Transparence des résultats de l'apprentissage profond

Bouteggui Sara, Bruel Lucile, Gimenez Florent, and Saulières Léo

Étudiants en Master 1 IARF, Université Paul Sabatier, Toulouse

Abstract. Aujourd'hui, les réseaux de neurones sont utilisés dans de nombreuses applications. Depuis 2010, les résultats obtenus avec ces réseaux dépassent largement les autres algorithmes dans les domaines du traitement d'image, de la reconnaissance vocale et le traitement du langage. Cependant, la compréhension de ces résultats est souvent difficile du fait de la complexité des réseaux de neurones. En effet, le nombre de neurones, de liaisons entre eux et le nombre de calculs effectués rendent difficile l'interprétation des résultats. La transparence des réseaux de neurones est donc primordiale afin de comprendre, utiliser correctement les réseaux de neurones et résoudre les problèmes qu'ils peuvent rencontrer.

Keywords: Réseaux de neurones · Transparence · Intelligence Artificielle · Apprentissage profond

1 Introduction

Les réseaux de neurones sont beaucoup utilisés aujourd'hui, que ce soit pour du traitement d'image tel que la reconnaissance faciale [1], pour de la reconnaissance audio comme la reconnaissance d'accords en musique [2] et même pour le traitement du langage naturel pour les moteurs de recherche (tel que l'algorithme Bert de Google [3]). Il est alors essentiel de comprendre les règles et fonctionnements divers qui régissent les réseaux de neurones pour les utiliser et les optimiser au mieux.

Cependant, il est encore aujourd'hui difficile de comprendre le raisonnement interne de ces réseaux et d'expliquer précisément comment ils produisent de tels résultats avec une donnée en entrée. Les multiples couches cachées de neurones et l'important nombre de calculs d'une couche à l'autre renforcent le fait que le fonctionnement interne des réseaux de neurones est assimilé à une boîte noire. Cette incompréhension entraîne une difficulté dans l'interprétation des résultats et l'amélioration des réseaux. Ainsi, avec l'explosion de l'emploi de ces systèmes, les études sur la transparence des réseaux de neurones se sont multipliées [4–8]. En effet, plusieurs techniques ont été développées pour permettre de mettre en lumière (en partie) ces boîtes noires et pour ainsi expliquer, d'une meilleure façon, les résultats obtenus [7].

Dans ce document, nous présenterons d'abord le fonctionnement général des réseaux de neurones (Section 2) ainsi que les différentes manières de les utiliser. Ensuite, nous illustrerons leurs utilisations avec des exemples concrets (Section 3) . Enfin, nous analyserons les différentes techniques utilisées (Section 4) pour rendre les réseaux de neurones plus transparents.

2 Les réseaux de neurones

2.1 Intelligence Artificielle

L'intelligence artificielle est "l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine" [9]. Celle-ci est scindée en deux mouvements de pensées : l'approche connexionniste et l'approche symbolique.

L'*approche symbolique* se caractérise par un ensemble de règles logiques, formelles, permettant de manipuler des symboles. Tout est basé sur le raisonnement. Les données n'impactent pas le monde (espace où utiliser l'ensemble des symboles) dans l'approche symbolique [10]. La création de symboles, représentation à haut niveau, a permis par exemple la création de langages de programmation, de traduire un langage machine en langage de haut niveau plus compréhensible pour l'humain.

L'*approche connexionniste*, à laquelle nous nous intéressons dans ce document, est l'utilisation de réseaux de neurones. Chaque neurone "prend des variables en entrées, y applique un poids pour produire une somme qui, si elle dépasse un certain seuil, déclenche l'activation du neurone." [10]. Cependant ces réseaux sont

constitués de plusieurs centaines de neurones connectés entre eux, ce qui rend difficile la compréhension de ces systèmes et donc l'interprétation de leurs résultats. On les assimile donc à des "boîtes noires" qui apprennent en fonction de ses entrées et sorties. Les données impactent réellement le fonctionnement du monde (ici le réseau de neurones). On constate une baisse d'intérêt dans les années 90, car les résultats [10] et la puissance de calcul des machines [11] sont limités. Il y a un essor des réseaux de neurones depuis qu'il existe des machines suffisamment puissantes pour les calculs et que les réseaux de neurones soient supérieurs en performances sur les autres techniques."Depuis 2010, domaine après domaine, les réseaux de neurones profonds provoquent la même perturbation au sein des communautés informatiques traitant du signal, de la voix, de la parole ou du texte" [10].

2.2 Fonctionnement global

Un réseau de neurones est un ensemble de noeuds connectés, disposés de manière à ce que chaque neurone soit un point de réseau qui reçoit de l'information entrante et émet de l'information sortante. Un noeud, dans notre cas, est un neurone qui fonctionne de la manière suivante : il calcule la somme pondérée de ses entrées multipliées par leurs poids respectifs, en rajoutant ensuite un biais [12]. Cette valeur est transmise à une fonction d'activation qui va décider de la valeur en sortie du neurone. Il existe différentes fonctions d'activations : Sigmoid, Tanh, ReLU, Maxout,... Dans le réseau de neurones, les poids et les biais sont d'abord attribués de manière aléatoire. Afin que le réseau obtienne la sortie attendue, on doit l'entraîner. Pour ce faire, on modifie les poids des connexions afin que le réseau minimise son erreur.

