Introduction à l'apprentissage profond

Résultats fondamentaux et applications à la médecine

Ayoub Ajarra

14 Mar 2024

Philosophie de cette présentation

- Notre objectif est d'apprendre ce qu'est **l'apprentissage automatique** en général et plus spécifiquement de vous faire découvrir les principes de **l'apprentissage profond**, en mettant l'accent sur les applications dans le domaine de la santé publique et de la médecine.
- **NB**: il ne s'agit pas d'une analyse théorique des algorithmes d'apprentissage profond, ni de leur aspect pratique.

Introduction & Motivation

Quand pouvons-nous parler d'un problème d'apprentissage?

Compte tenu d'une certaine quantité de **données**, comment comprendre le processus général de conception de solutions algorithmiques pour prendre des décisions **précises** avec une grande **confiance**.

Introduction & Motivation

- Le terme "profond"dans apprentissage signifie plusieurs couches d'un réseau neuronal.
- L'apprentissage profond permet de réaliser des applications réelles dont les performances sont bien supérieures à celles des autres méthodes.
- Capacité à traiter des ensembles **larges** de données
- Pas besoin de "feature engineering", l'apprentissage profond le fait pour vous.
- Réduction de l'expertise nécessaire pour résoudre le problème
- Prend en compte le fléau
- Calcul dans des manifolds de grandes dimension.
- Logiciels établis et robustes, faciles à mettre en œuvre et à utiliser tel que: PyTorch (Populaire), Keras, TensorFlow (moins populaire)

Principes fondamentaux de l'apprentissage profond

Les types d'apprentissage

Apprentissage supervisé

Données du problème:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$

Hypothèses du statisticien:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$

Hypothèses du statisticien:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$

Données accessibles:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$
- Classe d'hypothèses: $\mathcal{H}$

Données accessibles:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$
- Classe d'hypothèses: $\mathcal{H}$
- Séparabilité linéaire des données (en cas de classification).

Données accessibles:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$
- Classe d'hypothèses: $\mathcal{H}$
- Séparabilité linéaire des données (en cas de classification).

Données accessibles:

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$
- Classe d'hypothèses: $\mathcal{H}$
- Séparabilité linéaire des données (en cas de classification).

Données accessibles:

- Echantillon fini de points étiquetés.

Apprentissage supervisé

Données du problème:

- Domaine: $\mathcal{X}$
- Codomaine: $\mathcal{Y}$
- Classe de concept: $\mathcal{C}$

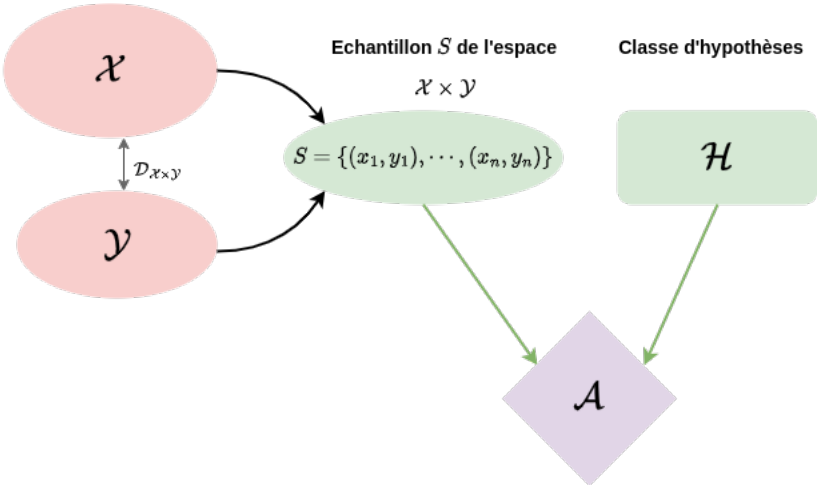
Hypothèses du statisticien:

- $S^{\infty} \overset{iid}{\sim} \mathcal{D}$
- Classe d'hypothèses: $\mathcal{H}$
- Séparabilité linéaire des données (en cas de classification).

