

## Gestion d'un réseau de neurones

Généré par Doxygen 1.8.13



# Table des matières

<b>1</b>	<b>Index hiérarchique</b>	<b>1</b>
1.1	Hiérarchie des classes . . . . .	1
<b>2</b>	<b>Index des classes</b>	<b>3</b>
2.1	Liste des classes . . . . .	3
<b>3</b>	<b>Documentation des classes</b>	<b>5</b>
3.1	Référence de la classe Apprentissage . . . . .	5
3.1.1	Description détaillée . . . . .	5
3.1.2	Documentation des fonctions membres . . . . .	6
3.1.2.1	getDonnees() . . . . .	6
3.1.2.2	getErreur() . . . . .	6
3.1.2.3	getOptimisateur() . . . . .	6
3.1.2.4	getParam() . . . . .	7
3.1.2.5	getRN() . . . . .	7
3.1.2.6	setDonnees() . . . . .	7
3.1.2.7	setParam() . . . . .	7
3.2	Référence de la classe Boite . . . . .	8
3.2.1	Description détaillée . . . . .	8
3.3	Référence de la classe BoiteArchitecture . . . . .	9
3.3.1	Description détaillée . . . . .	9
3.4	Référence de la classe BoiteChoixCouche . . . . .	9
3.4.1	Description détaillée . . . . .	10
3.5	Référence de la classe BoiteChoixDonnees . . . . .	10

3.5.1	Description détaillée	11
3.6	Référence de la classe BoiteChoixDonnées	11
3.6.1	Description détaillée	12
3.7	Référence de la classe BoiteChoixErreur	12
3.7.1	Description détaillée	13
3.8	Référence de la classe BoiteChoixFichier	14
3.8.1	Description détaillée	15
3.8.2	Documentation des constructeurs et destructeur	15
3.8.2.1	BoiteChoixFichier()	15
3.8.3	Documentation des fonctions membres	16
3.8.3.1	getNomFichier()	16
3.9	Référence de la classe BoiteChoixMultiple	16
3.9.1	Description détaillée	17
3.9.2	Documentation des constructeurs et destructeur	17
3.9.2.1	BoiteChoixMultiple()	17
3.9.3	Documentation des fonctions membres	18
3.9.3.1	ajouterChoix() [1/2]	18
3.9.3.2	ajouterChoix() [2/2]	18
3.9.3.3	getValeurSectionnee()	18
3.9.3.4	supprimerChoix()	18
3.10	Référence de la classe BoiteChoixReseauNeurones	19
3.10.1	Description détaillée	20
3.11	Référence de la classe BoiteConnexionCouche	20
3.11.1	Description détaillée	21
3.11.2	Documentation des constructeurs et destructeur	21
3.11.2.1	BoiteConnexionCouche()	21
3.11.3	Documentation des fonctions membres	22
3.11.3.1	getCoucheFinale()	22
3.11.3.2	getCoucheInitiale()	22
3.11.3.3	setCoucheFinale()	22

3.11.3.4	setCoucheInitiale()	23
3.12	Référence de la classe BoiteParametrage	23
3.12.1	Description détaillée	24
3.12.2	Documentation des constructeurs et destructeur	24
3.12.2.1	BoiteParametrage()	24
3.12.3	Documentation des fonctions membres	24
3.12.3.1	getParametrage()	24
3.13	Référence de la classe Couche	25
3.13.1	Description détaillée	25
3.13.2	Documentation des fonctions membres	26
3.13.2.1	afficher()	26
3.13.2.2	derivee()	26
3.13.2.3	getDimInput()	26
3.13.2.4	getDimOutput()	27
3.13.2.5	getNom()	27
3.13.2.6	propagation()	27
3.13.2.7	setDimInput()	28
3.13.2.8	setDimOutput()	28
3.14	Référence de la classe CoucheActivation	28
3.14.1	Description détaillée	29
3.15	Référence de la classe CoucheCombinaison	30
3.15.1	Description détaillée	31
3.15.2	Documentation des fonctions membres	31
3.15.2.1	getParams()	31
3.16	Référence de la classe CoucheConnectee	32
3.16.1	Description détaillée	33
3.16.2	Documentation des fonctions membres	33
3.16.2.1	propagation()	33
3.17	Référence de la classe CoucheConnectée	33
3.17.1	Description détaillée	33

3.18	Référence de la classe <code>CoucheConvulsive</code>	34
3.18.1	Description détaillée	35
3.18.2	Documentation des constructeurs et destructeur	36
3.18.2.1	<code>CoucheConvulsive()</code>	36
3.18.3	Documentation des fonctions membres	36
3.18.3.1	<code>propagation()</code>	36
3.18.3.2	<code>setParams()</code>	37
3.19	Référence de la classe <code>DimTenseur</code>	37
3.19.1	Description détaillée	37
3.19.2	Documentation des fonctions membres	37
3.19.2.1	<code>getDim()</code>	38
3.19.2.2	<code>getOrdre()</code>	38
3.20	Référence de la classe <code>Donnee</code>	38
3.20.1	Description détaillée	39
3.20.2	Documentation des fonctions membres	39
3.20.2.1	<code>getEntree()</code>	39
3.20.2.2	<code>getSortie()</code>	39
3.20.2.3	<code>setEntree()</code>	39
3.20.2.4	<code>setSortie()</code>	40
3.21	Référence de la classe <code>Donnees</code>	40
3.21.1	Description détaillée	41
3.21.2	Documentation des fonctions membres	41
3.21.2.1	<code>ajouterDonnee()</code>	41
3.21.2.2	<code>ajouterDonnees()</code>	41
3.21.2.3	<code>getDonnee()</code>	42
3.21.2.4	<code>getDonnees()</code>	42
3.21.2.5	<code>getNbDonnees()</code>	42
3.22	Référence de la classe <code>Erreur</code>	43
3.22.1	Description détaillée	43
3.22.2	Documentation des fonctions membres	43

