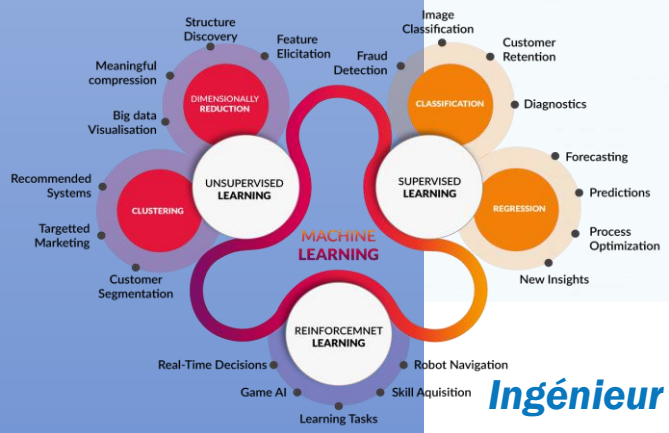
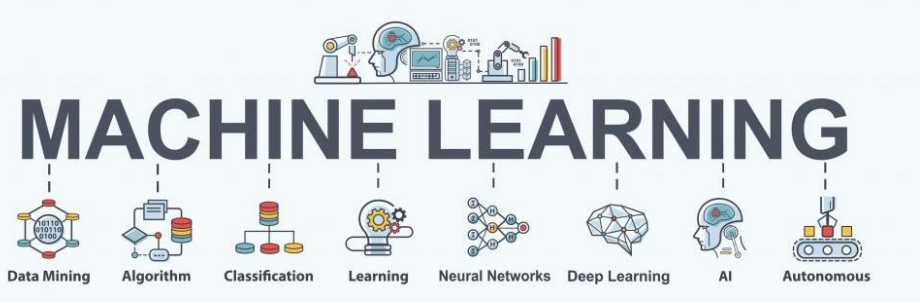




Corentin MADRE

Ingénieur Machine Learning / Développeur Python

Compétences Techniques

LANGAGES DE PROGRAMMATION

Python	●●●●●
C	●●●●●
C#	●●●●○
Java	●●●○○
Bash	●●●○○
C++	●●●○○
OCaml	●●●○○
SQL	●●○○○
VBA	●●○○○
JavaScript	●○○○○

Librairies ML	TensorFlow, Keras, Scikit-Learn, Numpy, Pandas
Librairies Graphiques	TKinter, PyGame, SDL, OpenGL
Outils / Middleware	Elasticsearch, MatLab, Git, Netica, NetLogo, ROS, Unity
IDE	Spyder, Jupiter, Eclipse, NetBeans, Visual Studio
Bureautique	Word, Excel, PowerPoint, Latex
CMS	Wordpress
Modélisation	UML
Réseaux	MobaXterm
SGBD	MySQL, MongoDB
Systèmes	Linux (Ubuntu, Mint, Raspberry Pi OS), Windows

DOMAINES CLÉS DE L'IA :

Apprentissage supervisé, Machine Learning / Deep Learning
Réseaux de neurones
Apprentissage par renforcement
Algorithmes Génétiques
Système Multi-Agents
Planification & Graphes
Optimisation & Estimation
Aide à la Décision
Programmation Orientée Objet
Automatisation & Robotique

Savoir-être

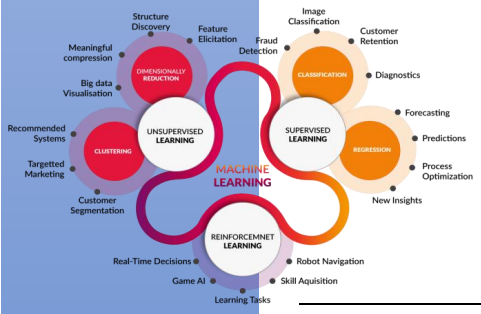
- ***Autonome***
- ***Capacité d'Adaptation***
- ***Créatif / Imaginatif***
- ***Curieux***
- ***Dynamique***
- ***Esprit d'équipe***
- ***Persévérant***
- ***Rigoureux***
- ***Sens de l'Observation / Analyse***

Formations

2017 - 2019	***Master (IARF) Intelligence Artificielle & Reconnaissance des Formes*** *– Université Paul Sabatier, Toulouse*
2014 - 2017	***Licence Informatique*** *– Université Paul Sabatier, Toulouse*
2014	***Baccalauréat Scientifique****, spécialité Informatique et Sciences du Numérique – Lycée Lapérouse, Albi*

Langues

Anglais	*B2 – Intermédiaire Supérieur*
Espagnol	*Notions*



Corentin MADRE

Ingénieur Machine Learning

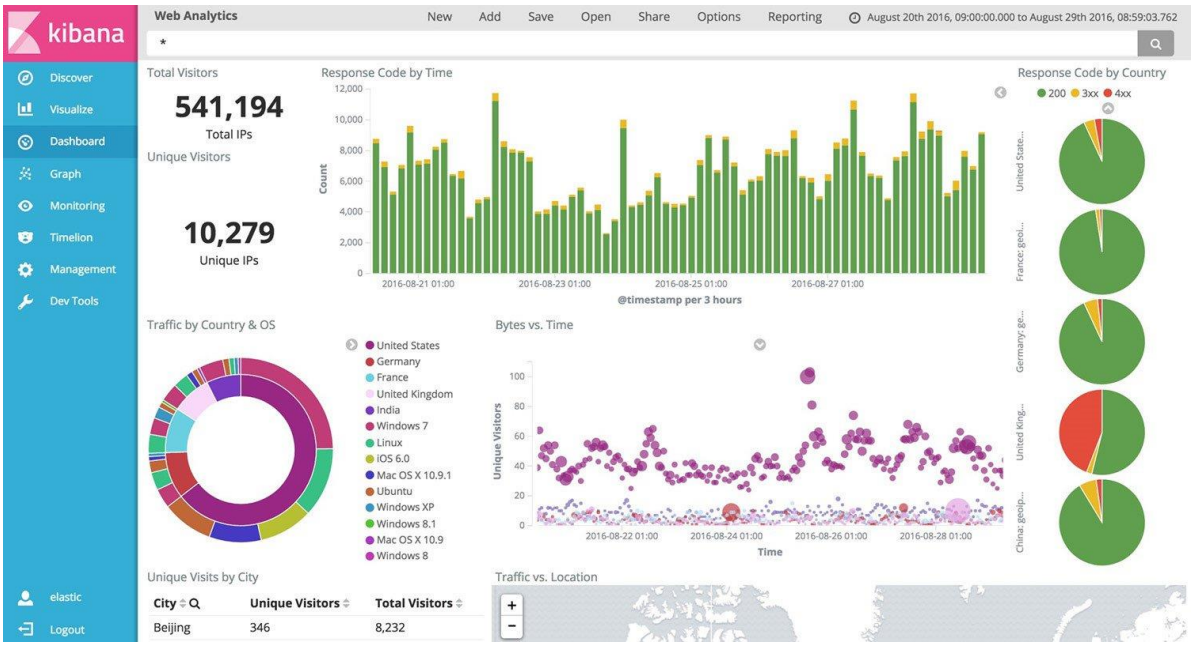
Développeur Python

Data Science

Du 05/2020 au 08/2020 ALTEN / SYNGENTA Développeur Python / Elasticsearch

Poste occupé : *Consultant **Elasticsearch** auto-formé en charge de la mise en place d'un moteur de recherche spécifique aux phénotypes des plantes et aux semences agricoles pour la société Syngenta.*

- Installation et configuration du serveur **Elasticsearch** : choix de détermination du nombre de nœuds, de shards et de réplicas, attribution de la taille d'allocation mémoire pour la JVM, définition des rôles et des caractéristiques pour chacun des nœuds, paramétrage du nombre de fichiers descripteurs, …
- Changement d'une version obsolète (2.4) à une version récente (7.6) d'Elasticsearch.
- Modification de l'architecture du code et des requêtes par conséquence.
- Indexation des données depuis une base de données **MongoDB** vers Elasticsearch, pilotée par des scripts Python.
- Augmentation de la performance du flux de données en modifiant les scripts **Python** d'indexation des données déjà existants.
- Adaptation des requêtes côté Back-end de recherche des documents pour la compatibilité avec la nouvelle version d'Elasticsearch installée.
- Installation et configuration de l'extension **Kibana** pour la visualisation des données indexées sur le serveur Elasticsearch.
- Documentation des scripts Python d'indexation des données.
- Mise en place d'un Wiki indiquant la configuration et l'architecture du serveur Elasticsearch ainsi que la structure des documents indexés (mapping).



