Introduction au Deep Learning

Présentation et histoire du Deep Learning

J. Rynkiewicz

Université Paris 1

2022

2012 : Le Big Bang du Deep Learning

- Les modèles réseaux de neurones artificiels existent depuis longtemps (plus de 60 ans), mais ils sont tombés en désuétude au début des années 2000.
- En 2012, à la plus grande conférence de vision artificiel un réseau de neurone profonds écrase toutes les autres techniques.
- Dominique Cardon, Jean-Philippe Cointet et Antoine Mazières, « La revanche des neurones. L'invention des machines inductives et la controverse de l'intelligence artificielle », Réseaux 2018/5 (n° 211), p. 173-220.
- Depuis, le Deep Learning est l'état de l'art dans de nombreux domaine.

Deep learning

Plusieurs familles de réseaux de neurones profonds sont utilisés en pratique actuellement :

- Les réseaux convolutifs qui sont l'état-de-l'art pour la classification d'images.
- Les réseaux récurrents qui peuvent modéliser des séries temporelles. Ils ont longtemps été utilisés avant pour le traitement du language naturel (reconnaissance de parole, traduction, etc...).
- Les "transformers" qui sont l'état-de-l'art pour le traitement du language naturel.

Ce cours à pour but d'étudier ces familles de réseaux de neurones profonds. Il existe d'autres modèles intéressants mais qui ne sont pas étudiés dans ce cours :

- Les modèles génératifs adversaires (pour la synthèse d'image).
- L'apprentissage par renforcement (AlphaGo, AlphaZero, etc...)

Algorithme d'apprentissage

Il faut prédire à quelle classe appartient une une variable aléatoire Y, à partir de la valeur de données explicatives X.

$X =$  $\Longrightarrow$ $Y =$ "chat"

- Les paramètres du réseau de neurones sont estimés sur une base d'apprentissage. Il doit faire peu d'erreurs de classification sur cet ensemble.
- Il faut que le modèle "généralise" bien : Il doit faire peu d'erreurs sur des données nouvelles.
- La différence entre l'erreur d'apprentissage et l'erreur de généralisation est appelée "surapprentissage".

Le fléau de la grande dimension

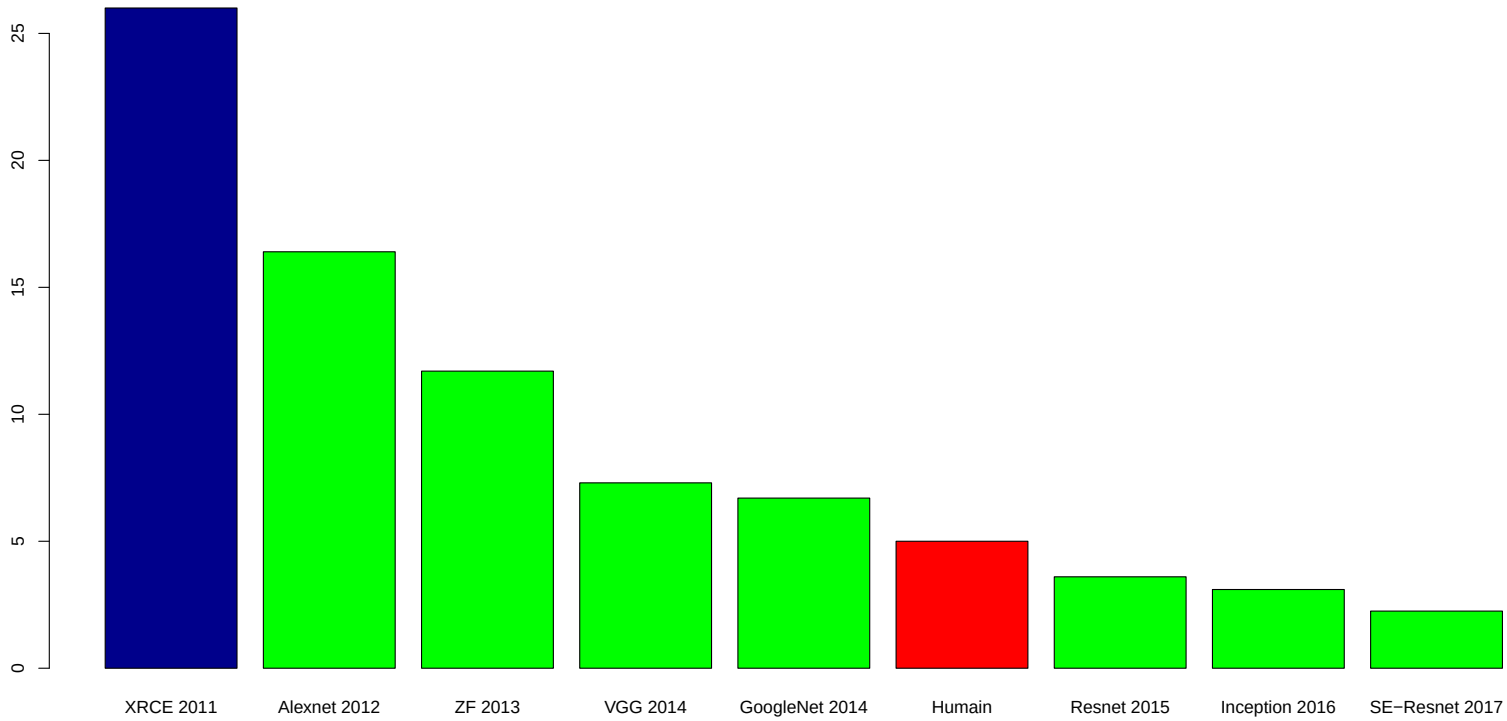
- La fonction de classification $x \mapsto f(x)$ peut être approximée à l'aide d'exemples :

