

# Conception d'une librairie de réseaux neuronaux

Samedi 23 Octobre 2021

**Hugo EYNARD**  
**Thomas BLUSSON**  
**Romain MOREAU**  
**Gabriel CHAVANON**

Responsable référent:  
Gabriel CHAVANON

Dépot GITEA :

<https://dwarves.iut-fbleau.fr/gitiut/blusson/PT21-22-Reseau-Neurones>

# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>Cahier des charges fonctionnelles</b>	<b>3</b>
Contexte	3
Etudes détaillées des objectifs (analyses des besoins)	4
Calendrier et priorisation des objectives	5
<b>Cahier des charges techniques et méthodologique</b>	<b>6</b>
Bibliographie	7

# Cahier des charges fonctionnelles

## Contexte

Client : Pierre VALARCHER (tuteur)

Description:

Nous comptons concevoir notre propre librairie de réseaux neuronaux et l'optimiser par la suite à travers différents tests (reconnaître des caractères manuscrites). Pour ceci nous ne comptons pas nous appuyer sur des solutions déjà existantes (comme [TensorFlow](#)) mais bien tout réaliser de A à Z. Pour ce qui est de l'optimisation (une meilleure vitesse d'exécution et une meilleure précision des résultats) on pourra utiliser des tests et des outils comparatifs (ancienne version de notre projet ou encore des librairies déjà existantes).

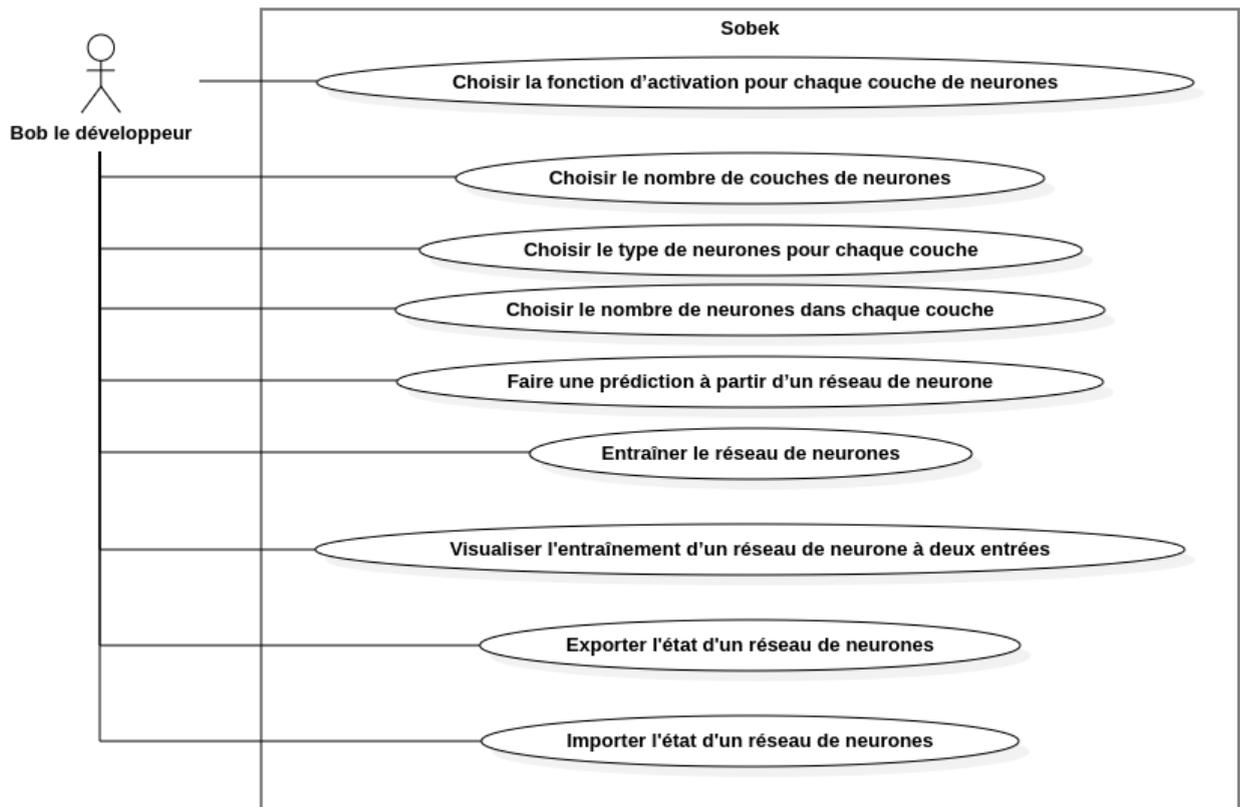
Contraintes :

On réalisera ce projet en Python orienté objet et grâce à ses librairies, tout notre travail sera disponible sur [notre dépôt GITEA](#).