Dans les réseaux à couches, les noeuds ou neurones sont organisés sous forme de plusieurs couches. Chaque noeud dans le réseau est lié avec un ou plusieurs noeuds de la couche précédente et un ou plusieurs de la couche suivante par des connexions. Lorsque tous les neurones d'une couche sont connectés à chaque neurone de la couche précédente, elle est dite totalement connectée [13]. La connexion entre les neurones est caractérisée par des poids qui sont des valeurs pour propager l'information. C'est à partir d'eux et de leur modification que l'apprentissage du réseau s'effectuera. Il existe une multitude de structures différentes pour construire des réseaux de neurones. Dans certains réseaux (cf. Figure 3), l'ensemble des couches est divisé en trois parties. La première est la couche d'entrée ('*Input Layer*' en anglais), la dernière est la couche de sortie ('*Output Layer*') et toute autre couche se situant entre les deux est une couche cachée ('*Hidden Layer*'). Les entrées sont présentées au réseau par la couche d'entrées. Ces informations seront converties grâce aux poids et vont se propager d'une couche à l'autre jusqu'à atteindre la dernière couche.

Avant de pouvoir utiliser un réseau de neurones, il y a une phase dite d'apprentissage qui se met en place de deux façons différentes. Tout d'abord, l'apprentissage non supervisé présente un avantage majeur car il utilise des données non labélisées. Il est connu pour avoir de meilleures performances en restauration et en mise en correspondance d'images [14]. Les domaines d'application entre l'apprentissage supervisé et non supervisé sont différents.

Contrairement à cet apprentissage, l'apprentissage supervisé consiste à fournir un jeu de données labélisées au réseau et comparer le résultat attendu avec le résultat obtenu. Cette comparaison entre le résultat en sortie et le résultat attendu donne une valeur appelée le coût. Plus le coût est grand, plus le résultat est mauvais. Lorsque le résultat est mauvais, une rétropropagation a lieu [15]. Elle consiste donc à reparcourir le réseau de neurones et à corriger les poids et les biais. Ce processus est répété de manière itérative jusqu'à ce que l'erreur du réseau soit minimisée ou ait atteint une valeur acceptable. À travers ces modifications successives, le réseau construit un modèle de processus de génération de données afin qu'il soit capable de prédire et généraliser les sorties depuis des entrées non déjà vues.

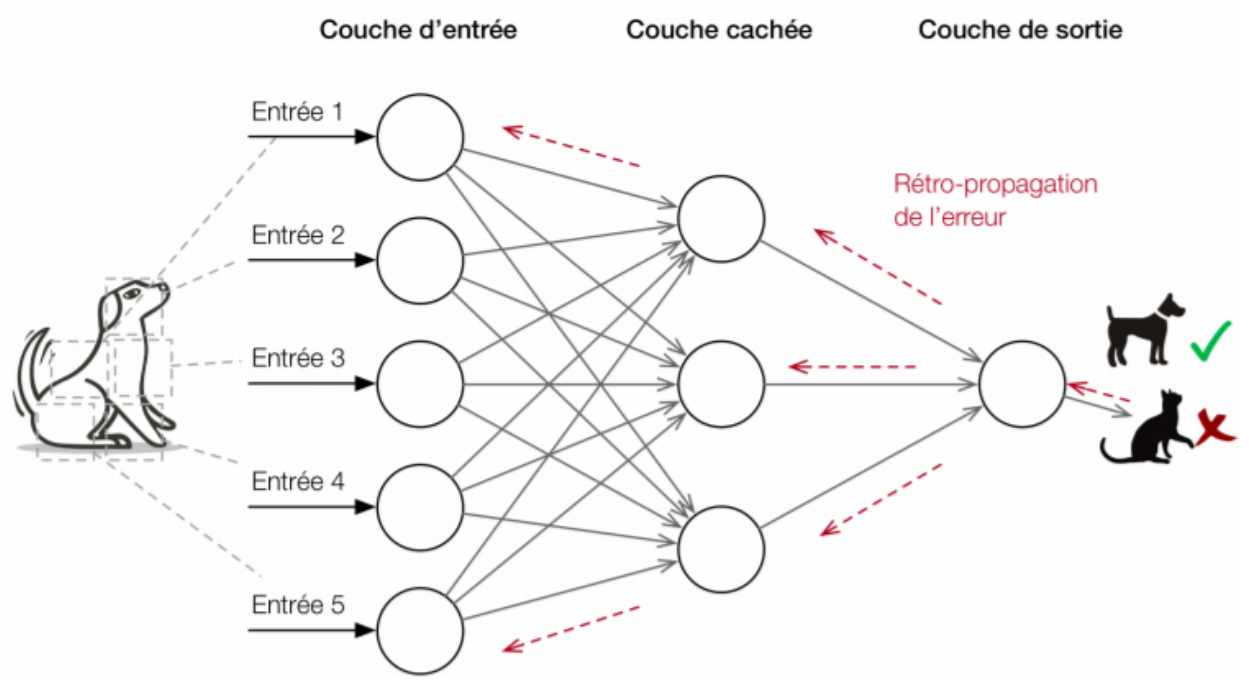

Fig. 1. Exemple de réseau de neurones en apprentissage supervisé avec rétropropagation [10]

2.3 Différentes structures de réseaux de neurones

Les réseaux de neurones ne sont pas uniformes, il existe différentes "variantes" permettant d'obtenir de meilleurs résultats dans certains secteurs d'applications. Nous allons en présenter qui sont majeures dans le domaine de l'intelligence artificielle. Depuis les réseaux multicouches des années 90, les architectures de réseaux de neurones ont évoluées. Notamment, il existe de nos jours des Convolutive Neural Networks (CNN) qui sont des réseaux neuronaux convolutifs que nous étudions dans cet état de l'art.

