potentiel d'action) le long de son axone vers d'autres neurones biologiques.
- ❖ Ce signal peut renforcer ou diminuer l'activité des neurones qui le reçoivent, selon que les synapses sont excitatrices ou inhibitrices.
- ❖ Un neurone biologique est connecté à plusieurs milliers d'autres neurones.
- ❖ Un neurone formel ou artificiel simule le fonctionnement d'un neurone biologique :
 - ❖ Les synapses sont modélisées par des poids
 - ❖ Le corps cellulaire est modélisé par une fonction de transfert, dite fonction d'activation
 - ❖ L'axone est l'élément de sortie.



IA et apprentissage profond

Les principes (de base) d'un réseau neuronal



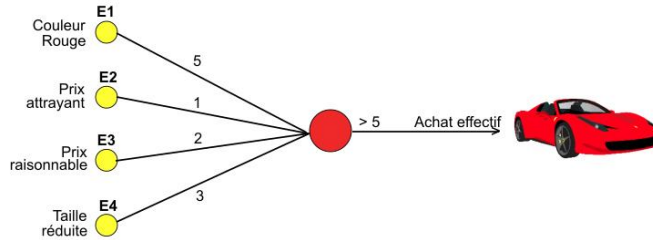
L'objectif d'un réseau neuronal est de trouver les valeurs de pondération $\sum(P_{ij} * X_i)$ et un biais B pour faire le lien entre les entrées et la sortie, connue ou pas :

$$f(X) = \sum(P_{ij} * X_i) + B$$

- ❖ Les poids sont notés P_{ij} , qui vont du neurone i au neurone j : on voit bien que chaque poids témoigne de l'importance de l'entrée sur le résultat
- ❖ Chaque poids pondère le stimuli émis par le neurone source X_i
- ❖ Le neurone j fait la somme des stimuli pondérés : $\sum(P_{ij} * X_i)$
- ❖ S'il y a n neurones sources i liés au neurone j , celui-ci va calculer : $\sum(P_{ij} * X_i)$, avec i de 1 à n
- ❖ C'est cette somme pondérée que le neurone j va traiter par une fonction de transfert
- ❖ La fonction de transfert (activation, seuillage, combinaison) est essentielle : c'est elle qui va déclencher un stimuli vers le niveau $j+1$ si la somme des pondérée niveau j est supérieure au seuil d'excitation
- ❖ Il en existe un grand nombre : seuil, linéaire, linéaire saturée, linéaire positive, sigmoïde, tangente hyperbolique...
- ❖ Rétro-propagation : si le résultat n'est pas celui attendu et ne se situe pas dans une marge d'erreur acceptable, les poids et biais sont mis à jour à partir du résultat obtenu : ce processus récursif est dit de rétro-propagation.
- ❖ Les neurones formels (artificiels) organisés en couches successives deviennent des réseaux de neurones

IA et apprentissage profond

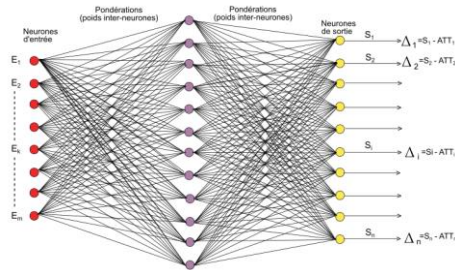
Comprendre la pondération



Valeurs d'entrée : 0 ou 1
 Pondérations (poids) : 5 pour favoriser la couleur rouge, 1, 2 et 3 pour les autres données d'entrée
 Fonction d'activation : somme supérieure à 5, sortie valorisée à 1, achat effectif, en-dessous non achat
 Si la sortie ne correspond pas aux souhaits de l'utilisateur, pour favoriser le prix, on modifie la pondération. On fonctionne par tâtonnements, jusqu'à ce que le perceptron réponde correctement

- ❖ La pondération permet de donner plus ou moins d'importance aux entrées incidentes
- ❖ Par tâtonnements ou de manière automatique, en fonction d'un modèle et des informations rétropropagées, on se rapproche de la cible par des itérations successives

