

DIC9150 Concepts fondamentaux de l'informatique cognitive

Réseaux Neuronaux Artificiels (RNA)

Roger Villemaire

Département d'informatique
UQAM

7 novembre 2023



© 2016-2023 Roger Villemaire, villemaire.roger@uqam.ca

Creative Commons Paternité - Pas d'Utilisation Commerciale - Pas de Modification 3.0 non transcrit.

Plan

- 1 Sources de Connaissance
- 2 Approches Naturalistes
- 3 RNA

Source d'information

- Méthodes Symboliques :
 - formalisation manuelle, par ex. www.cyc.com
 - travail considérable, mais qui peut valoir la peine !
 - apprentissage, par ex. arbres de décision
 - il faut néanmoins déterminer les attributs à évaluer.
- Il reste toutefois des tâches pour lesquelles ces approches ne semblent pas faciles à appliquer :
 - par ex., déterminer si une photo représente une fleur !
 - répondre à une question en langage naturel (français, anglais, etc.) !

Principe Connexionniste

- Apprentissage machine par la détermination de poids de connexions :
 - détermination de la représentation à partir d'exemples
 - usage d'ensembles de données correctes pour l'entraînement (détermination des poids), puis application sur des données quelconques.
 - Abstraction, déterminer une fonction f
 - à partir de valeurs correctes $f(x_1) = y_1, \dots, f(x_n) = y_n$,
 - usage de f pour prédire de nouvelles valeurs $f(x)$.
 - La fonction f est déterminée par le poids des connexions.

Intelligence

- Dès les débuts de l'IA (années 1950) plusieurs chercheurs promettaient une véritable “intelligence artificielle” pour un proche futur.
- Malgré les énormes progrès en résolution de problèmes, jeux, inférence logique, planification etc.,
- on réalisa que
 - il fallait intégrer beaucoup de connaissances du monde,
 - que des tâches “faciles” pour l'humain offraient des difficultés considérables au traitement par la machine,
 - Un jeune enfant reconnaît facilement une photo de fleur !
 - que le cerveau pouvait être une source d'inspiration pour la machine intelligente.

IA et Sciences cognitives

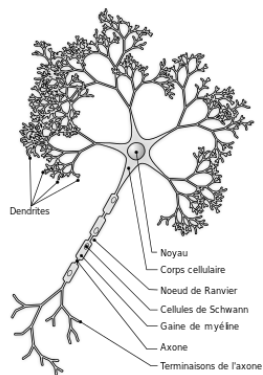
- L'approche connexionniste ne cherche pas à produire un modèle détaillé, ni même physiologiquement plausible du cerveau,
- s'inspire toutefois (sommairement) du développement des connaissances en cognition,
- peut donc éclairer les possibles processus sous-jacents de la connaissance.

Cerveau et Machine

- Il reste toutefois des différences considérables d'échelle :
 - le cerveau humain contient presque 100 fois plus de neurones qu'un processeur de transistors,
 - la simulation informatique d'un seul neurone artificiel prendrait des centaines de cycles et seul un nombre limités (un par cœur) pourra être simulé simultanément alors que tous les neurones, bien que lents, sont tous simultanément en fonction.
- L'intelligence ou la conscience pourrait être un effet d'échelle, mais :
 - les simulations informatiques de modèles plus réalistes du cerveau se font à des échelles très petites par rapport aux tailles des cerveaux des mammifères, voir par ex. European Human Brain Project, Brain Atlas du Allen Institute.

Cerveau

- Centre de la pensée, de la connaissance, de l'intelligence,
 - on connaît la localisation de certaines fonctions dans le cerveau (voir Oliver Sacks, par ex.).
- Structure :
 - le neurone (cellule nerveuse) est la composante fondamentale du cerveau,
 - l'influx nerveux, un signal bio-électrique, se décharge d'un neurone à l'autre.



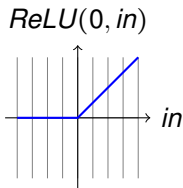
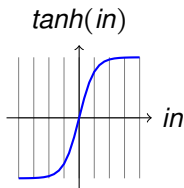
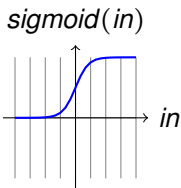
© Nicolas Rougier

Neurone : Fonctionnement Sommaire

- Les *dendrites* (des dizaines de milliers) conduisent l'influx nerveux venant des autres neurones au corps du neurone,
- en fonction des signaux reçus, le neurones va développer un *potentiel d'action* (variation de potentiel de quelques millisecondes),
- l'unique axone (quelques millimètres à un mètre de longueur) décharge périodiquement le potentiel d'action par ses *terminaisons synaptiques* (en moyenne 10 000) vers les autres neurones.