Environnement technique : Elasticsearch, Python, Javascript / NodeJS, MongoDB, Git, Linux, MobaXterm

Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

Du 10/2018 au 10/2019 ARCYS Apprenti Machine Learning / IA

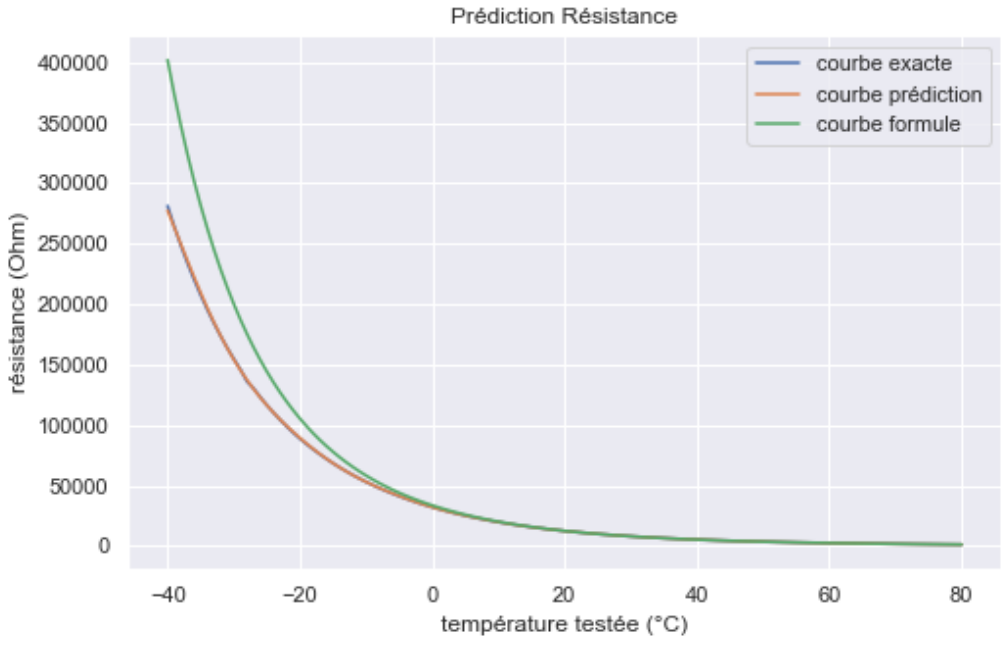
Poste occupé : *Alternant en Intelligence Artificielle appliquée au domaine de l'électronique sécuritaire pour les secteurs de la Défense Nationale, du Nucléaire Civil et du Transport Ferroviaire.*

Contexte général : *Conception, fabrication, qualification et maintenance des équipements électroniques sécuritaires en environnements sévères.*

Sujet des projets : *Implanter l'Intelligence Artificielle chez Arcys, explorer les opportunités et les possibilités d'applications dans le domaine de l'électronique sécuritaire, et démontrer l'efficacité du Machine Learning pour la conception de circuit électronique***.**

1. Prédiction de la résistance d'une thermistance en fonction de la température

Entraînement d'un réseau de neurones afin de prédire les valeurs de résistance d'une thermistance. Cette dernière est un composant électronique de type résistance dont la valeur change en fonction de la température. Une fonction de transfert utilisée par la société Arcys permet justement de calculer cette résistance en fonction de la température. L'objectif est de démontrer que les réseaux de neurones peuvent être plus efficaces que la formule actuellement utilisée. Après apprentissage des données, on obtient les résultats suivants :

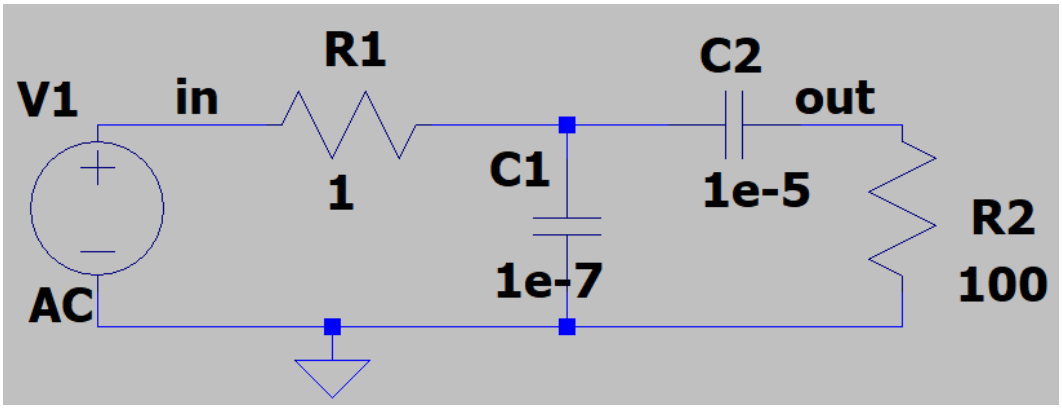


Le score de prédiction du réseau de neurones est de ***99.99%*** *tandis que celui de la formule théorique est de* ***83.38%****. Les prédictions faites sont alors plus précises que les résultats calculés par la formule théorique. Le réseau de neurones entrainé sera ensuite implémenté en FPGA pour remplacer l'utilisation de la formule.*

Corentin MADRE
Ingénieur Machine Learning
Développeur Python
Data Science

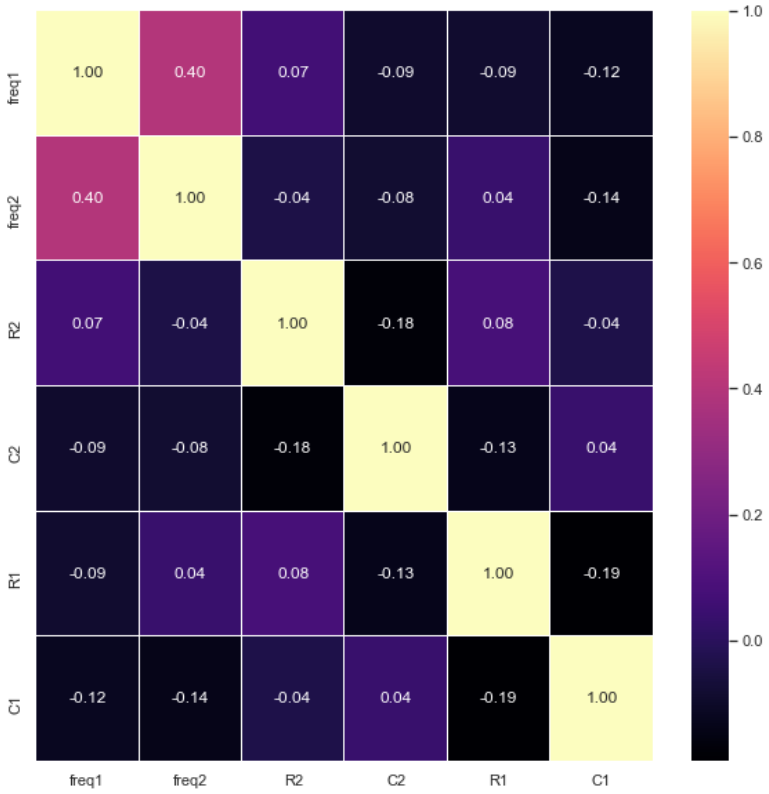
2. Prédiction des fréquences de coupure en fonction des valeurs de composants d'un circuit électronique

