

Manuel de référence

Généré par Doxygen 1.7.4

Sun May 15 2011 16 :14 :39

Table des matières

1	Index des classes	1
1.1	Hiérarchie des classes	1
2	Index des classes	3
2.1	Liste des classes	3
3	Documentation des classes	5
3.1	Référence de la classe FonctionTransfert	5
3.1.1	Description détaillée	5
3.2	Référence de la classe IHM	5
3.2.1	Description détaillée	6
3.2.2	Documentation des constructeurs et destructeur	6
3.2.2.1	IHM	6
3.2.2.2	~IHM	6
3.2.3	Documentation des fonctions membres	6
3.2.3.1	menu	6
3.3	Référence de la classe Image	6
3.3.1	Description détaillée	7
3.3.2	Documentation des constructeurs et destructeur	7
3.3.2.1	Image	7
3.3.2.2	~Image	7
3.3.3	Documentation des fonctions membres	7
3.3.3.1	centerImage	7
3.3.3.2	LigneColonne	7
3.3.3.3	obtenirPixel	7
3.3.3.4	obtenirPixels	8

3.4	Référence de la classe Neurone	8
3.4.1	Description détaillée	9
3.4.2	Documentation des constructeurs et destructeur	9
3.4.2.1	Neurone	9
3.4.3	Documentation des fonctions membres	9
3.4.3.1	calculeValeurTransfert	9
3.4.3.2	chargerNeurone	9
3.4.3.3	getIdentifiant	10
3.4.3.4	getPoidsSynaptic	10
3.4.3.5	getValeurSortie	10
3.4.3.6	sauvegarderNeurone	10
3.4.3.7	setPoidsSynaptic	10
3.4.3.8	setValeurEntree	11
3.5	Référence de la classe NeuroneChaine	11
3.5.1	Description détaillée	11
3.5.2	Documentation des fonctions membres	11
3.5.2.1	calculerErreur	11
3.5.2.2	calculerValeur	12
3.6	Référence de la classe ReseauNeurone	12
3.6.1	Description détaillée	12
3.6.2	Documentation des fonctions membres	12
3.6.2.1	apprendre	13
3.6.2.2	calculerResultat	13
3.6.2.3	chargerReseau	13
3.6.2.4	sauvegarderReseau	13
3.7	Référence de la classe ReseauNeuroneChainees	14
3.7.1	Description détaillée	14
3.8	Référence de la classe ReseauNeuronesChainees	14
3.9	Référence de la classe ReseauNeuronesCouches	14
3.9.1	Documentation des constructeurs et destructeur	15
3.9.1.1	ReseauNeuronesCouches	15
3.9.1.2	ReseauNeuronesCouches	15
3.9.1.3	ReseauNeuronesCouches	15
3.9.2	Documentation des fonctions membres	15

3.9.2.1	apprendre	16
3.9.2.2	calculerResultat	16
3.9.2.3	chargerReseau	16
3.9.2.4	sauvegarderReseau	16
3.10	Référence de la classe Sigmoide	17
3.10.1	Description détaillée	17

Chapitre 1

Index des classes

1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

FonctionTransfert	5
Sigmoides	17
IHM	5
Image	6
Neurone	8
NeuroneChainees	11
ReseauNeurone	12
ReseauNeuronesChainees	14
ReseauNeuronesCouches	14
ReseauNeuroneChainees	14

Chapitre 2

Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

FonctionTransfert (Class FonctionTransfert -)	5
IHM (Cette classe a pour but de charger l'interface en mode console)	5
Image (Cette classe a pour but de charger facilement des images et de réaliser des traitements basiques dessus)	6
Neurone	8
NeuroneChaine	11
ReseauNeurone	12
ReseauNeuroneChainees	14
ReseauNeuronesChainees	14
ReseauNeuronesCouches	14
Sigmoide (Class Sigmoïde -)	17

Chapitre 3

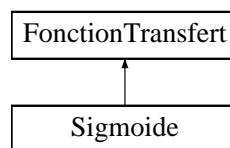
Documentation des classes

3.1 Référence de la classe FonctionTransfert

class `FonctionTransfert` -

```
#include <FonctionTransfert.h>
```

Graphe d'héritage de `FonctionTransfert` :