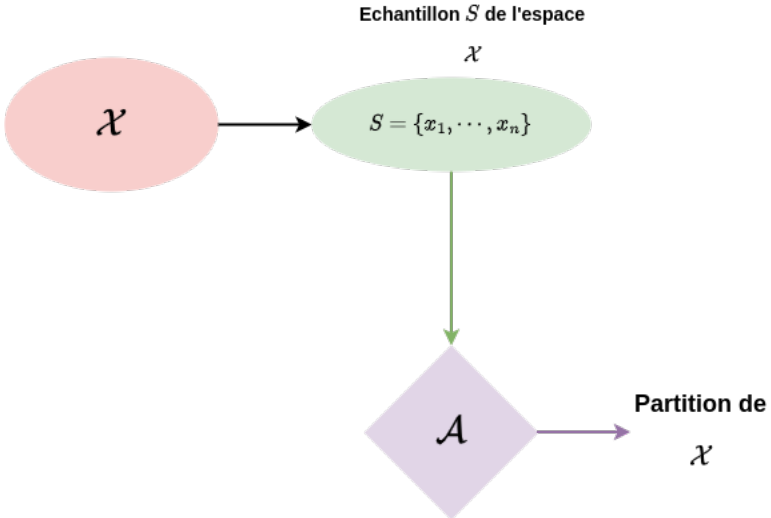
Données accessibles:

- Echantillon fini de points étiquetés.
- Mesure d'erreur.

