

Les réseaux de neurones et l'apprentissage profond

1.1 Introduction

Un réseau de neurones artificiels[1] est un système dont la conception est à l'origine, inspirée du fonctionnement des neurones biologiques. Si nous comparons l'ordinateur et le cerveau, pratiquement, l'ordinateur est bien plus puissant. Il est largement supérieur au cerveau humain pour effectuer des calculs, mémoriser et classer des informations. Mais, pour des opérations cérébrales comme raisonner, analyser son environnement et communiquer, l'être humain dépasse très largement la machine.

Les expériences montrent que l'être humain peut reconnaître l'image d'une personne ou d'un objet en une fraction de seconde. Ce qui correspond au temps d'une communication entre les neurones du cerveau. Un ordinateur suivant l'architecture de Von Neuman ne peut rien effectuer dans un temps aussi court, sur cette tâche spécifique. C'est alors que rentrent dans la partie, les réseaux de neurones artificiels. Basés sur l'architecture d'un cerveau, ce paradigme informatique permet de concurrencer l'homme sur les tâches citées précédemment.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage. Ils sont placés dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes qui permettent de créer des classifications rapides (comme par exemple : les réseaux de Kohonen). En modélisation des circuits biologiques, ils permettent de tester quelques hypothèses fonctionnelles issues de la neurophysiologie, ou encore les conséquences de ces hypothèses pour les comparer au réel.

1.2 Description des réseaux de neurones

Il existe plusieurs types de réseaux suivant la façon dont sont connectés les neurones entre eux. En informatique, les neurones sont calculés via un programme informatique, mais ils peuvent être parfois réalisés sur un circuit électronique. Un réseau de neurones possède deux états, un état d'apprentissage et un état de fonctionnement optimal. Afin de former le réseau, il doit y avoir une période d'enseignement et de répétition, tout comme un enfant

qui développe son cerveau grâce à l'enseignement de ses parents.

Les réseaux de neurones apprennent par un processus de rétro propagation. Cela implique la comparaison de la sortie du réseau de neurones avec la sortie pour laquelle il a été conçu. Cette différence sert à modifier les poids des connexions entre les neurones du réseau et à le rendre plus précis. Une fois le réseau suffisamment calibré, il atteint un niveau d'autonomie exceptionnel où il n'est plus nécessaire de le superviser.

Prenons un exemple, où nous supposons que nous avons enseigné à un réseau de neurones à distinguer une voiture automobile d'un engin de travaux publics. Pour cela nous lui avons montré 15 voitures et 15 engins différents, en lui expliquant bien à quoi correspondait chaque image.

Il atteint alors un état d'autonomie et est capable de distinguer par lui-même si les nouvelles entrées correspondent plutôt à une voiture ou à un engin. Si on lui montre une variante, par exemple, une voiture plus grande il va être capable de catégoriser cet objet, en voiture et non en engin de travaux publics. Simplement sur la seule base de son expérience et sans aucune aide, tout comme un être humain. Au final, nous avons appris à un ordinateur à apprendre. . . Les réseaux de neurones sont utilisés dans la vie de tous les jours pour plusieurs problématiques complexes comme :

- L'estimation des trafics maritime, ferroviaire, routier ;
- Les prévisions météorologiques ;
- La classification d'espèces animales et végétales, par analyse de l'ADN ;
- Les banques, afin de vérifier les montants des transactions financières ;
- La poste pour trier le courrier en fonction du code postal ;
- Les opérations boursières et bien d'autres utilisations ;

La recherche sur les réseaux de neurones a gagné en popularité et a atteint le sommet au début des années 1990. Jusqu'en 2006, les scientifiques n'ont pas su former les réseaux de neurones à dépasser les approches plus traditionnelles, sauf pour quelques problèmes spécialisés. Ce qui a changé à partir de cette date, c'est la découverte de techniques d'apprentissage dans les réseaux de neurones profonds. Ces techniques sont maintenant connues comme apprentissage profond .

Elles ont été développées d'avantage et aujourd'hui les réseaux de neurones et l'apprentissage profond ont obtenus des performances exceptionnelles sur de nombreux problèmes importants. Ils sont déployés à grande échelle par des entreprises telles que Microsoft, Google , IBM et Facebook.

Dans l'approche conventionnelle de la programmation, nous disons à l'ordinateur ce qu'il faut faire, diviser un grand problème en de nombreuses petites tâches définies avec précision, que l'ordinateur peut facilement effectuer.

En revanche, dans un réseau de neurones, nous ne disons pas à l'ordinateur comment résoudre

notre problème. Au lieu, il apprend des données d'observation, trouvant sa propre solution au problème en question.

1.3 Les neurones biologiques et artificiels

Le cerveau est l'organe le plus complexe du corps humain. L'unité fondamentale du cerveau humain est le neurone. Un petit morceau de cerveau de la taille du grain de riz, contient plus de 10. 000 neurone.[2] A sa base, le neurone est optimisé pour recevoir des informations d'autres neurones, traiter ces informations de manière unique et envoyer son résultat à d'autres cellules.

Le neurone reçoit ses entrées le long d'une structure en forme d'antenne appelée dendrite. Après avoir été pondéré par la force de leurs connexions, les entrées sont additionnées dans le corps de la cellule. Cette somme est ensuite transformée en un nouveau signal qui se propage le long de l'axone de la cellule et envoyée à d'autres cellules.