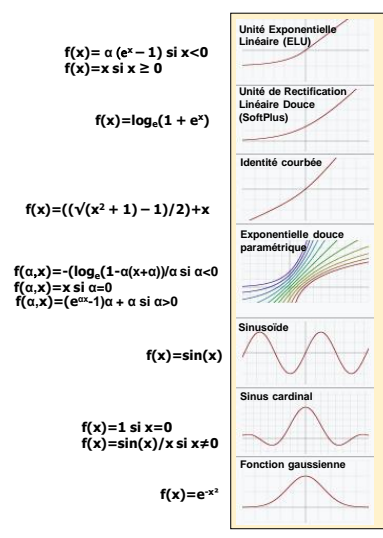
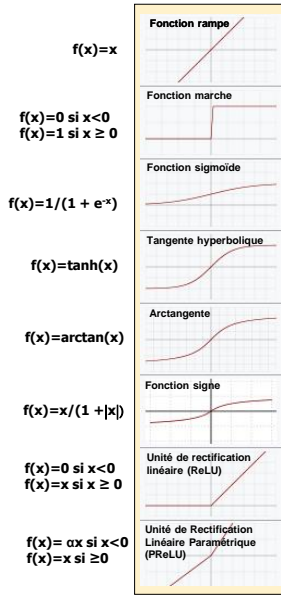
Le concept d'apprentissage



- ❖ Tout est là : faire converger le réseau, après un certain nombre d'itérations, vers ce que l'on attend qu'il produise (en mode supervisé, on connaît le résultat).
- ❖ Le principe commun est de comparer le résultat obtenu pour chaque neurone de sortie par rapport à ce que l'on attend et de constater l'écart (erreur)
- ❖ Pour se rapprocher de la cible, on changera la pondération de tout ou partie des liens (poids) inter-neurones.
- ❖ Il existe de nombreuses solutions, plus ou moins adaptées à la nature de l'application et chronophages. De même, il existe de nombreuses manières de mesurer les écarts : définition de l'erreur et la manière de la corriger.
- ❖ Plus il y aura de niveaux intermédiaires cachés, plus l'implémentation de l'algorithme de correction sera difficile à mettre en œuvre et prendra du temps en exploitation : c'est l'un des problèmes du "deep learning" dont l'apprentissage peut prendre plusieurs heures, voire plusieurs jours.

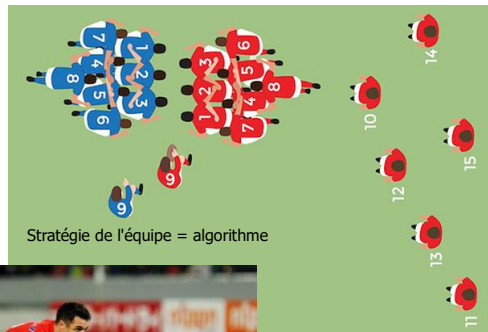
Les fonctions de transfert (activation)

- ❖ Chaque neurone applique une fonction de transfert, qui est la même pour tous les neurones appartenant à une même couche.
- ❖ Cette fonction mathématique est très importante, qui permet de faire passer l'information transmise par une couche i à une couche $i+1$, de façon discriminée pour chacun des neurones, en appliquant une transformation bien précise, qui dépendra de la nature du traitement que l'on voudra effectuer et de la finalité du réseau de neurones.
- ❖ On transforme une valeur en entrée en une valeur de sortie, qui elle-même va devenir la valeur d'entrée des neurones de niveau plus élevé.
- ❖ Les fonctions d'activation auront des propriétés mathématiques bien précises, parmi lesquelles la monotonie, l'étendue, la continuité et surtout la simplicité de dérivabilité, qui aura un impact direct sur la lourdeur (ou non) de l'algorithme d'apprentissage.



