

20 octobre 2016

Chapitre 1

Deep Learning

1.1 Machine Learning

1.1.1 Principes généraux

Le Machine Learning, également connu sous le terme d'apprentissage automatique, est un sous-domaine de l'intelligence artificielle, ayant vu le jour dans les années 1950[1][2]. Son champ d'application est aujourd'hui très vaste, et sa principale limite réside en la quantité d'informations exploitables, disponible au sein d'un domaine donné.

L'accroissement de la collecte, du stockage et de la mise à disposition des données que nous connaissons depuis quelques années a permis l'essor de ces algorithmes et leur transposition à de nombreux problèmes :

- la classification de contenu audio-visuel au sens large, allant de l'image au sujet d'un texte ou d'une revue
- le filtrage de contenus, tels que les spams ou les intrusions sur les systèmes d'informations
- le tri et la sélection d'informations les plus pertinentes à délivrer via la publicité ou les flux de contenus des médias sociaux
- l'analyse de sentiments

Plus précisément, ce concept recouvre les systèmes constitués de paramètres réglables, typiquement vus sous la forme de valeurs vectorielles, en vue de fournir la sortie attendue pour une série de valeurs données en entrée. En outre, ce type d'apprentissage se distingue par sa capacité à ajuster ses paramètres de manière autonome, en se basant sur l'expérience des données précédemment traitées.

Dans ce qui va suivre, cette technique sera abordée à la lumière des problèmes de reconnaissance de formes, les entrées dont il sera alors question étant des images ou des vidéos.

1.1.2 Systèmes de reconnaissance de formes

L'architecture d'un système de reconnaissance de formes comprend deux éléments principaux :

- un extracteur des caractéristiques de l'entrée
- un classifieur qui donne le résultat en sortie, associant généralement l'entrée à une classe

Dans les années 50, les premiers modèles de reconnaissance étaient constitués d'un extracteur de caractéristiques "fait-main", peu modulaire et fastidieux à implémenter[3]. C'est lui qui est chargé de traduire l'image, ou partie de l'image, en une représentation vectorielle des motifs qu'elle contient.

Dans un deuxième temps, le classifieur détermine la classe résultante selon une somme pondérée des composantes du vecteur de caractéristiques. Une valeur seuil est fixée de telle sorte à ce que la sortie soit activée ou non en fonction du calcul précédent. Lorsque des erreurs sont repérées en sortie, cet algorithme va réajuster ses paramètres internes en vue d'améliorer les réponses suivantes. On parle alors d'entraînement.

Dans le cadre d'une classification dite linéaire, ces réglages s'effectuent sur la valeur des poids associés aux caractéristiques. L'enjeu du classifieur est de réduire la différence entre les résultats attendus \hat{y} , et effectifs y . La phase d'entraînement revient à minimiser une fonction objectif¹, en modulant les pondérations (θ). D'un point de vue mathématique, la fonction objectif peut différer en fonction de l'approche adoptée. On la trouvera par exemple sous forme de moindres carrés de résidus[4] :

$$J(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

ou d'erreur quadratique moyenne[5] :

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Un tel comportement correspond à un apprentissage supervisé, où la réponse correcte est connue, permettant à la machine de s'améliorer. Une fois cette phase achevée, le système est théoriquement apte à classifier avec exactitude de nouvelles entrées, jusqu'alors inconnues : c'est la généralisation.

D'autre part, comme représenté figure 1.1, l'approche par apprentissage profond vise à étendre les capacités d'entraînement et de généralisation à l'ensemble de la chaîne de reconnaissance de formes. Notre étude se base sur ce dernier modèle, où l'extracteur de caractéristiques, aussi appelé noyau, peut être entraîné.

