

Introduction aux réseaux de neurones convolutifs : qu'est-ce que le Machine Learning ?— 1^e partie

Par Ole Dreessen, Staff Engineer, Field Applications, Analog Devices

Synthèse

Le monde de l'intelligence artificielle (IA) évolue rapidement, rendant possible un nombre croissant d'applications longtemps inenvisageables ou dont la mise en œuvre s'avérerait hautement complexe. Dans cette série d'articles, nous tenterons d'expliquer le fonctionnement des réseaux de neurones convolutifs (CNN — *Convolutional Neural Networks*), ainsi que leur importance dans l'apprentissage automatique au sein des systèmes d'IA. Les réseaux de neurones convolutifs sont des outils puissants qui permettent d'extraire des motifs (*patterns*) à partir de données complexes, ce qui inclut entre autres la reconnaissance de formes complexes dans des signaux audio ou des images. Le présent article décrit les avantages des réseaux de neurones convolutifs par rapport à la programmation linéaire classique. Dans un prochain article intitulé « *Formation sur les réseaux de neurones convolutifs : qu'est-ce que l'apprentissage automatique ? 2^e partie* », nous aborderons la manière dont les modèles CNN sont entraînés. Un troisième article, enfin, sera consacré à un cas d'utilisation spécifique et au test du modèle à l'aide d'un microcontrôleur fondé sur l'intelligence artificielle.

Qu'est-ce qu'un réseau de neurones convolutif ?

Un réseau de neurones est un système ou une structure de neurones qui permet à l'intelligence artificielle (IA) de mieux comprendre les données en vue de résoudre des problèmes d'une grande complexité. S'il existe de nombreux types de réseaux de neurones, cette série d'articles porte uniquement sur les réseaux de neurones convolutifs (CNN), dont les principaux domaines d'application sont la reconnaissance des motifs et la classification des objets contenus dans les données d'entrée. L'expression « réseau de neurones convolutif » désigne une catégorie de réseaux de neurones artificiels utilisés dans l'apprentissage profond (*deep learning*). Ces réseaux sont composés d'une couche d'entrée, de plusieurs couches convolutives et d'une couche de sortie. Les couches convolutives sont les composants les plus importants, dans la mesure où elles utilisent un ensemble unique de poids et de filtres qui permettent au réseau d'extraire certaines caractéristiques à partir de données d'entrée pouvant se présenter sous différentes formes — images, son ou texte. C'est ce processus d'extraction qui permet au réseau CNN d'identifier des motifs dans les données. En extrayant des caractéristiques présentes dans les données, les CNNs donnent aux ingénieurs la possibilité de créer des applications plus efficaces et plus efficientes. Mais pour mieux comprendre le fonctionnement de ces réseaux, nous nous intéresserons dans un premier temps à la programmation linéaire classique.

Programmation linéaire dans l'ingénierie de contrôle classique

Dans le domaine de l'ingénierie de contrôle, la tâche consiste à lire des données collectées par un ou plusieurs capteurs, à les traiter et y réagir en fonction de règles définies, avant d'afficher ou de transmettre les résultats. Prenons un exemple : un régulateur de température mesure la température toutes les secondes grâce à un microcontrôleur qui lit les informations relevées par un capteur. Les valeurs fournies par ce dernier sont utilisées comme données

d'entrée par le système de régulation en boucle fermée et comparées à la température de consigne dans une boucle. C'est un exemple d'exécution linéaire exécutée par un microcontrôleur. Cette technique fournit des résultats concluants à partir d'un ensemble de valeurs préprogrammées et réelles. Or, des probabilités interviennent dans le fonctionnement des systèmes fondés sur l'intelligence artificielle.

Traitement des signaux et reconnaissance de motifs complexes

De nombreuses applications fonctionnent en utilisant des données d'entrée qui doivent, dans un premier temps, être interprétées par un système de reconnaissance des motifs (*pattern recognition*). La reconnaissance des motifs peut être appliquée à différentes structures de données. Dans les exemples que nous avons choisis, nous nous limitons à des structures de données unidimensionnelles (1D) et bidimensionnelles (2D) : signaux audio, électrocardiogrammes (ECG), photopléthysmographie (PPG), détection de vibrations pour les images et les données unidimensionnelles ; images thermiques et diagrammes en cascade pour les données bidimensionnelles.