Entraînement d'un réseau de neurones afin de prédire les valeurs des fréquences de coupure en fonction des valeurs des résistances et condensateurs du circuit électronique. Le circuit utilisé est un coupe-bande de type RC CR :

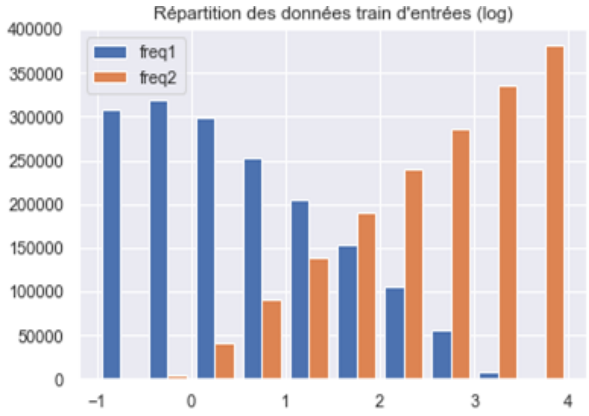


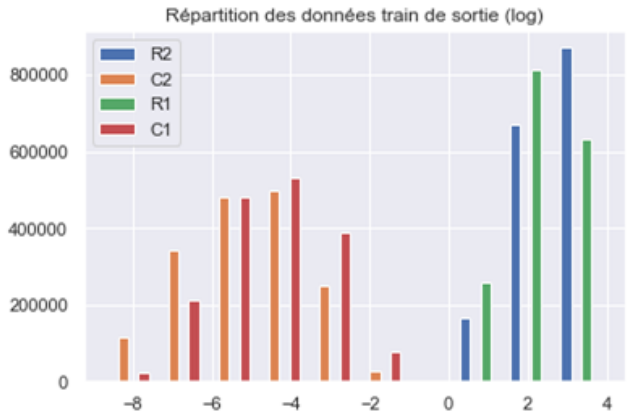
Les données sont générées avec des scripts Python permettant d'automatiser la simulation du circuit en pilotant le logiciel de simulation électronique LTSpice, permettant ainsi de calculer les fréquences de coupures correspondantes aux valeurs des composants du circuit. Les valeurs des composants sont ensuite modifiées de manière aléatoire afin d'avoir un large jeu de données différents.

Les données générées sont analysées avant la phase d'apprentissage afin de vérifier la pertinence des données fournies au réseau de neurones. Une matrice de corrélation permet d'identifier les dépendances entre les valeurs des composants et les fréquences de coupures calculées.

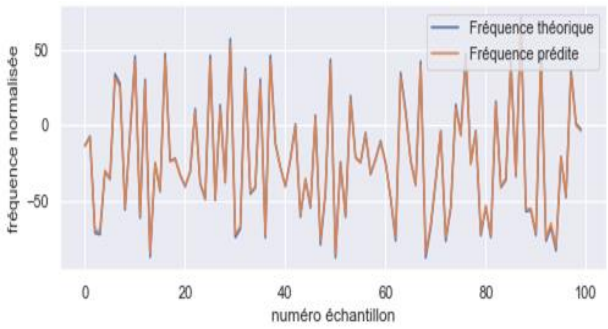


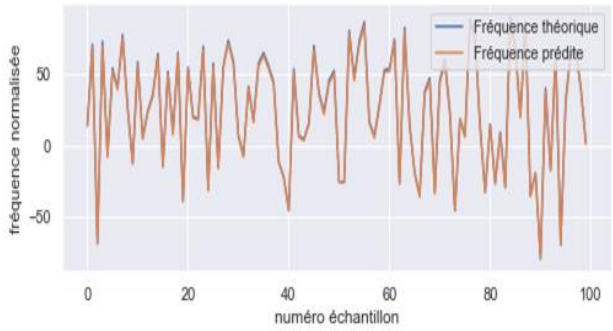
Il est aussi possible de vérifier l'ordre de grandeur et la répartition des données générées au moyen d'analyses statistiques ou en affichant les données sous forme d'histogrammes.





Après apprentissage des données, on obtient les résultats suivants :





*Le score de prédiction des basses fréquences (f1) est de **86,78%** et celui des hautes fréquences (f2) est de **90.86%**.*

3. Prédiction des valeurs de composants d'un circuit électronique en fonction des fréquences de coupures voulues à l'aide des algorithmes génétiques.

Implémentation d'un algorithme génétique permettant de déterminer une ou plusieurs solutions de circuit électronique dont les valeurs des composants déterminées permettent d'obtenir les fréquences de coupures demandées. Le circuit utilisé est le même que précédemment, c'est-à-dire un coupe-bande de type RC CR.

Un algorithme génétique s'inspire du principe de la sélection naturelle où une population d'individus évolue de génération en génération, et dont les gènes permettant la survie de l'individu se transmettent des individus parents aux individus enfants. Les individus sont ainsi soumis à une phase de sélection. Seuls les meilleurs pourront créer de nouveaux individus héritant d'un mélange des gènes parents (brassage génétique), soumis parfois à des mutations de manière aléatoire du génome afin d'avoir une population diversifiée.

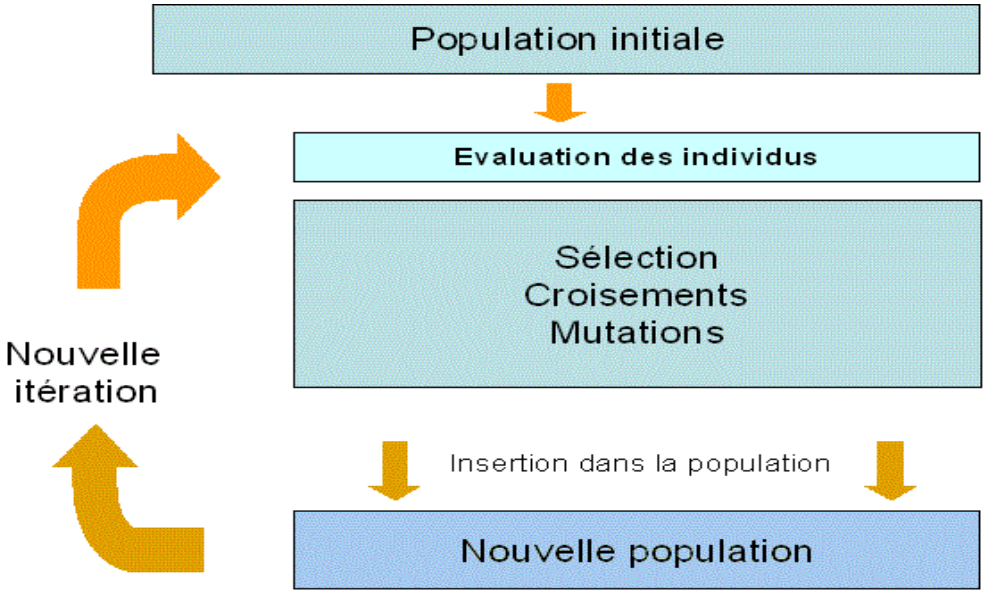
Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

Au fil des générations, les meilleurs individus sont ceux restants et sont considérés comme solution du problème.



Dans le cas du problème électronique, un individu constitue un circuit RC CR, et dont les gènes correspondent aux valeurs des composants. Son indice d'évaluation est la différence entre les fréquences demandées et les fréquences de coupures calculées au moyen de ses gènes. Les individus solutions sont donc ceux dont les gènes (les valeurs de résistances et de condensateurs) permettent d'arriver aux fréquences de coupures voulues (donc ceux ayant un indice d'évaluation proches de 0).

