

TP : Réseaux de neurones

Table des matières

Introduction :.....	3
I) Choix techniques.....	3
II) Algorithme d'apprentissage.....	4
III) Résultats obtenus.....	4

Introduction :

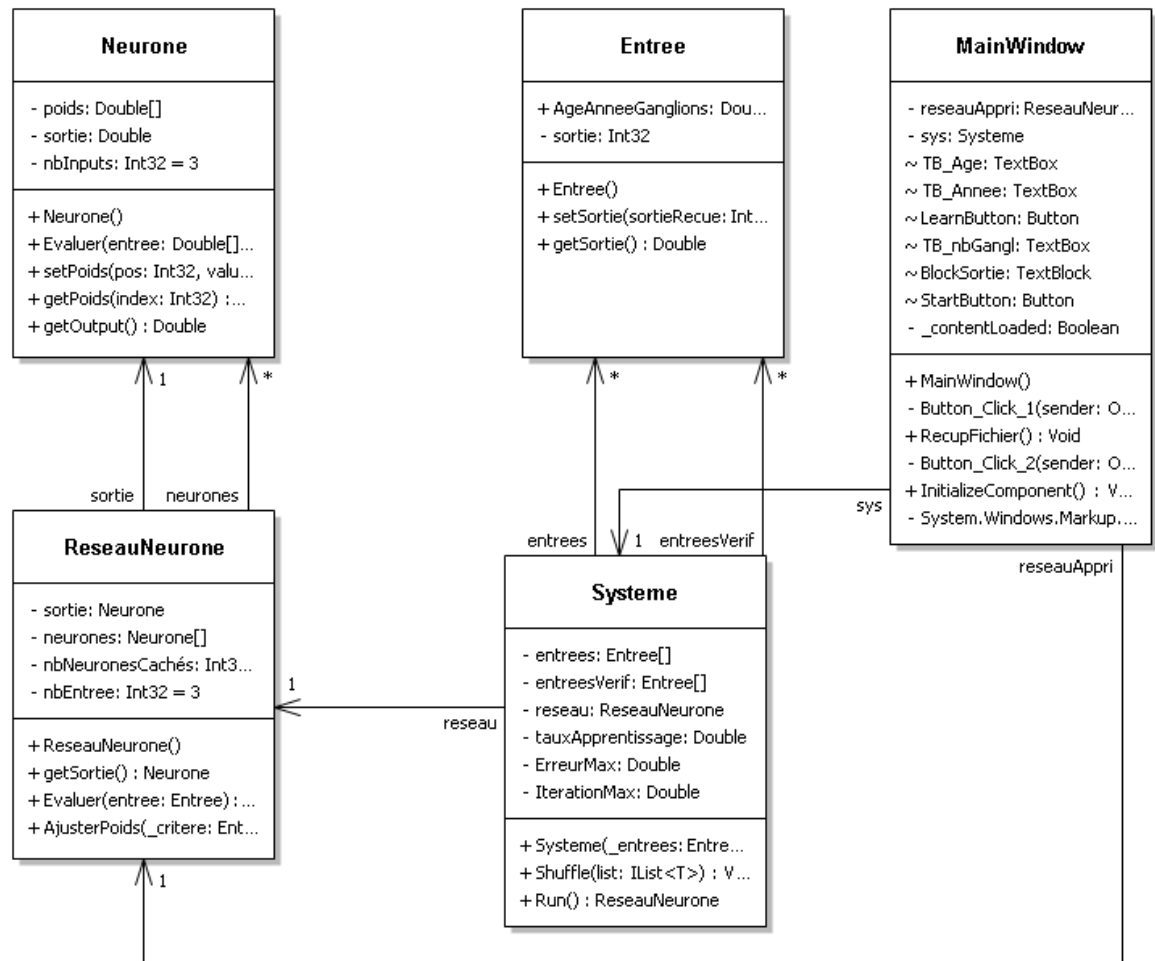
Ce TP a pour but de créer un réseau de neurones permettant de répondre à la question suivante : la personne atteinte d'un cancer du sein va elle survivre plus ou moins de 5 ans ? Pour ce faire, nous disposons de 3 critères : Age, année de l'opération, nombre de ganglions infectés.

I) Choix techniques

Notre réseau de neurones est composé de trois neurones cachés. Nous avons décidé de prendre trois neurones cachés parce qu'après un certain nombre de tests cela semblait le meilleur compromis entre performance et efficacité. Nous avons aussi choisi de fixer le taux d'apprentissage à 30% pour commencer, avant de progressivement être réduit. 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 sortie, 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 entier 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

Nous avons effectué une batterie de tests qui nous a permis de constater que quelque soit le nombre de neurones et paramètres d'entrée, le resultat du neurone de sortie vaut toujours 2. De même le taux d'erreur vaut toujours 146 et le taux d'erreur de généralisation vaut toujours 65.

Nombre de neurones cachés	3	3	6	6	2	2	10	10
Age du patient	25	67	25	67	25	67	25	67
Année de l'opération	93	65	93	65	93	65	93	65
Nombre de ganglions	2	7	2	7	2	7	2	7
Taux d'erreur d'apprentissage	146	146	146	146	146	146	146	146
Taux d'erreur de généralisation	65	65	65	65	65	65	65	65
Valeur du neurone de sortie	2	2	2	2	2	2	2	2

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.