

Les mathématiques des réseaux de neurones

Partie 1 : les réseaux discriminatifs

Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr

Résumé

Depuis 2012, les réseaux de neurones profonds ont révolutionné l'apprentissage automatique. Bien que relativement ancienne, cette technique a permis ces dernières années des avancées très spectaculaires pour la reconnaissance de textes, de sons, d'images et de vidéos. Comprendre les enjeux de ces méthodes soulève des questions à l'interface entre les mathématiques et l'algorithme. Dans cet article, je vais expliquer la structure de ces réseaux ainsi que les concepts clefs de leur apprentissage supervisé.

1 L'algorithme et les mathématiques de l'apprentissage

Les réseaux de neurones sont des algorithmes, qui permettent à partir d'une entrée x (par exemple une image) de calculer une sortie y . Comme montré à la figure 1, cette sortie est le plus souvent un ensemble de probabilités : par exemple la première sortie est la probabilité que l'image contienne un chat (plus ce nombre est proche de 100%, plus cela signifie que l'algorithme est sûr de lui), la deuxième est la probabilité que l'image contienne un chien, etc. Pour simplifier, on ne considèrera dans nos exemples que deux classes : les chats et les chiens, mais en pratique on peut considérer une sortie y avec plusieurs milliers de classes. On se restreint également à l'exemple des images, mais les réseaux de neurones sont également très performants pour reconnaître des textes ou des vidéos.

Mathématiquement, un tel algorithme définit une fonction f (c'est-à-dire que $y = f(x)$). Le programme informatique qui permet de calculer cette fonction est très simple : il est composé d'un enchaînement de plusieurs étapes, et chaque étape effectue des calculs élémentaires (des additions, des multiplications, et un maximum). En comparaison, les programmes informatiques que l'on trouve dans le système d'exploitation d'un ordinateur sont beaucoup plus complexes. Mais ce qui fait l'énorme différence entre un algorithme « classique » et un réseau de neurones, c'est que ce dernier dépend de paramètres, qui sont les poids des neurones. Avant d'utiliser un réseau de neurones, il faut modifier ces poids pour que l'algorithme puisse résoudre le mieux possible la tâche demandée. Ceci s'effectue à l'aide de méthodes mathématiques et algorithmiques que l'on va expliquer dans les sections suivantes. C'est ce que l'on appelle « entraîner » un réseau de neurone, et ceci nécessite beaucoup de temps, de calculs machine et d'énergie.

Utiliser à bon escient de tels algorithmes nécessite donc des compétences en informatique et en mathématiques. Il faut ainsi manipuler les concepts clefs de l'algorithme (méthodes itéra-