Nous pouvons traduire cette compréhension fonctionnelle des neurones de notre cerveau en un modèle artificiel que nous pouvons représenter sur notre ordinateur. Dans le domaine des technologies de l'information, un réseau de neurones est un système logiciel et / ou matériel qui imite le fonctionnement des neurones biologiques. Les réseaux neuronaux,[3] aussi appelés réseaux de neurones artificiels (RNA ou ANN, en anglais).

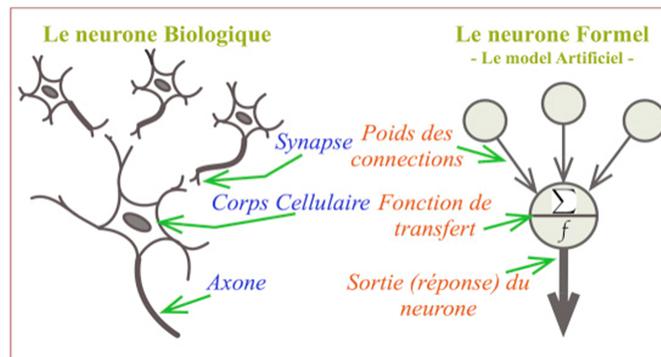


FIGURE 1.1 – Mise en correspondance neurone biologique /neurone artificiel

1.4 Le neurone formel

Le neurone formel[4][5] est un dispositif à plusieurs entrées et une sortie, qui prend pour modèle schématiquement le neurone biologique : il correspond respectivement aux dendrites et au point de départ de l'axone (cône d'émergence du neurone biologique).

Le neurone formel est une modélisation mathématique sous forme d'une fonction algébrique non linéaire, qui reprend les principes du fonctionnement du neurone biologique et dont la

valeur dépend des paramètres appelés coefficients ou poids. Il constitue l'unité minimale d'un réseau de neurones artificiels.

1.4.1 Le perceptron simple

Un réseau de neurones est une structure de réseau constituée d'un nombre de noeuds interconnectés par des liaisons directionnelles. Chaque noeud représente une unité de traitement et les liaisons représentent les relations causales entre les noeuds.

Le réseau le plus simple est celui de neurones monocouche appelé perceptron terme forgé par Franck Rosenblatt en 1958 [6] et l'assimilant à un modèle de rétine artificiel. Ce réseau est représenté par une couche de neurones d'entrée et de plusieurs neurones de sortie. Il constitue l'un des tous premiers algorithmes d'apprentissage supervisé. Il s'agit d'un neurone artificiel inspiré par la théorie cognitive de Friedrich Hayek et celle de Donald Hebb [7]. Dans sa version la plus simple, le perceptron n'a qu'une seule sortie.

La structure d'un perceptron est représentée dans la figure 1.2. Les valeurs d'entrée (x_1, x_2, \dots, x_n) et les poids associés aux entrées ($w_{1j}, w_{2j}, \dots, w_{nj}$) sont les variables de la fonction de combinaison qui détermine la somme pondérée du neurone (net_j). Cette somme est ensuite passée à la fonction d'activation qui détermine la valeur de sortie du neurone (o_j). Enfin, le neurone corrige ses poids afin de s'approcher de la valeur attendue (S).

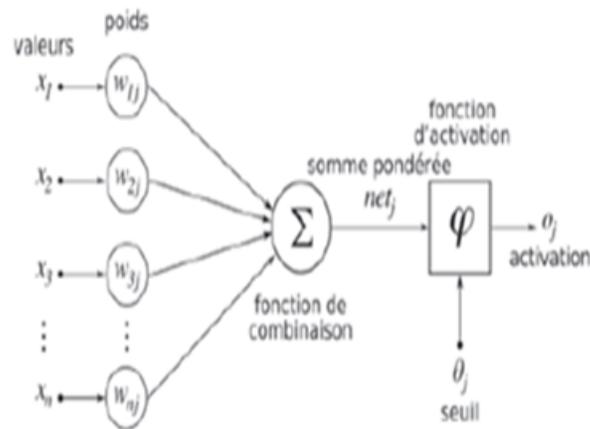


FIGURE 1.2 – Structure d'un perceptron simple

La somme pondérée des entrées par les poids (weights) w_i , associés aux entrées est appelée potentiel, noté S .

$$S = \sum_{i=0}^n w_{ij} x_j + b \quad (1.1)$$

S : est la somme pondérée du neurone net_j .

w_{ij} : est le poids de l'entrée i .

x_i : est la valeur de l'entrée i .

b : entrée spéciale (biais).

- **Biais(bias)** :c'est une valeur d'entrée qui est toujours égale à 1, ce qui permet une normalisation grâce à la valeur du poids correspondant ; [8]
- **La fonction d'activation** (ϕ) : transforme la somme pondérée pour obtenir la valeur de sortie. Le choix de la fonction de sortie dépend de l'application et du comportement souhaité ; [8]

1.4.2 Le perceptron multicouche (PMC)

Le perceptron multicouche [5] est un classifieur linéaire de type réseau neuronal formel organisv en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Les connexions sont toujours dirigées des entrées à travers le réseau vers les sorties, sans interconnexions entre les neurones de la même couche.

Les perceptrons multicouches sont capables de traiter des données qui ne sont pas linéairement séparables. Avec l'arrivée des algorithmes de rétro propagation, ils deviennent le type de réseaux de neurones le plus utilisé. Les MLP (Multi Layer Perceptron) sont généralement organisés en trois couches, la couche d'entrée, la couche intermédiaire (dite couche cachée) et la couche de sortie.

