

NEURAL RANDOM FORESTS

G erard Biau ¹ & Erwan Scornet ² & Johannes Welbl ³

¹ *Sorbonne Universit es, UPMC Univ Paris 06, CNRS, Paris, France*

Institut Universitaire de France

gerard.biau@upmc.fr

² *Centre de Math ematiques Appliqu ees,  cole Polytechnique, CNRS,*

UMR 7641, 91128 Palaiseau, France

erwan.scornet@polytechnique.edu

³ *University College London, London, England*

J. Welbl@cs.ucl.ac.uk

R esum e. Les for ets al eatoires, propos ees par L. Breiman (2001), comptent parmi les m ethodes les plus utilis ees dans les probl emes d'estimation de la r egression en grande dimension, particuli erement dans des domaines comme la g enomique. Un lien naturel existe entre ces for ets et les r eseaux de neurones, couramment utilis es pour r esoudre des probl emes de reconnaissance d'image par exemple. Dans cet expos e, nous expliciterons cette connexion et en tirerons parti pour construire des r eseaux de neurones plus rapides. Un r esultat de consistance sera  galement  nonc e.

Mots-cl es. Apprentissage, Arbres de d ecision, For ets al eatoires, R eseaux de neurones.

Abstract. Random forests, designed by Breiman (2001), are among the most powerful methods used in high dimensional regression problems, especially in genomic field. A natural connection exists between random forests and neural networks, a class of algorithms used, for example, to deal with image recognition problems. In this talk, we will recall this connection and use it to build a faster neural network. We will also prove the consistency of this new network.

Keywords. Decision tree, Machine learning, Neural networks, Random forest.

1 Introduction

Les arbres de d ecision sont l'un des outils dont dispose le statisticien pour r esoudre des probl emes d'estimation de la r egression et de classification supervis ee. L'algorithme CART, imagin e par Breiman et al. (1984), est l'un des arbres de d ecision les plus utilis es de nos jours.

Afin de pallier la grande sensibilit e de cette m ethode aux variations dans les donn ees (donn ees fausses ou incompl etes), Breiman [2001] proposa d'introduire de l'al eatoi-re dans

le processus de construction de cet arbre. L'agrégation des arbres aléatoires ainsi obtenus est appelée forêt aléatoire. L'estimation est alors effectuée grâce à la forêt tout entière et non plus grâce à un seul arbre. Elle est obtenue en calculant la moyenne des estimations des différents arbres de la forêt. Intuitivement, si les arbres sont assez distincts, les estimateurs seront suffisamment différents pour que la moyenne de ces estimateurs permettent de s'affranchir d'aberrations ponctuelles dans les données.

Depuis 2001, des avancées significatives ont été faites dans la compréhension du comportement de l'algorithme des forêts aléatoires. Ainsi, Biau, Devroye et Lugosi (2008) ont prouvé la convergence de plusieurs modèles de forêts; Biau (2012) a montré, dans un contexte adapté, que la vitesse de convergence d'une forêt aléatoire particulière ne dépendait pas de la dimension de l'espace ambiant mais seulement du nombre de variables réellement pertinentes. Les forêts ont également fait l'objet de nombreuses publications dans des domaines plus appliqués. Citons par exemple les travaux de Díaz-Uriarte et de Andrés (2006) ou de Genuer (2010) qui ont, tous deux, proposés des méthodes de sélection de variables utilisant les forêts aléatoires.

Les réseaux de neurones comptent également parmi les algorithmes de machine learning les plus utilisés ces dernières années, notamment en reconnaissance de forme, d'image ou d'écriture. Il existe certaines similitudes entre les réseaux de neurones et les forêts aléatoires [e.g., Bengio, 2009], les deux algorithmes proposant de diviser l'espace des co-variables pour effectuer une prédiction. Welbl [2014] et Richmond et al. [2015] ont su tiré parti de cette connexion afin d'améliorer les performances prédictives et le temps de calcul des réseaux de neurones.

Les résultats théoriques permettant d'expliquer les bonnes performances des réseaux de neurones ou des forêts aléatoires sont peu nombreux : l'analyse théorique souffre en effet de la structure récursive de ces algorithmes, ce qui en fait de véritables boîtes noires difficiles à décortiquer.

L'objet de cette présentation est de détailler le lien entre forêts aléatoires et réseaux de neurones et d'expliquer comment modifier les réseaux de neurones en utilisant cette connexion. Je présenterai également un résultat de consistance portant sur la version modifiée des réseaux de neurones.