$$\left\{ \begin{pmatrix} x_1 \\ f(x_1) \end{pmatrix}, \ldots, \begin{pmatrix} x_n \\ f(x_n) \end{pmatrix} \right\}$$

  par interpolation locale. Ainsi, si x est proche de x_i, en pense que $f(x)$ sera proche de $f(x_i)$.
- Pour les images, la dimension d de x va de quelques milliers à quelques centaines de milliers.
- Pour couvrir $[0,1]^d$ avec des boules de rayon 0.1, il faut 10^d points, hors il y a moins de 10^{100} atomes dans l'univers...
- La distance euclidienne entre deux images différentes de chats est très grande.
- Le modèle doit découvrir d'autres régularités que la proximité des distances euclidiennes.

Le Deep learning et Imagenet

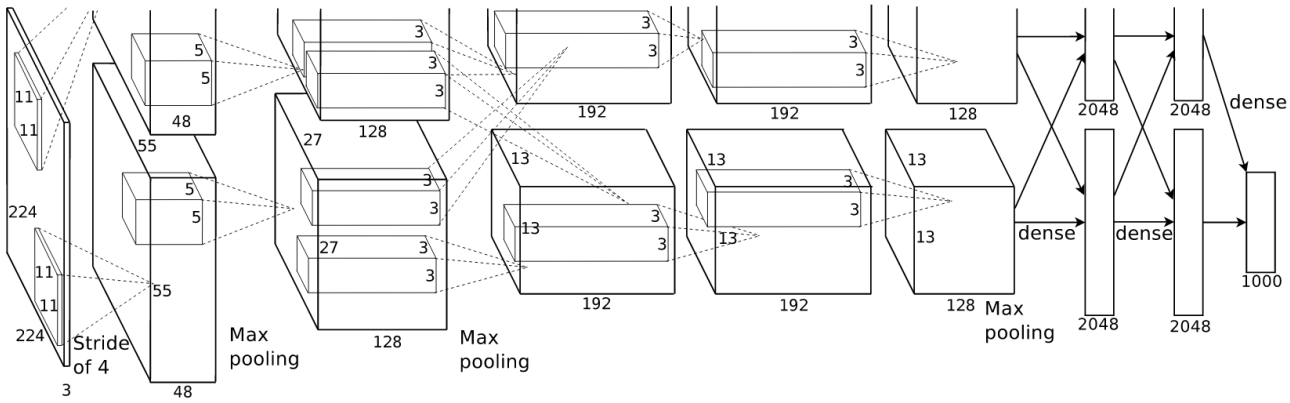
- Compétition de classification d'image sur une base de donnée avec $1,4$ millions d'images et 1000 classes.
- Taux d'erreur de classification sur imagenet (top 5).
- En bleu : avant le Deep-Learning, en vert : Modèles issus du Deep learning, en rouge : être humain.