Dans le domaine du traitement d'image, un filtre est une opération mathématique qui modifie la valeur d'un ou plusieurs pixels localement selon sa taille qui est fixe. Ce filtre contient des coefficients (différents ou non) pour chaque pixel situé dans sa zone. Prenons par exemple un filtre "moyenneur" de taille 3x3. Il effectue la moyenne des pixels dans cette zone et mémorise sa valeur dans le pixel situé au centre du filtre.

La convolution est une généralisation de la notion de filtre, car elle est appliquée sur l'ensemble de l'image en faisant "glisser" le filtre. Pour le traitement d'image et la classification en un nombre fini de groupes, ce sont des réseaux de neurones convolutifs qui sont utilisés [16]. Ce sont des réseaux qui appliquent une succession de convolutions, filtres à différentes régions d'une image.

Le sous-échantillonnage est une convolution qui permet de conserver les éléments les plus importants d'une image; cela réduit la dimension de l'image. Après chaque couche de neurones de convolution, il y a une couche de sous-échantillonnage qui sert à éviter le sur-apprentissage [16].

Il existe aussi des réseaux de neurones déconvolutifs qui parviennent à reconstruire une image détériorée ou avec des imperfections (flou, ...) [17]. La déconvolution consiste à utiliser un réseau neuronal convolutif qui prend en compte des dégradations. Hélas, ce réseau seul est insuffisant et doit ainsi être combiné avec un "modèle séparable" fiable pour la déconvolution. Comme le montrent les travaux de [17], un tel réseau a donc été séparé en deux parties entraînées de façon supervisée. Ainsi, ce réseau nommé Deconvolution-CNN (DCNN) apporte des résultats plus satisfaisants, mais Xu et al. [17] ont également créé une meilleure version nommée Outlier-rejection DCNN (ODCNN) qui est la combinaison d'un DCNN et d'un "Denoise CNN". Cette version parvient à supprimer les imperfections les plus fines. La dernière couche de ce DCNN est mise en entrée du Denoise CNN, ce qui produit des résultats vraiment satisfaisants comme le montre l'image ci-dessous :

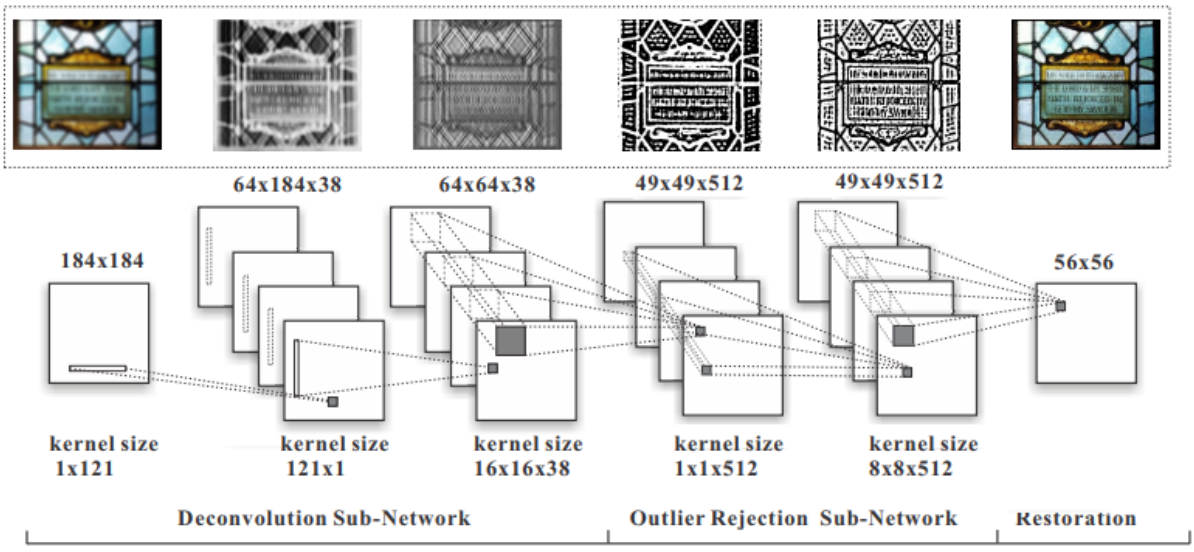

Fig. 2. Architecture complète de la déconvolution profonde par Xu et al. [17]

Les réseaux de neurones récurrents sont fortement utilisés dans la reconnaissance automatique de la parole [18] [19] [20]. Ils se différencient des autres réseaux par le fait que pour prédire le résultat à un instant T, les couches de neurones cachées ne se basent pas uniquement sur la couche de neurones en entrée, mais aussi sur le résultat des couches de neurones cachées à l'instant T-1. Nous pourrions par exemple [21], considérer la phrase "Comment allez-vous ?" afin de la reconnaître. Avec le mot "Comment" en entrée du réseau à l'instant 0, le réseau va prédire "allez" qui va se retrouver en entrée du réseau à l'instant 1. Pour prédire "vous", le réseau de neurones récurrents aura comme information "allez" en entrée et "comment" qui sont récupérés depuis les couches de neurones à l'instant 0.

