

Introduction

Introduction à l'apprentissage automatique
Master Sciences Cognitives
Aix Marseille Université

Alexis Nasr

Bienvenue

- Alexis Nasr
- `alexis.nasr@univ-amu.fr`
- Laboratoire Informatique et Systèmes
- Equipe Traitement Automatique du Langage Ecrit et Parlé
- Bâtiment TPR2 - Bureau 4.36 - Campus de Luminy
- Site du cours :
`pageperso.lif.univ-mrs.fr/~alexis.nasr/Ens/MASCO_AA`

Objectifs

- Connaître les principes de base de l'apprentissage automatique.
- Acquérir une bonne connaissance des réseaux de neurones.
- Maîtriser un outil de manipulation de réseaux de neurones (en l'occurrence keras).
- Être capable de réaliser une expérience complète à l'aide de keras.

Pré-requis

- Informatique : programmation en python (cours du premier semestre)
- Mathématique :
 - Fonctions
 - Dérivées
 - Fonctions de plusieurs variables
 - Gradient

Evaluation

- Trois projets : P_1, P_2, P_3
- Un examen : E
- Note finale : $\frac{1}{2}E + \frac{1}{2}\left(\frac{1}{3}P_1 + \frac{1}{3}P_2 + \frac{1}{3}P_3\right)$

Apprentissage Automatique

- L'apprentissage automatique (AA) est un ensemble de méthodes permettant de détecter automatiquement des **régularités** dans des **données**.
- Ces régularités peuvent être utilisées pour mieux **décrire** ces données ou **prédire** certaines de leurs caractéristiques.
- L'AA se présente généralement sous deux formes, l'apprentissage **supervisé** et l'apprentissage **non supervisé**.

Apprentissage supervisé I

- Etant donné des paires d'entrées-sorties

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$

- On souhaite apprendre une fonction f qui décrit la relation existant entre les entrées et les sorties :

$$y_i = f(x_i)$$

- \mathcal{D} est appelé **ensemble d'apprentissage** et N est le nombre d'**exemples**.
- En général, chaque entrée x_i est un vecteur de nombres, de dimension D .
- Ces nombres sont appelés des **traits** ou **caractéristiques** ou encore **features**.

Apprentissage supervisé II

- Les sorties y_i peuvent prendre des formes différentes.
- On s'intéressera aux deux cas suivants :
 - y_i est une **classe** (ou catégorie) matérialisée par un entier pris dans un ensemble fini ($y \in \{1, \dots, C\}$). On parle alors de **classification**.
 - y_i est un **nombre**. On parle alors de **régression**.

Exemples de régression

- Météorologie :
 - y = température
 - x = altitude d'une station météorologique
- Santé :
 - y = tension artérielle
 - x = âge d'un individu
- Agronomie :
 - y = nombre de nids de chenilles processionnaires par arbre
 - x_1 = altitude
 - x_2 = hauteur de l'arbre
 - x_3 = densité de peuplement
- Economie :
 - y = salaire horaire
 - x_1 = années d'étude
 - x_2 = années d'expérience

Exemples de classification

- Classification d'images d'animaux :
 - y = un animal
 - x = les pixels d'une image d'animal
- Analyse de sentiments :
 - y = un sentiment (*colère, joie, déception ...*)
 - x = une phrase
- Identification de personnes :
 - y = identité d'une personne prise dans un ensemble donné
 - x = enregistrement de la voix de la personne.

Apprentissage non supervisé

- En apprentissage non supervisé, on ne dispose que d'entrées $\mathcal{D} = \{(x_i)\}_{i=1}^N$ et l'objectif est de retrouver des régularités permettant, par exemple, de regrouper entre elles les entrées x_i pour constituer des sous ensembles, appelés aussi **clusters**.
- On parle dans ce cas de **clustering**.
- Il s'agit d'un problème moins bien spécifié que l'apprentissage supervisé car on ne sait pas quelles régularités sont intéressantes ni comment évaluer la qualité des prédictions.
- Nous n'aborderons dans ce cours que le cas de l'apprentissage supervisé.

Organisation du cours

sem.	AA	Math	Info.	Projet
1	Introduction Régression polynomiales	Fonctions		
2	Classifieur manuel	Dérivées		
3	Perceptron	Gradient		
4	NN1 - Perceptron multicouches		Keras	Projet 1
5	Algorithme de rétropropagation		numpy	
6	NN2 - Réseaux convolutionnels			Projet 2
7	Descente stochastique du gradient			
8	NN3 - Réseaux récurrents			
9	Régularisation			Projet 3
10	NN4 - Modèles d'attention			

Tous les slides sont disponibles sur le site du cours.

Bibliographie

- Christopher Bishop, *Pattern recognition and machine learning* Springer, 2006.
- Tom Mitchell, *Machine Learning*, McGraw Hill, 1997.
- Yoav Goldberg, *Neural Network Methods for Natural Language Processing*, Morgan & Claypool, 2017.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016.
- Le blog de Christopher Olah <http://colah.github.io>
- Stewart, James. *Essential calculus : Early transcendentals*. Cengage Learning, 8ème édition, 2012