Les trois couches d'un MLP et le rôle de chacune :

1. **Couche d'entrée** :La première couche elle est complètement connectée vers l'avant et reçoit toutes les valeurs d'entrées du réseau à des fins d'apprentissage. Donc le nombre de neurones dans la couche d'entrées est égal au nombre de données en entrées (ou égale au nombre de dimensions du problème) ;
2. **Couches cachées** :couches qui succèdent à la couche d'entrée, constituées d'une ou de plusieurs couches intermédiaires. Elles relient la couche d'entrées à la couche de sorties. Le choix du nombre de couches et la taille (nombre de neurones) de chaque couche est empirique ;
3. **Couche de sortie** :La troisième couche ou la couche de résultat. Elle donne le résultat obtenu par le réseau. Donc, le nombre de neurones dans la couche de sorties égale au nombre de classes ;

La figure 1.3 illustre la structure d'un MLP présentant trois neurones en entrée, deux couches cachées et deux neurones en sortie.

Lorsque tous les neurones d'une couche sont connectés aux neurones de la couche suivante, on parle alors de couches complètement connectées.[3]

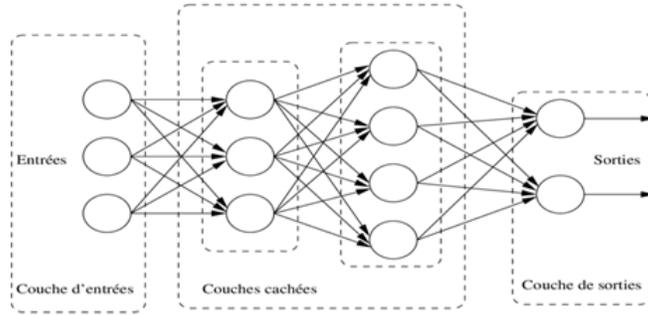


FIGURE 1.3 – Structure d'un perceptron multicouche. [9]

1.4.3 Fonctions d'activation

Dans le domaine des réseaux de neurones artificiels, la fonction d'activation est une fonction mathématique non linéaire appliquée à un signal en sortie d'un neurone artificiel. Le terme de fonction d'activation est synonyme de potentiel d'activation en biologie. C'est un seuil de stimulation qui une fois atteint entraîne une réponse du neurone. Les fonctions d'activation sont utilisées selon leurs caractéristiques d'étendue de non linéarité, de différentiabilité de convergence vers l'identité en 0 de continuité ou de monotonie.

Plusieurs fonctions d'activation ont été mises au point. Nous retenons quelques unes.

1. **Fonction Sigmoidé (Marche Douce ou Logistique) :**

Elle est souvent utilisée comme fonction d'activation pour les réseaux de neurones, dans lesquels les valeurs de sortie désirées sont soit binaires soit dans un intervalle compris entre 0 et 1. Le but de cette fonction est de réduire la valeur d'entrée à un intervalle entre 0 et 1.

Si la valeur en entrée est un très grand nombre positif, la fonction convertira cette valeur en une probabilité de 1. À l'inverse, si la valeur en entrée est un très grand nombre négatif, la fonction convertira cette valeur en une probabilité de 0. L'équation ci-dessous représente la fonction Sigmoidé

$$\sigma(x) = \frac{1}{1 + e^x} \quad (1.2)$$

2. **Fonction Tanh (Tangente Hyperbolique) :**

Cette fonction ressemble à la fonction Sigmoidé [5]. La différence avec la fonction Sigmoidé est que la fonction Tanh produit un résultat compris entre -1 et 1. La fonction Tanh est en terme général préférable à la fonction Sigmoidé car elle est centrée sur zéro. Les grandes entrées négatives tendent vers -1 et les grandes entrées positives tendent vers 1.

3. **Fonction ReLU (Unité de Rectification Linéaire) :**

Elle est inspirée par des motivations biologiques. [10] Elle a été introduite pour la première fois en 2011 par Glorot et al. [10], Les avantages de ReLU incluent le calcul efficace,

l'activation clairsemée, etc.

Elle est favorisée pour les réseaux neuronaux profonds et plus vite[10][11],L'équation ci-dessous représente la fonction ReLU :

$$f(x) = \max(0, x) \quad (1.3)$$

Si l'entrée est négative la sortie est 0 .Et si elle est positive alors la sortie est x. Cette fonction est la plus utilisée.

4. **Fonction Softmax :**

Est également appelée fonction exponentielle normalisée. Elle est couramment utilisée pour représenter une distribution de probabilité pour les K sorties possibles. Ainsi, la fonction softmax a été fortement utilisée dans les tâches de classification. Elle est définie comme :

$$F(x)_i = \frac{e^x}{\sum_{k=1}^k e^x k} \quad (1.4)$$

1.4.4 La rétropropagation

La technique de rétropropagation (Backpropagation en anglais) est une méthode qui permet de calculer le gradient de l'erreur pour chacun des neurones du réseau de la dernière couche vers la première. On appelle souvent technique de rétropropagation du gradient, l'algorithme classique de correction des erreurs basé sur le calcul du gradient grâce à la rétropropagation, mais cela n'est pas toujours le cas. La correction des erreurs peut se faire selon d'autres méthodes. Le plus souvent, dans le cas des réseaux de neurones, la méthode de correction d'erreurs agit en corrigeant de manière significative les coefficients synaptiques qui contribuent à engendrer une erreur importante tout en pondérant également les neurones générant une erreur moins conséquente.