Le réseau Alexnet (2012)

Il a ouvert la porte aux modèles de deep learning pour la reconnaissance d'image.

- Ce réseaux commence par des enchainements de couches convolutionelles et de max-pooling.
- Il finit par des couches denses.
- Il a été optimisé grâce à des cartes graphiques avec des algorithmes de calculs massivement parallèles.



Formalisation de la classification supervisée (1)

C'est une généralisation des modèles pour prédire des données catégorielles (régression logistique, polytomique).

- On cherche à prédire la probabilité d'une variable catégorielle Y connaissant une variable explicative X : $P(Y|X)$.
- On rappelle que, pour $k \in \{0, \cdots, K-1\}$, le modèle linéaire généralisé polytomique s'écrit :

$$P_\theta(Y = k|X) = \frac{\exp(X^T \beta_k)}{1 + \sum_{l=1}^{K-1} \exp(X^T \beta_l)}$$

  Ici, $\theta = (\beta_1, \cdots, \beta_{K-1})$.
- Pour la compétition Imagenet Y est dans $\{1, \cdots, 1000\}$ et X est dans $\mathbb{R}^{3\times 256\times 256}$. La dimension du paramètre du modèle polythomique serait donc $999 \times 256 \times 256 \times 3 \simeq 200 \times 10^6$.

Formalisation de la classification supervisée (2)

Pour un réseau de neurones, on remplace la fonction linéaire $X^T \beta_k$ par une fonction non-linéaire $F_{\beta_k}(X)$.

- Un réseau de neurones $f_\theta(.)$ va chercher à estimer $f_\theta(X) = P_\theta(Y|X) \simeq P(Y|X)$ avec

$$P_\theta(Y = k|X) = \frac{\exp(f_\theta^k(X))}{\sum_{l=1}^K \exp(f_\theta^l(X))}$$

  Ici, $f_\theta(X) = (f_\theta^1(X), \cdots, f_\theta^K(X))$ est un réseau de neurone avec une sortie de dimension K.

- Dans le cas d'un réseau de neurones, θ est aussi appelé l'ensemble des "poids". Ce poids vont être optimisés, numériquement, pour maximiser la vraisemblance du modèle à l'aide d'une descente de gradient.
- On va donc estimer $\widehat{\theta}_n = \arg\min - \sum_{t=1}^n \log P_\theta(y_i|x_i)$ grâce aux données d'apprentissage $\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \cdots, \begin{pmatrix} x_n \\ y_n \end{pmatrix} \right)$.
- Comme le paramètre θ est de très grande dimension (plusieurs millions), il y a un risque de sur apprentissage du modèle aux données d'apprentissage. Le modèle sera donc évalué sur des données totalement nouvelles, les données de test : $\left(\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix}, \cdots, \begin{pmatrix} x_{n+T} \\ y_{n+T} \end{pmatrix} \right)$.

