Introduction aux réseaux de neurones

CONSIDÉRATIONS ACTUARIELLES

ApSTAT TECHNOLOGIES

5255 Av. Decelles, suite 2030
Montréal (Québec) H3T 2B1
T: 514.592.9301
F: 514.340.6850
info@apstat.com
www.apstat.com

1 INTRODUCTION

Au cours de la dernière décennie, les algorithmes d'apprentissage statistique ont suscité beaucoup d'intérêt dans le milieu académique et au sein d'entreprises de diverses industries. Ils ont été implantés avec succès pour l'accomplissement de tâches prédictives reliées à des processus statistiques observés pour lesquels on peut identifier plusieurs variables explicatives. Ce document se concentre sur une classe particulière de ces algorithmes : les réseaux de neurones artificiels. Les réseaux de neurones tirent leur puissance de modélisation de leur capacité à capter les dépendances de haut niveau, c.-à-d. qui impliquent plusieurs variables à la fois.

Ce document vise à introduire le lecteur aux réseaux de neurones : premièrement, nous effectuons un survol historique de l'évolution des réseaux de neurones (section 2). Ensuite, nous décrivons formellement comment, à partir d'une série de variables explicatives, un réseau de neurones calcule la valeur prédite du processus que l'on tente de modéliser (section 3). Nous expliquons le processus d'optimisation des paramètres (section 4) et la méthodologie liée au choix de la complexité du réseau de neurones (section 5). Finalement, nous donnons une interprétation mathématique de la raison pour laquelle ces modèles peuvent capter les dépendances de haut niveau (section 6).

2 HISTORIQUE

De façon générale, on situe le début des réseaux de neurones artificiels en 1943 avec les travaux de McCulloch et Pitts qui montrent qu'un réseau de neurones discret, sans contrainte de topologie, peut représenter n'importe quelle fonction booléenne et donc émuler un ordinateur. En 1958, Rosenblatt propose le premier algorithme d'apprentissage, qui permet d'ajuster les paramètres d'un neurone. En 1969, Minsky et Papert publient le livre *Perceptrons* dans lequel ils utilisent une solide argumentation mathématique pour démontrer les limitations des réseaux de neurones à une seule couche. Ce livre aura une influence telle que la plupart des chercheurs quitteront le champ de recherche sur les réseaux de neurones. En 1982, Hopfield propose des réseaux de neurones associatifs et l'intérêt pour les réseaux de neurones renaît chez les scientifiques. En 1986, Rumelhart,

Hinton et Williams publient, l'algorithme de la *rétropropagation de l'erreur* qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches*. À partir de ce moment, la recherche sur les réseaux de neurones connaît un essor fulgurant et les applications commerciales de ce succès académique suivent au cours des années 90.

Aujourd'hui, on retrouve les réseaux de neurones solidement implantés dans diverses industries : dans les milieux financiers pour la prédiction des fluctuations de marché ; en pharmaceutique pour analyser le « QSAR » (quantitative structure-activity relationship) de diverses molécules organiques ; dans le domaine bancaire pour la détection de fraudes sur les cartes de crédit et le calcul de cotes de crédit ; dans les départements de marketing de compagnies de diverses industries pour prévoir le comportement des consommateurs ; en aéronautique pour la programmation de pilotes automatiques ; etc. Les applications sont nombreuses et partagent toutes un point commun essentiel à l'utilité des réseaux de neurones : les processus pour lesquels on désire émettre des prédictions comportent de nombreuses variables explicatives et surtout, il existe possiblement des dépendances non-linéaires de haut niveau entre ces variables qui, si elles sont découvertes et exploitées, peuvent servir à l'amélioration de la prédiction du processus. L'avantage fondamental des réseaux de neurones par rapport aux modèles statistiques traditionnels réside dans le fait qu'ils permettent d'automatiser la découverte des dépendances les plus importantes du point de vue de la prédiction du processus.