Petit cours de rugby

- ❖ Conceptuellement un réseau neuronal est un mélange de simulation du cerveau humain et de rugby...
- ❖ Les valeurs d'entrée : la disposition de l'équipe, ses forces et faiblesses
- ❖ Progression du ballon : vers l'avant s'il n'y a pas d'obstacle
- ❖ Retour arrière, en cas de blocage
- ❖ Stratégie de progression : les fonctions de transfert
- ❖ Progression en N tentatives : les itérations du réseau neuronal
- ❖ Essai : le réseau aboutit à la bonne conclusion
- ❖ Pénalité et transformation : bias du réseau



Bias



Fonction de transfert

Des exemples de réseaux neuronaux

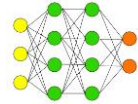
Propagation avant ("feed-forward")

- ❖ C'est le mode le plus simple, l'information n'est diffusée que dans un seul sens, il n'y a pas de cycles, ni de boucles et les neurones ne communiquent ni avec les nœuds précédents, ni avec d'autres neurones du même niveau (ex du perceptron, avec une entrée et une sortie). Un seul niveau caché, entraîné généralement en "rétropropagation", les paramètres de poids et de biais, passés à la fonction d'activation étant mis à jour de manière itérative, jusqu'à obtenir un résultat acceptable.



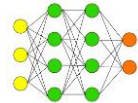
Le réseau MLP ("Multi-Layer Perceptron")

- ❖ Comporte une couche d'entrée, une couche de sortie et une ou plusieurs couches intermédiaires cachées. Avec n couches distinctes, le MLP comporte $n-1$ matrices de poids.
- ❖ C'est une organisation adaptée au concept d'approximateur universel de fonctions, car il suffit d'ajouter le nombre de neurones adéquat dans une couche cachée pour approximer n'importe quelle fonction non linéaire.



DFF ("Deep Feed Forward")

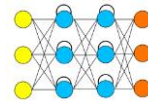
- ❖ Famille des FF, mais avec plusieurs couches cachées. Traite surtout le problème de la rétropropagation des erreurs qui nécessite plusieurs itérations. Ce type de réseau a été à l'origine du "deep learning" dans les années 90.



Des exemples de réseaux neuronaux

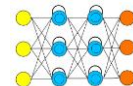
Réseaux de neurones récurrents RNN ("Recurrent Neural Network")

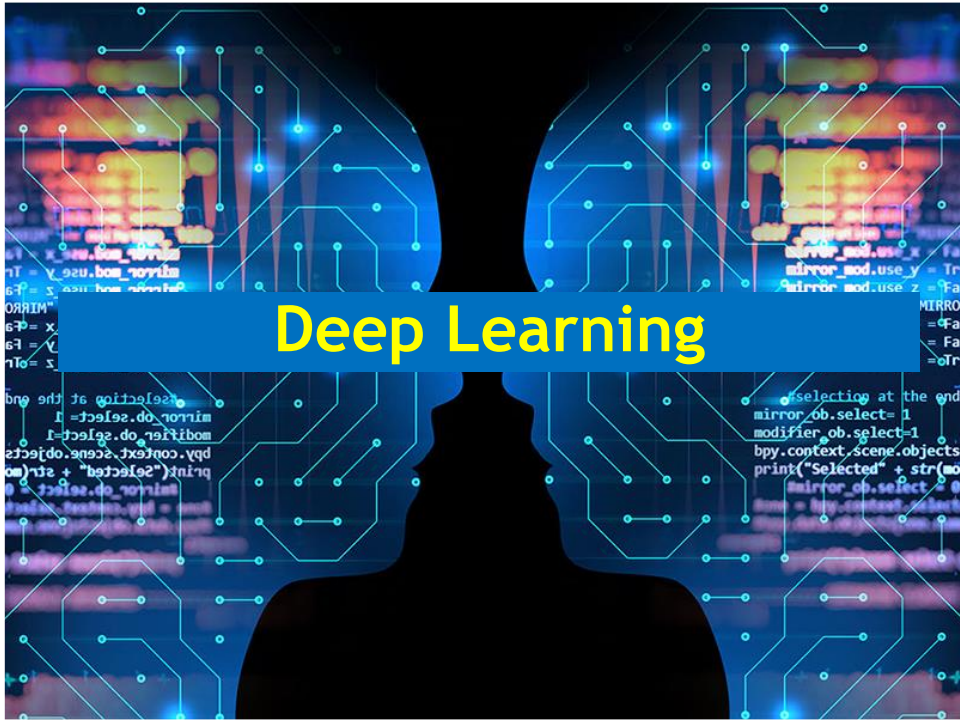
- ❖ Famille FF, mais chaque neurone des couches cachées reçoit ses propres sorties au bout d'un délai fixe, d'une ou plusieurs itérations. Très intéressant quand le contexte joue un rôle important, quand des décisions d'itérations peuvent influencer les décisions courantes. Exemple de l'analyse des textes quand un mot ne peut être analysé qu'en fonction du contexte des mots ou des phrases précédentes ou la prévision de données "timeseries", type évolution de valeurs mobilières.
- ❖ Différents modèles :
 - ❖ Hopfield : une seule couche (à la fois entrée et sortie) où toutes les unités sont interconnectées
 - ❖ Réseaux récurrents à couches : entrée, cachées, sortie. Propagation avant et rétro propagation de l'erreur lors de l'apprentissage. Ex : le modèle SRN ("Simple Recurrent Network"), dit aussi modèle de Elman.