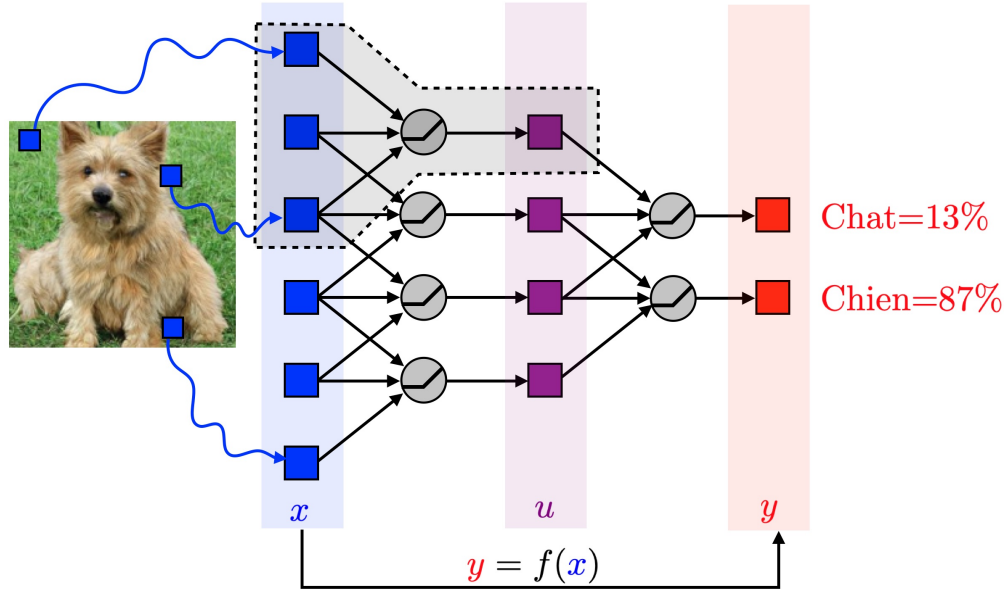


FIGURE 1 – Exemple d’un réseau de neurones discriminatif avec deux couches.

tives, temps de calcul, espace mémoire, implémentation efficace, ...) et des mathématiques (algèbre linéaire, optimisation, statistiques, ...).

2 Réseaux de neurones discriminatifs

Un réseau de neurones artificiel est construit autour d’une métaphore biologique. On connaît relativement bien la structure du cortex visuel primaire, et la découverte en 1962 de l’organisation des neurones dans les premières couches [?] a valu le prix Nobel en physiologie à Hubel et Wiesel. Ainsi, dans une vision extrêmement simplifiée du fonctionnement du cerveau, les neurones sont organisés en couches, chaque neurone récupère de l’information d’une couche précédente, effectue un calcul très simple, et communique son résultat à des neurones de la couche suivante. Il faut cependant garder à l’esprit qu’il ne s’agit que d’une métaphore et une source d’inspiration : les réseaux biologiques ont des connexions beaucoup plus complexes et les équations mathématiques qui les régissent sont également très complexes (elle ont été découvertes par Alan Hodgkin et Andrew Huxley en 1952 [?] et ils ont eu le prix Nobel). Il reste ainsi difficile de mettre précisément en relation les performances parfois surprenantes des neurones artificiels et les capacités cognitives du cerveau. Par exemple, les techniques d’entraînement des réseaux artificiels que l’on va maintenant expliquer sont très différentes de la façon dont un enfant apprend.

La figure 1 détaille un exemple d’un tel réseau artificiel. Pour simplifier, il est ici composé de seulement deux couches de neurones (la première couche entre x et u , la deuxième entre u et y), mais les réseaux actuels les plus performants peuvent comporter plusieurs dizaines de couches, on dit qu’ils sont plus profonds. Dans notre exemple, les entrées x sont les pixels d’une image. Une image contient typiquement des millions de pixels, et la figure n’en représente volontairement qu’un petit nombre : un réseau réaliste est très complexe. De plus, chaque pixel qui compose x est en fait composé de 3 valeurs (une pour chaque couleur primaire rouge, vert et bleu).

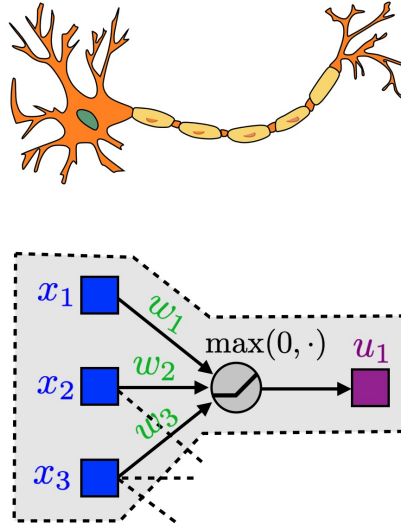


FIGURE 2 – Neurones biologique et artificiel.