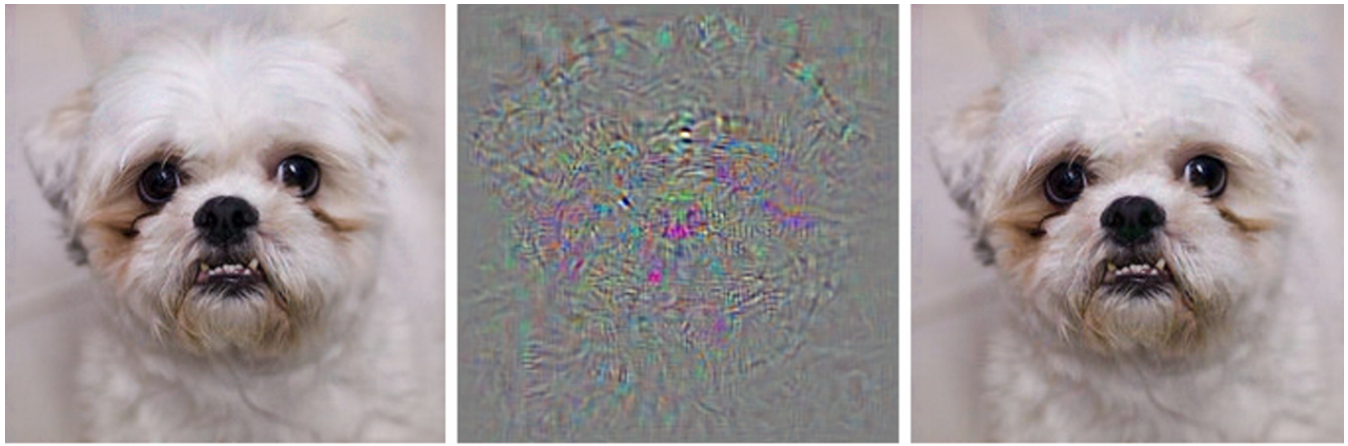
Différences avec les modèles statistiques classiques

Le cadre classique apporte peu d'informations sur les réseaux profonds.

- Le nombre de paramètres (le nombre de poids) peut être plus grand que le nombre de données disponibles pour l'apprentissage. Dans ce cas, les thèorèmes asymptotiques ne sont pas forcément justifiés.
- Pour éviter le surapprentissage l'ensemble d'apprentissage sera souvent découpé en ensemble d'entrainement pour estimer les paramètres $\widehat{\theta}_n$ et un ensemble de validation pour savoir quand on doit arrêter l'optimisation des poids.
- Pour avoir de bon résultats, les réseaux profonds doivent être très structurés à priori (convolutions de bonne taille, connexions "short-cut", fonctions de pooling...).
- L'optimisation est faite grâce à l'algorithme du gradient stochastique, sur des mini-batches.

Fragilité des modèles de Deep Learning

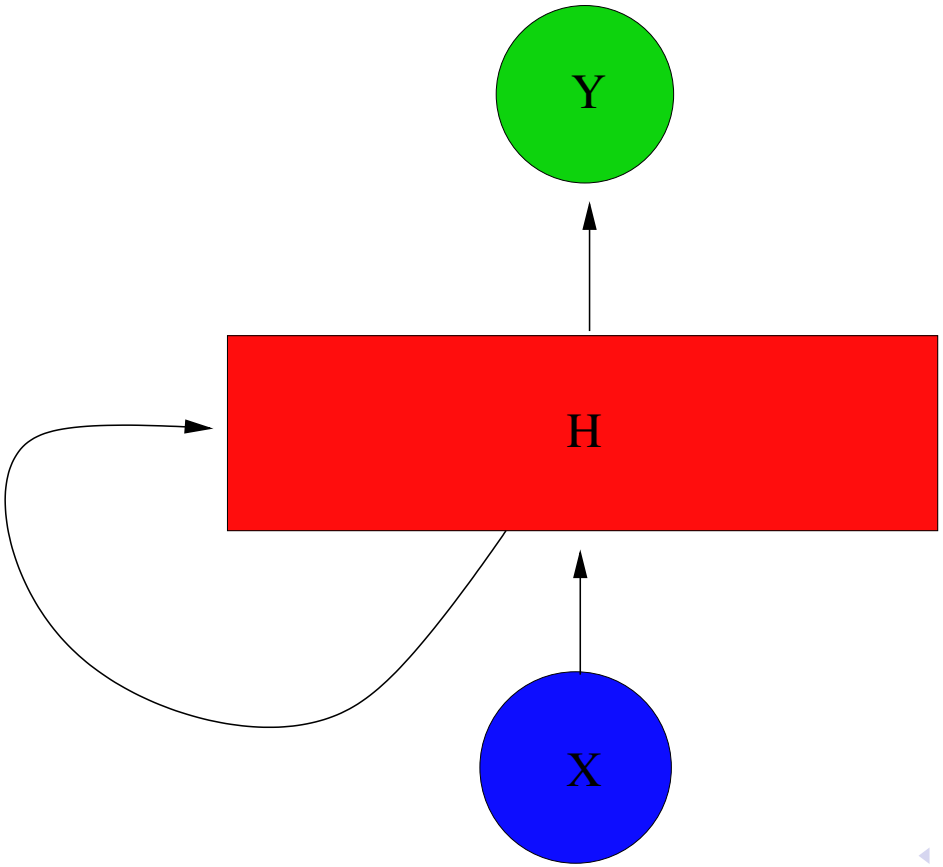
Même si ces modèles peuvent être très performants il est plutôt difficile d'avoir des bornes de confiance pour leurs prédictions :