LSTM ("Long/Short Term Memory")

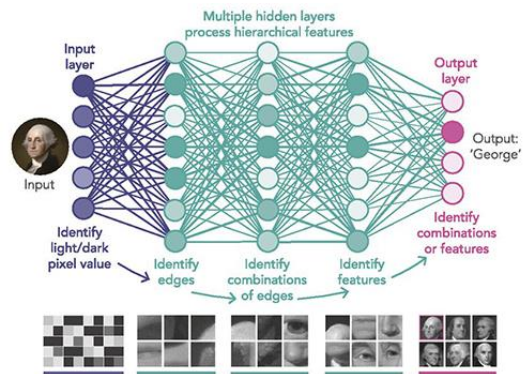
- ❖ Comporte des neurones à mémoire, qui peuvent traiter des données après un certain décalage. Par ex : traiter des textes en gardant en mémoire des dizaines de mots déjà utilisés, ou dans une vidéo, un élément qui est apparu plusieurs trames auparavant.
- ❖ Très utilisé en reconnaissance de la parole et de l'écriture.



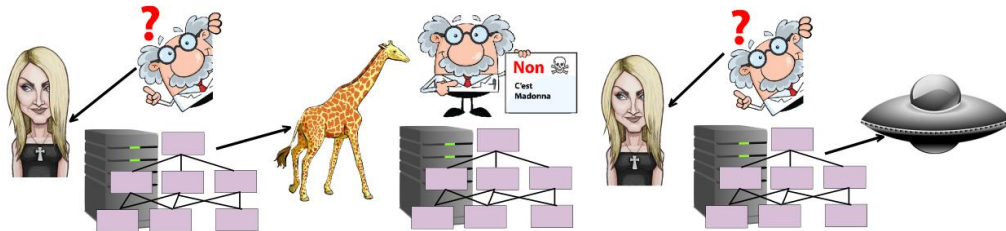


Deep Learning (apprentissage profond)

- ❖ Le « deep learning » ou apprentissage profond, est un ensemble de méthodes qui bénéficient des progrès des réseaux neuronaux, pour reconnaître des objets, images, images dans les images, sons, phrases dans une communication téléphonique, etc.
- ❖ C'est une branche du "Machine Learning", avec un grand nombre de couches "cachées", d'où l'idée de "deep" (100 à 200 couches contre quelques couches pour le "Machine Learning")
- ❖ Extraction automatique des caractéristiques d'entrée : l'algorithme analyse les données soumises
- ❖ Plusieurs différences avec le machine learning
 - ❖ Le gros volume de données traitées labellisées, pour l'entraînement
 - ❖ Nécessite une grande puissance de calcul : GPU parallèles, clusters et Cloud
- ❖ L'un des réseaux profonds les plus répandus est le mode CNN ou ConvNet, réseaux de neurones à convolution, des réseaux multi-niveaux
- ❖ Toujours deux phases :
 - ❖ Dans la première, le système cherche à appliquer une transformation non linéaire aux valeurs d'entrée et à créer un modèle de sortie, c'est la phase d'**éducation** ou d'**entraînement**, qui s'effectue sur des données labellisées
 - ❖ Les couches cachées sont dédiées à un traitement spécifique : par ex, détection des caractéristiques simples pour les premières couches (contours), combinaison de ces contours (edges) par les couches suivantes, pour aboutir à "quelque chose" de crédible
 - ❖ L'algorithme d'apprentissage va modifier les poids de manière à se rapprocher des prévisions et reconnaître l'objet présenté
 - ❖ La seconde phase, de **découverte**, consiste à améliorer le modèle (cible) avec une méthode mathématique (dite derivative) pour "coller" au plus près de la réalité. Elle s'effectue sur des données non labellisées.