En ce qui concerne la reconnaissance des motifs utilisée pour les exemples cités, il est très difficile de convertir l'application en code classique pour le microcontrôleur. C'est le cas de la reconnaissance d'un objet (un chat, par exemple) à l'intérieur d'une image. Dans ce scénario, peu importe si l'image à analyser provient d'un enregistrement antérieur ou si elle vient d'être lue par le capteur d'un appareil-photo. Le logiciel d'analyse lance une recherche basée sur des règles dans le but d'identifier des motifs pouvant correspondre à ceux d'un chat : oreilles pointues typiques, nez triangulaire ou moustaches. Si ces caractéristiques peuvent être reconnues dans l'image, le logiciel signale qu'il a découvert un chat. Dès lors, certaines questions se posent : que ferait le système de reconnaissance des motifs si le chat était présenté de dos ? Que se passerait-il s'il n'avait pas de moustaches, ou s'il avait perdu ses pattes dans un accident ? Malgré l'improbabilité de ces exceptions, le code de reconnaissance des motifs devrait vérifier un nombre important de règles supplémentaires couvrant toutes les anomalies possibles et imaginables. Même dans notre exemple simple, les règles définies par le logiciel deviendraient rapidement très nombreuses.

Comment l'apprentissage automatique remplace les règles traditionnelles

L'idée derrière l'IA est de reproduire l'apprentissage humain à petite échelle. Au lieu de formuler un grand nombre de règles de type « si-alors », nous modélisons une machine universelle de reconnaissance des motifs. La principale différence entre les deux approches réside dans le fait que contrairement à un ensemble de règles, l'intelligence artificielle ne fournit pas un résultat clair. À titre d'exemple, au lieu de dire « *J'ai reconnu un chat dans l'image* », le Machine Learning produira le résultat suivant : « *Il y a une probabilité de 97,5 % que l'image montre un chat. Il est possible qu'il s'agisse d'un léopard (2,1 %) ou d'un tigre (0,4 %)* ». En d'autres termes, il revient au développeur de ce type d'application de choisir à l'issue du processus de reconnaissance des motifs. Un seuil de décision est utilisé à cet effet.

Autre différence, une machine de reconnaissance des motifs ne dispose pas de règles fixes ; au contraire, elle est « entraînée ». Au cours de ce processus d'apprentissage, un réseau de neurones est confronté à un nombre élevé d'images représentant des chats. À la fin du processus, ce réseau est en mesure de reconnaître de manière indépendante si un chat est ou non présent sur une image. Or, la reconnaissance du futur ne sera pas limitée à des images déjà connues. C'est pourquoi ce réseau de neurones doit être implanté dans un microcontrôleur.

À quoi ressemble l'intérieur d'une machine de reconnaissance des motifs ?

Dans le domaine de l'intelligence artificielle, un réseau de neurones ressemble à son équivalent biologique du cerveau humain. Un neurone compte plusieurs entrées et une unique sortie. Fondamentalement, un neurone ne constitue ni plus ni moins qu'une transformation linéaire des entrées — multiplication des entrées par des nombres (poids, w) et addition d'une

constante (biais, b) —, suivie d'une fonction non linéaire fixe, également appelée « fonction d'activation ».¹ Seule composante non linéaire du réseau, cette fonction est utilisée pour définir la plage de valeurs dans laquelle un neurone artificiel se déclenche. La fonction d'un neurone peut être décrite mathématiquement de la manière suivante :

$$Out = f(w*x+b) \quad (1)$$

où f désigne la fonction d'activation ; w , le poids ; x , les données d'entrée ; et b , le biais. Les données peuvent se présenter sous la forme de produits scalaires individuels, de vecteurs ou de matrices. La figure 1 représente un neurone à trois entrées avec fonction d'activation ReLU². Les neurones d'un réseau sont toujours disposés en couches.