La procédure de rétropropagation peut se résumer dans les étapes suivantes :

- Initialiser tous les poids à de petites valeurs aléatoires dans l'intervalle $[-0.5, 0.5]$.
- Normaliser les données d'entraînement.
- Permuter aléatoirement les données d'entraînement.
- Pour chaque donnée d'entraînement n :
 1. Calculer les sorties observées en propageant les entrées vers l'avant.
 2. Ajuster les poids en rétropropageant l'erreur observée. Répéter les étapes 1 et 2 jusqu'à un nombre maximum d'itérations ou jusqu'à ce que l'erreur soit en dessous d'un certain seuil.

1.5 L'Apprentissage

L'apprentissage[1] est souvent considéré comme la caractéristique principale de l'intelligence. De nombreuses études en intelligence artificielle (IA) ont donc été consacrées à l'apprentissage et aux méthodes qui permettent à une machine, d'étendre ses connaissances. Le

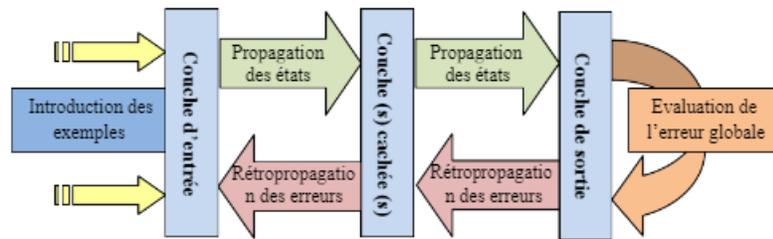


FIGURE 1.4 – Apprentissage des réseaux de neurone par l’algorithme de rétropropagation

principe de l’apprentissage devient alors le suivant : les informations captées ne servent pas uniquement à prendre une décision mais également à améliorer le réseau. Pour cette tâche, c’est l’apprentissage artificiel par induction qui est retenu par les scientifiques. Mais quelque soit le mode d’apprentissage nous distinguons habituellement deux situations : l’apprentissage supervisé et l’apprentissage non supervisé.

1.5.1 Apprentissage supervisé

Il est généralement destiné à reproduire un processus quelconque dont nous connaissons uniquement quelques variables et les résultats correspondants à obtenir.[12]. Dans ce cas, les données servant à l’apprentissage peuvent être représentées par le couple (entrée/sortie désirée, correspondante). Le processus est le suivant : nous modifions la force des connexions tant que le résultat obtenu correspondant à chaque entrée n’est pas assez proche de la sortie souhaitée.

Les réseaux à apprentissage supervisé [13] les plus utilisés sont les réseaux multicouches avec rétropropagation du gradient. La méthode la plus connue d’apprentissage supervisé est la classification.

1.5.2 Apprentissage non supervisé

Les réseaux à apprentissage non supervisé [13] sont utilisés lorsque les classes de données à obtenir ne sont pas connues à l’avance. Dans ce cas, les données servant à l’apprentissage sont uniquement des entrées. Comme c’est le cas pour les réseaux à apprentissage non supervisé, nous modifions la force des connexions entre les neurones du réseau. Or ici, il n’est pas possible de le faire en fonction d’une erreur rencontrée sur la réponse souhaitée. Dans ce type de réseaux, aucune réponse n’est connue a priori, les données étant uniquement des entrées. L’information utile se trouve donc uniquement dans les données servant de base pour l’apprentissage et en particulier dans les doublons de celles-ci

1.6 L'apprentissage profond

L'apprentissage profond [1] (deep Learning) est une classe de méthodes dont les principes sont connus depuis la fin des années 80, mais dont l'utilisation ne s'est vraiment généralisée que depuis environ 2012. L'apprentissage profond peut être vu comme un réseau multicouche avec de nombreuses couches intermédiaires.

Le Deep Learning est un ensemble de méthodes d'apprentissage qui tentent de modéliser des données avec architectures complexes, combinant différentes transformations non linéaires. Les bases élémentaires de l'apprentissage en profondeur sont les réseaux de neurones, qui sont combinés pour former les réseaux de neurones profonds.

Ces techniques ont permis des progrès significatifs dans les domaines du son et le traitement d'images, y compris la reconnaissance faciale, la reconnaissance vocale, la vision par ordinateur, le traitement automatique du langage, la classification des textes (par exemple reconnaissance du spam). Les applications potentielles sont très nombreuses. Un spectaculaire exemple est le programme Alpha Go[14], qui a appris à jouer le jeu de go par la méthode d'apprentissage en profondeur et a battu le champion du monde en 2016

Il existe plusieurs types d'architectures pour les réseaux de neurones :

- Les perceptrons multicouches, les plus anciens et les plus simples ;
- Les Réseaux de Neurones Convolutionnels (CNN), particulièrement adaptés au traitement d'images ;

L'architecture profonde est capable d'extraire automatiquement des caractéristiques de l'image.

Par exemple les premières couches extraient des contours que les couches suivantes forment en des concepts de plus en plus complexes et abstraits : des formes, des objets, des parties d'objets en objets, etc.