La mécanique neuronale du deep learning

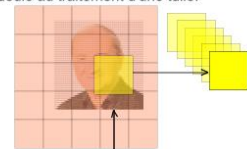


Focus sur les réseaux convolutifs

- ❖ CNN (Convolutional Neural Network) peut être considéré comme un empilement de perceptrons multicouches (*), chacun traitant une partie de l'information globale : reconnaissance d'images, vidéos, traitement naturel du langage.
- ❖ La convolution est un outil mathématique très utilisé en retouche d'image, car il en fait ressortir les caractéristiques (traits élémentaires...), mais aussi en interprétation climatique ou sismique
- ❖ L'image de départ est subdivisée en sous-images dont le traitement est assuré par un neurone artificiel
- ❖ Sur chaque sous-image, une fonction de convolution est appliquée à l'aide d'un filtre sur des échantillons de même taille qu'une matrice de filtrage, dont les valeurs évoluent avec l'apprentissage

- ❖ L'échantillon est caractérisé par :
 - ❖ Le kernel, la dimension de la matrice de filtrage
 - ❖ Le "slide", le nombre de pixels de décalage entre deux sous-ensembles d'un même échantillon
 - ❖ Le "padding" est l'adjonction de valeurs nulles autour de l'image, pour éviter les dépassements dans le décalage du filtrage
- ❖ L'apprentissage du réseau convolutif, c'est-à-dire l'affinement de son comportement applicatif, se fera ensuite sur trois critères :
 - ❖ Les matrices de filtres qui caractérisent les convolutions, dont les valeurs vont changer pour s'adapter, tout en gardant le même kernel et algorithme
 - ❖ Le biais : valeur ajoutée après la convolution à chaque pixel (le même pour un filtre donné)
 - ❖ Le poids entre les couples de neurones

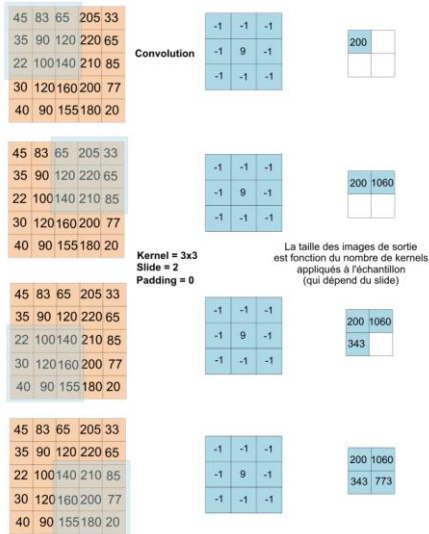
L'image à reconnaître est divisée en échantillons (16 x 16, 32 x 32...) ou "tuile". Un neurone artificiel est dédié au traitement d'une tuile.



