BOUCHET Léo
LE BISSONNAIS Gautier

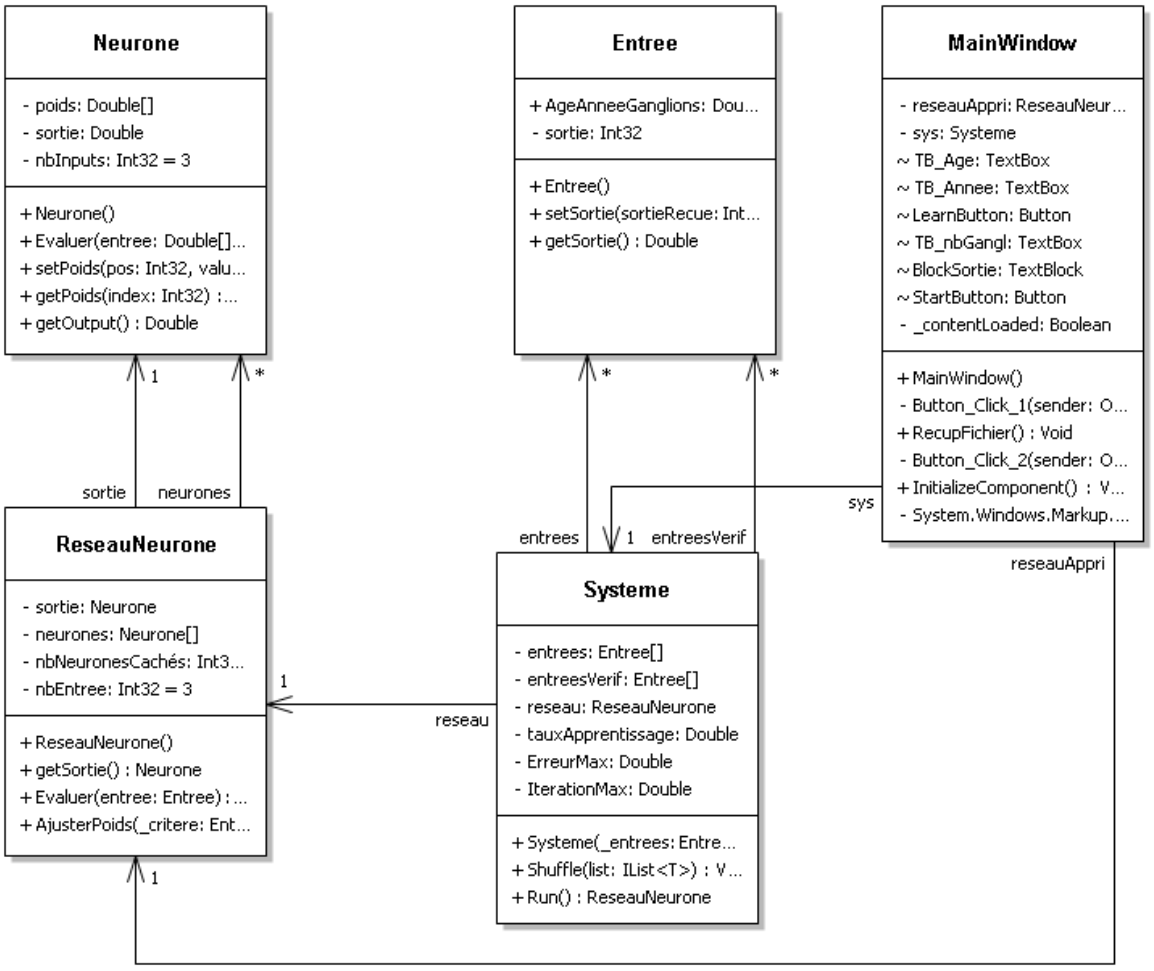
TP : Réseaux de neurones

I) Choix techniques

Notre réseau de neurones est composé de trois neurones cachés. La structure adoptée est la suivante :

- Système : Entité globale du programme, contient le réseau de neurones ainsi que les différentes informations de configurations importantes (taux d'apprentissage,..)
- RéseauNeurones : Contient les neurones cachés ainsi que le neurone de sorite, est responsable de l'évaluation des entrées ainsi que de l'ajustement des poids.
- Neurone : Entité de réflexion, la fonction évaluer est une sigmoïde.
- Entrée : Objet permettant de stocker les différents cas d'apprentissage et de tests contenus dans le fichier de texte fourni, une entrée est composée d'un tableau de 3 doubles (respectivement âge, année d'opération et nombre de ganglions infectés) ainsi que d'un int contenant la sortie attendue.
- MainWindow : Interface graphique et main du programme, effectue les différentes constructions, permet l'apprentissage ainsi que le test de nouveaux cas.

Ci-dessous le diagramme UML correspondant à notre programme.



II) Algorithme d'apprentissage

L'algorithme d'apprentissage choisi est la retro propagation du gradient. On analyse donc la différence entre le résultat d'évaluation de notre réseau de neurones et le résultat attendu, pour ce résultat on calcule ensuite quels neurones ont le plus influencé ce résultat et on ajuste les poids du neurone de sortie en fonction des résultats obtenus ainsi que les différents poids des neurones cachés.

III) Résultats obtenus

L'observation que l'on peut faire de notre réseau de neurones est que l'apprentissage n'est pas fonctionnel, bien que la méthode utilisée corresponde à celle fournie dans les sources, nous n'arrivons pas à obtenir un résultat concluant. Le reste du projet est cependant fonctionnel puisque l'interface permet le lancement de l'apprentissage ainsi que le test de nouvelles valeurs, malgré des résultats peu concluants compte tenu du non-apprentissage de notre réseau.