3.22.2.1	<a href="#">derivee()</a>	43
3.22.2.2	<a href="#">eval()</a>	44
3.23	<a href="#">Référence de la classe ErreurEntropieCroisee</a>	44
3.23.1	<a href="#">Description détaillée</a>	45
3.24	<a href="#">Référence de la classe ErreurL1</a>	46
3.24.1	<a href="#">Description détaillée</a>	46
3.25	<a href="#">Référence de la classe ErreurQuadratique</a>	47
3.25.1	<a href="#">Description détaillée</a>	48
3.26	<a href="#">Référence du modèle de la classe Graphe&lt; Type &gt;</a>	48
3.26.1	<a href="#">Description détaillée</a>	48
3.26.2	<a href="#">Documentation des fonctions membres</a>	49
3.26.2.1	<a href="#">ajouterArc()</a>	49
3.26.2.2	<a href="#">ajouterNoeud()</a>	49
3.26.2.3	<a href="#">contientCycle()</a>	50
3.26.2.4	<a href="#">estConnexe()</a>	50
3.26.2.5	<a href="#">supprimerArc()</a>	50
3.26.2.6	<a href="#">supprimerNoeud()</a>	51
3.27	<a href="#">Référence de la classe Matrice</a>	51
3.27.1	<a href="#">Description détaillée</a>	52
3.27.2	<a href="#">Documentation des fonctions membres</a>	52
3.27.2.1	<a href="#">produitMatriciel()</a>	52
3.28	<a href="#">Référence de la classe MaxPooling</a>	53
3.28.1	<a href="#">Description détaillée</a>	54
3.28.2	<a href="#">Documentation des fonctions membres</a>	54
3.28.2.1	<a href="#">getPoolX()</a>	54
3.28.2.2	<a href="#">getPoolY()</a>	54
3.28.2.3	<a href="#">propagation()</a>	54
3.28.2.4	<a href="#">setPoolX()</a>	55
3.28.2.5	<a href="#">setPoolY()</a>	55
3.29	<a href="#">Référence de la classe Neurone</a>	56

3.29.1 Description détaillée . . . . .	57
3.30 Référence de la classe Optimisateur . . . . .	57
3.30.1 Description détaillée . . . . .	57
3.31 Référence de la classe Panneau . . . . .	58
3.31.1 Description détaillée . . . . .	59
3.31.2 Documentation des fonctions membres . . . . .	59
3.31.2.1 afficher() . . . . .	59
3.31.2.2 BoiteArchitecture() . . . . .	59
3.31.2.3 getReseauNeurones() . . . . .	60
3.32 Référence de la classe PanneauArchitecture . . . . .	60
3.32.1 Description détaillée . . . . .	61
3.33 Référence de la classe ParametresApprentissage . . . . .	62
3.33.1 Description détaillée . . . . .	62
3.33.2 Documentation des fonctions membres . . . . .	62
3.33.2.1 getNbEpoques() . . . . .	62
3.33.2.2 getTauxApprentissage() . . . . .	63
3.33.2.3 setNbEpoques() . . . . .	63
3.33.2.4 setTauxApprentissage() . . . . .	63
3.34 Référence de la classe Pretraitement . . . . .	63
3.34.1 Description détaillée . . . . .	64
3.34.2 Documentation des fonctions membres . . . . .	64
3.34.2.1 chargerDonnees() . . . . .	64
3.34.2.2 chargerRN() . . . . .	65
3.34.2.3 csvToTenseur() . . . . .	65
3.34.2.4 denormaliser() . . . . .	65
3.34.2.5 imageToTenseur() . . . . .	66
3.34.2.6 normaliser() . . . . .	66
3.35 Référence de la classe ReLU . . . . .	67
3.35.1 Description détaillée . . . . .	67
3.35.2 Documentation des fonctions membres . . . . .	68



3.35.2.1	propagation()	68
3.36	Référence de la classe ReseauNeurones	68
3.36.1	Description détaillée	69
3.36.2	Documentation des fonctions membres	70
3.36.2.1	ajouterCoucheFinale()	70
3.36.2.2	ajouterCoucheInitiale()	70
3.36.2.3	getNom()	70
3.36.2.4	propagation()	70
3.36.2.5	supprimerCoucheFinale()	71
3.36.2.6	supprimerCoucheInitiale()	71
3.37	Référence de la classe Sigmoid	71
3.37.1	Description détaillée	72
3.37.2	Documentation des fonctions membres	73
3.37.2.1	propagation()	73
3.38	Référence de la classe TanH	74
3.38.1	Description détaillée	75
3.38.2	Documentation des fonctions membres	75
3.38.2.1	propagation()	75
3.39	Référence de la classe Tenseur	76
3.39.1	Description détaillée	76
3.39.2	Documentation des constructeurs et destructeur	77
3.39.2.1	Tenseur()	77
3.39.3	Documentation des fonctions membres	77
3.39.3.1	getDim()	77
3.39.3.2	getValeur()	77
3.39.3.3	produitTermeATerme()	77
3.39.3.4	setDim()	78
3.39.3.5	setValeur()	78
3.40	Référence de la classe TestCouche	78
3.40.1	Description détaillée	80
3.41	Référence de la classe TestDonnees	80
3.41.1	Description détaillée	81
3.42	Référence de la classe TestGraphe	81
3.42.1	Description détaillée	82
3.43	Référence de la classe TestReseauNeurones	83
3.43.1	Description détaillée	83
3.44	Référence de la classe TestTenseur	84
3.44.1	Description détaillée	84
3.45	Référence de la classe Vecteur	85
3.45.1	Description détaillée	85
3.45.2	Documentation des fonctions membres	86
3.45.2.1	produitScalaire()	86



# Chapitre 1

## Index hiérarchique

### 1.1 Hiérarchie des classes

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

Apprentissage . . . . .	5
Boite . . . . .	8
BoiteChoixFichier . . . . .	14
BoiteChoixDonnées . . . . .	11
BoiteChoixReseauNeurones . . . . .	19
BoiteChoixMultiple . . . . .	16
BoiteChoixCouche . . . . .	9
BoiteChoixErreur . . . . .	12
BoiteConnexionCouche . . . . .	20
BoiteParametrage . . . . .	23
Panneau . . . . .	58
PanneauArchitecture . . . . .	60
BoiteArchitecture . . . . .	9
BoiteChoixDonnees . . . . .	10
Couche . . . . .	25
CoucheActivation . . . . .	28
ReLU . . . . .	67
Sigmoid . . . . .	71
TanH . . . . .	74
CoucheCombinaison . . . . .	30
CoucheConnectee . . . . .	32
Neurone . . . . .	56
CoucheConvulsive . . . . .	34
MaxPooling . . . . .	53
ReseauNeurones . . . . .	68
CoucheConnectée . . . . .	33
DimTenseur . . . . .	37
Donnee . . . . .	38
Donnees . . . . .	40
Erreur . . . . .	43
ErreurEntropieCroisee . . . . .	44
ErreurL1 . . . . .	46
ErreurQuadratique . . . . .	47

Graphe< Type > . . . . .	48
Graphe< Couche > . . . . .	48
ReseauNeurones . . . . .	68
Optimisateur . . . . .	57
ParametresApprentissage . . . . .	62
Pretraitement . . . . .	63
Tenseur . . . . .	76
Matrice . . . . .	51
Vecteur . . . . .	85
TestFixture	
TestCouche . . . . .	78
TestCouche . . . . .	78
TestDonnees . . . . .	80
TestDonnees . . . . .	80
TestGraphe . . . . .	81
TestReseauNeurones . . . . .	83
TestReseauNeurones . . . . .	83
TestTenseur . . . . .	84
TestTenseur . . . . .	84
TestFixture	
TestGraphe . . . . .	81