Imager d'origine à analyser

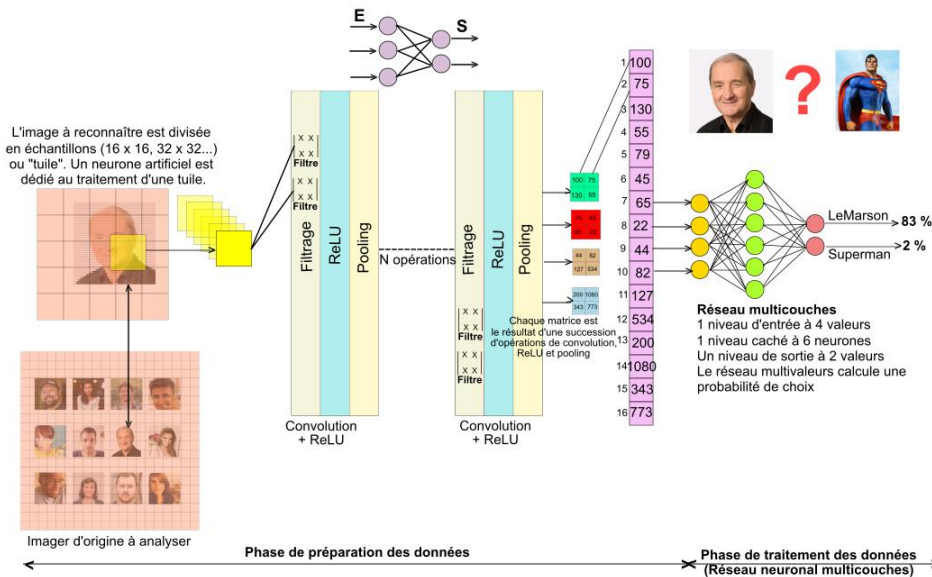
(*) : perceptron, le Réseau Neuronal Artificiel le plus simple

Réseau de convolution



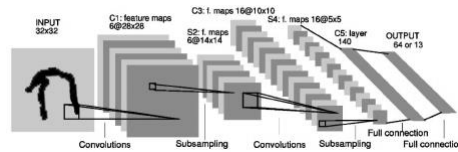
- ❖ La fonction de convolution s'applique aux sous-échantillons d'images
- ❖ Opération mathématique, qui va produire une matrice en sortie, généralement de taille réduite.
- ❖ Il existe plusieurs manières d'effectuer le calcul de convolution :
 - ❖ Convolution classique : chaque élément du kernel est multiplié par son équivalent dans le sous-échantillon, la valeur reportée étant la somme de ces calculs
 - ❖ La convolution "dilatée", identique à la précédente, mais on ne prend qu'un pixel sur X, le "dilation rate" étant le nombre de pixels à ignorer
 - ❖ La convolution transposée
 - ❖ "separable convolution"
- ❖ La convolution est associée à une fonction d'activation ReLU, qui se justifie pour augmenter les écarts entre deux caractéristiques d'images (suppression des valeurs négatives)
- ❖ Pourquoi : les opérations de convolution sont linéaires (additions, multiplications...), or les images ne sont pas linéaires, avec de gros écarts entre des pixels proches : il faut donc recréer cette non linéarité
- ❖ Le "pooling" est une opération qui revient à remplacer un carré de pixels (2x2, 3x3) par une seule valeur, ce qui diminue la taille de l'image et l'image est lissée
- ❖ Les types de "pooling" :
 - ❖ "max pooling" : on prend la valeur la plus élevée de la sélection
 - ❖ "mean pooling" : on prend la valeur moyenne de toute la sélection
 - ❖ "sum pooling" : on prend la somme des pixels
- ❖ Le "flattening" (mise à plat) revient à mettre bout à bout les images pour en faire un vecteur

Réseau de convolution

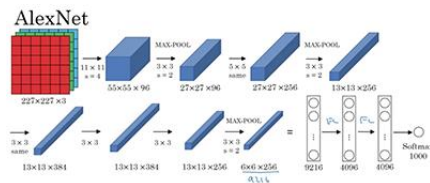


Les réseaux de convolution prêts à l'emploi

- ❖ Plusieurs CNN prêts à l'emploi sont proposés qui, après entraînement, peuvent être exploités pour reconnaître des objets
- ❖ Habituellement on les compare en appliquant le benchmark ILSVRC qui lui-même est fondé sur la base ImageNet et comporte plus de 14 millions d'objets classés en 20 000 catégories (ImageNet et ILSVRC sont libres de droits) et plus de 1 000 classes
- ❖ Le critère de comparaison est le "top-five error rate", le nombre de tests pour lesquels la solution exacte ne fait pas partie des cinq classes prédites par le réseau



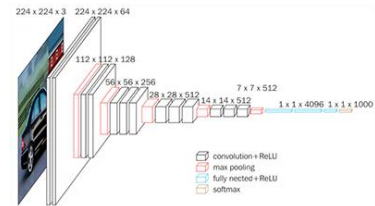
LeNet-5 de Yann LeCun date de 1998. C'est un pionnier, exploité au départ pour reconnaître les digits manuscrits sur les chèques, numérisés en images 32 x 32 pixels et niveaux de gris.