1. Dans le framework Caffe, une telle fonction est désignée sous le nom de loss function.

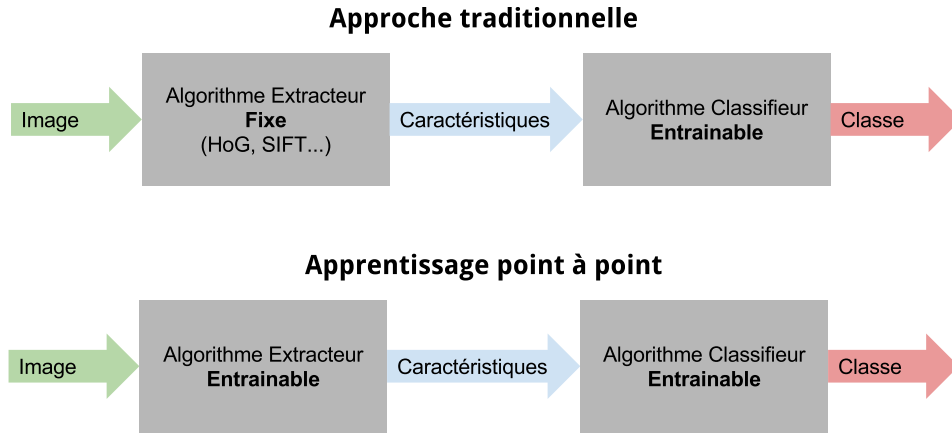


FIGURE 1.1: Différentes approches de reconnaissance de formes

1.2 Réseaux de neurones

1.2.1 Neurone artificiel

A la base des réseaux de neurones se trouve le concept de neurone formel représenté sur la figure 1.2.

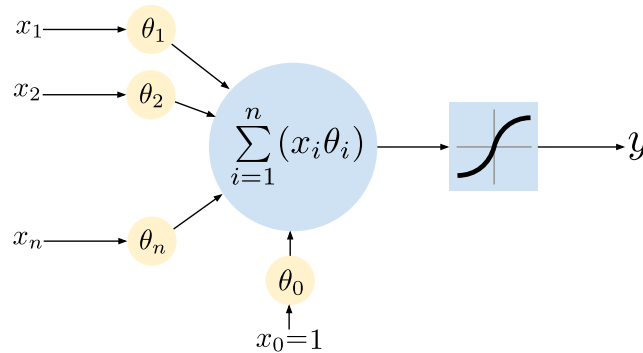


FIGURE 1.2: Schéma d'un neurone formel

Nous retrouvons les entrées x_i qui correspondent dans notre cas aux i composantes du vecteur de caractéristiques donné par l'extracteur. Ces entrées sont ensuite sommées avec leur poids θ_i (ou coefficient synaptique) respectif, ainsi qu'un coefficient θ_0 appelé biais. Le résultat pondéré est ensuite transmis à une fonction d'activation non linéaire[6]. Celle-ci activera la synapse suivante (ici, la sortie), si son résultat dépasse un seuil donné.

En pratique, plusieurs fonctions d'activation peuvent être choisies au regard de leurs caractéristiques. Parmi les plus utilisées, on peut citer les fonctions

sigmoïdes et tangentes hyperboliques. Leur efficacité tient notamment du fait qu'elles soient indéfiniment et rapidement dérivables et non polynômiales.

Admettons que ce système contienne des entrées sous forme de vecteurs à n dimensions, étiquetés selon les classes C_0 et C_1 . Le but de l'entraînement sera d'obtenir en sortie :

$$\begin{cases} y = 0, \text{ si } x \in C_0 \\ y = 1, \text{ si } x \in C_1 \end{cases}, \forall x$$

sachant que,

$$y = \phi\left(\sum_{i=1}^n (x_i \theta_i) + \theta_0\right), \text{ avec } \phi \text{ fonction d'activation}$$

le problème revient à trouver la valeur optimale du vecteur θ , permettant de séparer linéairement les classes. La figure illustre ce concept dans le cadre d'une classification d'éléments à deux dimensions, distribués entre deux classes.

1.2.2 Réseau multicouches

Les problèmes traditionnels de classification nécessitent des modèles plus élaborés, car ils sont souvent insolubles à l'aide de simples régressions linéaires. On dit que l'ensemble des éléments constituant nos classes sont non linéairement séparables (les résultats de la fonction XOR par exemple). On organise alors les neurones formels en successions de couches, ce qui permettra de diviser le plan en n sous-ensembles, avec $n > 2$. On appelle typiquement *couche d'entrée* la première couche du réseau, *couches cachées* les couches internes au réseau et *couche de sortie* la dernière couche. Issue du site de tensorflow[7] (la librairie de calcul de machine learning développée par Google[8]), la capture d'écran 1.3 illustre les jeux de données pour lesquelles les régressions linéaires s'avèrent inefficaces.