Toutes les différentes structures de réseaux de neurones ne sont pas présentées dans notre état de l'art. C'est pourquoi l'image ci-dessous (cf. Figure 3) permet de recenser et de constater une grande partie des structures existantes pour construire des réseaux de neurones.

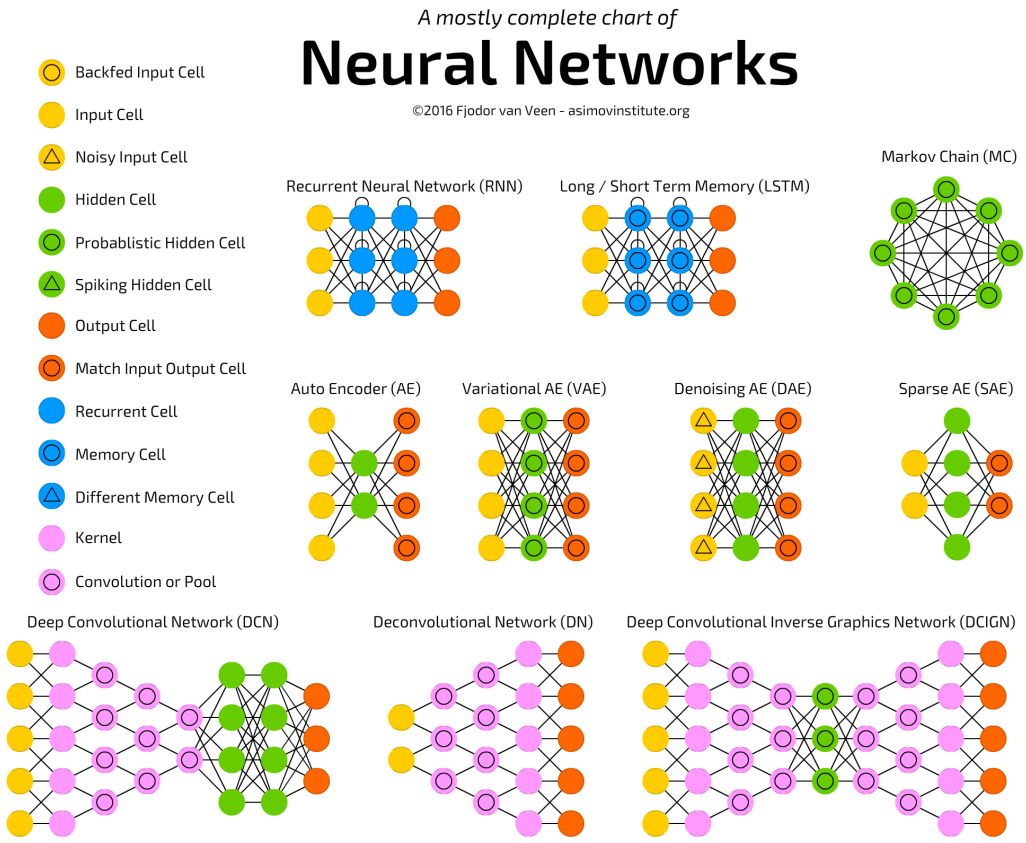

Fig. 3. Principales structures de réseaux de neurones pour notre état de l'art [22]

3 Exemples d'application des réseaux de neurones

Les réseaux de neurones sont en grande partie utilisés dans le domaine du traitement de l'image, reconnaissance vocale et langage naturel. Les exemples développés ci-dessous sont des utilisations des réseaux de neurones pour le traitement d'images et le traitement du langage naturel.

3.1 Reconnaissance des chiffres manuscrits et numérisés

La reconnaissance de chiffres est un exemple courant dans les réseaux de neurones tout comme le montre l'article de Yuchun Lee [23], ainsi que l'article de Knerr et al. [24] puis l'état de l'art de Liu et al. en 2003 [25].

Cet exemple a été proposé en 1990 [26]. Le réseau a été entraîné sur un ensemble d'images composées de véritables chiffres écrits à la main sur des lettres aux États-Unis (9298 lettres) et des chiffres écrits avec différentes polices d'écriture (3349 chiffres). Le réseau de neurones de l'exemple ci-dessus est un réseau de neurones qui utilise la rétropropagation du gradient. Ce réseau est performant (1% de taux d'erreur). Il y a un prétraitement des images avant qu'elles soient envoyées sur le réseau. En effet, la taille des images est de 40*60 pixels généralement, mais toutes les images sont modifiées pour avoir une taille unique avant d'être traitées par le réseau de neurones : 16*16 pixels. Ce réseau de neurones est composé d'une succession de convolutions qui sont finalement sous-échantillonnées pour obtenir 12 images de taille 4*4. La dernière couche totalement connectée avec l'avant-dernière est composée de 10 neurones qui vont prédire la sortie du modèle. [26]

La base de données utilisée dans les articles ci-dessous et plus généralement dans le domaine de la reconnaissance de chiffres est "MNIST" [27]. Elle se compose de 60 000 images d'entraînements et de 10 000 pour tester son classifieur. Les images en noir et blanc sont de dimension 28 par 28 pixels.

La reconnaissance digitale comprend de multiples manières d'être mise en place. Par exemple, les machines à vecteurs de support (SVM) sont des classifieurs linéaires qui sont utilisées dans la reconnaissance digitale [28]. Il y a également un algorithme simple de l'apprentissage automatique, nommé K-NN (ou k plus proches voisins). Cela consiste d'abord, à classer des données en fonction de leur classe. Ensuite, une fois entraîné, l'algorithme détermine pour une donnée passée en entrée, ses k plus proches voisins puis, en fonction de la classe majoritaire parmi ses voisins, sa classe. Cet algorithme a été utilisé par Yuchun Lee [23], Babu et al. [29].