Le passage d’une couche (par exemple la couche x des entrées) à une autre (par exemple la deuxième couche u , qui est une couche « cachée » au milieu du réseau) se fait par l’intermédiaire d’un ensemble de neurones artificiels. Un neurone est représenté sur la figure 2. C’est le premier neurone, celui qui calcule la première valeur u_1 qui compose la couche u . Ce neurone connecte un certain nombre d’éléments de la première couche (ici trois : x_1, x_2, x_3 , mais il peut y en avoir plus) à un seul élément de la deuxième, donc ici u_1 . La formule calculée par le neurone est

$$u_1 = \max(w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3, 0).$$

Le neurone effectue ainsi une somme pondérée des trois entrées, avec trois poids w_1, w_2, w_3 . Puis le neurone calcule le maximum entre cette somme et zéro. On peut également utiliser une autre fonction que la fonction maximum, mais celle-ci est la plus populaire. Il s’agit d’une opération de seuillage. On peut la comparer aux neurones biologiques qui laissent ou non passer l’information suivant s’ils sont suffisamment excités ou pas. Ainsi, si la somme pondérée $w_1x_1 + w_2x_2 + w_3x_3$ est plus petite que 0, alors le neurone renvoie la valeur $u_1 = 0$, sinon il renvoie la valeur de cette somme et la place dans u_1 .

De tels réseaux de neurones ont été introduits par Rosenblatt [?] en 1957, qui les a appelés « perceptrons ». Les premiers perceptrons ne contenaient qu’une seule couche. De telles architectures avec une seule couche sont trop simples pour pouvoir effectuer des tâches complexes. C’est en rajoutant plusieurs couches que l’on peut calculer des fonctions plus complexes. Les réseaux de neurones profonds utilisent ainsi un très grand nombre de couches. Depuis quelques années, ces architectures ont permis d’obtenir des résultats très impressionnants pour faire de la reconnaissance d’images et de vidéos ainsi que pour la traduction automatique de textes. Ce sont ces recherches sur les réseaux profonds qui ont permises au chercheur français Yann Le Cun ainsi qu’à Geoffrey Hinton et Yoshua Bengio [?] d’obtenir le prix Turing en 2019. Ce prix est considéré comme étant l’équivalent du prix Nobel en informatique.



FIGURE 3 – Exemples d’images issues de la base de données ImageNet [?] utilisées pour l’apprentissage.

3 L’apprentissage supervisé d’un réseau de neurones

L’entraînement d’un réseau de neurones consiste à choisir les « meilleurs » poids possibles de l’ensemble des neurones qui compose un réseau (par exemple en particulier les poids w_1 , w_2 et w_3 du neurone montré à la figure 2). Il faut ainsi choisir les valeurs de ces poids afin de résoudre le mieux possible la tâche étudiée, et ceci sur un ensemble de données d’apprentissage. Pour la reconnaissance d’objets dans les images, il s’agit d’un problème d’apprentissage supervisé : on dispose à la fois des images x et des y (les probabilités de présence d’un chat et/ou d’un chien dans l’image). La figure 3 montre quelques exemples d’images utilisées pour entraîner un réseau, pour lesquelles on sait ce qu’elles contiennent (la classe des chats et la classe des chiens). Il faut donc, en amont de la phase d’apprentissage, que des humains fasse un long et fastidieux travail d’étiquetage de milliers voir de millions d’images.

La procédure d’entraînement consiste ainsi à modifier les poids w tels que pour chaque x , le réseau f prédise aussi précisément que possible le y associé, c’est-à-dire que l’on souhaite à la fin de l’entraînement que $y \approx f(x)$. Un choix simple est de minimiser la somme $E(w)$ des carrés des erreurs, ce qu’on écrit mathématiquement comme