Fonctions membres publiques

- double **f** (double x)
- double **f_inverse** (double x)

3.1.1 Description détaillée

class `FonctionTransfert` -

La documentation de cette classe a été générée à partir du fichier suivant :

- /home/cocouf/Documents/gm4/gm11-neurones/src/FonctionTransfert.h

3.2 Référence de la classe IHM

Cette classe a pour but de charger l'interface en mode console.

```
#include <IHM.h>
```

Fonctions membres publiques

- `IHM()`
- `~IHM()`
- `void menu()`

3.2.1 Description détaillée

Cette classe a pour but de charger l'interface en mode console.

3.2.2 Documentation des constructeurs et destructeur

3.2.2.1 `IHM : :IHM ()`

Ceci est le constructeur par défaut de la classe `IHM`.

3.2.2.2 `IHM : :~IHM ()`

Ceci est le destructeur par défaut.

3.2.3 Documentation des fonctions membres

3.2.3.1 `void IHM : :menu ()`

Lance le menu principal de l'interface.

La documentation de cette classe a été générée à partir des fichiers suivants :

- `/home/cocouf/Documents/gm4/gm11-neurones/src/IHM.h`
- `/home/cocouf/Documents/gm4/gm11-neurones/src/IHM.cpp`

3.3 Référence de la classe Image

Cette classe a pour but de charger facilement des images et de réaliser des traitements basiques dessus.

```
#include <Image.h>
```

Fonctions membres publiques

- `Image` (string path)
- `~Image()`
- `void centerImage()`
- `int obtenirPixel` (int x, int y)
- `vector< double > obtenirPixels()`
- `vector< int > LigneColonne()`
- `void afficher()`

Cette méthode permet d'afficher l'image jusqu'à ce que l'utilisateur ferme la fenêtre.

– void `afficher` (int n)

Cette méthode permet d'afficher l'image n millisecondes.

Amis

– class `Image_Test`

3.3.1 Description détaillée

Cette classe a pour but de charger facilement des images et de réaliser des traitements basiques dessus.

3.3.2 Documentation des constructeurs et destructeur

3.3.2.1 `Image : :Image (string path)`

Ceci est le constructeur de la classe `Image` qui permet de charger une image.

Paramètres

<code>path</code>	le chemin (relatif) où se trouve l'image à charger
-------------------	--

3.3.2.2 `Image : :~Image ()`

Ceci est le destructeur de la classe `Image` qui permet de charger une image.

3.3.3 Documentation des fonctions membres

3.3.3.1 `void Image : :centerImage ()`

Cette méthode a pour but de recentrer l'image dans l'optique d'un prétraitement.

3.3.3.2 `std : :vector< int > Image : :LigneColonne ()`

Cette méthode retourne la somme des pixels sur chaque ligne et colonne.

Renvoie

renvois un tableau de données.

3.3.3.3 `int Image : :obtenirPixel (int x, int y)`

Retourne la valeur du pixel souhaité

Paramètres

<code>x</code>	ordonnée du pixel
<code>y</code>	abscisse du pixel

3.3.3.4 `vector< double > Image : :obtenirPixels ()`

Retourne le tableau de pixels de l'image

Renvoie

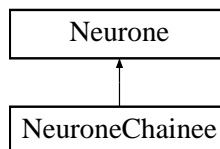
tableau avec tous les pixels.

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/cocouf/Documents/gm4/gm11-neurones/src/Image.h
- /home/cocouf/Documents/gm4/gm11-neurones/src/Image.cpp

3.4 Référence de la classe Neurone

Graphe d'héritage de Neurone :