Après implémentation de l'algorithme génétique, on affiche ici les 5 meilleurs individus à 3 générations différentes dans le cas où on impose une fréquence basse (f1) de 100 Hz et une fréquence haute (f2) de 1300Hz :

Indice d'évaluation	f1 (Hz)	f2 (Hz)	R2 (Ω)	C2 (Farad)	R1 (Ω)	C1 (Farad)
Génération 0						
17,342	123,415	1288,730	17,400	3,000E-05	11,300	2,700E-05
59,399	1,802	1320,600	2,050	6,200E-05	69,800	1,200E-03
61,646	6,218	1329,510	806,000	1,500E-07	76,800	3,300E-04
66,6548815	0,250237	1266,44	3,83	3,300E-05	102	6,200E-03
67,1367	72,9366	1407,21	6810	2,000E-08	6040	3,000E-07
Génération 20						
1,31275	99,8245	1297,55	1430	9,100E-07	22,1	6,800E-06
2,63135	96,8373	1302,1	12,1	4,300E-05	14,3	2,700E-05
2,7425	100,005	1294,52	5490	2,700E-08	11	1,200E-04
3,26175	96,4065	1302,93	261	5,600E-07	5,11	2,700E-04
3,653	104,116	1296,81	14,3	4,300E-05	11,3	2,700E-05
Génération 150						
0,03	100	1300,06	5,36	8,200E-05	7,15	6,200E-05
0,2093	99,8714	1300,29	17,4	1,100E-05	23,7	4,300E-05
0,31675	99,9665	1299,4	15,8	2,200E-05	25,5	2,200E-05
0,4311	99,1978	1300,06	124	1,200E-06	11	1,200E-04
0,521	100,382	1299,34	107	1,100E-05	11	1,500E-05

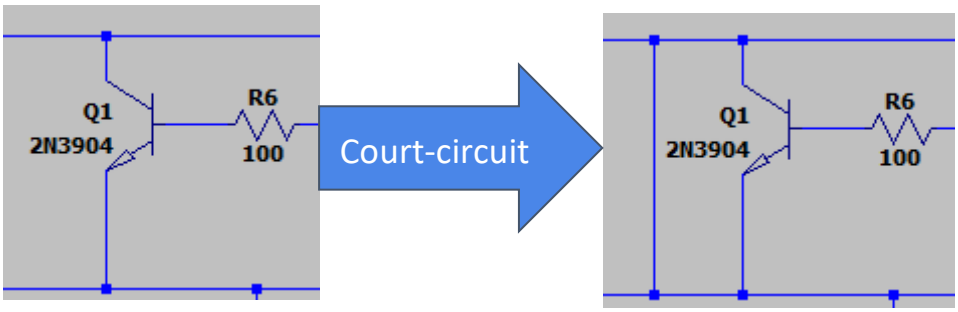
Corentin MADRE
Ingénieur Machine Learning
Développeur Python
Data Science

À la génération 150, la population contient des individus correspondants aux solutions demandées : les valeurs recherchées des composants (colonnes vertes) permettent d'obtenir des fréquences de coupures (colonnes bleues) très proches de celles attendues. Les individus obtenus sont ainsi considérés comme des solutions et l'avantage est que la diversité des individus permet de proposer plusieurs solutions différentes au problème donné.

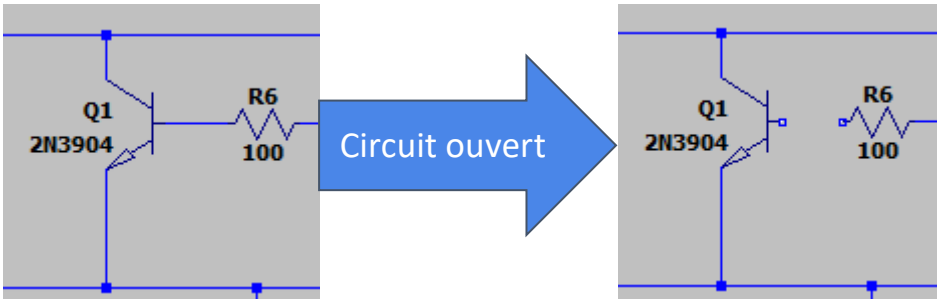
4. Détection d'anomalies à partir de l'intensité d'un composant

Implémentation d'un algorithme génétique permettant de déterminer les anomalies présentes dans un circuit électronique à partir de la mesure d'une intensité d'un composant. L'utilisation d'un algorithme génétique permet de détecter les défaillances d'un circuit à partir d'une intensité mesurée représentant la présence d'une anomalie non identifiée. Le but est alors, à partir d'un circuit sans défaillances, de reproduire ces anomalies pour arriver à l'intensité spécifiée. Ces anomalies ainsi reproduites correspondant à l'intensité spécifiée représentent donc les anomalies présentes dans le circuit et constituent ainsi la solution du problème. De la même manière que précédemment, un individu caractérise un circuit électronique. Les gènes de celui-ci sont non seulement les valeurs des composants du circuit, mais aussi la topologie du circuit, c'est-à-dire la connexion entre les composants. Les mutations des gènes représentent :

- *Une dérive d'un composant (modification de la valeur d'un composant)*
- *Un court-circuit d'un composant (connexion directe entre 2 broches d'un même composant)*