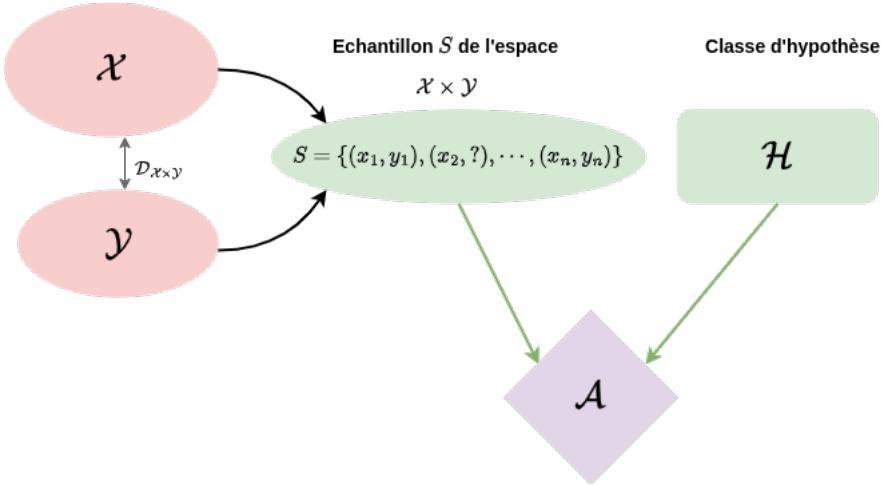
Apprentissage supervisé



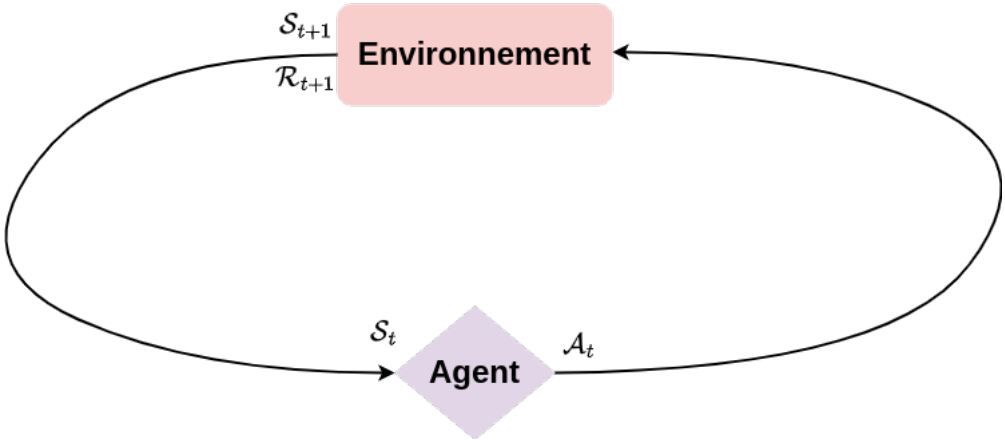
Apprentissage non-supervisé



Apprentissage semi-supervisé



Apprentissage par renforcement



Theorème d'approximation universelle & Architectures des réseaux neuronales

Expressivité de la profondeur
Vs
Expressivité de la largeur

Expréssivité des réseaux de neurones

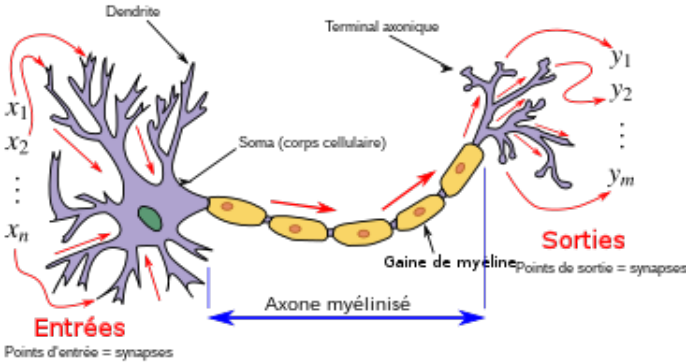
Question fondamentale

Quel type de fonctions peut-on approximer avec des réseaux de neurones?

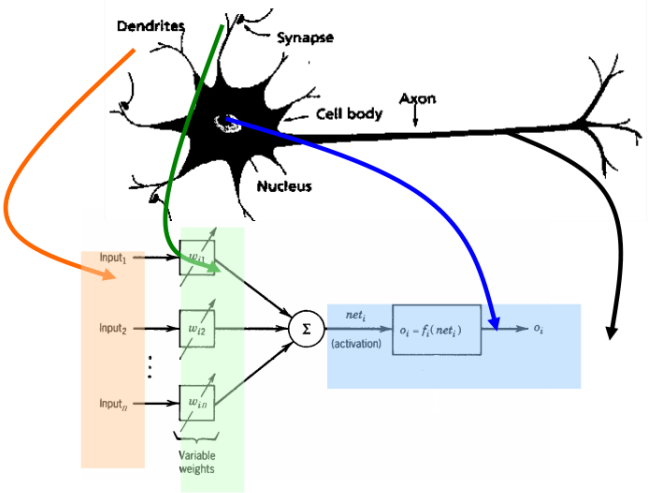
Les architectures des réseaux neuronnaux

- Feed-forward (output unique)
- Reccurent (données séquentielles)

Neurone biologique



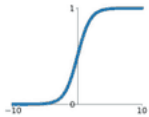
Neurone biologique & Analogie



Fonctions d'activation

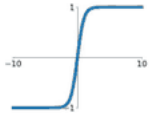
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



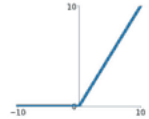
tanh

$$\tanh(x)$$



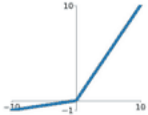
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

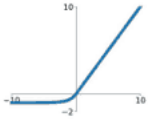


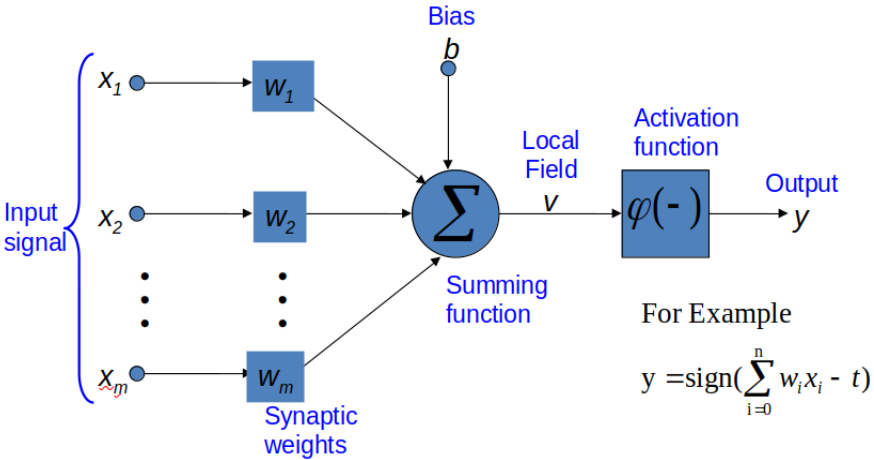
Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$





For Example

$$y = \text{sign}\left(\sum_{i=0}^{n} w_i x_i - t\right)$$

Fonctions d'activations: Sigmoid

- **Avantages:**
  - Ramène les valeurs entre 0 et 1.
  - Bien étudié.
- **Inconvénients:**
  - Un neurone saturé a pour effet de d'annuler les gradients.
  - Sortie d'une sigmoïde n'est pas centrée à zéro.
  - l'exponentielle est très coûteux en termes de calcul.