À la place de classer les images de chiffres directement, il existe d'autres méthodes qui améliorent le taux de reconnaissance comme le "Patch Autocorrelation Feature" (PAF) et le "Translation and Rotation Invariant Patch Autocorrelation Features" (TRIPAF) [30]. L'algorithme PAF renvoie un vecteur contenant le résultat de l'autocorrélation de l'image qui est ensuite utilisé en entrée du classifieur à la place de l'image de base. Pour les images, l'autocorrélation est la corrélation entre une image et elle-même. La corrélation sert à mesurer les déplacements (déformation et rotation) entre deux images. L'algorithme TRIPAF effectue la même chose sauf que l'autocorrélation n'est plus affectée par les translations et rotations présentes dans l'image [30].

3.2 Reconnaissance d'actions humaines

3.2.1 Utilisation de réseaux convolutifs en 3 dimensions [31]

Pour la reconnaissance d'actions humaines dans des images/vidéos contenant des environnements non contrôlés, les réseaux de neurones convolutifs sont des modèles profonds qui peuvent agir directement à partir de données "brutes" (images, ...) en appliquant des filtres de convolution en 3 dimensions puis des opérations comme le sous-échantillonnage. Contrairement aux modèles plus traditionnels qui sont très dépendants du contexte dans lequel la vidéo a été filmée car ces caractéristiques sont extraites par des algorithmes. Toutefois, il est difficile pour ces réseaux de travailler à partir de données 2D ou plus. C'est pour cette raison que par le biais de convolutions 3D dans le domaine spatial et temporel, ce nouveau modèle peut extraire les caractéristiques des mouvements sur plusieurs images adjacentes. Si l'on ne considère qu'une image isolée, tous les mouvements encodés/décrits sur plusieurs images ne sont pas pris en compte et ainsi la détection d'actions est faussée ou incomplète.

Comparaison des résultats de 4 méthodes (précision en %) [31]

Méthode	**FPR**	**CellToEar**	**ObjectPut**	**Pointing**	**Moyenne**
3D CNN	0.1%	**64,33**	**67,48**	82,30	**71,37**
	1%	**40,91**	**51,54**	74,70	**55,72**
2D CNN	0.1%	38,42	58,65	**85,47**	60,85
	1%	30,32	39,37	74,46	48,05
SPM [1]	0.1%	35,76	60,51	85,41	60,56
	1%	26,07	43,32	**75,11**	48,17

3.2.2 Estimation de postures humaines à partir d'un modèle graphique [32]

En effet, l'analyse de postures se révèle très difficilement reconnaissable dans un contexte quelconque, que ce soit des mouvements simples ou complexes. Pour résoudre ce problème, Tompson et al. [32] ont proposé une architecture hybride qui combine un réseau de neurones convolutif (RNC) et un modèle graphique nommé "champ aléatoire de Markov" qui représente les dépendances entre des objets/variables aléatoires. Grâce à cette caractéristique, il est possible d'utiliser des liaisons géométriques entre différentes parties du corps humain. La combinaison de ces deux architectures permet aisément de surpasser les méthodes traditionnelles en vision par ordinateur mais un nouveau problème se présente : les nombreux paramètres visuels (variations de corps humains, habits, éclairage, ...). Ce problème est décomposable en deux parties : premièrement, les différentes parties du corps humain (avec un RNC par Deformable Part Models), deuxièmement, des modèles discriminants basés sur de l'apprentissage profond plus tolérants aux changements de données en entrée.

[1] Spatial Pyramid Matching sur des images en niveau de gris

3.3 Reconnaissance de phrases [33]

La reconnaissance du langage naturel est effectuée à l'aide de réseaux de neurones. Le plus impactant récemment a été BERT [3], qui a été créé et utilisé par Google. Dans une phrase, il se sert des mots situés avant et après le mot inconnu pour le prédire. L'exemple ci-dessous a été effectué en 2014.

Le réseau de neurones proposé par Hu et al. [33] a pour but d'adapter les réseaux de neurones convolutifs pour la correspondance sémantique du langage naturel ; le réseau sert à reconnaître (ou non) la correspondance entre deux phrases. Ce modèle n'a besoin d'aucun prérequis préalable sur le langage naturel comme le montrent les expériences faites sur plusieurs langues.

Avant d'utiliser ce réseau, il a été enrichi par des bases de données de mots dans différentes langues : une base d'environ 1 milliard de mots anglais et une autre de 300 millions de mots chinois. Après cela, l'entrainement du réseau s'effectue en fonction des tests effectués par la suite (phrase longue, courte, ...).

Il est construit de la manière suivante. D'abord, les premières couches convolutives réduisent la phrase fournie en entrée en un ensemble de mots importants pour son sens (plusieurs sorties sont possibles). Ensuite, un ensemble de couches vont découper ces sorties en plusieurs segments puis les regrouper en un vecteur qui pourra être analysé. Pour la correspondance entre deux phrases, chaque phrase est d'abord analysée séparément grâce aux réseaux convolutifs sans qu'elles ne se connaissent puis elles sont ensuite comparées l'une à l'autre.