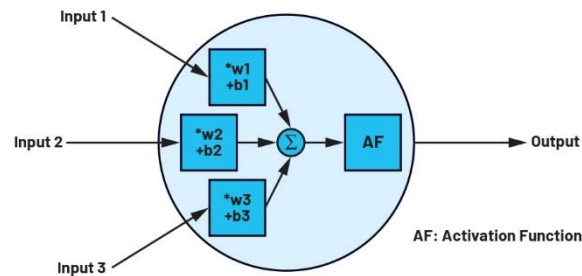


Figure 1. Représentation d'un neurone à trois entrées et une sortie

Comme nous l'avons indiqué, les réseaux de neurones convolutifs sont utilisés pour la reconnaissance des motifs et la classification des objets contenus dans des données d'entrée. Ils se composent de plusieurs sections : une couche d'entrée, plusieurs couches cachées et une couche de sortie. La figure 2 représente un petit réseau comprenant trois entrées, une couche cachée avec cinq neurones, et une couche de sortie avec quatre sorties. Toutes les sorties des neurones sont connectées à toutes les entrées de la couche suivante. Le réseau illustré à la figure 2 ne permet pas de traiter des tâches significatives et est uniquement représenté à des fins de démonstration. Même dans ce réseau de petites dimensions, l'équation utilisée pour décrire le réseau compte 32 biais et 32 poids.

Un réseau de neurones CIFAR³ est un type de réseau convolutif largement utilisé dans la reconnaissance d'images. Il comprend deux types de couches : des couches convolutives et des couches de « pooling »⁴, toutes deux étant utilisées avec une grande efficacité dans l'entraînement des réseaux de neurones. Comme son nom l'indique, la couche convolutive exécute une opération mathématique appelée « convolution » afin d'identifier des motifs dans un ensemble de valeurs de pixels. La convolution se produit dans les couches cachées, comme le montre la figure 3. Ce processus est répété à plusieurs reprises jusqu'à ce que le niveau de précision souhaité soit atteint. Il est à noter que la valeur de sortie d'une opération de convolution est toujours particulièrement élevée si les deux valeurs d'entrée à comparer (image et filtre, dans ce cas) sont similaires. On parle alors de « matrice de filtrage », de « noyau de filtrage » ou simplement de « filtre ». Les résultats sont ensuite transmis à la couche de pooling qui génère une carte de caractéristiques, c'est-à-dire une représentation des données d'entrée qui identifie les caractéristiques importantes. Cette carte est également considérée comme une matrice de filtrage. Après l'apprentissage — dans l'état opérationnel du réseau —, ces cartes de caractéristiques sont comparées aux données d'entrée. Étant donné que les cartes contiennent des caractéristiques spécifiques aux classes d'objets qui sont comparées aux images d'entrée, la sortie des neurones se déclenchera uniquement si

¹ Les fonctions les plus couramment utilisées sont les fonctions sigmoid, tanh et ReLU.

² ReLU (*Rectified Linear Unit*, soit Unité Linéaire Rectifiée) : la dérivée de cette fonction est nulle lorsque les valeurs d'entrée sont négatives, et inchangée lorsque la valeur est positive. [Les ReLUs laissent toutes les valeurs positives passer inchangées et attribuent simplement « 0 » aux valeurs négatives.]

³ Le CIFAR est une organisation de recherche mondiale basée au Canada.

⁴ La couche de pooling (POOL) permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage). [wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif](https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif).

les contenus sont similaires. En combinant ces deux approches, le réseau CIFAR peut être utilisé pour reconnaître et classer divers objets dans une image avec un haut niveau de précision.

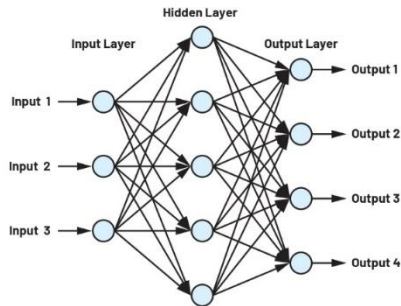


Figure 2. Réseau de neurones de petites dimensions.