1.7 Les réseaux de neurones convolutifs

les réseaux neuronaux convolutifs désignés par l'acronyme CNN (Convolutional Neural Network)[15] renvoient à un terme mathématique : le produit de convolution. En termes simples, l'idée est qu'on applique un filtre à l'image d'entrée, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert et Bleu). La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image, par une opération de maximum ou de moyenne(pooling). Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

en sortie de la partie convolutive est ensuite branchées en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN.

Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo.

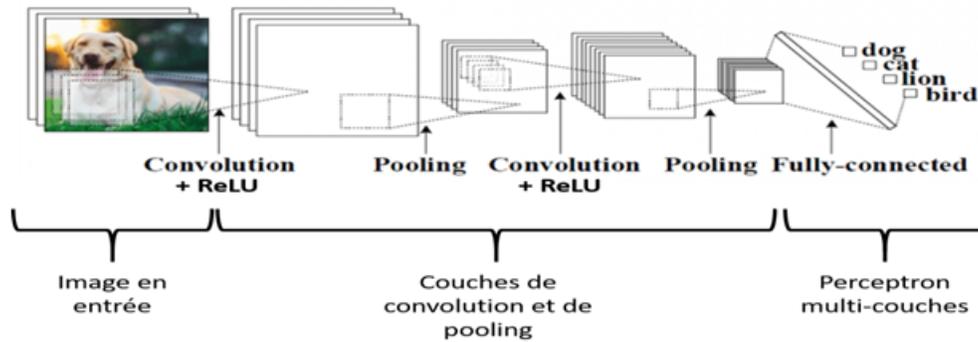


FIGURE 1.5 – Architecture standard d'un réseau de neurones convolutifs

1.7.1 Architecture des réseaux de neurones convolutifs

La couche de convolution(CONV)

La convolution est très classique en traitement d'images Elle consiste à déplacer un filtre (ou kernel en anglais) en le faisant glisser sur une image et à réaliser une convolution (un produit matriciel) de ce filtre avec l'image.

La couche de convolution est le bloc de construction de base d'un CNN. Trois paramètres permettent de dimensionner le volume de la couche de convolution : la profondeur, le pas et la marge.[15]

1. **La profondeur de la couche** :nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
2. **Le pas** :lorsque nous effectuons une convolution, nous choisissons ce qu'on appelle le pas. stride, il contrle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
3. **La marge(à 0) ou zero padding** :parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce zero-padding est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Avec un padding de 0, aucun contour de 0 n'est rajouté. Avec un padding de 1, un contour de 0 est rajouté. Avec un padding de 2, deux contours de 0 sont rajoutés, et ainsi de suite.

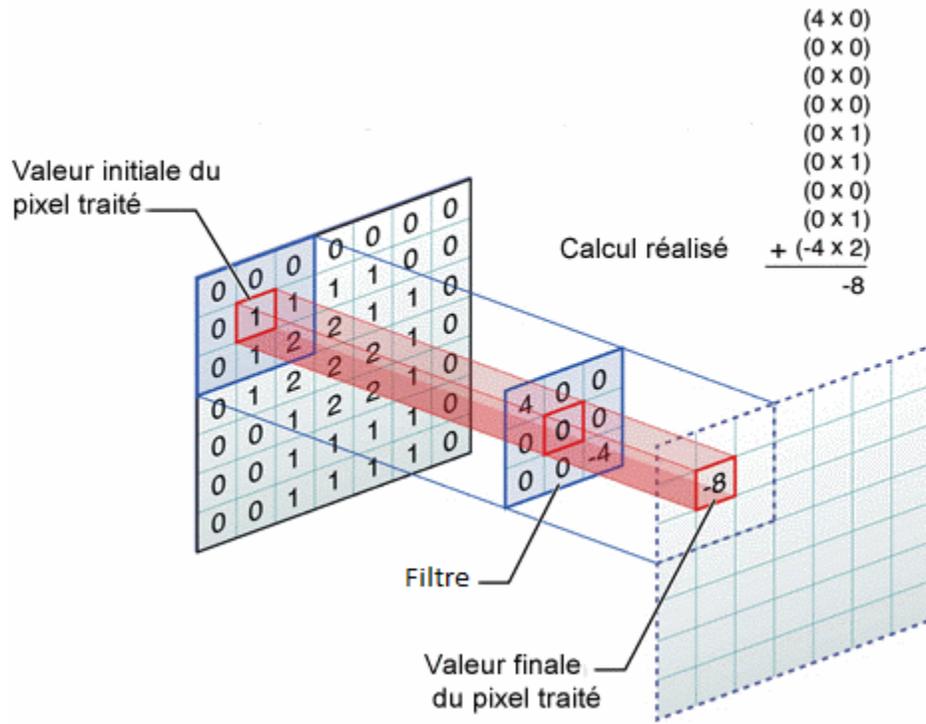


FIGURE 1.6 – Exemple de convolution avec un filtre 3 x 3

Avec un stride de 1, le filtre se déplace d'un pixel à la fois. Avec un stride de 2, de deux pixels à la fois.

Nous pouvons savoir la taille de l'image en sortie d'une convolution à partir de la taille de l'image d'entrée (W), la taille du filtre (F), le padding (P) et le stride (S) :

$$\text{Taille sortie convolution} = \left\{ \frac{W - F + 2P}{S} + 1 \right\} \quad (1.5)$$

Souvent, nous considérons un pas $S=1$. Nous calculons donc la marge de la manière suivante :

$$P = \frac{F - 1}{2} \quad (1.6)$$

si nous souhaitons un volume de sortie de même taille que le volume d'entrée, dans ce cas particulier la couche est dite : connectée localement.