3 CALCUL DE LA VALEUR PRÉDITE

Le calcul de la valeur prédite par un réseau de neurones se compose de quelques étapes simples. Premièrement, on calcule une série

*Notons que la thèse de Werbos, publiée en 1974 et restée longtemps sous silence, comprenait les développements mathématiques de la rétropropagation appliquée à des réseaux d'architectures quelconques dont les réseaux de neurones représentent un cas particulier.

de combinaisons linéaires des variables explicatives :

$$v_i \quad = \quad \alpha_{i,0} + \sum_{j=1}^{n} \alpha_{i,j} x_j, \tag{1}$$

où x_j est la $j^{\text{ème}}$ de n variables explicatives et $\alpha_{i,0}$ et $\alpha_{i,j}$ sont les coefficients de la $i^{\text{ème}}$ combinaison linéaire. Le résultat de la combinaison linéaire, v_i, représente une projection dans une direction de l'espace des variables explicatives. Chacune de ces projections combine de l'information provenant potentiellement de plusieurs variables.

La seconde étape consiste à appliquer une transformation non-linéaire à chacune des combinaisons linéaires afin d'obtenir les valeurs de ce que l'on appelle les *unités cachées* ou *neurones* qui forment ensemble la *couche cachée*. Typiquement, on utilise la tangente hyperbolique pour effectuer la transformation non-linéaire :

$$\begin{aligned} h_i &= \tanh(v_i) \\ &= \frac{e^{v_i} - e^{-v_i}}{e^{v_i} + e^{-v_i}}, \end{aligned} \tag{2}$$

où h_i est la $i^{\text{ème}}$ unité cachée. L'utilisation d'une telle fonction de transfert avec une expansion infinie dans ses termes joue un rôle fondamental dans la capacité d'un réseau de neurones de capter les dépendances de haut niveau entre les variables explicatives. C'est le sujet de la section 6.

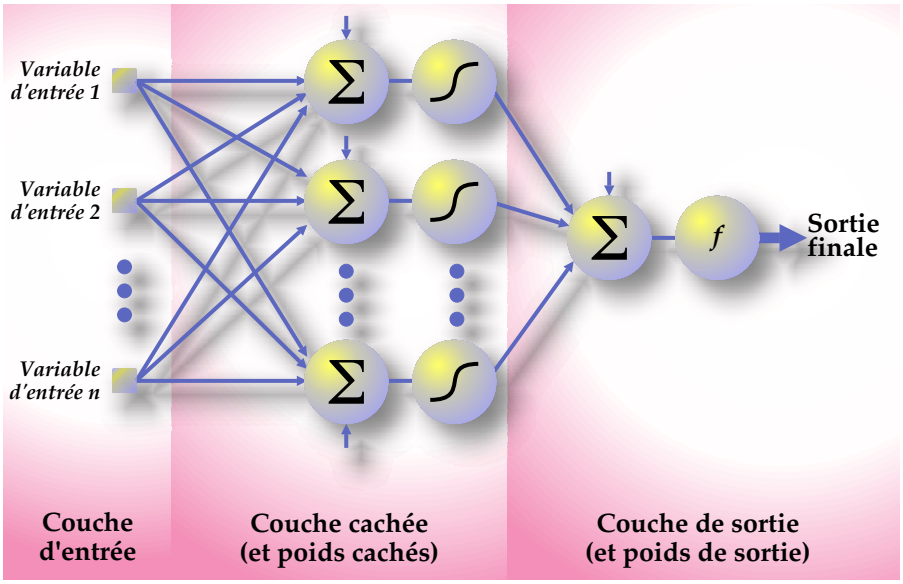
Finalement, les unités cachées sont recombinées linéairement afin de calculer la valeur prédite par le réseau de neurones :

$$p(\vec{x}) \quad = \quad \beta_0 + \sum_{i=1}^{n_h} \beta_i \, h_i, \tag{3}$$

où $p(\vec{x})$ est la valeur prédite par le réseau de neurones, n_h est le nombre d'unités cachées du réseau et β_0 et β_i sont les coefficients de la combinaison linéaire. Les n variables explicatives sont représentées sous forme vectorielle par $\vec{x}$. On peut résumer les étapes du calcul en une seule équation :

$$p(\vec{x}) \quad = \quad \beta_0 + \sum_{i=1}^{n_h} \beta_i \, \tanh\left(\alpha_{i,0} + \sum_{j=1}^{n} \alpha_{i,j} x_j\right). \tag{4}$$

Afin de permettre au lecteur de mieux visualiser la structure d'un réseau de neurones, la Figure 1 illustre, à l'aide d'un graphe de flot, un réseau de neurones tel que défini par les équations de cette section.