Les réseaux récurrents

Ces réseaux de neurones ont été utilisés pour la modélisation des séries temporelles. Ils ont été supplanté par les "transformers" pour le traitement du langage naturel. Leur principe est de tenir compte de l'état de la couche caché au temps précédent pour influencer la couche cachée au temps présent :

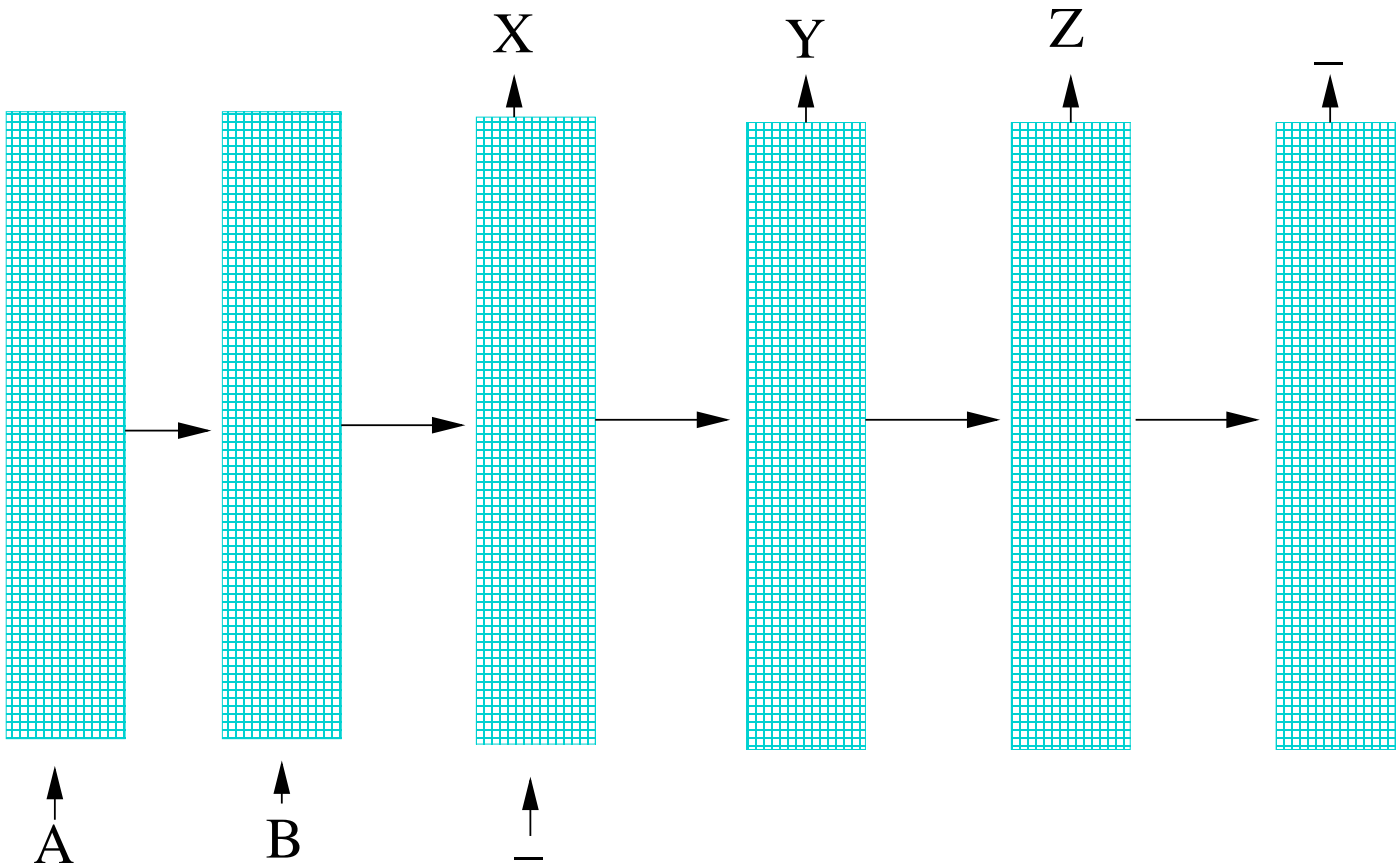
$$\begin{cases} \hat{Y}_t = f_\theta(h_t) \\ h_t = g_\theta(X_t, h_{t-1}) \end{cases}$$