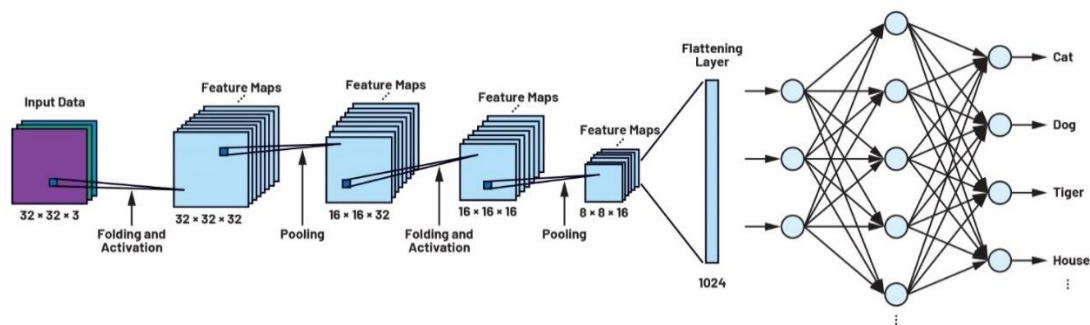


Figure 3. Modèle du réseau CIFAR entraîné avec le jeu de données CIFAR-10.

La base d'images CIFAR-10 est un jeu de données spécifique couramment utilisé pour l'apprentissage des réseaux de neurones CIFAR. Elle se compose de 60 000 images en couleur mesurant chacune 32x32, réparties en 10 classes et collectées dans différentes sources : pages Web, groupes de discussion et collections d'images personnelles. Chaque classe comporte 6 000 images réparties équitablement entre les ensembles d'apprentissage, de test et de validation, ce qui en fait un ensemble idéal pour tester de nouvelles architectures de vision par ordinateur et autres modèles d'apprentissage automatique.

Les réseaux de neurones convolutifs se distinguent essentiellement des autres types de réseaux par la manière dont ils traitent les données. Le filtrage permet d'examiner successivement les propriétés des données d'entrée. Plus le nombre de couches convolutives connectées en série est élevé, plus le niveau de détails pouvant être reconnus augmente. Le processus débute par les propriétés élémentaires d'un objet (bords ou points) après la première convolution, avant de passer aux structures détaillées (angles, cercles, rectangles, etc.) après la deuxième convolution. À partir de la troisième convolution, les caractéristiques représentent des motifs complexes qui ressemblent à des parties d'objets dans les images et correspondent généralement à une classe d'objets donnée. Dans notre exemple initial, il s'agit des moustaches ou des oreilles d'un chat. La visualisation des cartes de caractéristiques (figure 4) n'est pas indispensable à l'application elle-même, mais elle permet de bien comprendre la convolution.

Même les petits réseaux de type CIFAR comptent plusieurs centaines de neurones par couche et de nombreuses couches connectées en série. Le nombre de poids et de biais nécessaires augmente rapidement en fonction de la complexité et de la taille du réseau. Dans l'exemple de la base d'images CIFAR-10 illustrée à la figure 3, 200 000 paramètres nécessitent un ensemble déterminé de valeurs pendant le processus d'entraînement. Les cartes de caractéristiques peuvent bénéficier d'un traitement supplémentaire de la part des couches de pooling, ce qui réduit le nombre de paramètres à entraîner tout en préservant les informations importantes.