- *Un circuit ouvert d'un composant (suppression d'une broche d'un composant)*



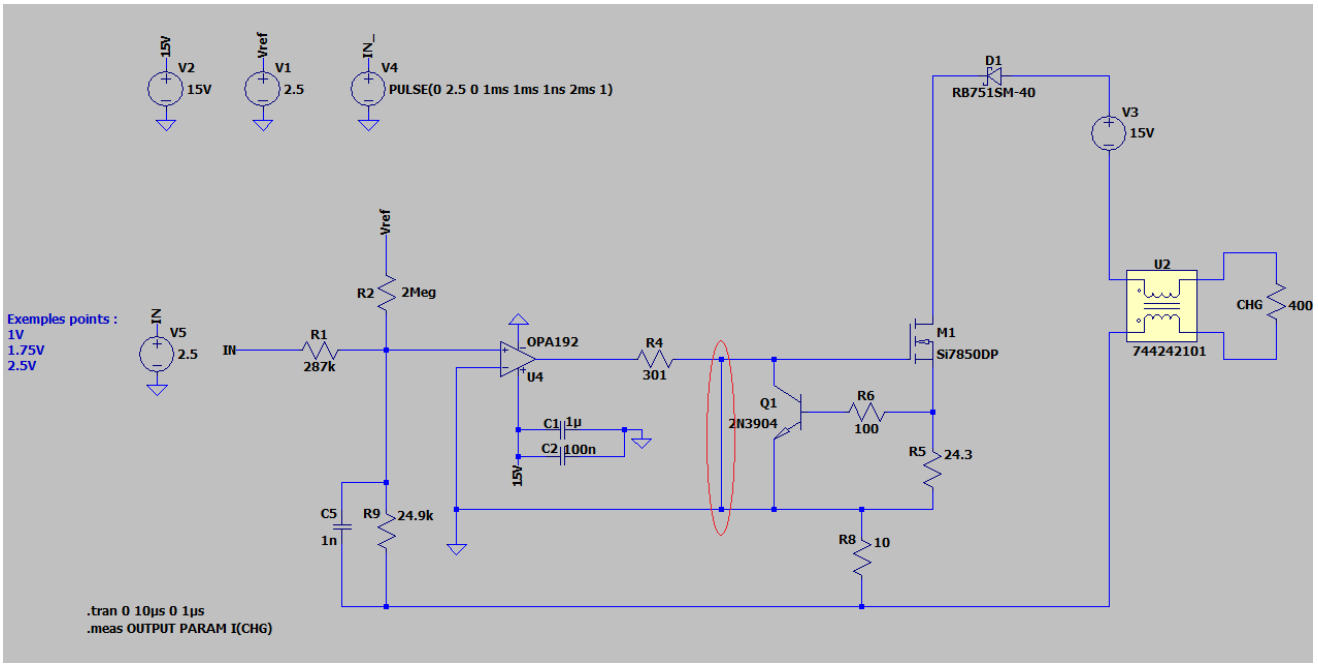
Corentin MADRE

Ingénieur Machine Learning

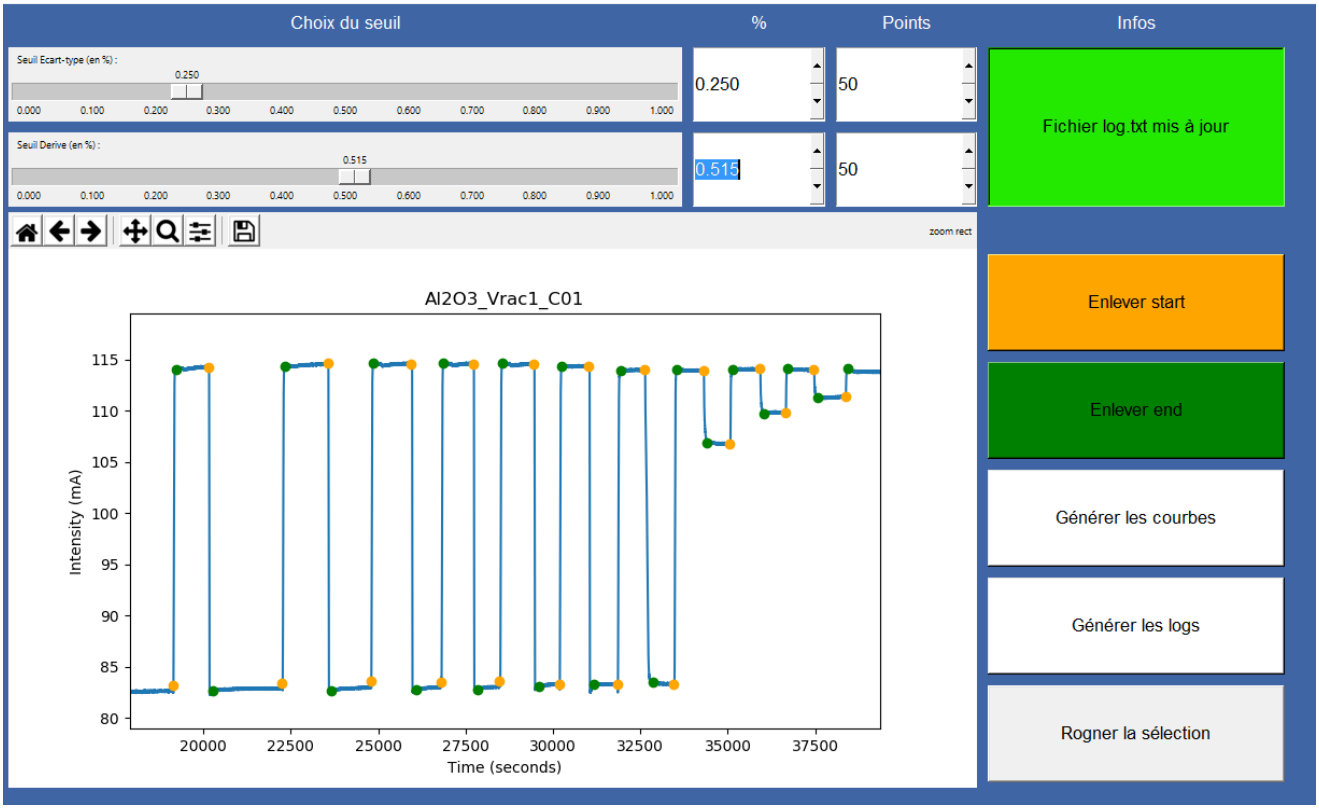
Développeur Python

Data Science

Après implémentation de l'algorithme génétique, une solution indique la présence d'un court-circuit au niveau du transistor Q1 pour une valeur d'intensité de -6.613e-11A de la résistance CHG du circuit suivant :



5. Développement d'un pack outil logiciel dans le cadre d'un projet de recherche et de développement de la conception de capteur d'Hydrogène à base de Palladium



Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

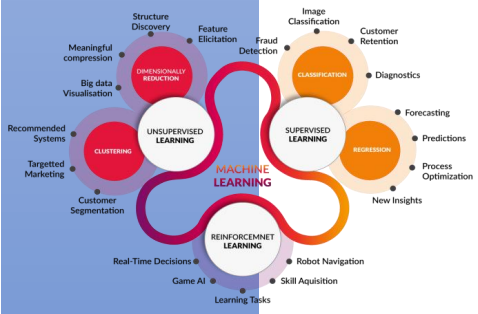
Développement d'une interface graphique avec la librairie TKinter de Python permettant d'aider les chercheurs de l'ICGM et du CNRS de Montpellier à visualiser et d'analyser les caractéristiques des capteurs prototypes conçus pour la détection du taux d'Hydrogène en centrale nucléaire.

L'objectif était de faciliter les recherches en laboratoire pour le choix du type de capteur le plus adéquat à utiliser concernant ce projet. Les intensités mesurées peuvent directement être lues par le logiciel développé afin de donner des informations telles que l'amplitude du signal et les temps de détection des phases de variations du taux d'Hydrogène.

```
**********************************************************************
*** focus ***
path = C:/Users/cmadre/Documents/ML_Autotune/trunk/H2 Palladium/Hydrogen/build/exe.win-amd64-3.5/focus.csv
Seuil Ecart-type utilisé : 0.323 avec 50 points
Seuil dérive utilisé : 0.2 avec 50 points
Valeur min : 82.245102
Valeur max : 114.782
Amplitude : 32.536897999999994
Phase 1 : 97.99999881349868 sec (from 19123.61071458412 to 19221.61071339762)
Phase 2 : 140.49999829893932 sec (from 20143.11070224084 to 20283.61070053978)
Phase 3 : 89.99999891035986 sec (from 22250.11067673098 to 22340.11067564134)
Phase 4 : 115.99999859556192 sec (from 23555.1106609311 to 23671.11065952666)
Phase 5 : 89.49999891640255 sec (from 24783.61064605741 to 24873.11064497381)
Phase 6 : 141.4999982868394 sec (from 25949.11063194647 to 26090.61063023331)
Phase 7 : 121.49999852896872 sec (from 26792.61062173406 to 26914.11062026303)
Phase 8 : 119.49999855319038 sec (from 27717.11061054096 to 27836.61060909415)
Phase 9 : 167.99999796598786 sec (from 28472.61060139397 to 28640.61059935996)
Phase 10 : 162.49999803257742 sec (from 29448.61058957735 to 29611.11058760993)
Phase 11 : 108.99999868030864 sec (from 30176.11058076937 to 30285.11057944968)
Phase 12 : 146.99999822023892 sec (from 31025.61057048431 to 31172.61056870455)
Phase 13 : 95.99999883771306 sec (from 31838.11056064721 to 31934.11055948492)
Phase 14 : 234.49999832897447 sec (from 32617.11055121571 to 32851.61054954468)
Phase 15 : 111.50000020768493 sec (from 33456.11055067065 to 33567.61055087834)
Phase 16 : 109.50000020395964 sec (from 34298.110552239 to 34407.61055244296)
Phase 17 : 92.00000017137063 sec (from 35056.11055365088 to 35148.11055382225)
Phase 18 : 124.50000023189932 sec (from 35916.11055525276 to 36040.61055548466)
Phase 19 : 70.5000001313092 sec (from 36644.6105566097 to 36715.11055674101)
Phase 20 : 99.50000018533319 sec (from 37458.61055812589 to 37558.11055831122)
Phase 21 : 64.50000012014061 sec (from 38362.11055980879 to 38426.61055992893)
```

L'application permet aussi de générer les images des courbes relatives aux caractéristiques du signal mesuré telles que :