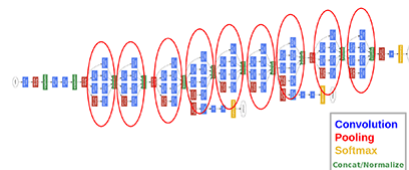
AlexNet (2012), conçu par le Supervision Group, a fait chuter le "top-five error rate" de 26 à 15 %. Ressemble à LeNet-5 mais plus profond et avec plus de filtres par niveaux. Fondé sur des découpages 11 x 11, 5x5, 3x3, 5 couches de convolutions, max pooling, dropout (technique de régularisation pour réduire le surapprentissage, l'activation ReLU, etc.

Les réseaux de convolution prêts à l'emploi

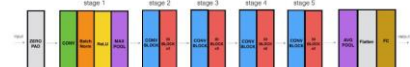
VGGNet (2014), a été développé par Simonyan et Zisserman. Comporte 16 niveaux de convolution, plus que AlexNet et 3 x 3 kernels. C'est le CNN le plus utilisé pour extraire les "features" des images à reconnaître. La pondération algorithmique de VGGNet est publique, d'où son usage dans de nombreux domaines. La difficulté est qu'il comporte 138 millions de paramètres...ce qui peut être "lourd" à prendre en charge
Points forts : Concept de bloc/modules, remplace les noyaux de grandes taille en plusieurs noyaux de petite taille. Diminution du nombre de paramètres.



GoogleNet (2014), à l'origine Inception V1, a atteint 6,67 % au "top-5 error rate". Proche d'un comportement humain. Conceptuellement dérivé de LeNet, mais avec la normalisation batch, la distorsion d'images et RMSProp (un algorithme de Geoffrey Hilton). Comporte 22 niveaux cachés mais avec "seulement" 4 millions de paramètres.



ResNet (Residual Network) a introduit en 2015 des concepts nouveaux : "skip connections" (les "gated units") et un meilleur usage de la normalisation batch. A atteint un taux de 3,57 %, supérieure à celui d'un être humain. 152 couches. Principale idée : introduction d'une connexion de raccourci, qui saute une ou plusieurs couches.



Deep Learning : les API

- ❖ La grande révolution du deep learning, tient à ce qu'il existe désormais une importante panoplie d'API dédiée, exploitables sans avoir nécessairement une connaissance fine des mécanismes internes de la technologie.
- ❖ Le trio sur lequel investir : Tensor Flow, Keras surtout, PyTorch
- ❖ Montée ne puissance des solutions Cloud : AWS, Google, IBM, Oracle, Microsoft, ExoScale, Alibaba... : ressources machines très importantes, API à jour, modèles pertinents implantés

| API | Logo | Ecrit en | Langages des API | Disposition CNN | Disposition RNN | Simplicité d'usage | Vitesse de traitement | Support parallèle GPU | Plate-forme d'usage |
|------------|------|-----------------|--|-----------------|-----------------|--------------------|-----------------------|-----------------------|--|
| CNTK | | C++ | Python, C++, BrainScript | ■ | ■ ■ ■ | ■ | ■ ■ ■ | ■ | Windows, Linux, OSX via Decker |
| MXNet | | C++ | R, Python, Julia, Scala, C++, Go, Perl, Matlab, JavaScript | ■ ■ ■ | ■ | ■ ■ ■ | ■ | ■ ■ ■ | Linux, OSX, Windows, AWS, Android, iOS |
| Caffee | | C++ | C++, Python, Matlab | ■ ■ ■ | | ■ | ■ | ■ | Linux, OSX, Windows |
| Torch | | Lua/JIT, Python | Lua, C, Python | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | Linux, OSX, Windows, Android, iOS |
| TensorFlow | | Python, C++ | Python, C, C++, Java, Go, R | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | ■ ■ ■ | Linux, OSX, Windows |
| Theano | | Python, C++ | Python | ■ ■ ■ | ■ ■ ■ | ■ | ■ ■ ■ | ■ ■ ■ | Cross-platform |