Fonctions d'activations: Tanh

- **Avantages:**
  - Ramène les valeurs entre -1 et 1.
  - Sortie centrée à zéro.
- **Inconvénients:**
  - Annulation du gradient lorsque la fonction saturée.

Fonctions d'activations: ReLU

- **Avantages:**
  - Rapide.
  - Aucune saturation.
  - permet une convergence $\sim 5 - 10$ plus rapide que sigmoid et tanh.
- **Inconvénients:**
  - Sortie non centrée à zéro.
  - Pour les valeurs de $x < 0$, le gradient est nul.

Fonctions d'activations: Leaky ReLU

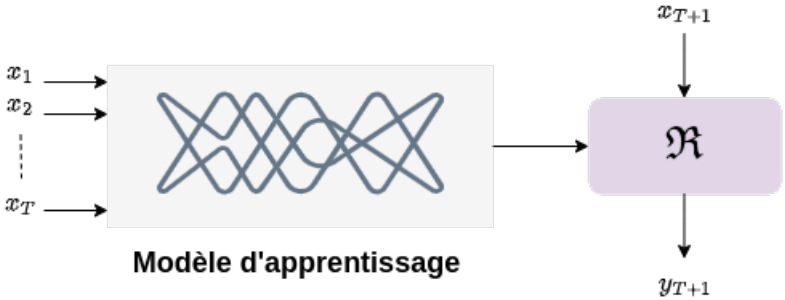
- **Avantages:**
  - Aucune saturation.
  - Très rapide.
  - Le gradient ne s'annule pas.
- **Inconvénients:**
  - 0.01 est un hyperparamètre

Fonctions d'activations: Parametric ReLU

- **Avantages:**
  - Même avantages que Leaky ReLU.
- **Inconvénients:**
  - l'hyperparamètre s'apprends durant la rétropropagation et est actualisé avec les autres hyperparamètres du problème.

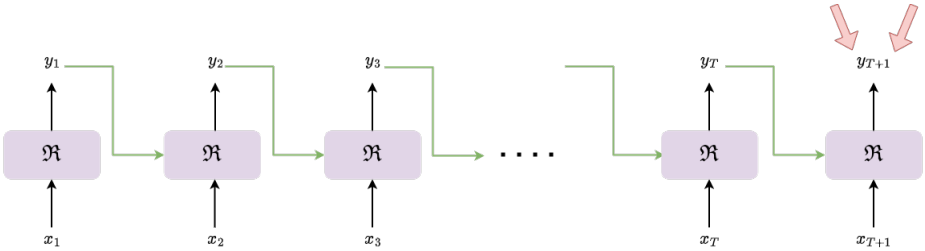
Fonctions d'activations: En pratique

- Par défaut, on utilise souvent la fonction d'activation ReLU.
- Les variations de la fonction ReLU sont encouragées.
- Sigmoid est uniquement utilisée à la sortie d'un réseau pour un problème de classification à deux classes.



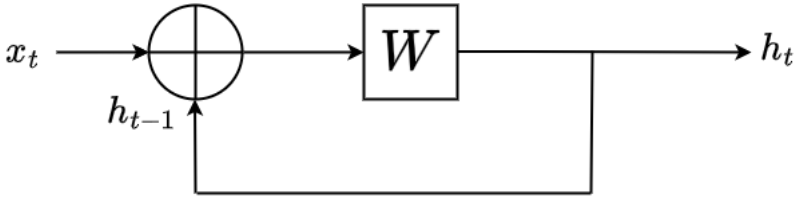
La structure générale de l'algorithme d'apprentissage dans le cas de données temporelles.

Réseaux de neurones réccurents



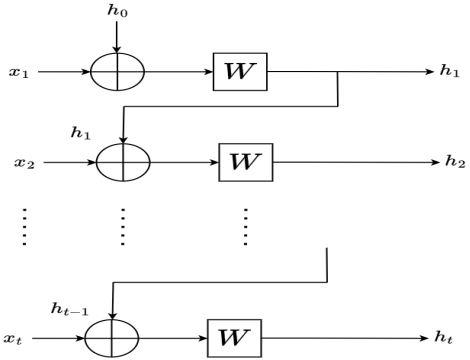
La nature séquentielle de l'algorithme d'apprentissage necessite que chaque entrée à $\mathfrak{R}$ dépendera de toutes les séquences temporelles qui précedent.