Fonctions membres publiques

- **Neurone** (int identifiant, int nbAntecedents)
Ceci est le créateur d'un neurone.
- **Neurone** (const **Neurone** &n)
- virtual void **chargerNeurone** (ifstream *sauvegarde)
Cette méthode permet de lire un neurone dans un fichier.
- virtual void **sauvegarderNeurone** (ofstream *sauvegarde)
Cette méthode permet de sauvegarder un neurone dans un fichier.
- virtual void **calculeValeurTransfert** (vector< double > entreeSynaptique)
Calcule la valeur de transfert à partir des entrée synaptiques qui lui sont données.
- virtual double **getValeurSortie** ()
Cette méthode retourne la valeur de sortie du neurone après passage dans la fonction de transfert.
- virtual int **getIdentifiant** ()
Cette méthode retourne la valeur de l'identifiant du neurone.
- virtual void **setValeurEntree** (double v)
Permet d'assigner la valeur d'entrée du neurone (utile quand le neurone est de la première couche).
- virtual void **setPoidsSynaptic** (vector< double > poids)
Permet d'assigner la valeur des poids synaptiques du neurone (pour la phase d'apprentissage).
- virtual vector< double > **getPoidsSynaptic** ()
Permet de récupérer la valeur des poids synaptiques du neurone (pour la phase d'apprentissage).
- virtual double **getValeurAvantTrasphert** ()

Attributs protégés

- int **identifiant**

- double **valeurTransfert**
- double **valeurErreur**
- vector< double > **poidsSynaptiques**
- [FonctionTransfert](#) **fonctionTransfert**

Amis

- class **Test_Neurone**
- class **Test_ReseauNeuronesCouche**

3.4.1 Description détaillée

représente un neurone.

3.4.2 Documentation des constructeurs et destructeur

3.4.2.1 Neurone : :Neurone (int *identifiant*, int *nbAntecedents*)

Ceci est le créateur d'un neurone.

Paramètres

<i>identifiant</i>	le numéro d'identification du neurone.
<i>nbAntecedents</i>	le nombre d'antecedents qu'aura le neurone (c'est à dire le nombre de co-efficient synaptique qu'il aura)

3.4.3 Documentation des fonctions membres

3.4.3.1 void Neurone : :calculeValeurTransfert (vector< double > *entreeSynaptique*) [virtual]

Calcule la valeur de transfert à partir des entrée synaptiques qui lui sont données.

Paramètres

<i>entreeSynaptique</i>	donne les valeurs aux entrées du neurones.
-------------------------	--

3.4.3.2 void Neurone : :chargerNeurone (ifstream * *sauvegarde*) [virtual]

Cette méthode permet de lire un neurone dans un fichier.

Paramètres

<i>sauvegarde</i>	le fichier dans lequel il faut lire le neurone.
-------------------	---

3.4.3.3 `int Neurone : :getIdentifiant () [virtual]`

Cette méthode retourne la valeur de l'identifiant du neurone.

Renvoie

l'identifiant du neurone.

3.4.3.4 `std : :vector< double, std : :allocator< double > > Neurone : :getPoidsSynaptic () [virtual]`

Permet de récupérer la valeur des poids synaptiques du neurone (pour la phase d'apprentissage).

Renvoie

la valeur des poids.

3.4.3.5 `double Neurone : :getValeurSortie () [virtual]`

Cette méthode retourne la valeur de sortie du neurone après passage dans la fonction de transfert.

Renvoie

la valeur de sortie.

3.4.3.6 `void Neurone : :sauvegarderNeurone (ofstream * sauvegarde) [virtual]`

Cette méthode permet de sauvegarder un neurone dans un fichier.

Paramètres

<i>sauvegarde</i>	le fichier dans lequel il faut écrire le neurone.
-------------------	---

3.4.3.7 `void Neurone : :setPoidsSynaptic (vector< double > poids) [virtual]`

Permet d'assigner la valeur des poids synaptiques du neurone (pour la phase d'apprentissage).

Paramètres

<i>poids</i>	les valeurs souhaitées.
--------------	-------------------------

3.4.3.8 void Neurone::setValeurEntree (double v) [virtual]

Permet d'assigner la valeur d'entrée du neurone (utile quand le neurone est de la première couche).

Paramètres

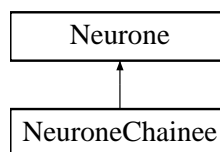
v	la valeur souhaitée.
---	----------------------

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/cocouf/Documents/gm4/gm11-neurones/src/Neurone.h
- /home/cocouf/Documents/gm4/gm11-neurones/src/Neurone.cpp

3.5 Référence de la classe NeuroneChaine

Graphique d'héritage de NeuroneChaine :



Fonctions membres publiques

- double [calculerValeur](#) ()
Cette méthode permet de calculer la valeur du neurone en faisant des appels récursifs. La méthode calcule le resultat et la met dans la variable.
- double [calculerErreur](#) ()
Cette méthode permet de calculer l'erreur du neurone en faisant des appels récursifs. La méthode calcule le resultat et la met dans la variable.