- *L'intensité du signal en fonction du temps*
- *L'écart type du signal*
- *La dérive du signal*
- *Les temps de croissances du signal*
- *Les temps de décroissance du signal*

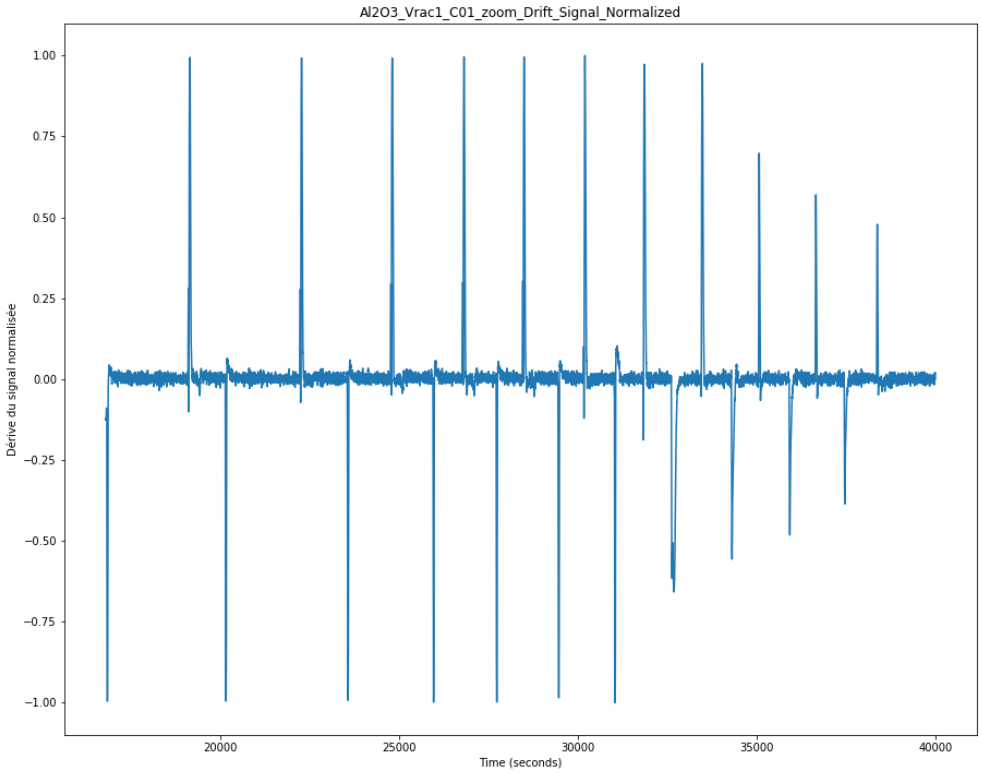


Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

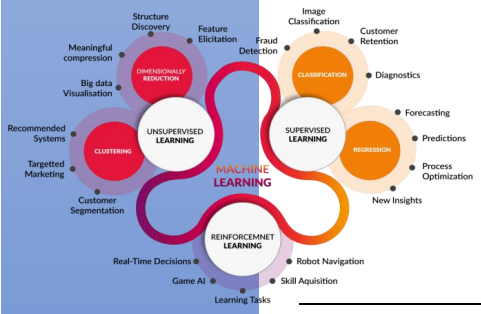


<u>Réalisations des projets</u> :

- Développement de scripts Python et Batch permettant de communiquer avec le simulateur LTSpice pour l'automatisation des simulations de circuits électroniques.
- Modélisation de réseaux de neurones pour des problèmes de régression et pouvant être implémentés en FPGA pour remplacer des fonctions de transfert inconnues.
- Implémentation d'algorithmes génétiques pour la recherche de plusieurs solutions différentes équivalentes pour un même problème, dans le cadre de la recherche de valeurs de composants ou de la détection d'anomalies dans un circuit.
- Développement d'une interface graphique en langage Python avec la librairie TKinter pour l'analyse de signaux et l'aide à la sélection de capteurs à base de Palladium.

<u>Environnement technique</u> :

Python, TensorFlow, Keras, Scikit-Learn, TKinter, Spyder, LTspice, TortoiseSVN, PowerPoint, Windows



Corentin MADRE

Ingénieur Machine Learning

Développeur Python

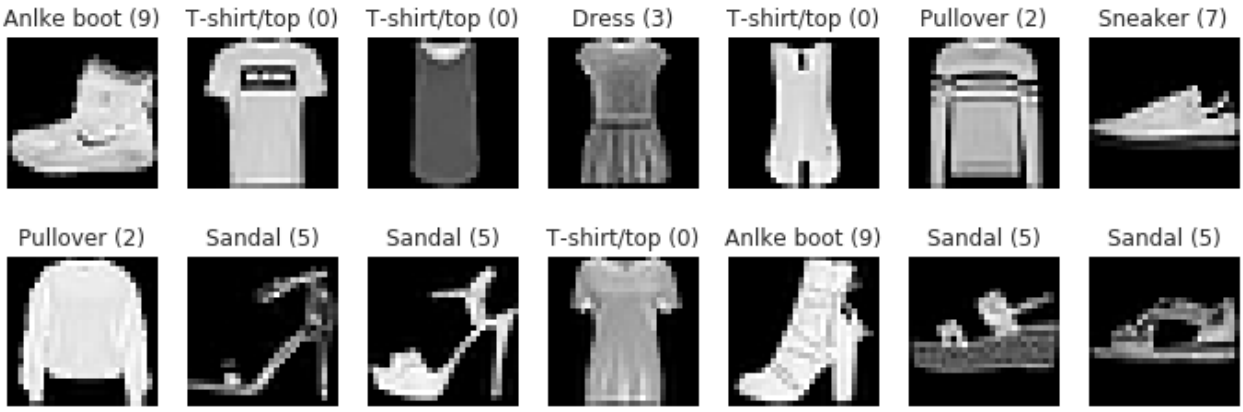
Data Science

Du 09/2018 au 11/2018 **Université Paul Sabatier** **M2 IARF**

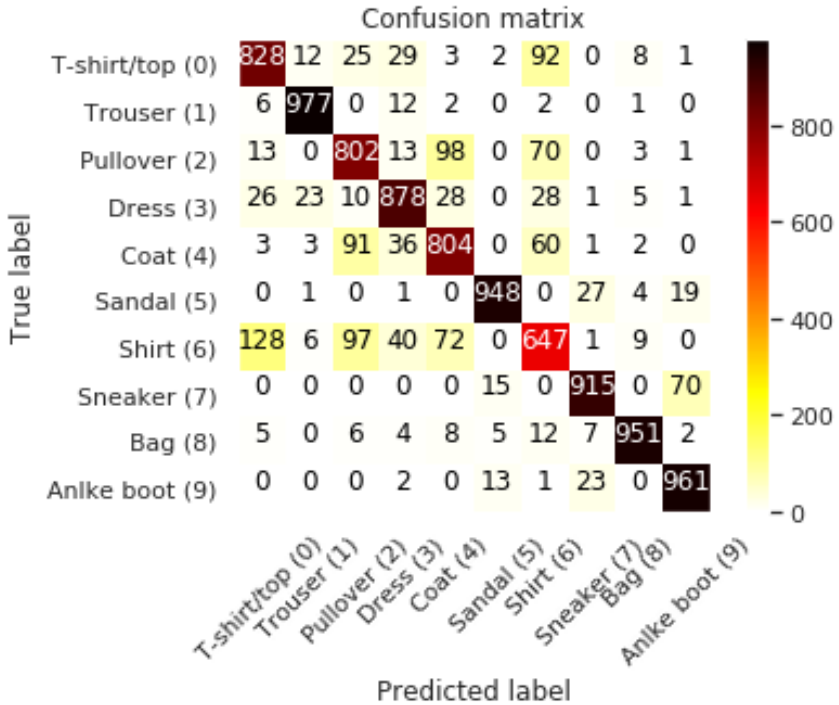
Poste occupé : *Étudiant en M2 IARF.*

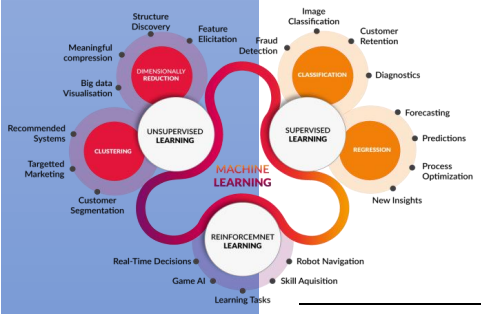
Contexte général : *Initiation aux réseaux de neurones dans le cadre de Travaux Pratiques : problème de classification des images.*