Blocks résiduels



Block résiduel (Version sans rollout)

Blocks résiduels



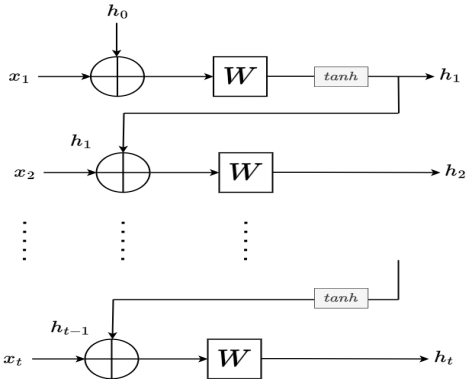
Block résiduel (Version rollout)

Cependant ...

La construction est linéaire en fonction des états de chaque bloc et donc susceptible de mal représenter le flux d'information temporel (Transformation affine des données d'entrée).

Solution: structure non-linéaire

La construction est linéaire en fonction des états de chaque bloc et donc susceptible de mal représenter le flux d'information temporel (Transformation affine des données d'entrée).



Structure finale.

La classe des réseaux de neurones $\mathcal{H}_\theta$ & Theorème fondamental d'approximation

Theorème d'approximation universelle (informel)

Toute fonction continue $f : [0,1]^n \to [0,1]$ peut être approximée arbitrairement par un réseau de neurones avec au moins une couche cachée et un nombre fini de poids.

Exemple: Approximation des fonctions Lipschitz

Pour approximer une fonction L-Lipschitz $f : \mathbb{R} \to \mathbb{R}$ dans l'interval $[0, 1]$ avec une erreur d'au plus ϵ, combien de bin a t-on besoin?

Exemple: Approximation des fonctions Lipschitz

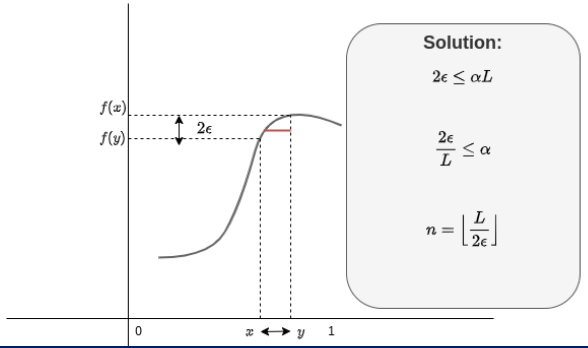
Une fonction $f : \mathbb{R} \to \mathbb{R}$ est L-Lipschitz quand elle vérifie la proprieté suivante:

$$\forall x, y \in \mathbb{R} : |f(x) - f(y)| < L|x - y|$$

Exemple: Approximation des fonctions Lipschitz

Une fonction $f : \mathbb{R} \to \mathbb{R}$ est L-Lipschitz quand elle vérifie la proprieté suivante:

$$\forall x, y \in \mathbb{R} : |f(x) - f(y)| < L|x - y|$$



Exercice: Approximation des fonctions Lipschitz dans un espace de grande dimension

Pour approximer une fonction L-Lipschitz $f : \mathbb{R}^n \to \mathbb{R}$ dans l'interval $[0, 1]^n$ avec une erreur d'au plus ϵ, combien de bin a t-on besoin?

Approximation de Cybenko par superposition de famille de fonctions Sigmoid

Famille de fonctions Sigmoid

Une famille de fonctions Sigmoid est une famille de fonctions σ qui vérifient les proprietés suivantes:

$$\lim_{x \to +\infty} \sigma(x) = 1$$
$$\lim_{x \to -\infty} \sigma(x) = 0$$

Approximation de Cybenko par superposition de famille de fonctions Sigmoid

Approximation de Cybenko (informel)

Toute fonction continue peut être approximée par un MLP avec une seule couche intermédiaire et une fonction arbitraire de la famille Sigmoid.

Approximation de Cybenko par superposition de famille de fonctions Sigmoid

Approximation de Cybenko (formel)

Pour toute fonction de la famille Sigmoid σ,
l'ensemble de fonctions $\left\{f : x \to \sum_{i=1}^{N} \alpha_i \sigma(w_i x + b_i)\right\}$ (MLP avec une couche intermédiaire) est dense dans $\mathcal{C}([0,1]^n)$

NB: N dépend de la complexité de la fonction à approximer et de l'erreur d'approximation.