4 La transparence des réseaux de neurones

Les réseaux de neurones sont largement développés aujourd'hui notamment dans le traitement d'image, et sont efficaces mais également peu robustes. Il a été montré que pour certaines perturbations parfois minimes, ces réseaux peuvent donner des résultats aberrants [34]. Pour améliorer ces réseaux et pouvoir leur faire confiance, la transparence est primordiale. La transparence d'un système est le fait, pour un utilisateur humain, de pouvoir comprendre totalement les résultats de ce système et la manière dont ils ont été obtenus [4]. Dans de nombreuses situations, les systèmes de prédiction ne sont que très peu transparents [4]. La transparence influe sur leur complexité et parfois sur leur efficacité. Ainsi, le but est de trouver un compromis entre l'efficacité et la transparence [8]. Cette transparence, en plus d'aider l'utilisateur à comprendre les résultats et de permettre une meilleure interprétation, est également utile pour la sécurité et la maintenance. En effet, grâce à cela, on peut comprendre les erreurs commises lors d'une prédiction et les corriger [4]. Différentes méthodes ont donc été développées comme celles qui sont présentées ci-dessous.

4.1 Méthode par visualisation

Il existe plusieurs techniques basées sur la visualisation des réseaux de neurones convolutifs. Certaines s'appuient sur une classification de pixels d'une image comme pour Grad-Cam et la méthode présentée par Pedro Pinheiro et Ronan Collobert. D'autres sont une classification plus générale des objets appelée "carte de saillance" des objets, que nous allons présenter dans le paragraphe suivant.

La méthode Grad-CAM est une méthode de visualisation pour la classification d'images qui permet de faire une carte sur la prédiction du réseau [8]. Cette technique montre les pixels ayant la plus forte influence sur la décision du réseau (cf. Fig. 1). Un humain est donc en mesure de comprendre pour quelles raisons le réseau a établi une prédiction précise. Cette approche est utile pour la différenciation de deux réseaux ayant les mêmes résultats. En effet, elle montre les différences faites lors de la prédiction et peut ainsi aider à discriminer deux systèmes pour identifier le plus robuste d'entre eux [8]. Cependant, comme dit précédemment, il y a un compromis à faire entre l'interprétabilité et la fidélité du modèle. Ainsi, la transparence n'est présente que sur certains points du modèle [35]. En effet, cette méthode met en évidence seulement les zones de pixels utilisées par les réseaux mais n'explique pas totalement son fonctionnement. Une grande partie de la prédiction reste encore opaque pour l'utilisateur. Cette solution est donc une solution locale qui dépend de l'entrée du modèle [8].

La "carte de saillance des objets" est une technique qui s'appuie sur la visualisation [4]. Comme la précédente, cette méthode consiste, pour le traitement d'image, à classer des pixels en fonction de leur influence sur la classification. Elle permet donc de savoir quelles perturbations vont agir sur la prédiction. Ainsi, on peut constater la robustesse du réseau de neurones. Cette technique a une application plus générale que

Fig. 4. Exemple d'utilisation de Grad-Cam [8]

la première car elle est applicable sur des objets. Il est possible d'établir une carte de saillances des objets qui influent plus ou moins sur le système. La carte de saillance des objets peut-être également difficile à comprendre pour n'importe quelle personne, mais elle permet cependant l'interprétation des réseaux de neurones même si, comme la technique précédente, elle ne met qu'en lumière certains aspects de l'interprétation du réseau. Cette méthode est peut être moins compréhensible que l'autre visuellement mais elle est plus générale.

Il existe également une technique similaire basée sur les pixels avec des réseaux neuronaux convolutifs [36]. En effet, pour effectuer une segmentation en objets sur une librairie d'images, chaque image doit avoir un pixel (ou aucun) correspondant à une étiquette, ainsi pour l'entraînement tous les pixels appartenant à cette classe d'objets sont déduits par le réseau de neurones. Le réseau de neurones utilisé donne plus d'importance à ces pixels en question avec la couche d'agrégation, ce qui entraîne convenablement le réseau pour la classification [36]. L'extraction de caractéristiques discriminantes est dans un premier temps faite par la première couche d'un réseau basée sur Overfeat [37]. Overfeat est un classifieur d'images créé en 2014 par Sermanet et al. [37], basé sur un réseau de neurones convolutif, et qui permet donc d'extraire les caractéristiques souhaitées d'une banque d'images. Les données issues de cette couche passent ensuite dans la couche LSE (Log-Sum-Exp) pour convertir les labels sur les pixels à des labels sur les images. Mais cette approche produit des faux positifs, ainsi, deux post-traitements (image-level prior (ILP) et smoothing priors (SP)) y sont appliqués pour augmenter le nombre d'informations. Ce modèle entraîné à partir de données faiblement supervisées est meilleur que différentes méthodes existantes (MIM, GMIM, PGC) pour le même type d'entraînement [36]. De plus, comparé à des réseaux entraînés de manière complètement supervisée, ce modèle donne des résultats proches des performances déjà montrées.

4.2 Méta-prédicteurs [5]

Contrairement aux techniques précédentes, la technique que nous allons présenter ici n'est pas basée sur la visualisation des réseaux mais sur leur comportement global.