Existant :

Nous avons à notre disposition un jeu de données contenant des images de chiffres annotés avec le chiffre correspondant. Il nous permettra d'entraîner notre réseau neuronal pour le tester.

## Etudes détaillées des objectifs (analyses des besoins)



### Fonctionnalités :

- Choix de la fonction d'activation pour chaque couche de neurones
- Choix du nombre de couches de neurones
- Choix du type de neurones pour chaque couche
- Choix du nombre de neurones dans chaque couche
- Faire une prédiction à partir d'un réseau de neurone
- Entraîner le réseau de neurones
- Exporter et importer l'état (modèle, biais et poids) d'un réseau de neurones
- Visualiser l'entraînement d'un réseau de neurone à deux entrées

Bob le développeur possède un jeu de données comportant des images de chats et de chiens annotés. Il commence par importer Sobek. Il choisit ensuite pour son modèle de réseau neuronal d'avoir une matrice 360 par 360 pour entrée. Il décide de mettre 2 premières couches de 180 neurones convolutifs, puis 2 couches de 64 neurones denses (ou classiques) et enfin une sortie de 2 neurones denses (Un pour chien et un pour chat). Il sépare son jeu de données en 2 parties: il utilise les 3 premiers quarts pour entraîner son réseau neuronal et après quelques minutes d'attente, il utilise le dernier quart pour estimer sa précision. Après quelques modifications et tests de l'architecture de son réseau pour obtenir une meilleure

précision, Bob est satisfait et peut maintenant utiliser son réseau pour faire des prédictions.

## Calendrier et priorisation des objectives

Jalon 0 : Cahier des charges (à signer) début novembre

Jalon 1 : Perceptron multicouche (première itération du projet), mi décembre M

1.1 : Partie prédiction du perceptron multicouche M

1.2 : Partie apprentissage du perceptron multicouche M

1.3 : Estimation de la précision du perceptron multicouche S

1.4 : Exportation et importation de l'état du réseau neuronal M

Jalon 2 : Utilisation concrète du réseau neuronal, fin décembre M

Jalon 3 : Visualisation 2D de l'entraînement du réseau neuronal M

Jalon 4 : Réseau de neurones convolutif (version optimisée du perceptron multicouche), fin janvier S

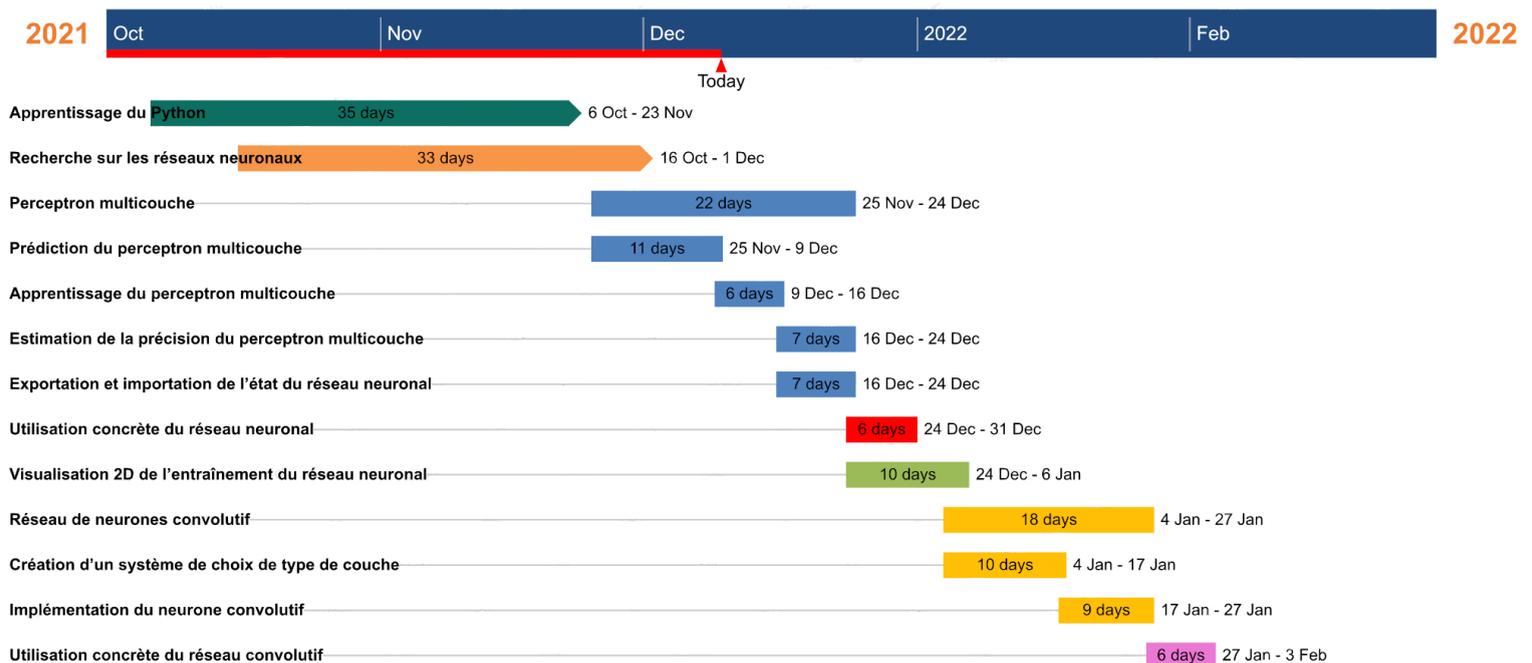
4.1 : Création d'un système de choix de type de couche M