## Chapitre 2

# Index des classes

### 2.1 Liste des classes

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

<a href="#">Apprentissage</a>	Gère l'apprentissage automatique des paramètres du réseau de neurones . . . . .	5
<a href="#">Boite</a>	Gestion du type <a href="#">Boite</a> . . . . .	8
<a href="#">BoiteArchitecture</a>	Gestion de l'interaction Homme/Machine liée au choix de l'architecture . . . . .	9
<a href="#">BoiteChoixCouche</a>	Gestion de l'interaction Homme/Machine liée au choix des couches . . . . .	9
<a href="#">BoiteChoixDonnees</a>	Gestion de l'interaction Homme/Machine liée au choix du fichier de données . . . . .	10
<a href="#">BoiteChoixDonnées</a>	. . . . .	11
<a href="#">BoiteChoixErreur</a>	Gestion de l'interaction Homme/Machine liée au choix de l'erreur . . . . .	12
<a href="#">BoiteChoixFichier</a>	Gestion de l'interaction Homme/Machine liée au choix de fichier . . . . .	14
<a href="#">BoiteChoixMultiple</a>	Gestion de l'interaction Homme/Machine liée au choix de l'utilisateur . . . . .	16
<a href="#">BoiteChoixReseauNeurones</a>	Gestion de l'interaction Homme/Machine liée au choix du fichier de RN . . . . .	19
<a href="#">BoiteConnexionCouche</a>	Gestion de l'interaction Homme/Machine liée à la liaison entre 2 couches . . . . .	20
<a href="#">BoiteParametrage</a>	. . . . .	23
<a href="#">Couche</a>	Classe liée à la couche . . . . .	25
<a href="#">CoucheActivation</a>	Gestion d'une couche d'activation . . . . .	28
<a href="#">CoucheCombinaison</a>	Gestion d'une couche de combinaison . . . . .	30
<a href="#">CoucheConnectee</a>	. . . . .	32
<a href="#">CoucheConnectée</a>	Création d'une couche connectee . . . . .	33
<a href="#">CoucheConvulsive</a>	Création d'une couche convolutive . . . . .	34
<a href="#">DimTenseur</a>	Classe stockant les dimensions d'un tenseur . . . . .	37

<a href="#">Donnee</a>	Gère une des donnees necessaires a l'apprentissage . . . . .	38
<a href="#">Donnees</a>	Gère les donnees necessaire a l'apprentissage . . . . .	40
<a href="#">Erreur</a>	Gère le calcul de l'erreur commise lors d'un apprentissage . . . . .	43
<a href="#">ErreurEntropieCroisee</a>	Gère le calcul de l'erreur commise suivant l'entropie croisee lors d'un apprentissage . . . . .	44
<a href="#">ErreurL1</a>	Gère le calcul de l'erreur commise lors d'un apprentissage suivant la norme L1 . . . . .	46
<a href="#">ErreurQuadratique</a>	Gère le calcul de l'erreur commise de facon quadratique lors d'un apprentissage . . . . .	47
<a href="#">Graphe&lt; Type &gt;</a>	Gestion du type <a href="#">Graphe</a> . . . . .	48
<a href="#">Matrice</a>	Classe qui crée une matrice . . . . .	51
<a href="#">MaxPooling</a>	Classe gérant une couche de <a href="#">MaxPooling</a> . . . . .	53
<a href="#">Neurone</a>	. . . . .	56
<a href="#">Optimisateur</a>	Gère l'optimisation d'un apprentissage . . . . .	57
<a href="#">Panneau</a>	Gestion du type <a href="#">Panneau</a> . . . . .	58
<a href="#">PanneauArchitecture</a>	???	60
<a href="#">ParametresApprentissage</a>	Gère les parametres d'apprentissage du réseau de neurones . . . . .	62
<a href="#">Pretraitement</a>	Gère le pretraitement des donnees necessaires a l'apprentissage . . . . .	63
<a href="#">ReLU</a>	. . . . .	67
<a href="#">ReseauNeurones</a>	Classe liée à la création du réseau de neurones . . . . .	68
<a href="#">Sigmoid</a>	Création de la fonction <a href="#">ReLU</a> . . . . .	71
<a href="#">TanH</a>	Création d'une fonction tangente hyperbolique . . . . .	74
<a href="#">Tenseur</a>	Classe liée à la manipulation de tenseurs . . . . .	76
<a href="#">TestCouche</a>	Test des méthodes de la classe <a href="#">Couche</a> . . . . .	78
<a href="#">TestDonnees</a>	Test des méthodes de la classe <a href="#">Donnees</a> . . . . .	80
<a href="#">TestGraphe</a>	Test des methodes de la classe <a href="#">Graphe</a> . . . . .	81
<a href="#">TestReseauNeurones</a>	. . . . .	83
<a href="#">TestTenseur</a>	. . . . .	84
<a href="#">Vecteur</a>	Classe qui crée un vecteur . . . . .	85

## Chapitre 3

# Documentation des classes

### 3.1 Référence de la classe Apprentissage

Gère l'apprentissage automatique des paramètres du réseau de neurones.

```
#include <Apprentissage.hpp>
```

#### Fonctions membres publiques

- [Apprentissage](#) ()  
*Constructeur d'une session d'apprentissage vide.*
- [Apprentissage](#) ([ReseauNeurones](#) reseauN, [Erreur](#) erreur)  
*Constructeur d'une session d'apprentissage avec un reseau de neurones et une erreur donnees.*
- void [apprendre](#) ()  
*Lance l'apprentissage.*
- [ReseauNeurones](#) [getRN](#) ()  
*Recupere le reseau de neurones.*
- [Optimisateur](#) [getOptimisateur](#) ()  
*Recupere l'optimisateur.*
- [Erreur](#) [getError](#) ()  
*Recupere l'erreur.*
- [Donnees](#) [getDonnees](#) ()  
*Recupere les donnees d'apprentissage.*
- [ParametresApprentissage](#) [getParam](#) ()  
*Recupere les parametres d'apprentissage du reseau.*
- void [setOptimisateur](#) ([Optimisateur](#) optimisateur)
- void [setDonnees](#) ([Donnees](#) d)  
*Met a jour les donnees de l'apprentissage.*
- void [setParam](#) ([ParametresApprentissage](#) paramApp)  
*Met a jour les parametres d'apprentissage du reseau de neurones.*

#### 3.1.1 Description détaillée

Gère l'apprentissage automatique des paramètres du réseau de neurones.

Auteur

Marion

**Version**

1.0

**Date**

avril 2019

Module permettant le lancement de l'apprentissage. Il permet également la récupération des données et paramètres d'apprentissage.

Définition à la ligne 23 du fichier Apprentissage.hpp.

### 3.1.2 Documentation des fonctions membres

#### 3.1.2.1 getDonnees()

```
Donnes Apprentissage::getDonnees ( )
```

Recupere les donnees d'apprentissage.