Approximation de Cybenko par superposition de famille de fonctions Sigmoid

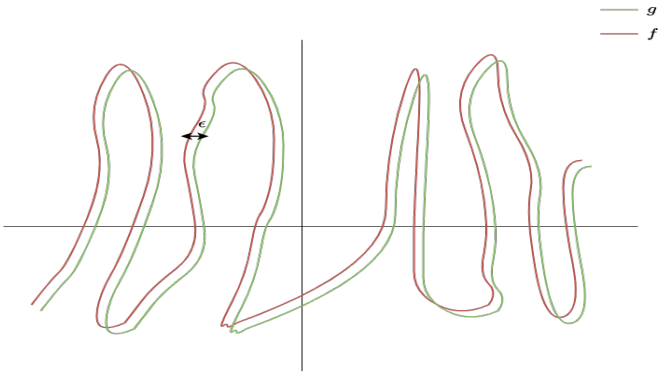


Illustration du theorème de Cybenko

Théorème de représentation de Kolmogorov-Arnold

Theorème de Kolmogorov-Arnold (Informel)

Toute fonction continue à plusieurs variables peut être représentée comme une superposition de fonctions continues d'une seule variable.

Théorème de représentation de Kolmogorov-Arnold

Theorème de Kolmogorov-Arnold (Formel)

Toute fonction continue $f : [0, 1]^n \to \mathbb{R}$ peut être représentée de la façon suivante:

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{2n+1} \phi_i \left(\sum_{j=1}^{n} \psi_{ij}(x_j) \right)$$

Expressivité de la largeur

Theorème d'approximation universelle pour les réseaux ReLU avec une largeur borné

Pour tout $\epsilon > 0$, et toute fonction $f : \mathbb{R} \to \mathbb{R}$ Lesbegue intégrable, il existe un réseau de neurones ReLU $\mathfrak{N}$ (entièrement connecté) avec une largeur $\leq n + 4$ tel que la fonction représenté par ce réseau de neurones $F_{\mathfrak{N}}$ vérifie l'approximation:

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathfrak{N}}(x) dx| < \epsilon$$

Expressivité de la largeur Vs Expressivité de la profondeur

L'importance de la profondeur dans l'apprentissage (Telgarsky 2015)

Soit $m \geq 2^{\frac{k-3}{l-1}}, k \in \mathbb{N}$,
Il existe un échantillon $S = (x_i, y_i)_{i=1}^{2^k}$, avec $x_i \in [0,1], y_i \in \{0,1\}$ pour tout $i \leq 2^k$ tel que:

$$\min_{f \in \mathfrak{N}(\sigma_R, m, l)} \mathcal{R}(f) = \frac{1}{6} \quad \& \quad \min_{f \in \mathfrak{N}(\sigma_R, 2, 2k)} \mathcal{R}(f) = 0$$

$\mathfrak{N}(\sigma_R, m, l)$ désigne l'ensemble de fonctions données par un feedforward réseau de neurones avec une fonction d'activation σ, à l couches avec au plus m nœuds.
σ_R désigne la fonction d'activation ReLU.

Expressivité de la largeur Vs Expressivité de la profondeur

Qu’en est t-il pour les réseaux réccurents?

Expressivité de la largeur Vs Expressivité de la profondeur

Qu'en est t-il pour les réseaux réccurents?

Définition (t-sawtooth fonction d'activation)

Pour tout $t \in \mathbb{N}$, une function $f : \mathbb{R} \to \mathbb{R}$ est dite t-sawtooth quand elle est linéaire par morceaux avec t morceaux.

Example: La fonction ReLU est 2-sawtooth.

- Notation: $\mathfrak{R}(\sigma, m, l, k)$ désigne k itérations d'une réseau de neurones à l couches d'au plus m noeuds.
- Conséquence: $\mathfrak{R}(\sigma, m, l, k) \subseteq \mathfrak{N}(\sigma, m, lk)$

Expressivité de la largeur Vs Expressivité de la profondeur

Qu’en est t-il pour les réseaux réccurents?

Expressivité de la largeur Vs Expressivité de la profondeur

Qu'en est t-il pour les réseaux réccurents?