4.2 : Implémentation du neurone convolutif M

Jalon 5 : Utilisation concrète du réseau convolutif, fin février C

## Diagramme de Gantt

### Librairie de réseau de neurones



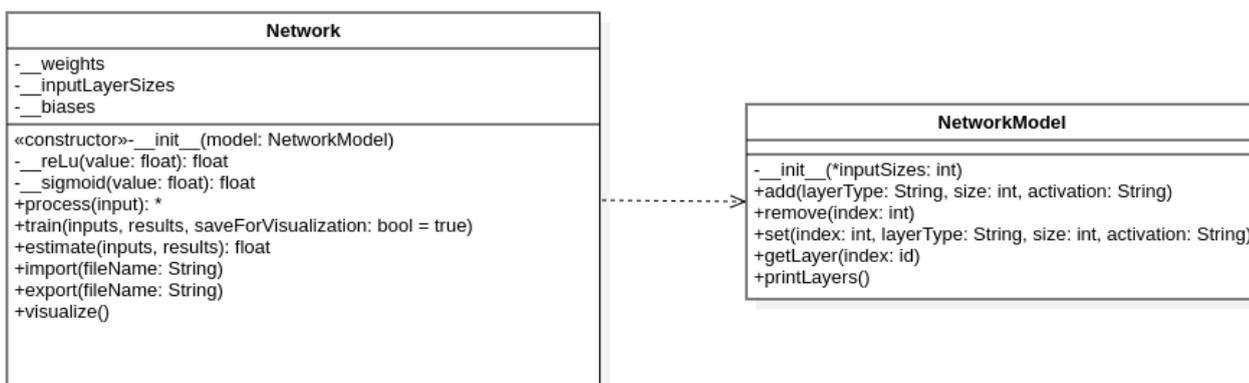
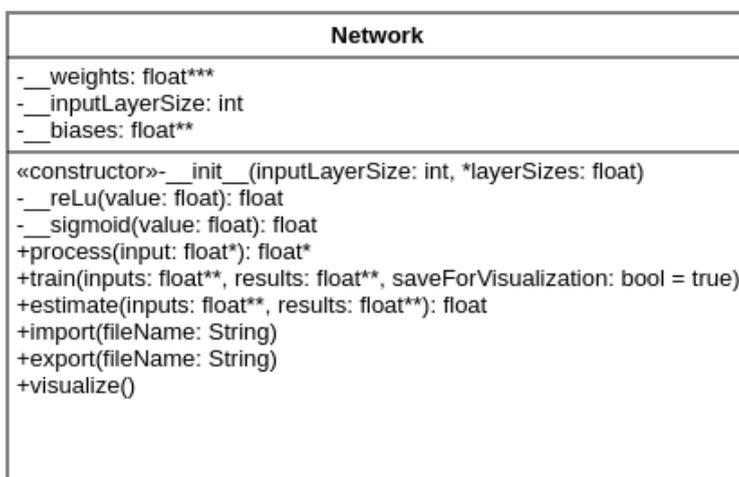
# Cahier des charges techniques et méthodologique

Nous avons choisi d'utiliser le Python car c'est un langage plutôt abordable qui est communément utilisé pour le machine learning. On compte également utiliser Scipy (NumPy (gestion des tableaux), Matplotlib (visualisation de données sous forme graphique)).

Méthodologie de travail :

On envisage d'utiliser la méthode agile (1 sprint correspond à 1 jalon) ainsi que le pair programming.

On compte utiliser très fréquemment GIT, et faire des tests unitaires dès que possible. On utilisera make pour l'exécution des tests et des mises en application.



## Bibliographie

[https://en.wikipedia.org/wiki/Types\\_of\\_artificial\\_neural\\_networks](https://en.wikipedia.org/wiki/Types_of_artificial_neural_networks)

<https://youtu.be/bVQUSndDIU>

<https://youtu.be/aircAruvnKk>

<https://www.mygreatlearning.com/blog/open-source-python-libraries/>

<https://fr.wikipedia.org/wiki/Matplotlib>

<https://brilliant.org/wiki/backpropagation/>

*Apprentissage machine Clé de l'intelligence artificielle* - Rémi Gilleron, 2019

*Deep Learning with Python, 2nd Edition* - François Chollet, 2021