Couche de pooling

La couche de pooling est une forme de sous-échantillonnage de l'image entrée. En extrayant les valeurs importantes des pixels, elle permet de réduire une image (réduit la taille spatiale) tout en conservant les caractéristiques pertinentes. La méthode la plus utilisée est le Max Pooling . Elle consiste à extraire les valeurs les plus grandes des pixels. Pour ce faire, nous avons un filtre qui se déplace sur la surface de notre image. à chaque position de filtre, nous extrayons la valeur la plus haute et nous ne retenons que celle là. Cela produit une

nouvelle image avec uniquement les valeurs remarquables de cette image.

Le Pooling est très important. En réduisant l'image, le nombre de données traitées diminue et donc le temps de calcul sera lui aussi réduit. Cela n'est pas négligeable. Il existe d'autres méthodes que le Max Pooling par exemple :

- **Average Pooling** : Moyenne de toutes les valeurs recouvertes par la tuile ;
- **Pooling stochastique** : Ne retient qu'une seule valeur, comme le Max Pooling, mais en se basant sur une méthode probabiliste ;

Cependant, en pratique le Max Pooling est la méthode qui donne le plus souvent les meilleurs résultats. Elle est donc généralement privilégiée.

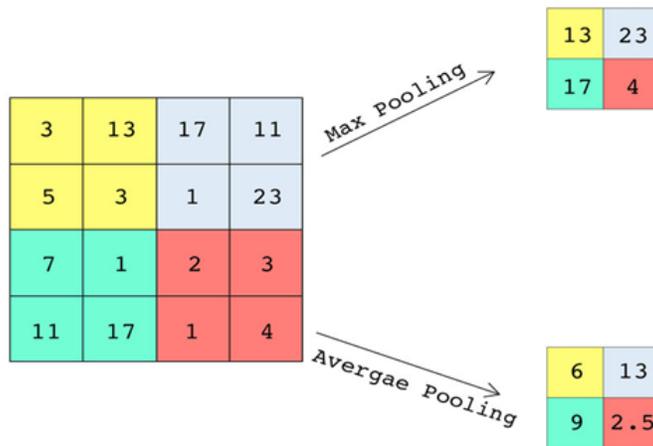


FIGURE 1.7 – Exemples de max et moyenne pooling sur une fenêtre 2x2 et pas 2

Couches de correction (RELU)

La fonction ReLU (abréviation de Unités Rectifié linéaires) désigne la fonction réelle non-linéaire définie par :

$$F(x) = \max(0, x) \quad (1.7)$$

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation. La fonctionnalité ReLU est illustrée à la figure ci-dessous.

Subsampling

C'est la fonction inverse de Max-Pooling, ou l'image sera reconstruite en répétant les lignes et les colonnes des données respectivement. Il a pour but d'obtenir la même résolution que l'image d'entrée.

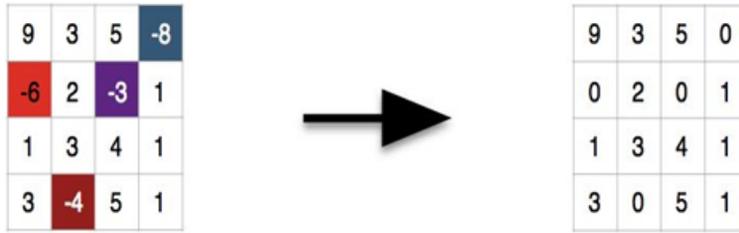


FIGURE 1.8 – Exemple de la fonction ReLu

Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.

Les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est capable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels.

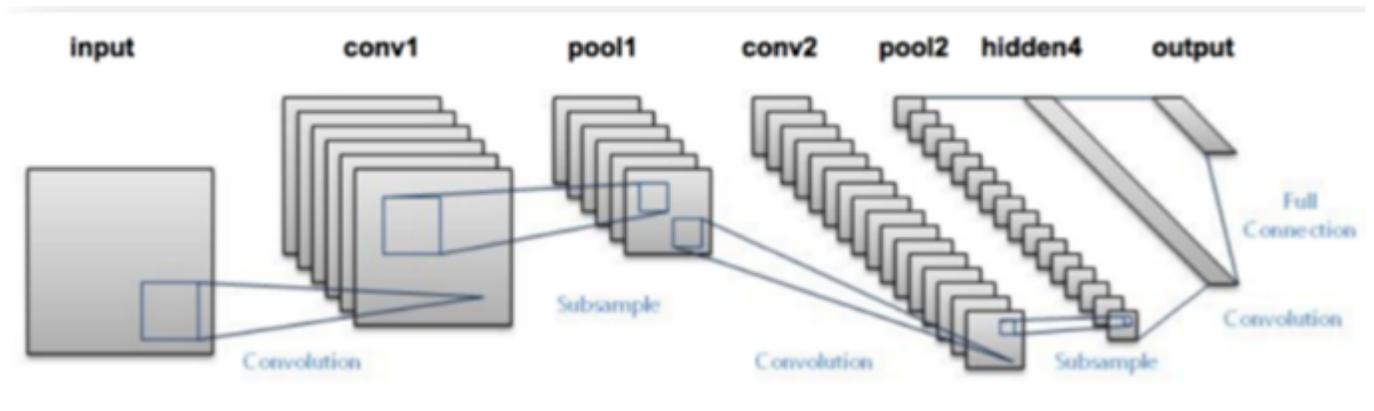


FIGURE 1.9 – Les différentes couches d'un CNN

1.8 Auto-encodeur (AE)

Les réseaux de neurones ne se limitent pas forcément à des problématiques d'apprentissage supervisé, même si cela reste leur utilisation principale. En effet, il existe aussi des réseaux de neurones qui permettent de faire de l'apprentissage non supervisé.