► **FIG. 1.** *Topologie d'un réseau de neurones. Dans chaque unité de la couche cachée les variables sont combinées de façon linéaire. Le réseau de neurones applique une transformation non-linéaire à chacune de ces combinaisons. Finalement, les valeurs résultantes des unités cachées sont combinées de façon linéaire pour obtenir la valeur prédite.*

4 OPTIMISATION DES PARAMÈTRES

Tel que mentionné plus haut, la transformation non-linéaire joue un rôle prépondérant dans le réseau de neurones. Par contre, la présence de cette transformation dans les équations du calcul de la valeur prédite fait en sorte que l'on ne peut trouver de solution analytique pour le calcul des paramètres optimaux d'un réseau de neurones . Dans cette section, nous décrivons brièvement la technique d'optimisation la plus couramment utilisée, celle du gradient stochastique.

Tout d'abord, les paramètres du réseau de neurones sont généralement initialisés à des valeurs aléatoires. La distribution de ces valeurs aléatoires suit certaines règles dictées par la théorie. Ensuite, on présente au réseau de neurones un *exemple d'entraînement*, c.-à-d. une paire qui comprend l'ensemble des valeurs des variables explicatives, $\vec{x}$ et la valeur observée du processus, y. Dans une première

étape, on calcule la valeur prédite par le réseau de neurones, $p(\vec{x})$ et l'erreur de prédiction, e, qui découle du fait que $y \neq p(\vec{x})$. Généralement, on utilise l'erreur quadratique qui jouit de certaines propriétés statistiques intéressantes :

$$e \quad = \quad (p(\vec{x}) - y)^2.$$

Dans une seconde étape, on utilise le calcul différentiel pour obtenir la dérivée de l'erreur par rapport à chacun des paramètres du réseau de neurones. Ces valeurs servent à modifier les paramètres de sorte que si l'exemple d'entraînement était présenté à nouveau au réseau de neurones, l'erreur serait moindre. Ces deux étapes (calcul de la valeur prédite et modification des paramètres) sont répétées pour chaque exemple d'entraînement qui est fourni. Ces exemples forment ce que l'on appelle l'*ensemble d'entraînement*. Pour certaines applications, un ensemble d'entraînement peut comprendre plusieurs millions d'exemples d'entraînement. Une fois que chaque exemple de l'ensemble d'entraînement a été présenté au réseau de neurones, on a terminé une *époque* d'entraînement. Les paramètres sont modifiés jusqu'à ce que l'*erreur d'entraînement*, c.-à-d. l'erreur totale commise sur tous les exemples de l'ensemble d'entraînement se stabilise, ce qui peut nécessiter quelques centaines d'époques d'entraînement. La stabilité finale est théoriquement assurée si certaines règles relatives à la modification des paramètres du réseau sont respectées.

5 CHOIX DU NOMBRE D'UNITÉS CACHÉES

Le nombre d'unités cachées (n_h, ci-haut) joue un rôle crucial dans le contrôle de la *capacité* du réseau de neurones. Si la valeur de n_h est trop petite, alors le réseau possède trop peu de paramètres et ne peut capter toutes les dépendances qui servent à modéliser et prédire les valeurs du processus observé. À l'inverse, si l'on choisit une valeur trop grande pour n_h, alors le nombre de paramètres du modèle augmente et il devient possible, pendant la phase d'optimisation des paramètres, de modéliser certaines relations qui ne sont que le fruit de fluctuations statistiques propres à l'ensemble d'entraînement utilisé plutôt que des relations fondamentales de dépendance entre les variables. Il faut comprendre que les réseaux de neurones sont des approximateurs universels, c.-à-d. qu'ils peuvent modéliser

n'importe quelle fonction si le nombre d'unités cachées est suffisant. Autrement dit, un réseau de neurones peut apprendre par coeur un ensemble d'entraînement. Afin de s'assurer que le réseau de neurones s'en tient aux relations fondamentales de dépendance, on utilise, en plus de l'ensemble d'entraînement, un second ensemble appelé *ensemble de validation* : à la fin de chaque époque d'entraînement, on mesure non seulement l'erreur d'entraînement mais aussi l'*erreur de validation*, c.-à-d. l'erreur totale commise sur tous les exemples de l'ensemble de validation. Cette erreur de validation est calculée une fois que la phase d'optimisation des paramètres est terminée.