3.5.1 Description détaillée

hérite de la classe neurone et permet de de réaliser des réseaux de neurones en graphe

3.5.2 Documentation des fonctions membres

3.5.2.1 double NeuroneChaine::calculerErreur ()

Cette méthode permet de calculer l'erreur du neurone en faisant des appels récursifs. La méthode calcule le resultat et la met dans la variable.

Renvoie

donne l'erreur.

3.5.2.2 double NeuroneChaine : :calculerValeur ()

Cette méthode permet de calculer la valeur du neurone en faisant des appels récursifs. La méthode calcule le resultat et la met dans la variable.

Renvoi

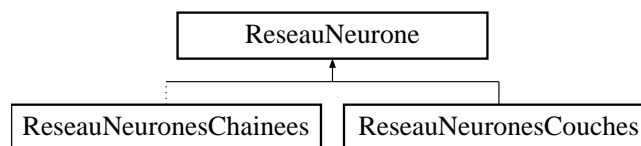
donne la valeur.

La documentation de cette classe a été générée à partir du fichier suivant :

– /home/cocouf/Documents/gm4/gm11-neurones/src/NeuroneChaine.h

3.6 Référence de la classe ReseauNeurone

Graphe d'héritage de ReseauNeurone :



Fonctions membres publiques

- virtual bool **chargerReseau** (string path)=0
Cette méthode permet de charger un réseau lu dans un fichier.
- virtual bool **sauvegarderReseau** (string path)=0
Cette méthode permet de sauvegarder un réseau dans un fichier.
- void **apprendre** (vector< double > *entrees, vector< double > *valeursAttendues, int n)
Cette méthode permet de réaliser une étape d'apprentissage.
- vector< double > **calculerResultat** (vector< double > valeurEntree)
Cette méthode permet de calculer le résultat sur la couche de sortie.

Attributs protégés

- vector< **Neurone** > **Neurones**

3.6.1 Description détaillée

représentant un réseau de neurones.

3.6.2 Documentation des fonctions membres

3.6.2.1 `void ReseauNeurone : :apprendre (vector< double > * entrees, vector< double > * valeursAttendues, int n)`

Cette méthode permet de réaliser une étape d'apprentissage.

Paramètres

<i>entrees</i>	liste des entrées pour le jeu d'apprentissage.
<i>valeursAttendues</i>	les valeurs attendues en sortie pour chacune de ses entrées.
<i>n</i>	la taille de l'échantillon d'apprentissage.

Réimplémentée dans [ReseauNeuronesCouches](#).

3.6.2.2 `vector<double> ReseauNeurone : :calculerResultat (vector< double > valeurEntree)`

Cette méthode permet de calculer le résultat sur la couche de sortie.

Renvoie

la valeur des résultats.

Réimplémentée dans [ReseauNeuronesCouches](#).

3.6.2.3 `virtual bool ReseauNeurone : :chargerReseau (string path) [pure virtual]`

Cette méthode permet de charger un réseau lu dans un fichier.

Paramètres

<i>path</i>	une chaîne de caractère décrivant le nom du fichier dans lequel est stocké le réseau.
-------------	---

Renvoie

Si oui ou non le chargement s'est bien déroulé.

Implémenté dans [ReseauNeuronesCouches](#).

3.6.2.4 `virtual bool ReseauNeurone : :sauvegarderReseau (string path) [pure virtual]`

Cette méthode permet de sauvegarder un réseau dans un fichier.

Paramètres

<i>path</i>	une chaîne de caractère décrivant le nom du fichier dans lequel on stocke le réseau.
-------------	--

Renvoie

Si oui ou non la sauvegarde s'est bien déroulée.

Implémenté dans [ReseauNeuronesCouches](#).

