

# Machine Learning

Martin Guyard et Guillaume Brizolier

TP2 : Renewable Energy Prediction

## 1 Preprocessing des données

### 1.1 Problèmes éventuels dans les données

1. Redondance : cela peut éventuellement ralentir les calculs mais ce n'est pas un problème en soi.
2. Normalisation : Il faut toujours normaliser pour des problèmes informatiques ! On divise chaque colonne par la valeur max et on soustrait la moyenne pour centrer les données.
3. Indépendance : fortement lié à la redondance. Dans notre cas, ce n'est pas un problème.
4. Format : Utilisation de one-hot vector pour convertir les chaînes ASCII en un nombre de colonnes égales au nombre de villes. Il s'agit d'un vecteur où toutes les valeurs sont nulles sauf 1 à 0 qui correspond à l'indicateur de catégorie.

0 | 0 | 0 | 1 | 0 | 0

On utilise la même technique pour la date avec cette fois-ci 3 one-hot vectors : un de 12 cases pour les mois, un de 31 pour les jours, et un de 24 pour les heures.

5. Trous : Si on a des trous, on peut soit faire des moyennes pondérées en fonction de points suivants et précédents, ou tout simplement remplacée par la moyenne (nulle ici car nous avons normalisé les données).

### 1.2 Input

[9, Nbcol, 1] : on a un seul neurone qui est un réel. Output : [batch, 1] Dans notre couche finale, on fait une combinaison linéaire qui sort un nombre sans mettre de fonction d'activation, puisqu'on veut des valeurs dans R.

Au milieu : on met une couche dense fully connected, et ça marche. Les fonctions d'activation au sein du réseau sont celles qu'on veut, mais par contre à celle de la fin on n'en met pas.

## 2 Prévision

target : une ligne!

on peut utiliser une fenêtre de 4 lignes. Input : [9 : nb de lignes, 4 : nb de dimensions tempo que je veux, NbCol : nbre de features] : b(batch), t(time), c(colonnes)

Je ne peux prédire qu'une valeur ! (pas toute la ligne). Du coup la sortie ne change pas.