

## APPLICATION ET PRÉVISIONS PAR RÉSEAUX DE NEURONES

**Vous utiliserez les librairies R suivantes :** *MASS*, *e1071*, *neuralnet*, *nnet*, *caret*.

L'objectif de ce TP est tout d'abord d'implémenter un cas simple de réseau de neurones, et d'en réaliser les limites. Si le temps le permet, tarifier ensuite un contrat d'assurance automobile en ayant implémenté un réseau neuronal au préalable afin de prédire la sinistralité. Pour rappel, n'importe quelle fonction régulière peut être approchée par un réseau de neurones avec une couche cachée.

### Fonction racine carrée et réseau neuronal

On se propose ici d'approcher la fonction racine carrée par un réseau de neurones. Pour ce faire, charger la librairie *neuralnet* dans un premier temps.

- (1) Simuler uniformément 50 nombres entre 0 et 100. Ce seront les entrées du réseau.
- (2) En calculer la racine carrée. Créer le jeu de données d'apprentissage composé de ces 50 observations et de leur racine carrée (sortie).
- (3) Entraîner le réseau de neurones avec 10 neurones cachés. Pour cela, utiliser la fonction **neuralnet** : quels sont ses arguments ? Quels sont les attributs de l'objet construit ? Lister ces éléments et donnez en leur signification.
- (4) Afficher les résultats complets de la construction du réseau de neurones en accédant aux attributs de l'objet créés. Interpréter les résultats.
- (5) A l'aide de la fonction **plot**, faire un graphe du réseau. Quels sont les éléments représentés en bleu ? En noir ?
- (6) Retrouver à la main la prévision du réseau pour la 1<sup>ère</sup> entrée générée (1<sup>er</sup> individu).
- (7) Evaluer la qualité de prévision du réseau sur un échantillon de validation. Les données test seront les entiers de 1 à 10 au carré. Calculer la sortie prévue par le réseau de neurones construit. En déduire l'erreur commise par le réseau neuronal.
- (8) Essayer maintenant de prédire par le réseau la racine carrée de 225, 400, 2500. Que constatez-vous ? Quel phénomène est illustré ici ?
- (9) Entraîner d'autres réseaux de neurones avec des structures (nombre de couches cachées, nombre de neurones par couche cachée) différentes et tenter de trouver les paramètres de tuning optimaux correspondants. Pour cela, vous pourrez utiliser la librairie *caret* et la fonction **train**. La fonction **nnet** permet également d'accéder à un autre paramétrage.