La technique proposée Ruth C. et Andrea Vedald vise à formaliser la compréhension des boîtes noires des systèmes [5]. Elle introduit les métas prédicteurs. Cette méthode se concentre sur ce que le modèle a appris et la façon dont il l'a appris pour pouvoir tirer des conclusions sur ce qu'il produit ensuite et, pour quelles raisons. Son principe est de repérer quelles parties de l'image d'entrée a une influence sur la sortie, puis de tirer certaines règles sur la façon de prédire du réseau en perturbant de différentes façons les données d'entrée (pour lesquelles la prédiction est correcte) et en étudiant l'influence de ces perturbations sur le résultat. Ces règles permettent ensuite de savoir ce que le système va prédire sur d'autres entrées. Ces différentes entrées ont un lien avec les entrées sur lesquelles les règles ont été traduites. La déduction de ces règles peut être automatisée car elles sont similaires à un problème d'apprentissage ou une minimisation du risque empirique (trouver la règle la plus pertinente).

Cette technique met également en évidence les points faibles des systèmes opaques en montrant les perturbations qui les font varier trop fortement. Elle est beaucoup plus générale que les précédentes car elle s'applique à n'importe quel modèle d'apprentissage. De plus, aucune modification du réseau n'est nécessaire pour appliquer cette méthode. Cependant, elle demande plus de temps dans le sens où, plus les perturbations

sont nombreuses et différentes, plus les informations et règles extraites de cette boîte noire sont nombreuses et précises.

4.3 Fonctions d'influence [7]

Cette technique proposée par Koh et al. [7] comme la précédente cherche à expliquer les réseaux en les perturbants. Cependant elle introduit une autre méthode pour interpréter les résultats.

Les fonctions d'influence sont un moyen efficace de comprendre les perturbations des images d'entrée qui ont une forte influence sur le résultat du système sans devoir le réentrainer au préalable. "La fonction d'influence est une mesure de l'importance de la dépendance des paramètres du modèle ou des prédictions par rapport à une instance de formation" [38]. L'idée mathématique derrière ces fonctions est de trouver une fonction qui calcule l'influence que peut avoir la modification de pondération des entrées (poids). Cette technique permet la différenciation de deux modèles à l'apparence identique en montrant la façon dont ils prédisent deux entrées et les différentes perturbations qui influent sur ces prédictions.

Cette technique est fiable car elle s'appuie sur des méthodes statistiques dites robustes c'est-à-dire qui ne prennent pas en compte les données aberrantes [38]. Elle permet également de voir la vulnérabilité du modèle en mettant en évidence certaines perturbations parfois minimes mais ayant une grande influence sur les prédictions. Ainsi, elle aide à l'amélioration de ces modèles et à leur maintenance. Cependant, elle ne fonctionne pas sur tous les modèles d'apprentissage.

4.4 Neural Interaction Transparency (NIT) [6]

Cette méthode se différencie de toutes celles que nous venons de présenter car elle s'intéresse plus précisément aux réseaux de neurones et aux interactions entre les neurones pour essayer d'expliquer le comportement du réseau.

NIT proposé par Tsang et al. [6] est un environnement qui permet d'augmenter la transparence des réseaux de neurones. En effet, il a pour objectif principal la compréhension du fonctionnement de ces réseaux et de leurs résultats. Pour cela, il tente de mettre en lumière les liens cachés des neurones en les classant sous forme de couches de groupes de neurones avec des interactions. Pour arriver à former cette structure, l'environnement démêle les interactions entre les neurones en appliquant des pénalités sur certaines et ainsi mettre en évidence les neurones utilisés (phase de démêlage).

Par rapport aux autres techniques développées ici, celle-ci s'intéresse directement à la structure du réseau. Plutôt que de s'intéresser seulement aux résultats et de laisser le reste du comportement opaque pour l'utilisateur, elle étudie les couches cachées des réseaux de neurones pour comprendre leurs interactions. De plus ce modèle statistique appelé GAM (Generalized Additive Models) [39] est construit très rapidement par rapport aux autres méthodes qui elles ont besoin de parcourir chaque interaction pour construire le GAM. Cependant la phase de démêlage lors de l'entraînement peut prendre beaucoup de temps en fonction de certains paramètres. De plus avec cette méthode on ne peut pas savoir si les interactions apprises entre neurones sont correctes.

5 Conclusion

L'intelligence artificielle utilise les réseaux de neurones qui se sont fortement développés ces dernières années. Ils sont de plusieurs types en fonction de l'utilisation qui en est faite. De nombreux domaines d'application utilisent ces réseaux.

Les réseaux de neurones sont, par exemple, largement développés pour la reconnaissance de chiffres manuscrits et numérisés, la reconnaissance d'actions humaines ou encore celle de phrases. Il existe de nombreux autres exemples non développés dans cet état de l'art.

Le problème majeur de ces réseaux est leur transparence, il est difficile d'interpréter correctement leurs résultats mais aussi de comprendre leurs erreurs. Ainsi, certaines techniques utilisant différents aspects des neurones ont été développées. Ces techniques sont plus ou moins efficaces et interprétables par un utilisateur. Les techniques de visualisation sont par exemple les plus faciles à comprendre et à interpréter pour un utilisateur. Cependant des méthodes plus formelles et générales ont été développées telles que les méta-prédicteurs ou encore les fonctions d'influence, qui sont peut-être moins faciles à interpréter et plus longues à

mettre en place, mais plus fiables. D'autres techniques telles que les NIT ont été développées. Cette dernière technique s'intéresse vraiment au fonctionnement des neurones entre eux et tente ainsi de mettre en lumière complètement le comportement de ces réseaux.