La documentation de cette classe a été générée à partir du fichier suivant :

– /home/cocouf/Documents/gm4/gm11-neurones/src/ReseauNeurone.h

3.7 Référence de la classe ReseauNeuroneChainees

3.7.1 Description détaillée

du réseau de neurone permettant de réaliser un réseau de neurone chaîné.

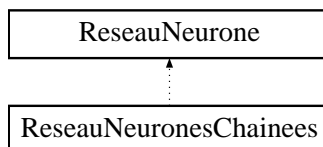
de réseau de neurone permettant de réaliser un réseau de neurone par couche.

La documentation de cette classe a été générée à partir du fichier suivant :

– /home/cocouf/Documents/gm4/gm11-neurones/src/ReseauNeuronesChainees.h

3.8 Référence de la classe ReseauNeuronesChainees

Graphe d'héritage de ReseauNeuronesChainees :

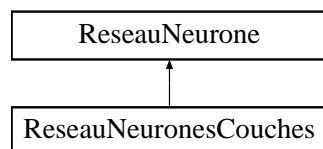


La documentation de cette classe a été générée à partir du fichier suivant :

– /home/cocouf/Documents/gm4/gm11-neurones/src/ReseauNeuronesChainees.h

3.9 Référence de la classe ReseauNeuronesCouches

Graphe d'héritage de ReseauNeuronesCouches :



Fonctions membres publiques

- [ReseauNeuronesCouches](#) (int nbNeuronesParCouche, int nbCouches)
- [ReseauNeuronesCouches](#) (vector< int > nbNeuronesParCouche)
- [ReseauNeuronesCouches](#) (const [ReseauNeuronesCouches](#) &u)

- bool `chargerReseau` (string path)
Cette méthode permet de charger un réseau lu dans un fichier.
- bool `sauvegarderReseau` (string path)
Cette méthode permet de sauvegarder un réseau dans un fichier.
- vector< double > `calculerResultat` (vector< double > valeurEntree)
Cette méthode permet de calculer le résultat sur la couche de sortie.
- void `apprendre` (vector< double > *entrees, vector< double > *valeursAttendues, int n)
Cette méthode permet de réaliser l'apprentissage complet à partir d'un jeu de données.

Amis

- class `Test_ReseauNeuronesCouche`

3.9.1 Documentation des constructeurs et destructeur

3.9.1.1 `ReseauNeuronesCouches : :ReseauNeuronesCouches (int nbNeuronesParCouche, int nbCouches)`

Ceci est le constructeur de la classe reseau neurone par couche.

Paramètres

<code>nbNeuronesParCouche</code>	le nombre de neurones par couche (dans ce cas le nombre de neurones par couche est fixé)
<code>nbCouches</code>	le nombre de couches

3.9.1.2 `ReseauNeuronesCouches : :ReseauNeuronesCouches (vector< int > nbNeuronesParCouche)`

Ceci est le constructeur de la classe reseau neurone par couche.

Paramètres

<code>nbNeuronesParCouche</code>	un tableau qui contient le nombre de neurone de chaque couche (ce nombre peut varier d'une couche à l'autre). Le nombre de couches retenues est la taille du tableau.
----------------------------------	---

3.9.1.3 `ReseauNeuronesCouches : :ReseauNeuronesCouches (const ReseauNeuronesCouches & u)`

Ceci est le constructeur par recopie de la classe reseau neurone par couche.

3.9.2 Documentation des fonctions membres

3.9.2.1 `void ReseauNeuronesCouches : :apprendre (vector< double > * entrees, vector< double > * valeursAttendues, int n)`

Cette méthode permet de réaliser l'apprentissage complet à partir d'un jeu de données.

Paramètres

<i>entrees</i>	liste des entrées pour le jeu d'apprentissage.
<i>valeursAttendues</i>	les valeurs attendues en sortie pour chacune de ses entrées.
<i>n</i>	la taille de l'échantillon d'apprentissage.

Réimplémentée à partir de [ReseauNeurone](#).

3.9.2.2 `vector< double > ReseauNeuronesCouches : :calculerResultat (vector< double > valeurEntree)`

Cette méthode permet de calculer le résultat sur la couche de sortie.

Renvoie

la valeur des résultats.

Réimplémentée à partir de [ReseauNeurone](#).

