

UNIVERSITÉ DE STRASBOURG  
INTELLIGENCE ARTIFICIELLE  
Licence 3 - UFR Mathématique - Informatique  
**Projet : classification supervisée avec un réseau de neurones**


**Consignes** – Vous travaillerez en binôme et vous soumettrez, sur moodle, votre code source ainsi qu'un court rapport présentant votre travail (en particulier, vous y incluez les réponses aux questions posées dans le présent sujet). La date limite de dépôt de vos projets est le 30 avril 2019 à 20h.

## 1 Les données

Vous allez travailler sur le jeu de données “Cleveland Heart Disease”. Ce jeu de données contient 303 instances comprenant chacune 14 attributs présentés dans le tableau 1. L'objectif de l'apprentissage sur ce jeu de données est de pouvoir dire si un patient est susceptible de développer (ou d'avoir développé) une maladie cardiaque en fonction de différentes mesures physiologiques.

Attributs	Description
age	The person's age in years
sex	The person's sex (1 = male, 0 = female)
chest_pain_type	The chest pain experienced (Value 1 : typical angina, Value 2 : atypical angina, Value 3 : non-anginal pain, Value 4 : asymptomatic)
resting_blood_pressure	The person's resting blood pressure (mm Hg on admission to the hospital)
cholesterol	The person's cholesterol measurement in mg/dl
fasting_blood_sugar	The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
rest_ecg	Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
max_heart_rate_achieved	The person's maximum heart rate achieved
exercise_induced_angina	Exercise induced angina (1 = yes; 0 = no)
st_depression	ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)
st_slope	the slope of the peak exercise ST segment (Value 1 : upsloping, Value 2 : flat, Value 3 : downsloping)
num_major_blood_vessels	The number of major vessels (0-3)
thalassemia	A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversible defect)
target	Diagnosis of a heart disease (0 = no, 1 = yes)


TABLE 1 – Liste des attributs du jeu de données et leur description

 **Observez le tableau 1 pour répondre aux questions suivantes :**

1. Quel est le nom de l'attribut que l'on souhaite prédire ?
2. S'agit-il d'une classification multi-classe ou binaire ?
3. Listez tous les attributs catégoriels.
4. Décrivez deux façons de gérer les attributs catégoriels dans un modèle d'apprentissage.

En fait, en observant ces attributs catégoriels, on note qu'ils sont aussi ordinaux (*i.e.* ordonnés). Donc, pour ce projet, vous n'aurez pas à transformer ces attributs catégoriels.

Les autres attributs sont numériques. On sait que les algorithmes d'optimisation du type "descente de gradients" sont sensibles à ce type d'attributs : un coup d'œil rapide au fichier `.csv` contenant les données nous permet de noter que les magnitudes des attributs numériques sont variées : pour que votre descente de gradient soit efficace, il faut normaliser ces attributs.

 **Normalisation des données**

1. Étant donné que l'on souhaite préserver la variance du jeu de donnée, quelle méthode de normalisation allez-vous employer ?
2. Allez-vous appliquer la normalisation avant ou après avoir séparé vos données en un jeu d'entraînement et un jeu de test ?
3. Dans le jeu de données, identifiez les indices des colonnes que vous allez normaliser.
4. Mettez en œuvre une fonction pour normaliser ces colonnes avec la méthode de normalisation adaptée. Pour cela, on vous propose le prototype suivant :

```
/**
 * Normalisation des attributs numériques
 * @param data    données à traiter
 * @param indices tableau contenant les indices des colonnes à normaliser
 */
public void normalize(float[][] data, int[] indices){ /* ... */ }
```

## 2 Modèles de réseaux et leurs évaluations

### 2.1 Réseau de neurones avec 1 couche cachée

Dans un premier temps, on propose d'utiliser **5 unités** sur la couche cachée.



 **Questions**

1. Quelles sont les dimensions des matrices (entrées, paramètres et sorties) représentant votre réseau ? (on suppose pour l'instant que l'on utilise des lots (batches) de taille 1)
2. Dessinez votre réseau (vous pouvez utiliser cet outil en ligne : <http://alexlenail.me/NN-SVG/index.html>)



Vous fixerez la taille de vos lots à 4 et le taux d'apprentissage  $\eta$  à 0.01.


### Mise en œuvre

1. Pour pouvoir évaluer votre modèle final, vous allez devoir enregistrer quelques métriques : lors de la dernière époque d'entraînement, vous calculerez et afficherez :
  - (a) la précision de votre modèle (pourcentage de prédiction justes sur le jeu de test)
  - (b) le taux de faux positifs (ratio de prédictions positives lorsque les données de test correspondantes sont négatives)
  - (c) le taux de faux négatifs (ratio de prédictions négatives lorsque les données de test correspondantes sont positives)
  - (d) le taux de vrais positifs (ratio de prédictions positives lorsque les données de test correspondantes sont positives)
  - (e) le taux de vrais négatifs (ratio de prédictions négatives lorsque les données de test correspondantes sont négatives)
2. Entraînez votre modèle sur :
  - (a) 100 époques
  - (b) 200 époques
  - (c) 500 époqueset reportez l'évaluation de votre modèle pour chaque entraînement.

  Refaites ce travail (questions et mise en œuvre) pour **10 unités** dans la couche cachée du réseau.

## 2.2 Réseau de neurones avec 2 couches cachées

  Refaites le travail de la partie 2.1 (questions et mise en œuvre) pour un réseau à 2 couches cachées et 5 unités par couche cachée.

 Au final, quel modèle vous a permis d'obtenir les résultats les plus intéressants ? Discutez.