2 Position du problème

Étant donné un échantillon $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ de variables aléatoires à valeurs dans $[0, 1]^d \times \mathbb{R}$, on cherche dans ce travail à estimer la fonction de régression de Y sur \mathbf{X} , définie par

$$m(\mathbf{x}) = \mathbb{E}(Y | \mathbf{X} = \mathbf{x}), \quad \mathbf{x} \in [0, 1]^d.$$

Pour ce faire, on construit une partition de $[0, 1]^d$ en hyperrectangles. La construction de cette partition est récursive. Chaque cellule est obtenue par découpage d'une cellule

pré-existante. La structure naturellement associée à ce type de construction est un arbre. Une fois la partition construite, on estime, pour tout $\mathbf{x} \in [0, 1]^d$, $m(\mathbf{x})$ par la moyenne des valeurs Y_i associées aux points X_i tombant dans la même cellule que \mathbf{x} . On note cette estimation $m_n(\mathbf{x})$. Afin d'évaluer la qualité de l'estimation, on s'intéresse à l'erreur en moyenne quadratique, définie par

$$\mathbb{E} [(m_n(\mathbf{X}) - m(\mathbf{X}))^2].$$

Les arbres de décision CART qui composent la forêt de Breiman sont caractérisés par les éléments suivants.

1. Le critère de coupure CART [Breiman et al., 1984] permet de déterminer la manière dont les coupes sont effectuées à chaque étape. Il contribue à créer des cellules à l'intérieur desquelles la variance des Y_i est faible.
2. Le critère d'arrêt détermine quand l'algorithme doit arrêter de couper une cellule. L'algorithme CART s'arrête lorsque chaque cellule contient moins de `nodesize` données, où `nodesize` est un paramètre, typiquement égal à 5.

Les forêts aléatoires procèdent de la manière suivante. À partir d'un unique jeu de données, on construit plusieurs jeux de données par sous-échantillonnage. On construit ensuite un arbre sur chacun de ces jeux de données. De plus, pour construire chaque arbre, dans chaque cellule, le critère CART est optimisé non pas sur chacune des variables mais sur un sous-ensemble de `mtry` variables, tirées uniformément parmi les d variables disponibles. Une fois l'ensemble d'arbre construit, l'estimation n'est plus donnée par un seul arbre mais par la moyenne des estimations de chaque arbre. Autrement dit, si l'on veut construire une forêt contenant M arbres, il faut :

1. Se donner M variables aléatoires $\Theta_1, \dots, \Theta_M$ indépendantes, de même loi Θ , correspondant à l'aléatoire introduit dans l'arbre, que ce soit celui lié au sous-échantillonnage des données ou à celui lié à la présélection des variables dans chaque cellule.
2. Construire les M arbres associés et les estimations correspondantes, notées $m_n(\mathbf{x}, \Theta_1), \dots, m_n(\mathbf{x}, \Theta_M)$ pour tout $\mathbf{x} \in [0, 1]^d$.
3. Construire l'estimateur associé à la forêt, noté $m_{M,n}(\mathbf{x})$, en effectuant la moyenne des estimations précédentes.

En pratique, on s'intéresse à l'estimateur $m_{M,n}(\mathbf{x})$ car on ne peut construire qu'un nombre fini d'arbres. Cependant, d'après la loi des grands nombres, on sait que, pour tout \mathbf{x} ,

$$m_{M,n}(\mathbf{x}) \xrightarrow{M \rightarrow \infty} \mathbb{E}_{\Theta} [m_n(\mathbf{x}, \Theta)].$$

L'estimateur limite $m_{\infty,n} = \mathbb{E}_{\Theta} [m_n(\mathbf{x}, \Theta)]$ est l'estimateur de la forêt aléatoire infinie. Contrairement à $m_{M,n}$ qui dépend des M arbres qui composent la forêt et est donc aléatoire, $m_{\infty,n}$ ne dépend pas de l'aléatoire Θ . Il est donc plus facile à étudier.

3 Les arbres de décisions vus comme réseaux de neurones

Voyons maintenant comment un arbre peut se réécrire sous la forme d'un réseau de neurones. On suppose qu'à partir d'un certain jeu de données, on a construit l'arbre CART décrit en Figure 1.