- ❖ Tensor Flow : Google, le plus connu, mais difficile à mettre en œuvre
- ❖ Keras : conçu pour simplifier les mécanismes de deep learning, très utilisé pour les POCs, très prometteur dans le futur. API écrite par François Chollet en Python, peut interagir avec Tensor Flow et Theano.
- ❖ PyTorch : créé par Nvidia. Très rapide car il exploite la carte graphique des machines. Une valeur sûre dans le futur.
- ❖ Caffee : plutôt en retrait désormais. Pose des problèmes avec Python 3.
- ❖ Theano : le plus ancien, à abandonner
- ❖ DeepLearning4J : le seul vrai framework pour ceux qui développent en Java. Peut importer des modèles Keras (Python).
- ❖ CNTK (Microsoft Cognitive Toolkit), bien intégré avec les autres outils de Microsoft
- ❖ FasAI est l'équivalent sur PyTorch de Keras sur Tensor Flow. A suivre...peut-être...


Les applications du deep learning



On peut aussi raisonner à l'envers...

Approche des développeurs : partir de l'application et choisir la meilleure technologie neuronale

| | | | |
|---|---|--|--|
| <p>Contrôle et modélisation de processus</p>  <p>Réseaux à fonctions radiales de base. Activation par fonction radiale de base</p> | <p>Diagnostics médicaux</p>  <p>Diagnostics de machines</p>  <p>Perceptron mult niveaux. Activation par tangente ou sigmoïde</p> | <p>Diagnostics médicaux</p>  <p>Perceptron mult niveaux. Activation par tangente ou sigmoïde</p> | <p>Ciblage marketing</p>  <p>WHO IS MY CUSTOMER?</p> <p>Réseau avec rétro-propagation. Activation par fonction logistique</p> |
| <p>Gestion de portefeuilles</p>  <p>Algorithme supervisé de classification Activation par tangente ou sigmoïde</p> | <p>Détermination des taux de crédit</p>  <p>Credit Rating</p> <p>SVM (Support Vector Machine) Analyse discriminante. Activation par fonction logistique</p> | <p>Détermination des taux de crédit</p>  <p>Credit Rating</p> <p>Reconnaissance vocale</p>  <p>Perceptron mult niveaux Réseaux convolutifs. Activation par fonction logistique</p> | <p>Détection des fraudes</p>  <p>Algorithme du gradient Algorithme LMS (Least Mean Square) Activation par fonction logistique</p> |
| <p>Reconnaissance de cible</p>  <p>Réseau neuronal constitué de réseaux indépendants, modérés par un intermédiaire Activation par tangente ou sigmoïde</p> | <p>Prévisions financières</p>  <p>Réseau avec rétro-propagation Activation par fonction logistique</p> <p>IA et apprentissage profond</p> | <p>Reconnaissance vocale</p>  <p>Perceptron mult niveaux Réseaux convolutifs. Activation par fonction logistique</p> | <p>Recherche intelligente</p>  <p>Perceptron mult niveaux Activation par fonction logistique</p> |



Intelligence Artificielle Apprentissage Profond

25 septembre 2020

Nos prochains rendez-vous

| | |
|----------------------|---|
| Vendredi 2 octobre | : Le grave danger que représentent les GAFAM |
| Vendredi 9 octobre | : Au cœur des backbones Internet, comprendre... |
| Vendredi 16 octobre | : Cyberguerre, entre fantasmes et réalités |
| Vendredi 23 octobre | : Les avancées concrètes des villes intelligentes |
| Vendredi 30 octobre | : Les algorithmes de chiffrement, ces inconnus |
| Vendredi 6 novembre | : L'IA et la fin de la démocratie |
| Vendredi 13 novembre | : Les certifications pour remplacer les diplômes |
| Vendredi 20 novembre | : IA et la démocratie |
| Vendredi 27 novembre | : La médecine du futur, les barrières explosent |
| Vendredi 4 décembre | : La transformation digitale, mythe ou réalité |
| Vendredi 18 décembre | : Panorama des architectures globales du TI |
| Mercredi 23 décembre | : Une journée comme les autres en... 2070 |

IA et apprentissage profond

24 / 24