**Renvoie**

les données d'apprentissage.

#### 3.1.2.2 getErreur()

```
Erreur Apprentissage::getErreur ( )
```

Recupere l'erreur.

**Renvoie**

l'erreur. \*

#### 3.1.2.3 getOptimisateur()

```
Optimisateur Apprentissage::getOptimisateur ( )
```

Recupere l'optimisateur.

**Renvoie**

l'optimisateur.'



#### 3.1.2.4 getParam()

```
ParametresApprentissage Apprentissage::getParam ( )
```

Recupere les parametres d'apprentissage du reseau.

**Renvoie**

les parametres d'apprentissage actuels.

#### 3.1.2.5 getRN()

```
ReseauNeurones Apprentissage::getRN ( )
```

Recupere le reseau de neurones.

**Renvoie**

le reseau de neurones.' \*

#### 3.1.2.6 setDonnees()

```
void Apprentissage::setDonnees (
    Donnees d )
```

Met a jour les donnees de l'apprentissage.

**Paramètres**

<i>d</i>	les donnees de l'apprentissage.
----------	---------------------------------

#### 3.1.2.7 setParam()

```
void Apprentissage::setParam (
    ParametresApprentissage paramApp )
```

Met a jour les parametres d'apprentissage du reseau de neurones.

**Paramètres**

<i>paramApp</i>	les parametres d'apprentissage.
-----------------	---------------------------------

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

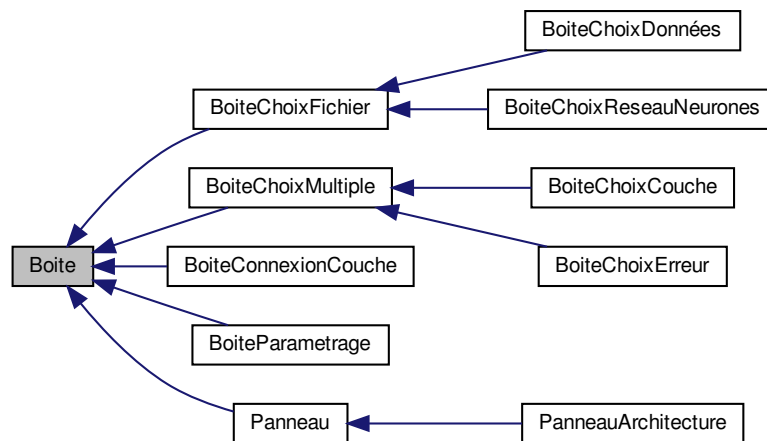
— src/deeplearn/train/Apprentissage.hpp

## 3.2 Référence de la classe Boite

Gestion du type [Boite](#).

```
#include <Boite.hpp>
```

Graphe d'héritage de Boite :



### Fonctions membres publiques

— [Boite](#) (string nom)  
*Constructeur d'une boite avec un nom.*

#### 3.2.1 Description détaillée

Gestion du type [Boite](#).

Auteur

Samra

Version

1.0

Date

avril 2019

Cette classe hérite de `getkmm : :Frame`

Définition à la ligne 19 du fichier `Boite.hpp`.

La documentation de cette classe a été générée à partir du fichier suivant :  
— src/ihm/Boite.hpp

### 3.3 Référence de la classe BoiteArchitecture

Gestion de l'interaction Homme/Machine liée au choix de l'architecture.

```
#include <BoiteArchitecture.hpp>
```

#### 3.3.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix de l'architecture.

**Auteur**

Marion

**Version**

1.0

**Date**

avril 2019

Ce module gere le choix de l'architecture du reseau de neurones utilisé par le logiciel Cette classe hérite de [Boite](#)

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

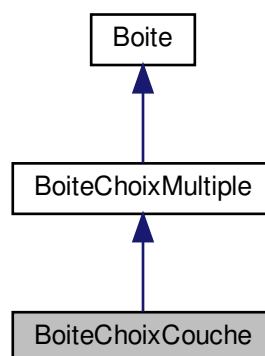
— src/ihm/BoiteArchitecture.hpp

### 3.4 Référence de la classe BoiteChoixCouche

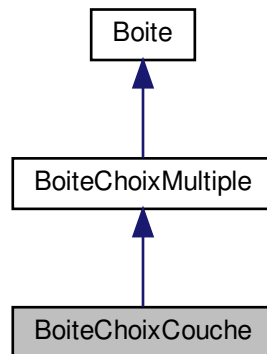
Gestion de l'interaction Homme/Machine liée au choix des couches.

```
#include <BoiteChoixCouche.hpp>
```

Graphe d'héritage de BoiteChoixCouche :



Graphe de collaboration de BoiteChoixCouche :



## Membres hérités additionnels

### 3.4.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix des couches.

#### Auteur

Samra

#### Version

1.0

#### Date

avril 2019

Cette classe permet d'indiquer à l'utilisateur les différentes couches qu'il peut modifier et permet d'accéder à celle qu'il choisit Cette classe hérite de [BoiteChoixMultiple](#)

Définition à la ligne 19 du fichier BoiteChoixCouche.hpp.

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

— src/ihm/BoiteChoixCouche.hpp

## 3.5 Référence de la classe BoiteChoixDonnees

Gestion de l'interaction Homme/Machine liée au choix du fichier de données.

```
#include <BoiteChoixDonnees.hpp>
```

### 3.5.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix du fichier de données.

Auteur

Samra

Version

1.0

Date

avril 2019

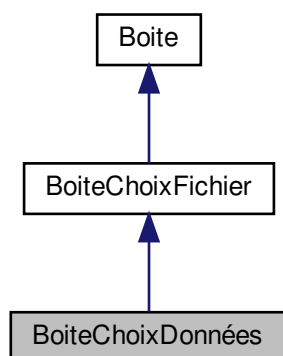
Ce module gere le choix des fichiers de données utilisés par le logiciel Cette classe hérite de [BoiteChoixFichier](#)

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

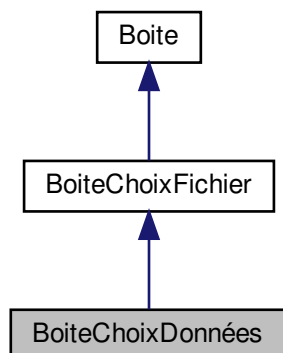
— src/ihm/BoiteChoixDonnees.hpp

## 3.6 Référence de la classe BoiteChoixDonnées

Graphe d'héritage de BoiteChoixDonnées :



Graphe de collaboration de BoiteChoixDonnées :



## Membres hérités additionnels

### 3.6.1 Description détaillée

Définition à la ligne 20 du fichier BoiteChoixDonnees.hpp.