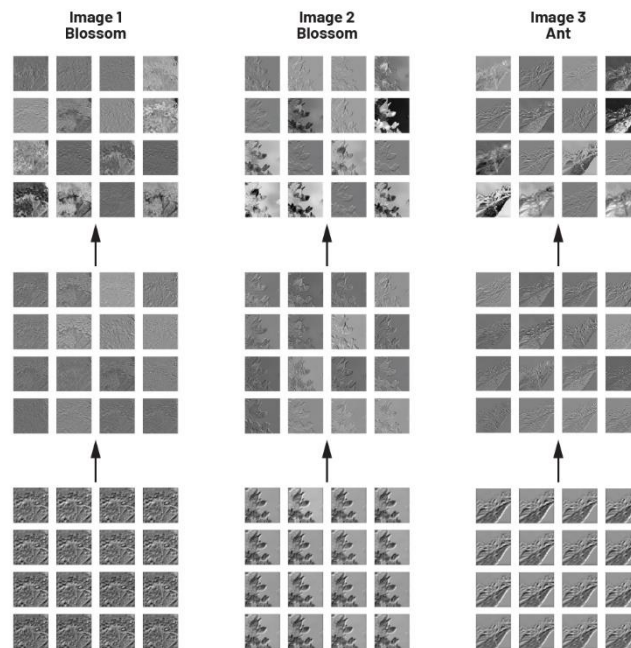


Figure 4. Cartes de caractéristiques d'un réseau de neurones convolutif.

Comme nous l'avons mentionné précédemment, le mécanisme de pooling, également appelé « sous-échantillonnage », a lieu après chaque convolution. Le pooling permet de réduire les dimensions des données. Si l'on observe les cartes de caractéristiques présentées à la figure 4, on remarque que les grandes régions contiennent peu, voire pas d'information significative, ce qui s'explique par le fait que les objets ne constituent pas la totalité, mais seulement une petite partie de l'image. Le reste de l'image n'est pas utilisé dans cette carte de caractéristiques et, à ce titre, n'est pas pertinent pour la classification. Dans une couche de pooling, le type de pooling (maximum ou moyen) et la taille de la matrice de fenêtre sont spécifiés. La matrice de fenêtre est déplacée progressivement sur les données d'entrée au cours du processus de pooling. Dans le cas d'un sous-échantillonnage maximum, par exemple, c'est la plus grande valeur de données de la fenêtre qui sera retenue, toutes les autres valeurs étant écartées. Cette méthode permet de réduire continuellement le nombre des données, de sorte qu'elles forment à terme, avec les convolutions, les propriétés uniques de la classe d'objets correspondante.

Toutefois, le résultat de ces groupes de convolution et de pooling représente un nombre conséquent de matrices bidimensionnelles. Pour remplir concrètement notre objectif de classification, nous devons convertir les données bidimensionnelles en un long vecteur unidimensionnel. La conversion est effectuée dans une couche dite d'aplatissement, suivie d'une ou deux couches entièrement connectées. Les neurones des deux derniers types de couches sont similaires à la structure illustrée à la figure 2. La dernière couche de notre réseau de neurones comporte autant de sorties que de classes à distinguer. En outre, dans la dernière couche, les données sont également normalisées pour générer une distribution de probabilités (97,5 % de chat, 2,1 % de léopard, 0,4 % de tigre, etc.).

Cette étape conclut la modélisation de notre réseau de neurones. Toutefois, les poids et le contenu des matrices (noyau et filtrage) sont encore inconnus et doivent être déterminés en entraînant le réseau pour que le modèle fonctionne. Ce point sera expliqué dans l'article suivant intitulé « *Formation des réseaux de neurones convolutifs : qu'est-ce que le Machine Learning ? 2^e partie* ». Dans la troisième partie, nous expliquerons l'implémentation matérielle du réseau de neurones évoquée précédemment, par exemple pour la reconnaissance des chats. Nous utiliserons dans cette optique le microcontrôleur pour applications d'intelligence artificielle MAX78000 avec accélérateur CNN matériel développé par Analog Devices.

À propos de l'auteur

Ole Dreessen est ingénieur Applications chez Analog Devices. Avant de rejoindre la Société en 2014, Ole a occupé différents postes chez Avnet Memec et Macnica dans les domaines des technologies de communications et des microprocesseurs de haute performance. Il possède une solide expertise des sujets liés aux microcontrôleurs et à la sécurité, et intervient régulièrement dans le cadre de conférences et d'évènements auprès des distributeurs. Pendant son temps libre, Ole participe aux activités du Chaos Computer Club, où il a travaillé sur des concepts tels que l'ingénierie inverse et la sécurité embarquée.