Propriétés des réseaux récurrents

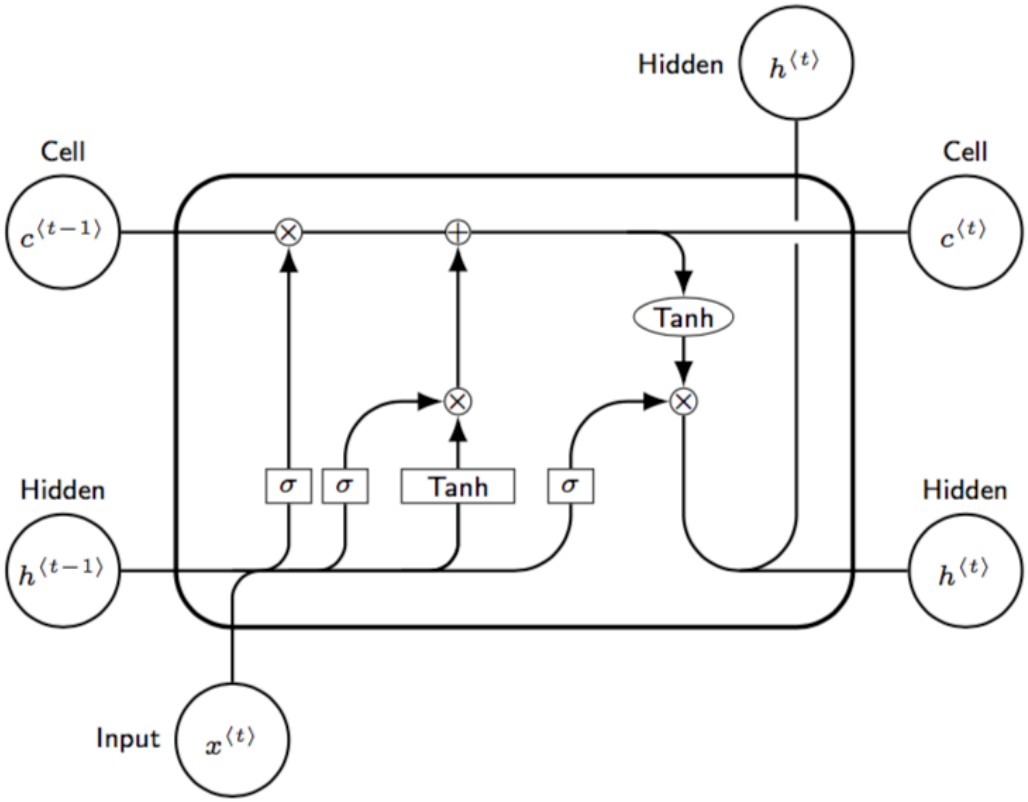
Les réseaux récurrents sont particulièrement adaptés aux séquences temporelles :

- Il peuvent mémoriser les observations passées.
- Les entrées et les sorties de ces réseaux peuvent être de longueur variables.