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

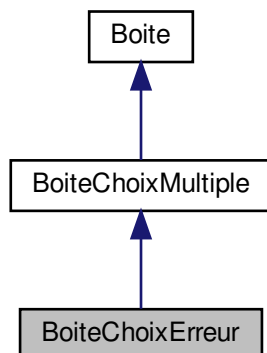
— src/ihm/BoiteChoixDonnees.hpp

## 3.7 Référence de la classe BoiteChoixErreur

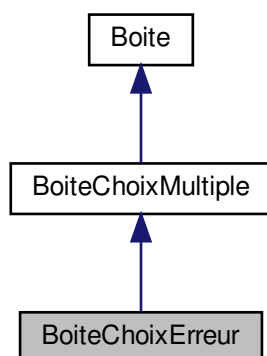
Gestion de l'interaction Homme/Machine liée au choix de l'erreur.

```
#include <BoiteChoixErreur.hpp>
```

Graphe d'héritage de BoiteChoixErreur :



Graphe de collaboration de BoiteChoixErreur :



### Membres hérités additionnels

#### 3.7.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix de l'erreur.

##### Auteur

Samra

##### Version

1.0

##### Date

avril 2019

Cette classe permet à l'utilisateur de choisir la maniere dont il veut quantifier l'erreur Cette classe hérite de [BoiteChoixMultiple](#)

Définition à la ligne 19 du fichier BoiteChoixErreur.hpp.

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

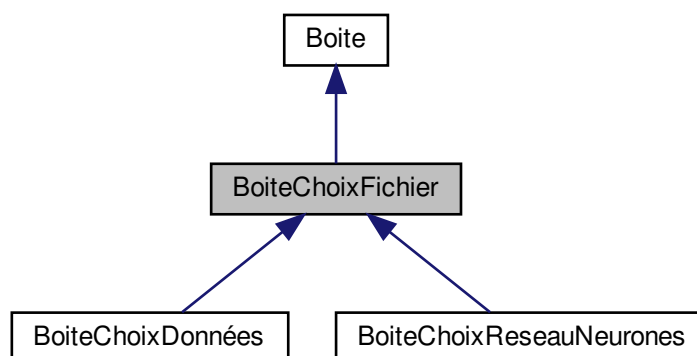
— src/ihm/BoiteChoixErreur.hpp

### 3.8 Référence de la classe BoiteChoixFichier

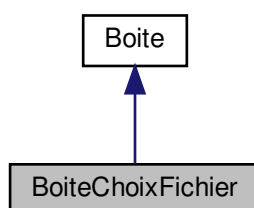
Gestion de l'interaction Homme/Machine liée au choix de fichier.

```
#include <BoiteChoixFichier.hpp>
```

Graphe d'héritage de BoiteChoixFichier :



Graphe de collaboration de BoiteChoixFichier :



#### Fonctions membres publiques

- [BoiteChoixFichier](#) ([Panneau](#) parent)  
*Constucteur prenant en paramètre un [Panneau](#).*
- string [getNomFichier](#) ()  
*getteur permettant d'accéder au nom du fichier en attribut*



### 3.8.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix de fichier.

Gestion de l'interaction Homme/Machine liée au choix des paramètres.

**Auteur**

Samra

**Version**

1.0

**Date**

avril 2019

Ce module gere le choix des fichiers (de donnée ou de reseau de neurones) utilisés par le logiciel Cette classe hérite de [Boite](#)

**Auteur**

Samra

**Version**

1.0

**Date**

avril 2019

Cette classe hérite de [Boite](#)

Définition à la ligne 21 du fichier BoiteChoixFichier.hpp.

### 3.8.2 Documentation des constructeurs et destructeur

#### 3.8.2.1 BoiteChoixFichier()

```
BoiteChoixFichier::BoiteChoixFichier (
    Panneau parent )
```

Constucteur prenant en paramètre un [Panneau](#).

**Paramètres**

<i>parent</i>	un <a href="#">Panneau</a> préalablement défini
---------------	---

### 3.8.3 Documentation des fonctions membres

#### 3.8.3.1 getNomFichier()

```
BoiteChoixFichier::getNomFichier ( )
```

getteur permettant d'accéder au nom du fichier en attribut

Renvoie

le string correspondant au nom du fichier

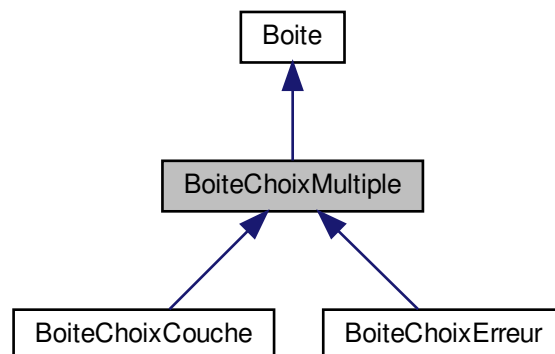
La documentation de cette classe a été générée à partir du fichier suivant :  
— src/m/m/BoiteChoixFichier.hpp

## 3.9 Référence de la classe BoiteChoixMultiple

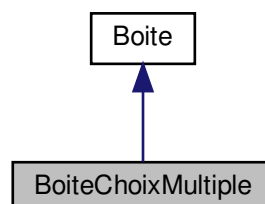
Gestion de l'interaction Homme/Machine liée au choix de l'utilisateur.

```
#include <BoiteChoixMultiple.hpp>
```

Graphe d'héritage de BoiteChoixMultiple :



Graphe de collaboration de BoiteChoixMultiple :



## Fonctions membres publiques

- [BoiteChoixMultiple](#) (string nom)  
*Constucteur prenant en paramètre un string.*
- void [ajouterChoix](#) (string nom)  
*Méthode permettant d'ajouter un string correspondant à un nouveau choix au vecteur de choix.*
- void [ajouterChoix](#) (vector< string > liste\_nom)  
*Méthode permettant d'ajouter un plusieurs string correspondant à des nouveaux choix au vecteur de choix.*
- void [supprimerChoix](#) (string nom)  
*Méthode permettant de supprimer un choix du vecteur de choix.*
- string [getValeurSectionnee](#) ()  
*Méthode permettant la lecture du choix de l'utilisateur.*

### 3.9.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix de l'utilisateur.

#### Auteur

Samra

#### Version

1.0

#### Date

avril 2019

Cette classe permet d'indiquer à l'utilisateur les différentes options qui lui sont proposées et permet d'accéder à celle qu'il choisit Cette classe hérite de [Boite](#)

Définition à la ligne 20 du fichier BoiteChoixMultiple.hpp.

### 3.9.2 Documentation des constructeurs et destructeur

#### 3.9.2.1 BoiteChoixMultiple()

```
BoiteChoixMultiple::BoiteChoixMultiple (
    string nom )
```

Constucteur prenant en paramètre un string.