Plusieurs techniques ont été développées chacune ayant leur particularité et cherchant à mettre en lumière certains aspects des boîtes noires. Elles ont toutes à différents niveaux des avantages et des inconvénients, car un compromis doit être fait entre la fiabilité et la complexité des réseaux et leur transparence.

Remerciements

Nous voudrions remercier Madame la professeure Josiane Mothe du laboratoire de recherche IRIT UMR5505 CNRS Lab, Université de Toulouse, INSPEE pour ses précieux conseils ainsi que ses commentaires tout au long du module, ce qui nous a permis d'aboutir à ce document.

References

1. Lawrence, S., Giles, C., Ah Chung Tsoi, Back, A.: Face recognition: a convolutional neural-network approach. IEEE Transactions on Neural Networks **8**(1) (January 1997) 98–113
2. Boulanger-lew, N., Bengio, Y., Vincent, P.: Audio Chord Recognition with Recurrent Neural Networks. (2013)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] (October 2018) arXiv: 1810.04805.
4. Iyer, R., Li, Y., Li, H., Lewis, M., Sundar, R., Sycara, K.: Transparency and Explanation in Deep Reinforcement Learning Neural Networks. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. AIES '18, New Orleans, LA, USA, Association for Computing Machinery (December 2018) 144–150
5. Fong, R.C., Vedaldi, A.: Interpretable Explanations of Black Boxes by Meaningful Perturbation. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, IEEE (October 2017) 3449–3457
6. Tsang, M., Liu, H., Purushotham, S., Murali, P., Liu, Y.: Neural Interaction Transparency (NIT): Disentangling Learned Interactions for Improved Interpretability. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., eds.: Advances in Neural Information Processing Systems 31. Curran Associates, Inc. (2018) 5804–5813
7. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17, Sydney, NSW, Australia, JMLR.org (August 2017) 1885–1894
8. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-CAM: Why did you say that? arXiv:1611.07450 [cs, stat] (January 2017)
9. Larousse, : Encyclopédie Larousse en ligne - intelligence artificielle Library Catalog: www.larousse.fr.
10. Cardon, D., Cointet, J.P., Mazières, A.: La revanche des neurones: L'invention des machines inductives et la controverse de l'intelligence artificielle. Réseaux **n 211**(5) (2018) 173
11. : Artificial neural networks technology. `http://andrei.clubcisco.ro/cursuri/f/f-sym/5master/aac-nnga/AI_neural_nets.pdf` (Accessed on 04/22/2020).
12. Jain, A., Mao, J., Mohiuddin, K.: Artificial neural networks: a tutorial. Computer **29**(3) (March 1996) 31–44
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 1097–1105
14. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 766–774
15. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. (1986)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 1097–1105
17. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 1790–1798
18. Graves, A., Jaitly, N.: Towards End-to-End Speech Recognitionwith Recurrent Neural Networks. (2014) 9
19. Sak, H., Senior, A., Rao, K., Beaufays, F.: Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. arXiv:1507.06947 [cs, stat] (jul 2015) arXiv: 1507.06947.
20. Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J.R., Schuller, B.: Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR. In Vincent, E., Yeredor, A., Koldovský, Z., Tichavský, P., eds.: Latent Variable Analysis and Signal Separation. Volume 9237. Springer International Publishing, Cham (2015) 91–99 Series Title: Lecture Notes in Computer Science.
21. Neveu: Comprendre les réseaux de neurones récurrents (RNN) - YouTube (jan 2019)
22. Tch, A.: The mostly complete chart of Neural Networks, explained (aug 2017) Library Catalog: towardsdatascience.com.
23. Lee, Y.: Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. Neural Computation **3**(3) (1991) 440–449 PMID: 31167319.
24. Knerr, S., Personnaz, L., Dreyfus, G.: Handwritten digit recognition by neural networks with single-layer training. IEEE transactions on neural networks **3 6** (1992) 962–8
25. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern recognition **36**(10) (2003) 2271–2285
26. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D.: Handwritten Digit Recognition with a Back-Propagation Network. In Touretzky, D.S., ed.: Advances in Neural Information Processing Systems 2. Morgan-Kaufmann (1990) 396–404

27. Li Deng: The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. IEEE Signal Processing Magazine **29**(6) (November 2012) 141–142
28. Decoste, D.: Training Invariant Support Vector Machines. (2002) 30
29. Ravi Babu, U., Kumar Chintha, A., Venkateswarlu, Y.: Handwritten Digit Recognition Using Structural, Statistical Features and K-nearest Neighbor Classifier. International Journal of Information Engineering and Electronic Business **6**(1) (February 2014) 62–68
30. Ionescu, R.T., Ionescu, A.L., Mothe, J., Popescu, D.: Patch autocorrelation features: a translation and rotation invariant approach for image classification. Artificial Intelligence Review **49**(4) (2018) 549–580
31. Ji, S., Xu, W., Yang, M., Yu, K.: 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1) (January 2013) 221–231
32. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 1799–1807
33. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 2042–2050
34. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, IEEE (June 2015) 427–436
35. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
36. Pinheiro, P.O., Collobert, R.: From Image-Level to Pixel-Level Labeling With Convolutional Networks. (2015) 1713–1721
37. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv:1312.6229 [cs] (February 2014) arXiv: 1312.6229.
38. Molnar, C.: Interpretable Machine Learning. (2019) `https://christophm.github.io/interpretable-ml-book/`.
39. Hastie, T.J.: Generalized additive models. In: Statistical models in S. Routledge (2017) 249–307