3.9.2.3 `bool ReseauNeuronesCouches : :chargerReseau (string path) [virtual]`

Cette méthode permet de charger un réseau lu dans un fichier.

Paramètres

<i>path</i>	une chaîne de caractère décrivant le nom du fichier dans lequel est stocké le réseau.
-------------	---

Renvoie

Si oui ou non le chargement s'est bien déroulé.

Implémente [ReseauNeurone](#).

3.9.2.4 `bool ReseauNeuronesCouches : :sauvegarderReseau (string path) [virtual]`

Cette méthode permet de sauvegarder un réseau dans un fichier.

Paramètres

<i>path</i>	une chaîne de caractère décrivant le nom du fichier dans lequel on stocke le réseau.
-------------	--

Renvoie

Si oui ou non la sauvegarde s'est bien déroulée.

Implémente [ReseauNeurone](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

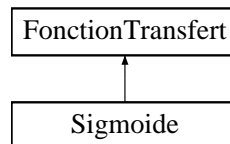
- /home/cocouf/Documents/gm4/gm11-neurones/src/ReseauNeuronesCouches.h
- /home/cocouf/Documents/gm4/gm11-neurones/src/ReseauNeuronesCouches.cpp

3.10 Référence de la classe Sigmoid

class Sigmoid -

```
#include <Sigmoid.h>
```

Graphe d'héritage de Sigmoid :



Fonctions membres publiques statiques

- static double **f** (double x)
- static double **f_inverse** (double x)
- static double **f_derive** (double x)

3.10.1 Description détaillée

class Sigmoid -

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/cocouf/Documents/gm4/gm11-neurones/src/Sigmoid.h
- /home/cocouf/Documents/gm4/gm11-neurones/src/Sigmoid.cpp

Index

- ~IHM
 - IHM, 6
- ~Image
 - Image, 7
- apprendre
 - ReseauNeurone, 12
 - ReseauNeuronesCouches, 15
- calculerErreur
 - NeuroneChaine, 11
- calculerResultat
 - ReseauNeurone, 13
 - ReseauNeuronesCouches, 16
- calculerValeur
 - NeuroneChaine, 11
- calculeValeurTransfert
 - Neurone, 9
- centerImage
 - Image, 7
- chargerNeurone
 - Neurone, 9
- chargerReseau
 - ReseauNeurone, 13
 - ReseauNeuronesCouches, 16
- FonctionTransfert, 5
- getIdentifiant
 - Neurone, 9
- getPoidsSynaptic
 - Neurone, 10
- getValeurSortie
 - Neurone, 10
- IHM, 5
 - ~IHM, 6
 - IHM, 6
 - menu, 6
- Image, 6
 - ~Image, 7
 - centerImage, 7
 - Image, 7
 - LigneColonne, 7
 - obtenirPixel, 7
 - obtenirPixels, 8
- LigneColonne
 - Image, 7
- menu
 - IHM, 6
- Neurone, 8
 - calculeValeurTransfert, 9
 - chargerNeurone, 9
 - getIdentifiant, 9
 - getPoidsSynaptic, 10
 - getValeurSortie, 10
 - Neurone, 9
 - sauvegarderNeurone, 10
 - setPoidsSynaptic, 10
 - setValeurEntree, 10
- NeuroneChaine, 11
 - calculerErreur, 11
 - calculerValeur, 11
- obtenirPixel
 - Image, 7
- obtenirPixels
 - Image, 8
- ReseauNeurone, 12
 - apprendre, 12
 - calculerResultat, 13
 - chargerReseau, 13
 - sauvegarderReseau, 13
- ReseauNeuroneChainees, 14
- ReseauNeuronesChainees, 14
- ReseauNeuronesCouches, 14
 - apprendre, 15
 - calculerResultat, 16
 - chargerReseau, 16
 - ReseauNeuronesCouches, 15

- sauvegarderReseau, [16](#)
- sauvegarderNeurone
 - Neurone, [10](#)
- sauvegarderReseau
 - ReseauNeurone, [13](#)
 - ReseauNeuronesCouches, [16](#)
- setPoidsSynaptic
 - Neurone, [10](#)
- setValeurEntree
 - Neurone, [10](#)
- Sigmoide, [17](#)