Après avoir entraîné quelques modèles, chacun avec un nombre différent d'unités cachées, on peut comparer les erreurs d'entraînement et de validation. On obtient généralement le résultat suivant : l'erreur d'entraînement diminue au fur et à mesure que le nombre d'unités cachées augmente. L'erreur de validation, quant à elle, est élevée lorsque le nombre d'unités cachées est faible, décroît avec l'augmentation du nombre d'unités cachées, atteint un minimum pour un certain nombre optimal d'unités cachées, puis croît lorsque le nombre d'unités devient trop grand. C'est donc l'utilisation d'un ensemble de validation, distinct de l'ensemble d'entraînement, qui nous permet de choisir le nombre optimal d'unités cachées ou neurones.

6 CAPTER LES DÉPENDANCES NON-LINÉAIRES DE HAUT NIVEAU

Pour le lecteur intéressé par les aspects plus mathématiques, cette section explique comment il est possible pour les réseaux de neurones de représenter les dépendances non-linéaires de haut niveau entre les variables explicatives. Pour simplifier, supposons que seulement deux variables explicatives, x_1 et x_2, sont utilisées. Dans le cadre de la régression linéaire classique, une technique simple et courante consiste à inclure des combinaisons de degrés supérieurs telles que $x_1^2, x_2^2, x_1x_2, x_1^2x_2, \ldots$ parmi les variables explicatives. Par contre, il est clair que cette approche ajoute un nombre croissant de variables à la régression, à mesure que l'on cherche à inclure des termes d'ordres de plus en plus élevés.

Considérons maintenant une unité cachée d'un réseau de neurones dont les paramètres sont α_0, α_1 et α_2 :

$$h \;=\; \tanh(\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2).$$

Le point central de l'argument est le suivant : si l'on développe la série de Taylor de h en fonction des variables explicatives x_1 et x_2, on obtient tous les termes de tous les ordres, c.-à-d. un nombre infini de termes, chacun étant multiplié par un certain coefficient ($\beta \equiv \tanh \alpha_0$) :

$$\begin{aligned}
&\tanh(\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2) = \\
&\quad \beta + (1-\beta^2)(\alpha_1 x_1 + \alpha_2 x_2) + (-\beta + \beta^3)(\alpha_1 x_1 + \alpha_2 x_2)^2 + \\
&\quad \left(-\tfrac{1}{3} + \tfrac{4\beta^2}{3} - \beta^4\right)(\alpha_1 x_1 + \alpha_2 x_2)^3 + \\
&\quad \left(\tfrac{2\beta}{3} - \tfrac{5\beta^3}{3} + \beta^5\right)(\alpha_1 x_1 + \alpha_2 x_2)^4 + \ldots .
\end{aligned}$$

Toutefois, tous ces termes ne peuvent être contrôlés indépendamment les uns des autres puisque ultimement, tous les coefficients qui multiplient ces termes ne dépendent que de trois valeurs : $\alpha_0, \alpha_1,$ et α_2. Le fait d'ajouter des unités cachées augmente la flexibilité de la fonction calculée par le réseau de neurones : chaque unité est liée aux variables explicatives avec ses propres coefficients, permettant ainsi au réseau de neurones de capter autant de dépendances non-linéaires que le nombre d'unités cachées le permet. Les coefficients qui relient les variables explicatives aux unités cachées peuvent aussi être interprétés comme étant des projections des variables explicatives. Chaque ensemble de coefficients d'une unité cachée représente une direction d'intérêt dans l'espace des variables explicatives.

7 CONCLUSION

Nous espérons que ce document aura permis au lecteur de mieux comprendre les réseaux de neurones et pourquoi ils ont pu susciter un engouement si fort. L'élément essentiel des réseaux de neurones est qu'ils peuvent capter les dépendances non-linéaires de haut niveau entre les variables explicatives, ce qui est possible grâce à la présence d'une transformation, elle-même non-linéaire, dans le calcul de la valeur prédite. Puisque les réseaux de neurones sont des

approximateurs universels, leur utilisation doit aller de paire avec une méthodologie stricte qui permet de capter les relations fondamentales des données tout en évitant de modéliser les fluctuations statistiques propres à un ensemble d'entraînement particulier. Les réseaux de neurones sont de puissants outils de modélisation et de prédiction. Ils ont été adoptés dans divers champs d'application et nous croyons que l'industrie de l'assurance est sur le point de faire de même.