#### Paramètres

Un	string correspondant à un premier choix offert à l'utilisateur
----	--

### 3.9.3 Documentation des fonctions membres

#### 3.9.3.1 ajouterChoix() [1/2]

```
BoiteChoixMultiple::ajouterChoix (
    string nom )
```

Méthode permettant d'ajouter un string correspondant à un nouveau choix au vecteur de choix.

##### Paramètres

<i>Un</i>	string correspondant au nom du nouveau choix
-----------	--

#### 3.9.3.2 ajouterChoix() [2/2]

```
BoiteChoixMultiple::ajouterChoix (
    vector< string > liste_nom )
```

Méthode permettant d'ajouter un plusieurs string correspondant à des nouveaux choix au vecteur de choix.

##### Paramètres

<i>liste_nom</i>	le vecteur contenant les nouveaux noms de choix
------------------	---

#### 3.9.3.3 getValeurSectionnee()

```
BoiteChoixMultiple::getValeurSectionnee ( )
```

Méthode permettant la lecture du choix de l'utilisateur.

##### Renvoie

le string choisie par l'utilisateur

#### 3.9.3.4 supprimerChoix()

```
BoiteChoixMultiple::supprimerChoix (
    string nom )
```

Méthode permettant de supprimer un choix du vecteur de choix.

## Paramètres

<i>un</i>	string correspondant au nom du choix à retirer
-----------	--

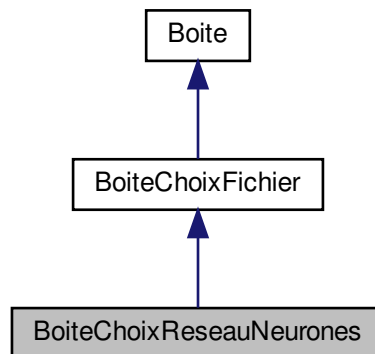
La documentation de cette classe a été générée à partir du fichier suivant :  
— src/ihm/BoiteChoixMultiple.hpp

### 3.10 Référence de la classe BoiteChoixReseauNeurones

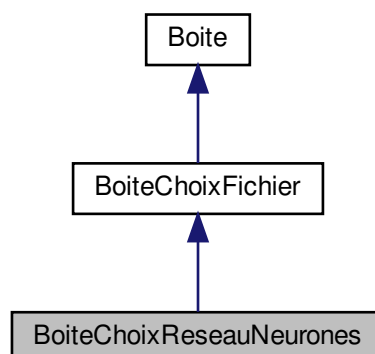
Gestion de l'interaction Homme/Machine liée au choix du fichier de RN.

```
#include <BoiteChoixReseauNeurones.hpp>
```

Graphe d'héritage de BoiteChoixReseauNeurones :



Graphe de collaboration de BoiteChoixReseauNeurones :



## Membres hérités additionnels

### 3.10.1 Description détaillée

Gestion de l'interaction Homme/Machine liée au choix du fichier de RN.

#### Auteur

Samra

#### Version

1.0

#### Date

avril 2019

Ce module gere le choix des fichiers de reseau de neurones utilisés par le logiciel Cette classe hérite de [Boite](#)↔  
[ChoixFichier](#)

Définition à la ligne 20 du fichier BoiteChoixReseauNeurones.hpp.

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

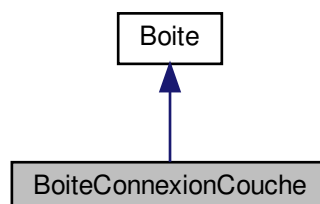
— src/ihm/BoiteChoixReseauNeurones.hpp

## 3.11 Référence de la classe BoiteConnexionCouche

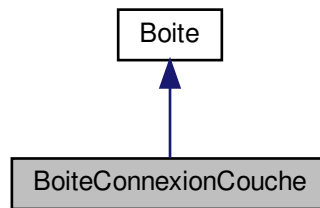
Gestion de l'interaction Homme/Machine liée à la liaison entre 2 couches.

```
#include <BoiteConnexionCouche.hpp>
```

Graphe d'héritage de BoiteConnexionCouche :



Graphe de collaboration de BoiteConnexionCouche :



### Fonctions membres publiques

- [BoiteConnexionCouche](#) ([Panneau](#) parent)  
*Constructeur prenant en paramètre un [Panneau](#).*
- void [setCoucheInitiale](#) ([Couche](#) couche)  
*Setteur permettant de changer la valeur de la [Couche](#) Initiale.*
- void [setCoucheFinale](#) ([Couche](#) couche)  
*Setteur permettant de changer la valeur de la [Couche](#) finale.*
- [Couche](#) [getCoucheInitiale](#) ()  
*Getteur permettant d'obtenir la valeur de la [Couche](#) Initiale.*
- [Couche](#) [getCoucheFinale](#) ()  
*getteur permettant d'obtenir la valeur de la [Couche](#) Finale*

#### 3.11.1 Description détaillée

Gestion de l'interaction Homme/Machine liée à la liaison entre 2 couches.

Auteur

Samra

Version

1.0

Date

avril 2019

Module permettant le choix des couches à relier et la mise en place de leur liaison

Définition à la ligne 21 du fichier BoiteConnexionCouche.hpp.

#### 3.11.2 Documentation des constructeurs et destructeur

##### 3.11.2.1 BoiteConnexionCouche()

```
BoiteConnexionCouche::BoiteConnexionCouche (
    Panneau parent )
```

Constructeur prenant en paramètre un [Panneau](#).

**Paramètres**

<i>parent</i>	un <a href="#">Panneau</a> contenant déjà un réseau de neurones
---------------	---

**3.11.3 Documentation des fonctions membres****3.11.3.1 `getCoucheFinale()`**

```
BoiteConnexionCouche::getCoucheFinale ( )
```

getteur permettant d'obtenir la valeur de la [Couche](#) Finale

**Renvoie**

la [Couche](#) `c_finale` en attribut de la classe

**3.11.3.2 `getCoucheInitiale()`**

```
BoiteConnexionCouche::getCoucheInitiale ( )
```

Getteur permettant d'obtenir la valeur de la [Couche](#) Initiale.

**Renvoie**

la [Couche](#) `c_init` en attribut de la classe

**3.11.3.3 `setCoucheFinale()`**

```
BoiteConnexionCouche::setCoucheFinale (
    Couche couche )
```

Setteur permettant de changer la valeur de la [Couche](#) finale.

**Paramètres**

<i>la</i>	nouvelle <a href="#">Couche</a> finale
-----------	--



## 3.11.3.4 setCoucheInitiale()

```
BoiteConnexionCouche::setCoucheInitiale (
    Couche couche )
```

Setteur permettant de changer la valeur de la [Couche](#) Initiale.

