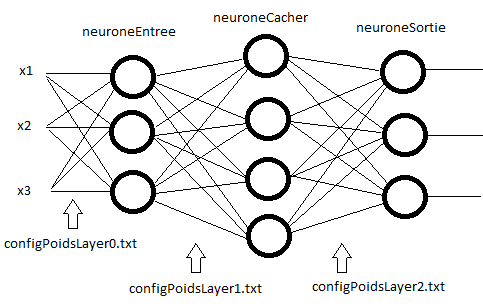
# Installation

-Suivez ce [tutoriel](https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html) pour installer la plus récente version de Python   
\*\*\*il est important de cocher la case « Add Python 3.7 to PATH »

-Ouvrer un terminal et exécuter la ligne de commande suivante :  
python -m pip install --user numpy

# Structure

Les dossiers les plus importants sont : **classe**, **config, SeqConfig et function**

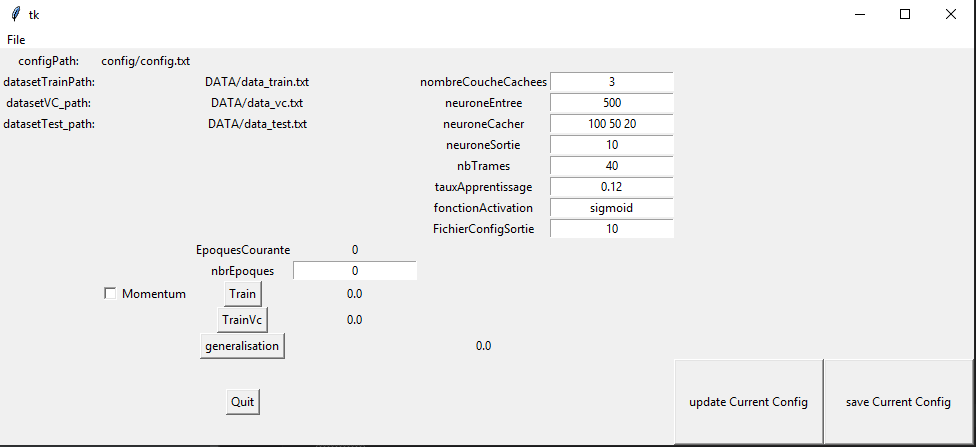
Dans le dossier de **config** on retrouve les fichiers de configurations qui sont chargé lors du lancement du programme. Le fichier **config.txt** contient les configs du réseau. Les lignes sont sensées s’expliquées elle-même mais pour mieux comprendre référer vous à la figure suivante.   
-La ligne « nombreCoucheCachees » est le nombre de couches entre la couche d’entrée et la couche de sortie.   
-La ligne « neuroneCacher » est le nombre de neurones pour chaque couche cachées. On doit simplement séparée les entrées par un espace.   
-La ligne « nbTrames» est le nombre trames en entrée (le nombre d’entrée sera égale à 26 x nbTrames). -La ligne « fonctionActivation » ne peut prendre que « sigmoid » ou « tanh ».  
-La ligne « FichierConfigSortie » doit toujours avoir le même nombre que la ligne « neuroneSortie ».   
Les fichiers **configPoidsLayer*x*.txt** sont les fichiers contenant le poids des liens de chaque couche.  
Lorsque vous enregistré une config, un nouveau sous dossier sera créé dans **SeqConfig** ayant le nom de la config créé. Le nom du dossier est expliqué par **configRM.txt**Si vous voulez utiliser un différent code de sortie que 10 ou 4, svp garder la nomenclature des fichiers de sortie comme configSortie10.txt

Dans le dossier **function** on retrouve les modules suivant :  
-**FuncActivation.py** comprend les fonctions d’activation possibles (ReLU ne fonctionne pas)  
-**fetch.py** comprend les méthodes pour aller lire les fichiers du dossier **DATA** et les méthodes pour charger les fichiers config.txt et configSortieX.txt  
-**configPoids.py** comprend les méthodes pour sauvegarder et charger les poids de liens de chaque couche

Dans le dossier de **classe** on retrouve le fichier **classe.py** qui est le cœur de tout le projet. On crée un réseau en lui passant une configuration. On utilisera ensuite la méthode train() pour faire l’apprentissage d’un set de donnée d’entrée et la méthode test() pour avoir la sortie obtenue d’un set de donnée d’entrée.

# Exécution

-Dans le même terminal, naviguer vers le dossier du projet ELE767 et exécuter la ligne de commande suivante :  
python main.py

Une fenêtre comme celle-ci-dessous devrait apparaitre  


Pour charger notre meilleure configuration trouvée, cliquer sur File->Open config et sélectionner le config.txt du seul dossier dans **SeqConfig.** Ensuite faites File->load poids neurones et sélectionner le sous dossier contenant le config.txt que vous venez de charger. Vous pouvez cliquer sur le bouton generalisation pour voir les pourcentages de réussites

Si vous voulez créer le réseau que vous voulez, vous pouvez jouer avec les boîtes d’entrée de droite. Référer vous aux règles décrites dans la section structure de ce LisezMoi. Ensuite cliquer sur le bouton « update Current Config » et ensuite sur « save Current Config » Vous remarquerez qu’un nouveau dossier a été créer pour votre config dans le dossier **SeqConfig**. Entrer dans la boîte « nbrEpoques » le nombre d’époques que vous voulez faire apprendre le réseau et cliquer ensuite sur « Train » Le pourcentage des tests sur les données d’apprentissage, de validation croisée et de test sont affiché juste en dessous.

SI JAMAIS VOUS AVEZ DES PROBLÈMES VOUS POUVEZ NOUS CONTACTER

[mathieu.fournier-desrochers.1@ens.etsmtl.ca](mailto:mathieu.fournier-desrochers.1@ens.etsmtl.ca)

bruno.de-lafontaine.1@ens.etsmtl.ca