Les plus utilisés dans ce domaine sont probablement les Auto-Encodeurs. Ces réseaux n'ont pas pour objectif de prédire ce qui se trouve dans une image mais seulement de chercher à

trouver une représentation pertinente et compressée de l'image en tirant justement parti du pouvoir de représentation des réseaux de neurones.

Auto Encodeurs : sont des algorithmes d'apprentissage non supervisé à base des réseaux de neurones artificiels, qui permettent de construire une nouvelle représentation d'une image. Généralement, celle-ci est plus compacte, et présente moins de descripteurs, ce qui permet de réduire la dimensionnalité de l'image. L'architecture d'un auto-encodeur est constituée de deux parties : l'encodeur et le décodeur.

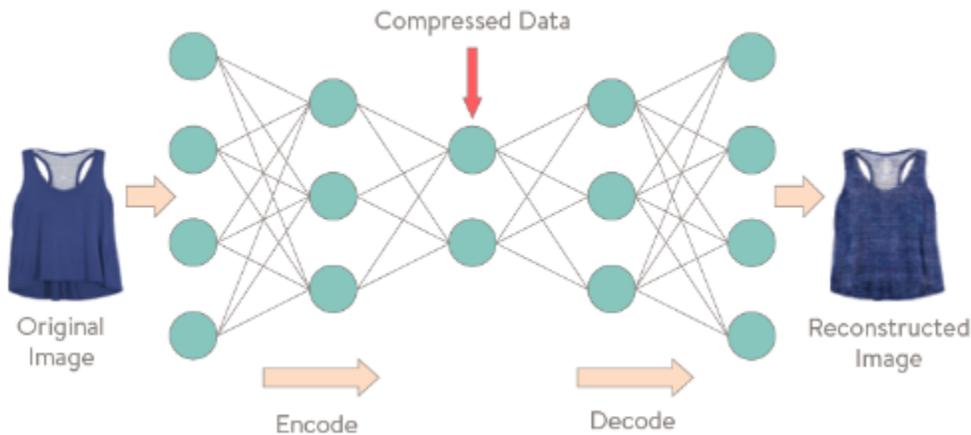


FIGURE 1.10 – Principe d'un auto-encodeur sur une image

1.8.1 Architecture des auto-encodeurs

Cette architecture peut être décomposée en deux parties :

L'encodeur : est constitué par un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites encodées. À leur tour, les couches de neurones du décodeur, reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ. Les différences entre les données reconstruites et les données initiales permettent de mesurer l'erreur commise par l'auto-encodeur. L'entraînement consiste à modifier les paramètres de l'auto-encodeur afin de réduire l'erreur de reconstruction mesurée sur les différents exemples du jeu de données.

Le décodeur : C'est la dernière couche qui contient uniquement la reconstruction des données initiales, mais plutôt à la nouvelle représentation créée par l'encodeur.

L'architecture la plus simple d'un auto-encodeur est semblable à un perceptron multicouche. Cependant, en fonction des données traitées, nous pouvons utiliser différentes topologies de réseaux de neurones. Par exemple, des couches convolutives afin d'analyser des images ou des couches de neurones récurrentes pour traiter des séries temporelles ou des séquences.

1.8.2 Apprentissage des auto-encodeurs

L'idée est simple mais ingénieuse : on entraînera un réseau de neurones à une couche cachée h à prédire en sa sortie z en son entrée x , Une tâche que nous nommons reconstruction. Le critère sera donc la minimisation d'une erreur de reconstruction $L(x, z)$. En principe, la couche cachée devra donc contenir de l'information pertinente à la reconstruction. Par exemple, si la couche cachée contient moins d'unités qu'il y a d'entrées, alors logiquement pour bien reconstruire elle doit apprendre à résumer l'entrée, puisque l'information pour la reconstruction est entièrement contenue dans la couche cachée. Ainsi des caractéristiques pertinentes à la reconstruction devront être extraites.

Il faut noter que les paramètres pour passer de h à z , W_0 , ont une forme matricielle qui est la transposée de la forme des paramètres W . Si les poids sont liés (tied) alors il s'agit effectivement de la transposée et durant l'entraînement le contenu de la matrice sera contraint de rester le même pour les deux directions. Autrement, s'ils ne sont pas liés, ils sont libres de prendre des valeurs différentes.

Il est aussi possible d'effectuer la reconstruction en faisant passer l'information dans plusieurs couches cachées h_1, \dots, h_n . Nous obtenons ainsi un auto-encodeur profond. Comme ce type de réseau contient plusieurs couches, il sera en général préférable de pré-entraîner les couches comme expliqué dans la section sur les architectures profondes, et de les faire correspondre par paires : la première et la dernière, la seconde et l'avant dernière, par les correspondances entre les noms des paramètres de chacune des couches.