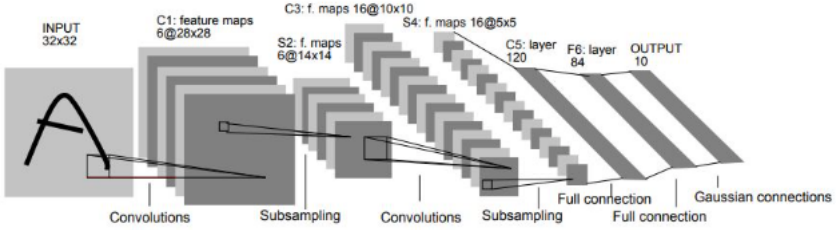
L'importance de la profondeur dans l'apprentissage (Telgarsky 2015)

Soit $k, t \in \mathbb{N}$, l: nombre de couches, m: nombre de noeuds (par couche)
Il existe un échantillon $S = (x_i, y_i)_{i=1}^{2^k}$, avec $x_i \in [0,1], y_i \in \{0,1\}$ pour tout $i \leq 2^k$ tel que:

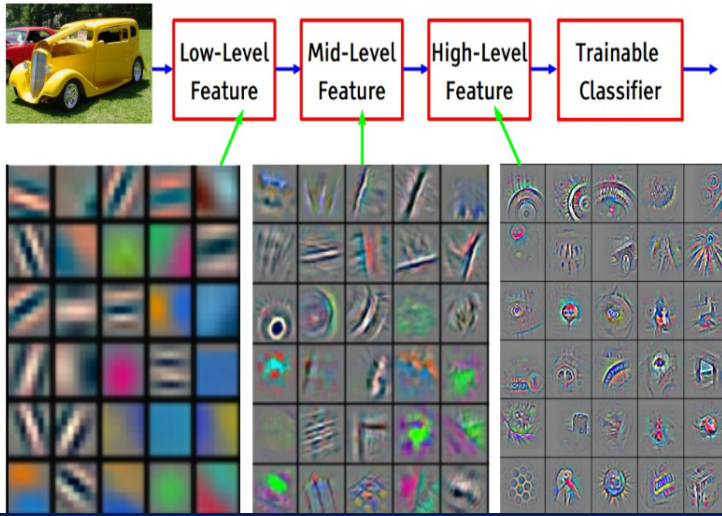
$$\min_{f \in \mathfrak{N}(\sigma_R, 2, 2, k)} \mathcal{R}(f) = 0 \quad \& \quad \min_{f \in \mathfrak{N}(\sigma_t, m, l)} \mathcal{R}(f) \geq \frac{n - 4(tm)^l}{3n}$$

Réseaux neuronales convolutifs (CNN)

- Intuition:
  - Traiter des données de très haute dimension :150 x150 pixels = 22500 entrées.
  - Peut exploiter la topologie 2D des pixels
  - Peut intégrer l'invariance à certaines variations
- Connectivité locale: Chaque unité cachée n'est connectée qu'à une sous-région (patch) de l'image d'entrée.
- Partage des paramètres de poids: Partager la matrice des paramètres entre certaines unités.
- Convolution.
- Pooling/sous-échantillonnage des unités cachées


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

CNN


Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier

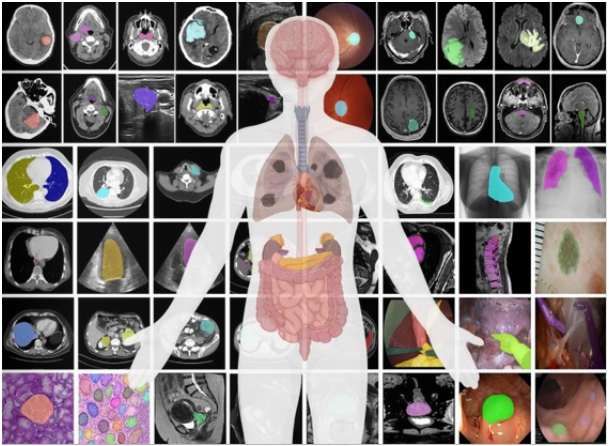
Comparaison aux RNN

- Les CNN traitent des données de nature spatialle: les grilles à valeurs, par exemple les images. Les RNN est spécialisé pour une séquence de valeurs.
- Les CNN s'adaptent facilement aux images de grandes dimensions, les CNN peuvent traiter des images de taille variable Les RNN s'adaptent à des séquences plus longues qu'il ne serait pas pratique pour des réseaux sans adaptations basées sur les séquences, ils peuvent également traiter des séquences de longueur variable.
- Les RNN sont caractérisés par un état, les CNN n'ont pas d'état. (sauf pendant l'entrainement - la quasi-totalité de l'entrainement se fait sans état)
- Les RNN ont une mémoire, contrairement aux CNN.