Neurone Artificiel

- Un neurone artificiel est formé
 - d'un nœud avec des entrées in_1, \dots, in_n de poids w_1, \dots, w_n (tous des réels)
- Le neurone (ou nœud) calcule sa sortie (ou *niveau d'activation*) en calculant :
 - la somme pondérée de ses entrées $in = \sum_{j=1}^n w_j \cdot in_j$,
 - la valeur $g(in)$ de sa *fonction d'activation* g appliquée à in .
- En général, la fonction d'activation est l'une de : *sigmoid*, *tanh*, *ReLU* ($\max(0, x)$, *Rectified Linear Unit*).



Réseau Neurones Artificiels (RNA)

- La sortie d'un neurone artificiel est connectée sur l'entrée de possiblement plusieurs (ou aucun) autres neurones pour former un *réseau de neurones artificiels* (RNA).
- En général tous les neurones, d'un même réseau, utilisent la même fonction d'activation.
- On distingue dans un RNA des nœuds d'entrée, de sortie ainsi que les autres nœuds qui sont dits *internes* ou *cachés*.
 - Un nœud peut être simultanément d'entrée et de sortie.

Calcul d'un RNA

- L'opération d'un RNA se produit de la façon suivante :
 - On affecte des valeurs aux nœuds d'entrée,
 - on effectue le calcul pour tous les nœuds (on itère jusqu'à la stabilisation des sorties, s'il y a des rétroactions),
 - finalement on récupère les valeurs de sortie.
- Chaque sortie est donc la valeur d'une fonction des entrées.

Exemples

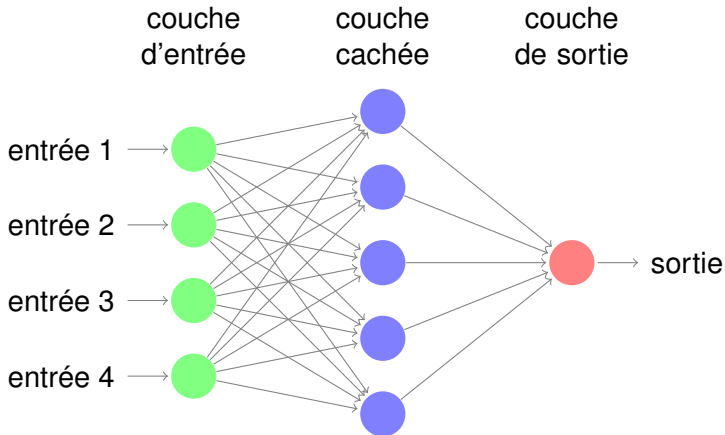
- Reconnaissance de l'écriture (codes postaux) :
 - entrées : matrice de pixels,
 - sortie : un nombre $[0, 9]$.
- Voiture qui se conduit elle-même :
 - entrées : matrice de pixels de l'image vidéo (éventuellement radar, détecteurs etc.),
 - sortie : instructions pour opérer le volant (tourner à droite, gauche), éventuellement d'autres sorties pour contrôler l'accélération, le freinage, etc.

Structure du réseau

- On distingue deux types de réseaux :
 - les réseaux *acycliques* (feed-forward) où il n'y a pas de cycles (rétroaction) :
 - les valeurs d'activation des nœuds sont calculées par propagation et la valeur de sortie ne dépend que des valeurs d'entrées, il n'y a donc pas d'état interne au système (pas de mémoire).
 - les réseaux *récurrents* où les liens sont quelconques :
 - les valeurs d'activation sont recalculées jusqu'à la stabilisation des sorties. Les valeurs d'activation des nœuds cachés peuvent servir à déterminer les valeurs d'activation des nœuds de sortie. Les nœuds cachés forment ainsi une forme de mémoire.
 - Un réseau récurrent peut très bien jamais se stabiliser et plutôt osciller ou avoir un comportement chaotique, exactement comme un système multi-agents.

Réseaux Multi-couches

- La plupart des applications utilisent un réseau (feed-forward) structuré en couches :



Fonctions Représentables

- Les réseaux multi-couches sans couche cachée (*perceptrons*) ont été étudiés dès les années 1960, mais il s'est avéré qu'ils étaient limités dans le type de fonctions qu'ils peuvent représenter (fonctions *linéairement séparables*).
- Néanmoins, déjà avec une couche cachée (suffisamment grande), on peut représenter n'importe quelle fonction continue des entrées,
- et avec deux couches cachées (suffisamment grandes) on peut représenter n'importe quelle fonction.

Apprentissage

- L'*apprentissage* consiste à déterminer, à l'aide d'un ensemble d'*entraînement* contenant des valeurs correctes de la fonction, des valeurs adéquates pour les poids des arêtes.
- Au niveau computationnel il s'agit de déterminer des valeurs des poids qui minimisent l'*erreur* (différence entre la valeur obtenue et celle désirée) pour l'ensemble d'entraînement,
- cette erreur est calculée à partir d'une fonction de perte (loss function) spécifique au problème.
- En termes statistiques, il s'agit d'un problème de *régression non-linéaire*.