## Paramètres

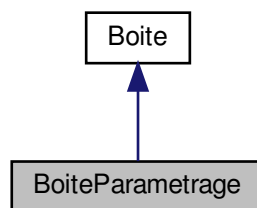
<i>la</i>	nouvelle <a href="#">Couche</a> initiale
-----------	--

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

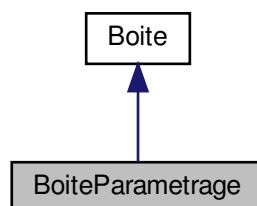
— src/ihm/BoiteConnexionCouche.hpp

## 3.12 Référence de la classe BoiteParametrage

Graphe d'héritage de BoiteParametrage :



Graphe de collaboration de BoiteParametrage :



## Fonctions membres publiques

- [BoiteParametrage](#) ([Panneau](#) parent)  
*Constucteur prenant en paramètre un [Panneau](#).*
- [ParametresApprentissage](#) `getParametrage ()`  
*Getteur permettant d'accéder aux paramètres en attribut.*

### 3.12.1 Description détaillée

Définition à la ligne 19 du fichier BoiteParametrage.hpp.

### 3.12.2 Documentation des constructeurs et destructeur

#### 3.12.2.1 BoiteParametrage()

```
BoiteParametrage::BoiteParametrage (
    Panneau parent )
```

Constucteur prenant en paramètre un [Panneau](#).

#### Paramètres

<i>parent</i>	un <a href="#">Panneau</a> contenant déjà un réseau de neurones
---------------	---

### 3.12.3 Documentation des fonctions membres

#### 3.12.3.1 getParametrage()

```
BoiteParametrage::getParametrage ( )
```

Getteur permettant d'accéder aux paramètres en attribut.

#### Renvoie

[ParametresApprentissage](#)

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

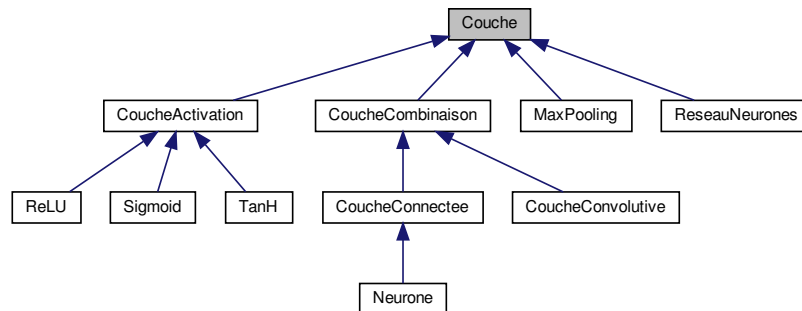
- `src/ihm/BoiteParametrage.hpp`

## 3.13 Référence de la classe Couche

Classe liée à la couche.

```
#include <Couche.hpp>
```

Graphe d'héritage de Couche :



### Fonctions membres publiques

- `Couche` (`DimTenseur` din, `DimTenseur` dout, `std::string` no)  
*Constructeur d'une couche à partir de la taille des tenseurs d'entrée/sortie.*
- `virtual Tenseur propagation` (`Tenseur` t)  
*Méthode virtuelle permettant la propagation d'une couche à une autre.*
- `virtual Tenseur derivee` (`Tenseur` t)  
*Méthode virtuelle pour avoir la dérivée de la couche.*
- `void setDimInput` (`DimTenseur` dimIn)  
*Méthode pour fixer la taille du tenseur à l'entrée de la couche.*
- `void setDimOutput` (`DimTenseur` dimOut)  
*Méthode pour fixer la taille du tenseur à la sortie de la couche.*
- `DimTenseur getDimInput` () `const`  
*Méthode pour obtenir la taille du tenseur à l'entrée de la couche.*
- `DimTenseur getDimOutput` () `const`  
*Méthode pour obtenir la taille du tenseur à la sortie de la couche.*
- `std::string getNom` () `const`  
*Méthode pour obtenir le nom de la couche.*
- `virtual bool afficher` ()  
*Méthode pour savoir si la couche est affichée ou non.*

### 3.13.1 Description détaillée

Classe liée à la couche.

Auteur

Adrien

Version

1.0

Date

avril 2019

Cette classe permet de fixer les tailles (entrée/sortie) de la couche. Une couche peut être dérivée et propagée.

Définition à la ligne 20 du fichier Couche.hpp.

### 3.13.2 Documentation des fonctions membres

#### 3.13.2.1 afficher()

```
bool Couche::afficher ( ) [virtual]
```

Méthode pour savoir si la couche est affichée ou non.

##### Renvoie

booléen

Définition à la ligne 60 du fichier Couche.cpp.

#### 3.13.2.2 derivee()

```
void Couche::derivee (
    Tenseur t ) [virtual]
```

Methode virtuelle pour avoir la derivee de la couche.

##### Paramètres

<i>t</i>	le tenseur pour lequel on veut la derivee
----------	---

##### Renvoie

la derivee de la couche

Définition à la ligne 19 du fichier Couche.cpp.

#### 3.13.2.3 getDimInput()

```
DimTenseur Couche::getDimInput ( ) const
```

Méthode pour obtenir la taille du tenseur à l'entrée de la couche.

##### Renvoie

La taille du tenseur d'entrée

Définition à la ligne 39 du fichier Couche.cpp.

#### 3.13.2.4 getDimOutput()

```
DimTenseur Couche::getDimOutput ( ) const
```

Méthode pour obtenir la taille du tenseur à la sortie de la couche.

##### Renvoie

La taille du tenseur de sortie

Définition à la ligne 46 du fichier Couche.cpp.

#### 3.13.2.5 getNom()

```
string Couche::getNom ( ) const
```

Méthode pour obtenir le nom de la couche.

##### Renvoie

Le nom de la couche

Définition à la ligne 53 du fichier Couche.cpp.

#### 3.13.2.6 propagation()

```
Tenseur Couche::propagation (
    Tenseur t ) [virtual]
```

Méthode virtuelle permettant la propagation d'une couche à une autre.

##### Paramètres

<i>t</i>	le tenseur d'entree
----------	---------------------

##### Renvoie

la sortie de la couche

Réimplémentée dans [ReseauNeurones](#), [CoucheConvolutive](#), [MaxPooling](#), [CoucheConnectee](#), [ReLU](#), [Sigmoid](#), et [TanH](#).

Définition à la ligne 13 du fichier Couche.cpp.

### 3.13.2.7 setDimInput()

```
void Couche::setDimInput (
    DimTenseur dimIn )
```

Méthode pour fixer la taille du tenseur à l'entrée de la couche.

#### Paramètres

<i>dimIn</i>	La dimension du tenseur d'entrée
--------------	----------------------------------

Définition à la ligne 25 du fichier Couche.cpp.

### 3.13.2.8 setDimOutput()

```
void Couche::setDimOutput (
    DimTenseur dimOut )
```

Méthode pour fixer la taille du tenseur à la sortie de la couche.

#### Paramètres

<i>dimOut</i>	La dimension du tenseur de sortie
---------------	-----------------------------------

Définition à la ligne 32 du fichier Couche.cpp.