Sujet du projet : *Mettre en place un réseau de neurones afin de résoudre un problème de classification de données. Cette tâche consiste à classifier correctement des images (de mêmes tailles) de vêtements (pull, pantalon, jupe, chaussures, …) proposés par Zalando. L'objectif est que les articles vendus par Zalando puissent être détectés automatiquement grâce à un apprentissage des images fournies à la machine. Voici un exemple des données utilisées pour entrainer le réseau de neurone :*



Le réseau de neurones est ensuite entraîné et des prédictions sont effectuées sur des images de vêtements qu'il n'a jamais rencontré lors de son apprentissage, mais qu'il sera en mesure de classifier par sa capacité de généraliser les données qu'il identifie. Voici le résultat des prédictions sur différentes images de vêtements, présentés sous la forme d'une matrice de confusion permettant de comparer la classe prédite avec la classe attendue. Le score de prédiction affiche une précision de ***87.11%****.*





Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

Du 09/2018 au 11/2018 **Université Paul Sabatier** **M2 IARF**

Poste occupé : *Étudiant en M2 IARF.*

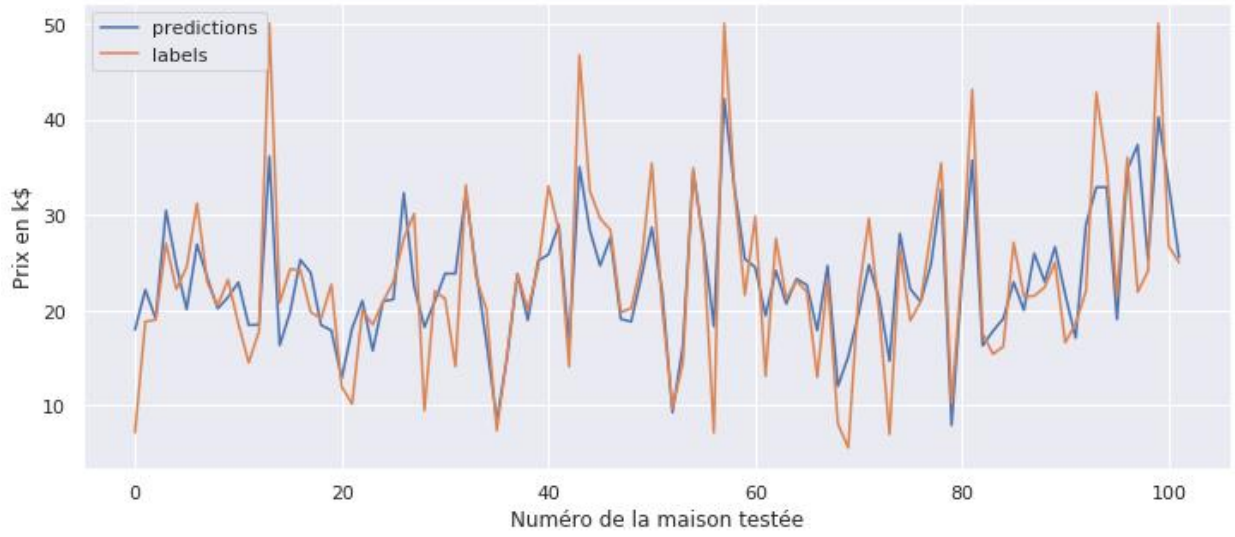
Contexte général : *Initiation aux réseaux de neurones dans le cadre de Travaux Pratiques : problème de régression.*

Sujet du projet :

Mettre en place un réseau de neurones dans le cadre d'un problème de régression des données, c'est-à-dire d'estimer une valeur en fonction de plusieurs paramètres d'entrées. Ici, l'objectif est de déterminer le prix des maisons situées à différents endroits dans la banlieue de Boston selon différents critères tels que :

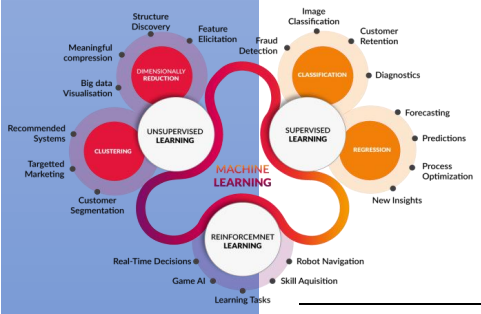
- *Le taux de criminalité ;*
- *La proportion de terrains résidentiels ;*
- *La concentration de monoxyde d'azote ;*
- *Le nombre moyen de pièces par logement ;*
- *Le taux d'imposition foncière ;*
- *L'indice d'accessibilité aux autoroutes radiales.*

Une partie des données est ensuite utilisée pour l'apprentissage du réseau de neurone, tandis que les données restantes vont permettre d'évaluer le modèle en comparant les prix prédits aux prix théoriques. Le score de prédiction affiche une précision de ***72.22%****.*



Réalisations :

- Modélisation d'un réseau de neurones pour un problème de régression.
- Apprentissage et évaluation du réseau de neurones.
- Analyse des données d'entrainement et des données prédites.



Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

Du 09/2018 au 11/2018 **Université Paul Sabatier** **M2 IARF**

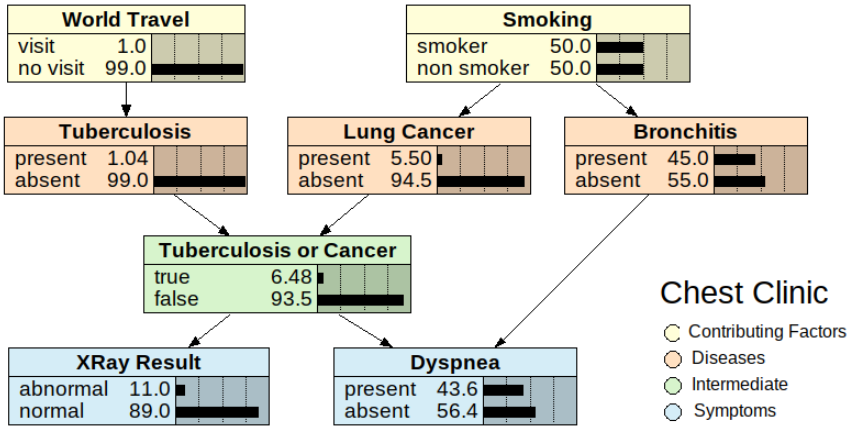
Poste occupé : *Étudiant en M2 IARF.*

Contexte général : Initiation aux arbres de probabilités pour l'Aide à la Décision dans le cadre de Travaux Pratiques.