Par contre, dans l'auto-encodeur simple, si le nombre d'unités dans la couche cachée est plus grand ou égal au nombre d'entrées, le modèle peut alors apprendre de façon triviale la fonction identité. C'est facile en copiant une à une les valeurs de l'entrée dans la couche cachée, par exemple. Nous n'obtenons donc pas de caractéristiques extraites intéressantes. Pour contourner ce problème, différentes contraintes peuvent être imposées durant l'apprentissage. C'est ce qui est fait avec l'auto-encodeur débruiteur (voir ci dessous).

Pour des modèles effectuant de la reconstruction, Nous pouvons évaluer l'erreur de reconstruction, la différence entre l'entrée originale et l'entrée reconstruite. Cela permet de comparer les modèles entre eux. Pour le faire numériquement, nous pouvons utiliser Le SSIM, EQM. (Voir chapitre 3) pour guider l'entraînement (évaluation objectif).

L'apprentissage de l'auto-encodeur se fait par rétropropagation du gradient.

1.8.3 Auto-encodeurs débruiteur

débruiteur

Bruit :

Le bruit [22][23] est un signal qui dégrade la qualité d'une image(vidéo),il est provoqué par une

perturbation externe. Dans le cas de la transmission d'une image (vidéo) par voie électronique on peut s'attendre à des erreurs se produisant dans le signal de l'image. Ces erreurs peuvent dégrader la qualité de l'image (vidéo) de différentes manières selon le type de canal de transmission (satellite, sans fil, câble réseau...).

Dans ce qui suit, nous allons présenter quatre types de bruit et leurs effets sur la qualité d'une image.

1. **Le bruit additif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit additif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j + b_j \quad (1.8)$$

b est une variable aléatoire de moyenne égale à 0.

type de bruit additif

- (a) Le bruit poivre et sel : est obtenu en ajoutant des pixels blancs et des pixels noirs (ou les deux) aléatoirement dans une image.



FIGURE 1.11 – Image avec le bruit poivre et sel

- (b) Le bruit Gaussienne : Est une forme idéalisée de bruit blanc, obtenu en ajoutant à chaque pixel une valeur aléatoire suivant une loi de probabilité Gaussienne.



FIGURE 1.12 – Image avec le bruit Gaussienne

2. **Le bruit multiplicatif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit multiplicatif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j * b_j \quad (1.9)$$

b est une variable aléatoire de moyenne égale à 0.

type de bruit multiplicatif

- (a) bruit de speckle : s'appelle également le bruit multiplicatif. il est un problème important dans quelques applications de radar et d'échographie.



FIGURE 1.13 – Le bruit de speckle (bruit multiplicatif)

3. **Le bruit convolutif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit additif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j * b_j \quad (1.10)$$

b est une variable aléatoire de moyenne égale à 1.

parmi ce type de bruit, nous trouvons le flow.

Un débruiteur : C'est un algorithme qui enlève le bruit dans l'image, vidéo, son, single.

Auto-encodeur débruiteur

Auto-encodeur débruiteur : est une combinaison d'un auto-encodeur en premier lieu et un débruiteur comme deuxième traitement. L'objectif de cette combinaison est l'amélioration de la convergence du réseau.

Il reçoit une donnée partiellement corrompue comme entrée et apprend à récupérer l'entrée originale débruitée. Cette technique a été introduite avec une approche spécifique d'une bonne représentation. Une bonne représentation est celle qui peut être obtenue de manière robuste à partir d'une entrée corrompue et qui sera utile pour récupérer l'entrée débruitée correspondante.

1.9 Conclusion

Dans ce chapitre nous avons abordé les notions de base d'un réseau neuronal, telles que, le perceptron et le perceptron multicouches. Ce dernier est efficace pour les problèmes logiques mais il est limité pour l'extraction des caractéristiques de l'image et de larges structures (le

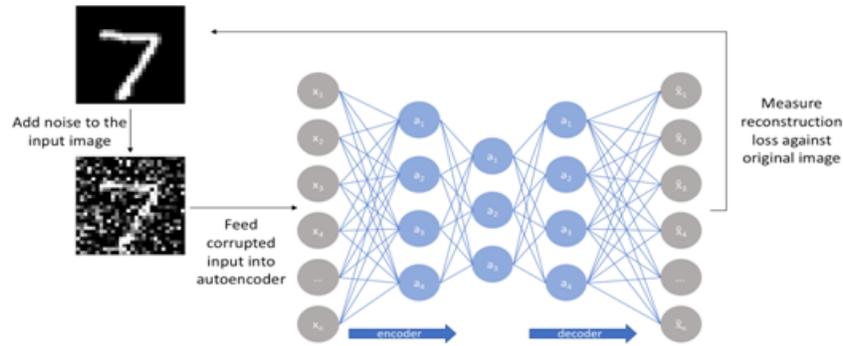


FIGURE 1.14 – Autoencodeur débruiteur

nombre des couches est grand). Nous avons également abordé les fonctions d'activation les plus utilisées puis nous avons expliqué la technique de rétropropagation. Nous avons présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures, etc.), ainsi qu'une vision générale sur l'apprentissage profond tel que, les réseaux de neurones convolutifs (CNN) qui est capable d'extraire les caractéristiques (les plus pertinentes) d'une image, par des séquences des couches convolutifs et de réduire les données non pertinentes (pooling). Nous avons aussi présenté un autoencodeur et un autoencodeur combiné avec un débruiteur. Cette combinaison peut être utilisée pour Accélérer l'apprentissage des neurones artificiels et minimiser le temps de travail.