Méthode d'apprentissage

- En pratique la méthode d'apprentissage la plus populaire est celle dite de la *descente de gradient* :
 - on fixe des poids initiaux au hasard.
 - on itère sur l'ensemble d'entraînement en déterminant pour chaque élément de l'ensemble d'entraînement :
 - l'erreur (différence entre la valeur obtenue et attendue) et
 - en distribuant une (petite) correction sur l'ensemble des poids, en corrigeant d'autant plus les poids dont la variation a le plus d'impact sur l'erreur.
- On teste la qualité (erreur) sur un ensemble de test qui ne sert pas à déterminer les poids et est donc indépendant de l'entraînement.

Méthodologie de validation

- Pour ne pas *biaiser* les résultats, il faut valider sur des données de test *distinctes* de celles utilisées pour l'entraînement.
- Il s'agit donc de :
 - obtenir un ensemble de données étiquetées,
 - en prendre une partie pour faire l'entraînement,
 - valider sur l'autre partie.

Méthodologie de validation

- Lorsqu'il y a peu de données étiquetées, on peut utiliser la *validation croisée* (10 folds cross-validation) :
 - diviser les données en 10 parties égales,
 - entraîner sur les 9 dixièmes et valider sur la dixième partie,
 - recommencer avec un autre dixième.

Apprentissage Profond (Deep learning) ¹

- Nombre très important de nœuds et de couches cachées !
- Architecture : choisir des types d'interconnexion entre chaque couche.
- En général nécessite beaucoup plus de données et de temps pour l'entraînement.
- Revient en force, car on a de plus en plus les données (Bigdata) et la puissance de calcul nécessaire (GPU).

Données Massives (Big Data)

- D'énormes quantités de données sont aujourd'hui disponibles :
 - en entreprise : bases de données,
 - par la contribution en ligne : Google, Facebook, Twitter, etc.
 - par des projets ciblés, par exemple :
 - ImageNet <https://www.kaggle.com/>, etc.
- Idée : L'intelligence humaine sera reproductible avec d'énormes quantités de données plutôt qu'avec de nouveaux algorithmes.

Optimisation et implémentations

- De nombreuses optimisations permettant un apprentissage plus rapide ont été introduites, par exemple, la méthode de descente de gradient peut être augmentée d'autres techniques pour faire converger plus rapidement les couches les plus basses, comme :
 - restrictions locales des variables,
 - utilisation d'algorithmes d'apprentissage non-supervisé pour les couches les plus profondes.
- Il y a de nombreuses implémentations efficaces, possiblement avec un langage spécifique de description du réseau, par exemple : <http://torch.ch/>, <https://www.tensorflow.org/>.

Yann LeCun vs. Gary Marcus²

- inné et acquis, plus/moins de structure
- réseaux profonds \neq intelligence humaine (de petites perturbations peuvent induire le réseau en erreur)
- mais c'est ce qui marche le mieux, du moins dans certains domaine
- la structure qui nous semble naturelle, n'est pas nécessairement utile, mieux vaut la faire découvrir lors de l'entraînement
- mais il y a de la structure aussi en apprentissage profond (nombre de couches, architecture)
- s'approche-t-on des limites de l'apprentissage profond ou est-ce le début d'un changement fondamental de paradigme qui ouvrira la porte vers une véritable intelligence artificielle ?

2. Oct5th, 2017, [https:](https://www.youtube.com/watch?v=aCCotxqxFSk)

[//www.youtube.com/watch?v=aCCotxqxFSk](https://www.youtube.com/watch?v=aCCotxqxFSk)

Voir aussi

- Adnan Darwiche, Human-level intelligence or animal-like abilities ?, Communications of the ACM, Volume 61 Issue 10, October 2018, Pages : 56-67, doi>10.1145/3271625

Conclusion

- Les RNA sont particulièrement adaptés aux domaines où il semble difficile d'établir les concepts et notions fondamentales, ce qui couvre un très vaste champ d'applications.
- Comme le temps de convergence de l'apprentissage peut être très grand, on voit émerger des modèles préentraînés qui ne nécessitent qu'un petit entraînement supplémentaire pour s'adapter à une nouvelle application.
- L'apprentissage profond a montré un potentiel important dans des domaines comme la reconnaissance de l'écriture, de la parole, d'images et la traduction automatique, principalement grâce aux données importantes dans ces domaines.
- Néanmoins, les RNA sont des boîtes noires. Ils n'offrent pas d'explication et ne donnent aucune signification à leurs paramètres (poids).