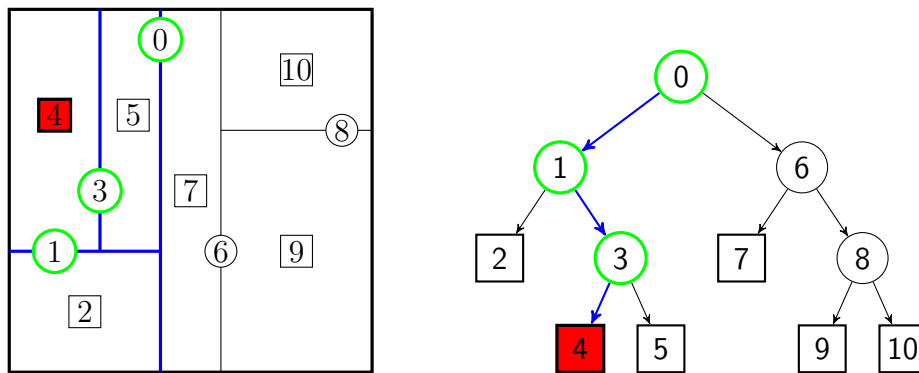


FIGURE 1 – Un exemple d'arbre de régression.

On peut alors construire un réseau de neurones à deux couches cachées dont la prédiction sera identique à celle de l'arbre de décision décrit en Figure 1.

- Chaque noeud de la première couche cachée correspond à une coupure de l'arbre de décision. Ainsi, la variable x_i ($i = 1, 2$) est connectée au noeud j si la coupure associée au noeud j a été effectuée selon la i -ème variable. Chaque noeud de la première couche est donc relié à une unique variable, les coupures étant effectuées parallèlement aux axes.
- Chaque noeud de la seconde couche correspond à une cellule. Plus précisément, le noeud i de la première couche est relié au noeud j de la seconde couche si et seulement si, la i -ème coupure a été utilisée pour construire la cellule j . Par exemple, les coupures 0, 1 et 3 ont servi à construire la cellule 4 : le noeud 4 de la seconde couche est donc relié aux noeuds 0, 1, 3 de la première couche.
- Les sorties des noeuds des deux couches cachées valent 0 ou 1. Les poids du réseaux sont choisis de sorte que seul le noeud correspondant à la cellule contenant l'entrée (x_1, x_2) produise un 1, les autres noeuds de la seconde couche produisant tous 0.
- On peut alors régler les poids entre la seconde couche et la couche de sortie pour prédire la moyenne dans chaque cellule.

Le réseau de neurone associé est décrit en Figure 2. J'expliquerai plus en détail dans l'exposé le lien entre réseau de neurones et arbres de décisions. Nous verrons comment étendre ce lien aux forêts aléatoires. Nous utiliserons alors la forêt aléatoire pour définir les poids non nul du réseau de neurones. Nous obtiendrons ainsi un réseau de neurones avec peu de connexions dont nous montrerons la consistance.

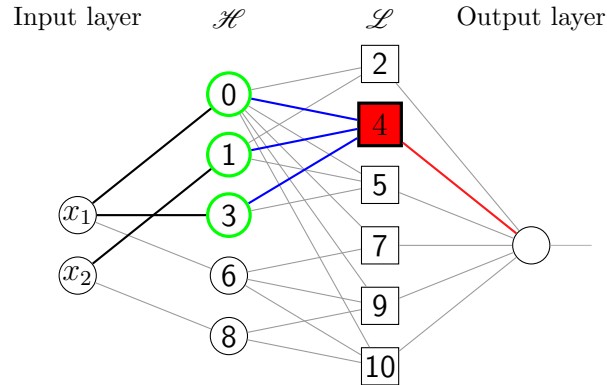


FIGURE 2 – Le réseau de neurone correspondant à l’arbre de régression de la Figure 1.

Bibliographie

Références

- Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2 :1–127, 2009.
- L. Breiman. Random forests. *Machine Learning*, 45 :5–32, 2001.
- L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, Boca Raton, 1984.
- D.L. Richmond, D. Kainmueller, M.Y. Yang, E.W. Myers, and C. Rother. Relating cascaded random forests to deep convolutional neural networks for semantic segmentation. *arXiv :1507.07583*, 2015.
- J. Welbl. Casting random forests as artificial neural networks (and profiting from it). In X. Jiang, J. Hornegger, and R. Koch, editors, *Pattern Recognition*, pages 765–771. Springer, 2014.