$$\min_w E(w) = \sum_{(x,y)} (f(x) - y)^2$$

où il faut se rappeler que la fonction f dépend de w . Ceci correspond à un problème d’optimisation, car il faut trouver un jeu de paramètres qui optimise une certaine quantité d’intérêt. C’est un problème difficile, car il y a beaucoup de paramètres, et ces paramètres, surtout ceux des couches cachées, influencent de façon très subtile le résultat. Heureusement, il existe des méthodes mathématiques et algorithmiques performantes pour résoudre de façon satisfaisante ce type de problème d’optimisation. Elles ne sont pas encore totalement comprises sur le plan théorique et c’est un domaine de recherche en pleine explosion. Ces méthodes d’optimisation modifient les poids w du réseau pour l’améliorer et diminuer l’erreur d’entraînement $E(w)$. La règle mathématique pour décider de la stratégie de mise à jour des poids s’appelle la rétro-propagation [?] et est une merveille d’ingéniosité, c’est un cas particulier d’une méthode mathématique et algorithmique qui s’appelle la différentiation automatique à l’envers [?].

Ces techniques d'apprentissage supervisé datent pour l'essentiel des années 1980. Mais c'est seulement en 2012 qu'un papier de Krizhevsky, Sutskever et Hinton [?] crée un coup de tonnerre en montrant que les réseaux profonds permettent de résoudre efficacement des problèmes de reconnaissance d'images. Cette révolution a été possible grâce à la combinaison de trois ingrédients : des nouvelles bases de données beaucoup plus grandes qu'auparavant [?]; des grosses puissances de calcul grâce aux processeurs graphiques (les « GPUs », qui étaient auparavant cantonnés aux jeux vidéo); l'introduction de plusieurs techniques d'optimisation qui stabilisent l'apprentissage [?].

4 L'efficacité des réseaux de neurones

George Cybenko a démontré en 1989 [?] qu'un réseau de neurones f avec deux couches peut approcher aussi précisément que l'on veut n'importe quelle fonction continue f_0 (donc en quelque sorte résoudre n'importe quelle tâche, représentée par la fonction f_0 inconnue, qui serait capable de reconnaître des objets dans n'importe quelle image) pour peu que la taille de la couche interne u (donc le nombre de neurones) soit arbitrairement grande. Ce n'est pas pour autant qu'un tel réseau f avec seulement deux couches marche bien en pratique. Pour appliquer le théorème de Cybenko, il faut pouvoir disposer d'un nombre de données d'apprentissage potentiellement infini, ce qui est très loin d'être le cas en pratique. Le but final de l'apprentissage n'est pas de minimiser l'erreur d'apprentissage $E(w)$, mais de pouvoir prédire aussi précisément que possible sur des nouvelles données. Si l'on dispose de peu de données, on risque de ne pas pouvoir apprendre assez précisément, et donc de faire des mauvaises prédictions : la fonction f sera en réalité très loin de la fonction f_0 idéale que l'on voudrait apprendre si on avait une infinité d'exemples.

Afin d'effectuer les meilleures prédictions possibles avec un nombre limité de données d'entraînement, on cherche donc les architectures de réseaux les plus adaptées, qui peuvent capter efficacement l'information présente dans les données. Pour les données images et les textes, les recherches de ces dernières années ont montré que certains réseaux très profonds, mais avec relativement peu de connexions entre les couches, marchent très bien. Ces réseaux de neurones profonds (avec de nombreuses couches) se sont ainsi avérés incroyablement efficaces sur les données très « structurées » comme les textes, les sons et les images. Par exemple, pour une image, les pixels ont des relations de voisinage, et on peut imposer des connexions spécifiques (une architecture) et ne pas connecter tous un neurone avec tous les autres mais seulement avec ses voisins (sinon il y aurait trop de connexions). De plus, on peut imposer que les poids associés à un neurone soit les mêmes que ceux associés à un autre neurone. On appelle ce type de réseaux les réseaux convolutifs [?]. Pour l'instant, il n'y a pas d'analyse mathématique qui explique cette efficacité des réseaux convolutifs profonds, et il y a donc besoin de nouvelles avancées mathématiques pour comprendre les comportements et les limitations de ces réseaux profonds.