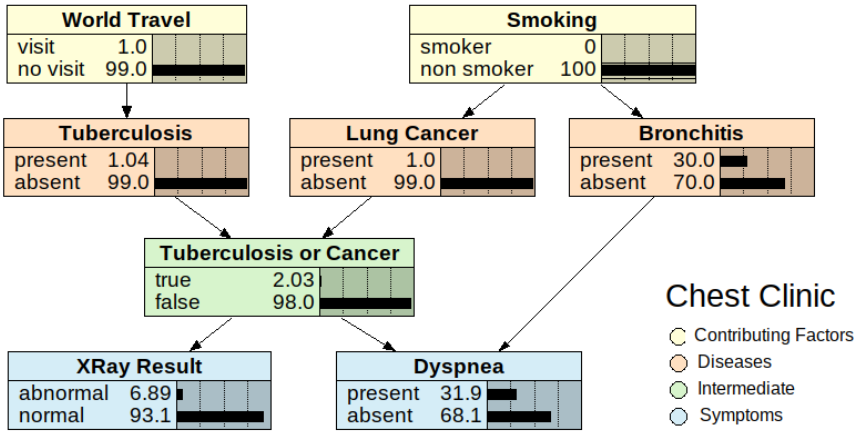
Sujet du projet :

Déterminer de manière probabiliste les facteurs d'un problème donné à partir de connaissances à priori fournies en modélisant des arbres de décisions, dont ces derniers sont le plus souvent utilisés dans le domaine de l'aide à la décision.

Par exemple, il est possible de modéliser un arbre de décision permettant de diagnostiquer la maladie d'un patient à partir des caractéristiques du malade (âge, fumeur, ...) et de ses symptômes.

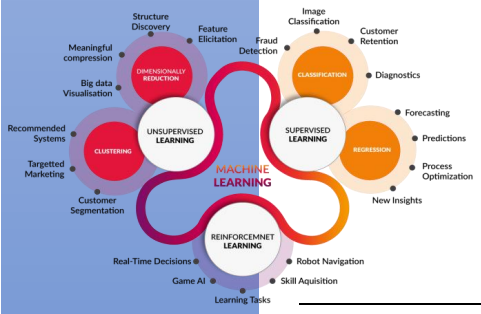


On apprend par la suite que le patient est non-fumeur, les probabilités des symptômes et des maladies sont ainsi actualisées.



Réalisations :

- Compréhension et modélisation d'un problème à partir d'un contexte donné.
- Modélisation d'un arbre de décision pour déterminer les causes probables d'un problème.



Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

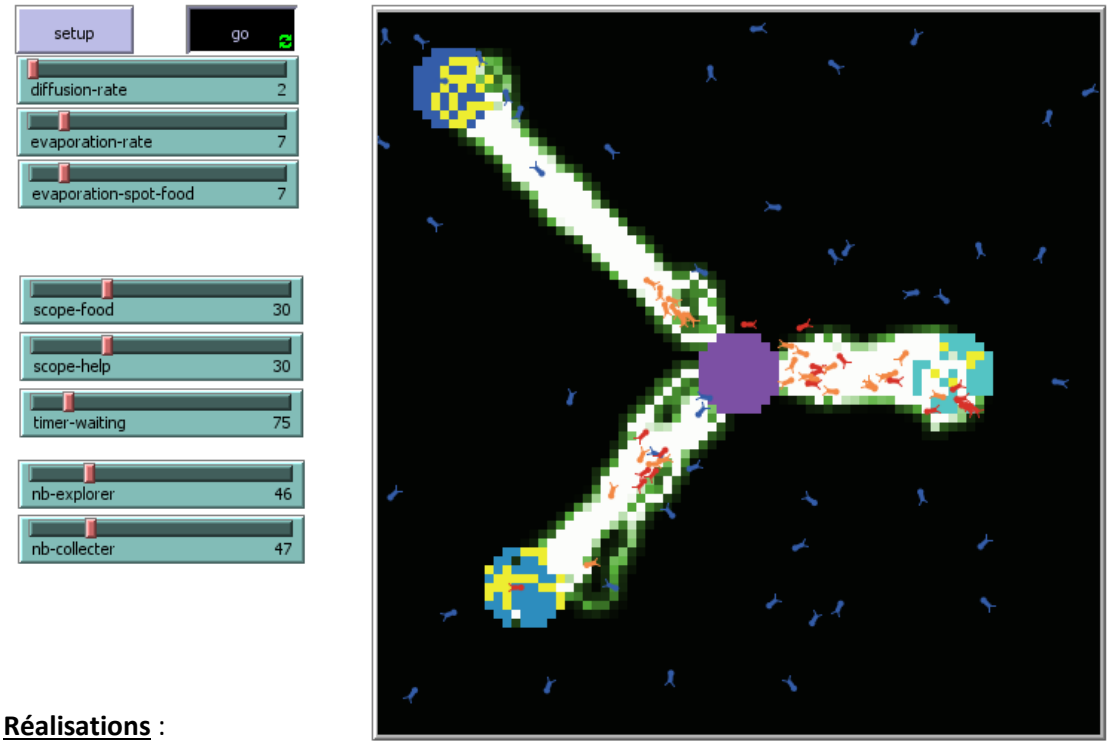
Du 02/2018 au 04/2018 **Université Paul Sabatier** **M2 IARF**

Poste occupé : *Étudiant en M2 IARF.*

Contexte général : *Initiation aux systèmes multi-agents collaboratifs dans le cadre de Travaux Pratiques.*

Sujet du projet :

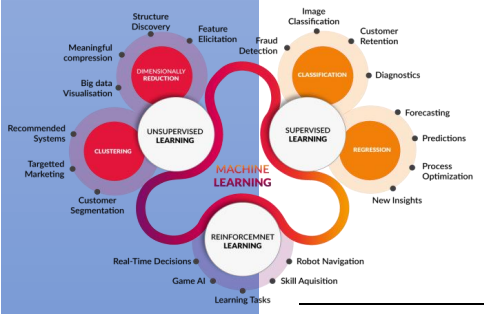
Mettre en place un système d'agents autonomes et collaboratifs en s'inspirant du fonctionnement de la société des fourmis. L'application repose sur des agents virtuels mais le principe peut être reproduit sur des robots collaboratifs. Ici, l'objectif est de pouvoir mettre en place une coopération entre deux types d'agents différents (les fourmis bleues et rouges) dont l'un a besoin de l'autre pour pouvoir effectuer les tâches confiées : le rôle des fourmis bleues est de marquer la nourriture présente sur le terrain, tandis que celui des fourmis rouges est de transporter cette nourriture marquée jusqu'à la fourmilière (le cercle violet au centre). Les fourmis rouges indiquent aux autres fourmis rouge la présence de nourriture en laissant derrière elles une trainée de phéromone (chemin vert/blanc).



Réalisations :

- Développement des deux types d'agents collaboratifs ayant chacun une fonction spécifique :
  - L'indication de la présence de nourriture ;
  - Le transport de la nourriture marquée.
- Mise en place de différents modes de communications entre les agents :
  - Communication par marquage au sol (phéromone) ;

Communication directe entre des agents voisins selon une certaine portée entre l'agent émetteur et récepteur d'informations.



Corentin MADRE

Ingénieur Machine Learning

Développeur Python

Data Science

Du 03/2017 au 05/2017 Université Paul Sabatier L3 Informatique

Poste occupé : *Étudiant en L3 Informatique.*

Contexte général : *Sujet de bureau d'étude de fin d'année dans le cadre d'une compétition organisée par l'**AFIA** (Association Française pour l'Intelligence Artificielle).*

Sujet du projet :

*Développement d'un bot validant le test de Turing sur le jeu « **Unreal Tournament 2004** » en simulant le comportement d'un joueur humain. **Unreal Tournament 2004** est un jeu vidéo de tir à la première personne dont les joueurs s'affrontent dans une arène à l'aide des armes présentes sur le terrain.*

Réalisations :

- Mise en place de la décision des actions que doit entreprendre un personnage contrôlé par l'ordinateur en imitant le comportement d'un joueur humain. Les actions possibles que peut avoir le bot sont :
  - Attaquer ;
  - Poursuivre ;
  - Esquiver ;
  - Fuir ;
  - Récupérer des soins ;
  - Récupérer des munitions.

- Gestion du déplacement du bot selon les situations rencontrées et à l'aide d'un système de RayCasting permettant de détecter et identifier les éventuels obstacles ou items présents dans son environnement.

- Implémentation de ces fonctions en programmation orientée objet (langage Java) et à l'aide de la bibliothèque **Pogamut**.