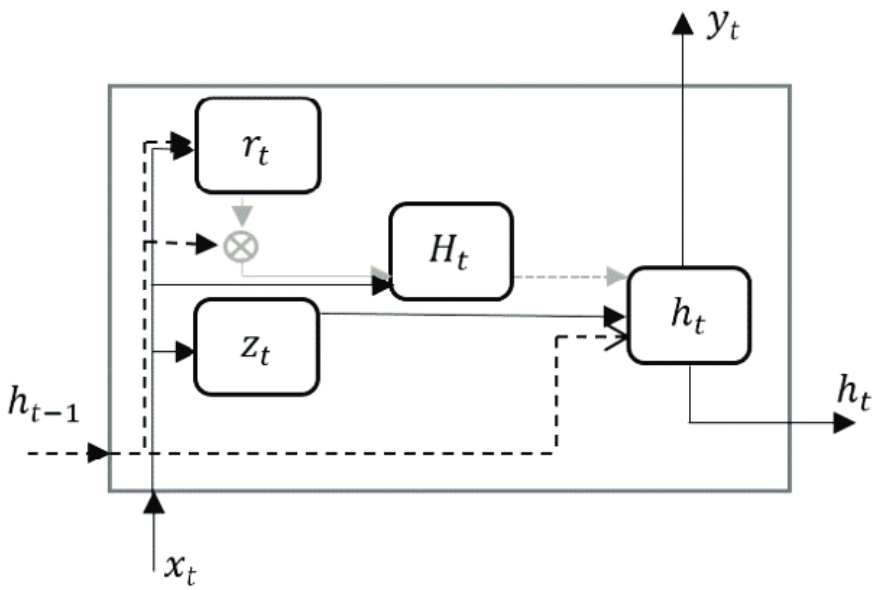
Modèle Long Short Time Memory (LSTM)

Certainement un des modèles le plus utilisé actuellement est le modèle LSTM. Celui-ci permet au réseau d'apprendre la profondeur de la mémoire utile pour une tâche donnée.



Modèle Gated Recurrent Unit (GRU)

C'est un modèle plus simple que Le modèle LSTM et qui a des performances comparables au LSTM.