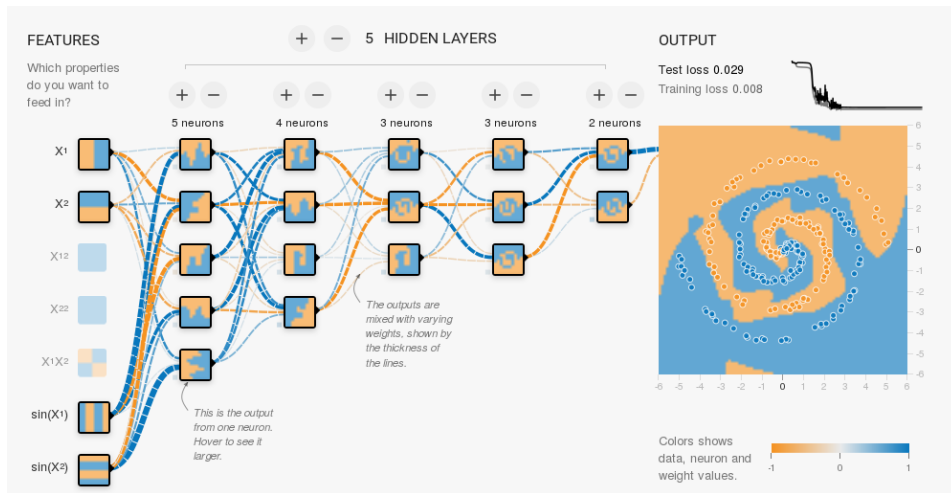


FIGURE 1.3: Schéma d'un neurone formel

Dans cette configuration, les interactions entre les différents neurones, et les différentes couches, correspondent à ce qui a été décrit précédemment. L'information se propage "en avant"

1.2.3 Autres

deep learning : connu depuis 1980 mais répandu depuis 2012
perceptron
capacité de généralisation : faculté d'un système à produire des résultats corrects sur des entrées inconnues, après une phase d'entraînement
rétropopagation : depuis les années 80
gardien : incrément ou décrement calculable de l'erreur correspondant à la modification des poids θ lors de la phase d'entraînement
vecteur de caractéristiques d'entraînement

Glossary

entraînement

Lorem ipsum dolor sit amet.

vecteur de caractéristiques

En reconnaissance de formes, un vecteur de caractéristiques est une représentation numérique de propriétés visuelles d'un objet. Il peut par exemple être constitué de la quantité de lignes présentes dans l'image selon leur orientation, chaque orientation constituant une dimension du vecteur. Ces représentations permettent d'effectuer des traitements statistiques, comme des régressions linéaires.

Bibliographie

- [1] Bernard Marr, *A short history of machine learning*, Forbes.com, 19 février 2016, <http://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/fdc5dfd323ff>.
- [2] John McCarthy, *Arthur Samuel : Pioneer in Machine Learning*, Stanford University, Stanford, California, USA - Stanford Computer Science, <http://infolab.stanford.edu/pub/voy/museum/samuel.html>.
- [3] Yann LeCun, *Recherches sur l'intelligence artificielle*, Collège de France, France, 7p, <https://www.college-de-france.fr/site/yann-lecun/Recherches-sur-l-intelligence-artificielle.htm>.
- [4] Wikipedia, *Least squares*, https://en.wikipedia.org/wiki/Least_squares.
- [5] Wikipedia, *Mean squared error*, https://en.wikipedia.org/wiki/Mean_squared_error.
- [6] Wikipedia, *Artificial neuron*, https://en.wikipedia.org/wiki/Artificial_neuron.
- [7] Playgroung TensorFlow, *Tinker with a neural network in your browser*, <http://playground.tensorflow.org>.
- [8] TensorFlow, *Open source software library for machine intelligence*, <https://www.tensorflow.org/>.
- [9] Matthew D Zeiler, Rob Fergus, 28 Nov 2013, Cornell University Library, Ithaca, NY, USA - Computer vision and pattern recognition, 11p *Visualizing and Understanding Convolutional Networks, v3*, <https://arxiv.org/abs/1311.2901>.