Applications dans la médecine

L'intégration de l'information au service de la médecine: de la prédiction à la segmentation

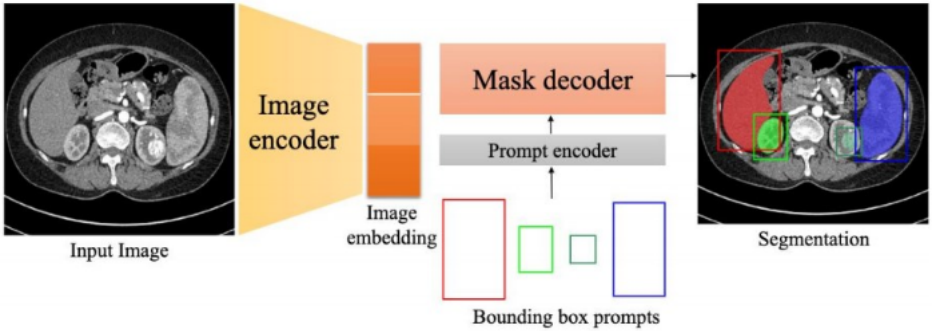
Segmentation sémantique dans les images médicales



Segmentation sémantique dans les images médicales

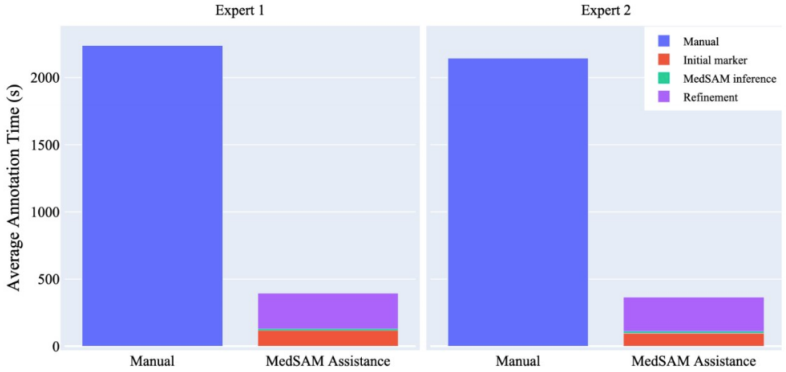
- Partie encoder: un "Vision transformer"pour extraire les caractéristiques de l'image et obtenir un embedding de l'image source, et un encodeur qui peut encoder l'information dans le box capturant l'information necessaire qui permet l'extraction du embedding.
- Partie decoder: décodeur de masque basé sur un transformateur pour fusionner l'incorporation d'image et l'incorporation de l'embedding afin de générer les masques finaux.

Segmentation sémantique dans les images médicales: MedSAM



Résultats

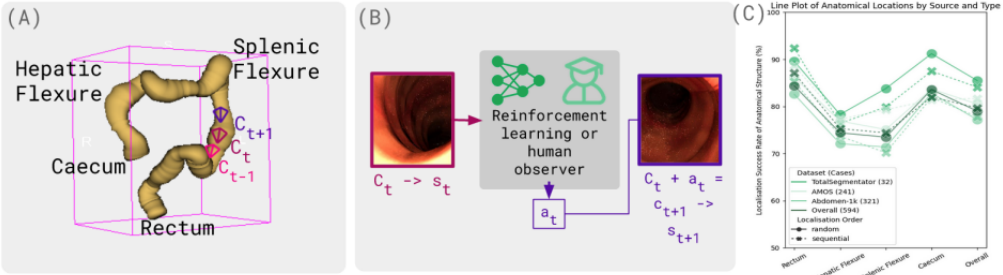
Un essai avec deux groupes de médecins a montré une augmentation de plus de 80 % de l'efficacité du flux de travail pour les médecins assistés par le modèle par rapport à ceux qui suivaient un flux de travail manuel standard.



SARAMIS: Simulation pour la chirurgie assistée par robot pour un traitement mini-invasif

- Collecte des données réelles à l'aide de capteurs fixés sur des caméras afin d'entraîner les algorithmes est souvent très couteux.
- SARAMIS permet de fournir le premier ensemble d'environnements et de données qui peuvent être utilisés pour générer une multitude de données synthétiques.

SARAMIS: Simulation pour la chirurgie assistée par robot pour un traitement mini-invasif



Inria

Université de Lille

That’s all folks!