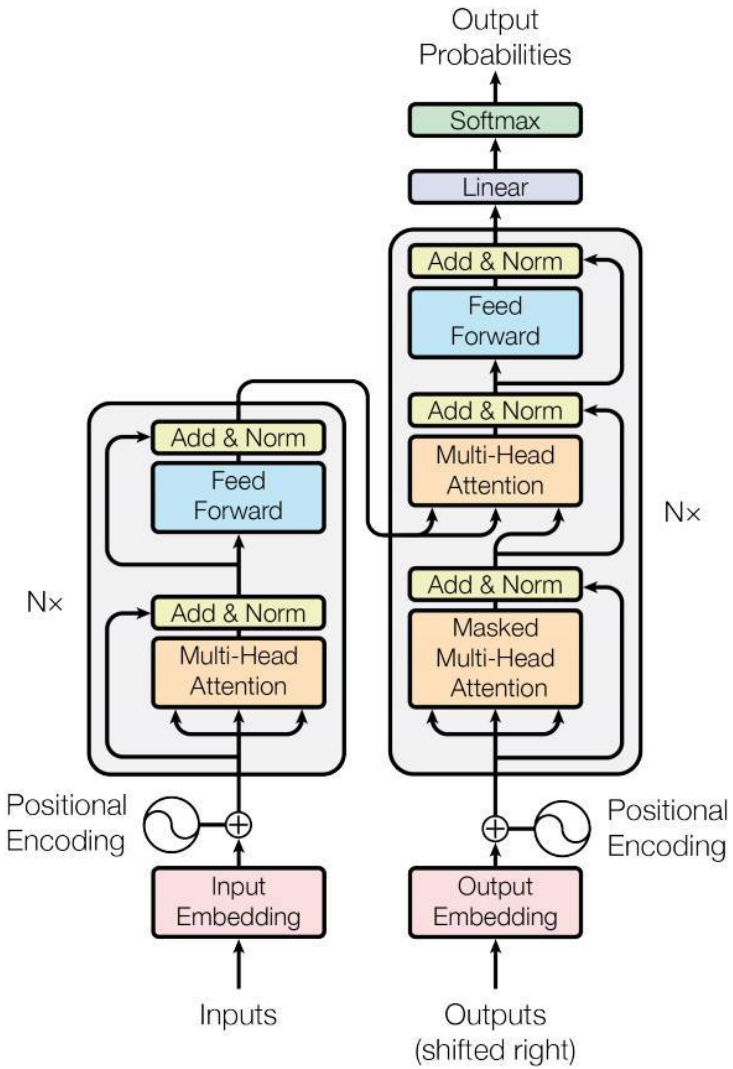
Les transformers

Ces réseaux de neurones sont l'état de l'art pour le traitement du langage naturel (Chatbot, analyse de sentiments, traduction, etc...). Voici un exemple de la traduction du francais en anglais d'un extrait de Wikipédia par "Google trad" :

- Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectés interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.
- A network of recurrent neurons is a network of artificial neurons with recurrent connections. A network of recurrent neurons consists of interconnected units (neurons) interacting non-linearly and for which there is at least one cycle in the structure. The units are connected by arches (synapses) that have a weight. The output of a neuron is a nonlinear combination of its inputs.

Principe des Transformers

Ces réseaux ont des modules d'attention qui permettent de tenir compte du contexte des mots.



Pytorch

Pytorch est une librairie Python libre et gratuite pour construire des modèles de Deep Learning.

- Ce langage est issu des équipes de recherche de Facebook.
- Il effectue des calculs tensoriels optimisés soit le le CPU mais aussi sur les carte graphiques (GPU)
- Il est dynamique, à tout moment on peut tester une petite partie du programme (comme R).
- Il permet d'échanger les tableaux de données facilement avec les autres librairie Python (numpy, etc...)
- Le calcul des gradients pour l'optimisation est automatisé et implicite.
- La plus grosse difficulté pour l'apprentissage de cette librairie est de comprendre que tous les calculs doivent pouvoir s'exécuter de façon massivement parallèle.

Tensorflow (Keras)

Tensorflow est aussi une librairie Python libre et gratuite pour construire des modèles de Deep Learning.

- Ce langage est issu des équipes de recherche de Google.
- Il effectue des calculs tensoriels optimisés soit le le CPU mais aussi sur les carte graphiques (GPU) ou bien TPU
- Il est statique, il faut donc faire tourner le programme en entier pour voir les résultats, cela le rend plus difficile à debugger que Pytorch.
- Il permet d'échanger les tableaux de données facilement avec les autres librairie Python (numpy, etc...)
- Le calcul des gradients pour l'optimisation est automatisé et implicite si on utilise sa "surcouche" Keras qui permet de programmer plus facilement les modèles.
- Keras est simple et efficace pour des modèles standards mais elle est moins souple que Pytorch si on veut faire des modèles plus exotiques.

Références

- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.
  - Il existe une traduction française.
  - Site web associé : https ://www.deeplearningbook.org/
- Cours du collège de France :
  - Yann Lecun : https ://www.college-de-france.fr/site/yann-lecun/course-2015-2016.htm
  - Stéphane Mallat : https ://www.college-de-france.fr/site/stephane-mallat/course.htm
- Pytorch : https ://pytorch.org/
  - On peut y télécharger le programme, pour l'installer sur l'ordinateur.
  - Trouver la documentation.
  - Trouver des tutoriaux.
- Tensorflow : https ://www.tensorflow.org/
  - On peut y télécharger le programme, pour l'installer sur l'ordinateur.
  - Trouver la documentation.
  - Trouver des tutoriaux.