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

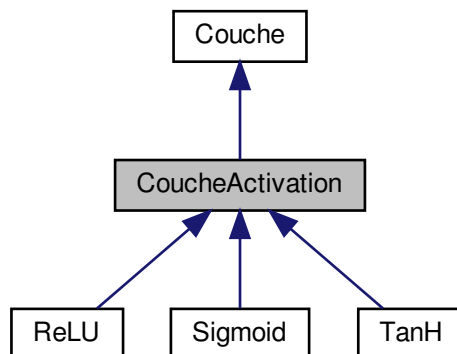
- src/deeplearn/archi/Couche.hpp
- src/deeplearn/archi/Couche.cpp

## 3.14 Référence de la classe CoucheActivation

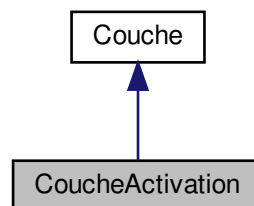
Gestion d'une couche d'activation.

```
#include <CoucheActivation.hpp>
```

Graphe d'héritage de CoucheActivation :



Graphe de collaboration de CoucheActivation :



### Fonctions membres publiques

- [CoucheActivation](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no)  
*Constructeur d'une couche d'activation avec une dimension d'entree.*

#### 3.14.1 Description détaillée

Gestion d'une couche d'activation.

Auteur

Adrien

Version

1.0

**Date**

avril 2019

Classe permettant la création d'une couche d'activation de trois types différents (sigmoid, tangente hyperbolique ou Rectified Linear Units) Cette classe hérite de la classe [Couche](#).

Définition à la ligne 18 du fichier CoucheActivation.hpp.

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

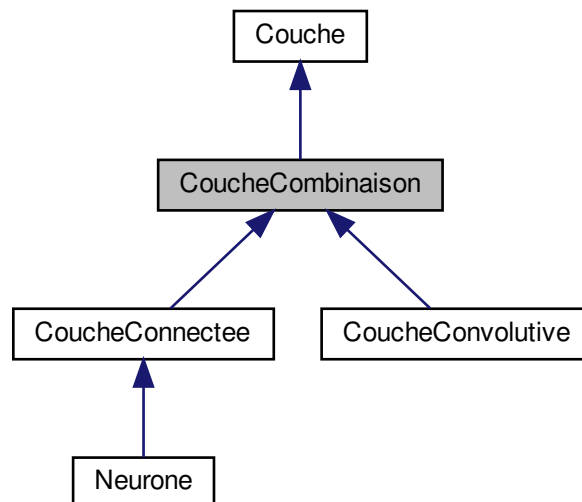
- `src/deeplearn/archi/CoucheActivation.hpp`
- `src/deeplearn/archi/CoucheActivation.cpp`

### 3.15 Référence de la classe CoucheCombinaison

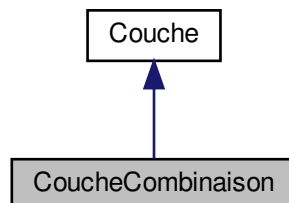
Gestion d'une couche de combinaison.

```
#include <CoucheCombinaison.hpp>
```

Graphe d'héritage de CoucheCombinaison :



Graphe de collaboration de CoucheCombinaison :





## Fonctions membres publiques

- `CoucheCombinaison` (`DimTenseur` din, `DimTenseur` dout, std : :string no, `Tenseur` par)  
*Constructeur d'une couche à partir de la taille des tenseurs d'entrée/sortie.*
- void `setParams` (`Tenseur` nouvParams)  
*setter les parametres de la couche.*
- `Tenseur` `getParams` ()  
*getter des parametres de la couche.*

### 3.15.1 Description détaillée

Gestion d'une couche de combinaison.

#### Auteur

Adrien

#### Version

1.0

#### Date

avril 2019

Classe permettant la création d'une couche de combinaison c'est-à-dire comportant les parametres du reseau

Définition à la ligne 16 du fichier CoucheCombinaison.hpp.

### 3.15.2 Documentation des fonctions membres

#### 3.15.2.1 `getParams()`

```
Tenseur CoucheCombinaison::getParams ( )
```

getter des parametres de la couche.

#### Renvoie

les parametres de la couche

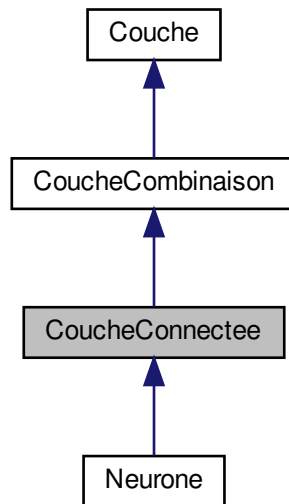
Définition à la ligne 16 du fichier CoucheCombinaison.cpp.

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

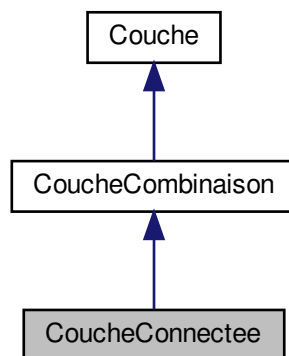
- `src/deeplearn/archi/CoucheCombinaison.hpp`
- `src/deeplearn/archi/CoucheCombinaison.cpp`

### 3.16 Référence de la classe CoucheConnectee

Graphe d'héritage de CoucheConnectee :



Graphe de collaboration de CoucheConnectee :



#### Fonctions membres publiques

- [CoucheConnectee](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no, [Tenseur](#) par)  
*Constructeur d'une couche connectée à partir du nombre de sorties.*
- [Tenseur propagation](#) ([Tenseur](#) t)  
*Methode permettant la propagation d'une couche à une autre.*

### 3.16.1 Description détaillée

Définition à la ligne 17 du fichier CoucheConnectee.hpp.

### 3.16.2 Documentation des fonctions membres

#### 3.16.2.1 propagation()

```
Tenseur CoucheConnectee::propagation (
    Tenseur t ) [virtual]
```

Methode permettant la propagation d'une couche à une autre.

#### Paramètres

<i>t</i>	le tenseur d'entree
----------	---------------------

#### Renvoie

la sortie de la couche

Réimplémentée à partir de [Couche](#).

Définition à la ligne 8 du fichier CoucheConnectee.cpp.

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

- src/deeplearn/archi/CoucheConnectee.hpp
- src/deeplearn/archi/CoucheConnectee.cpp

## 3.17 Référence de la classe CoucheConnectée

Création d'une couche connectee.

```
#include <CoucheConnectee.hpp>
```

### 3.17.1 Description détaillée

Création d'une couche connectee.

Création d'un neurone.

#### Auteur

Adrien

**Version**

1.0

**Date**

avril 2019

Classe permettant la création d'une couche de combinaison de "type" connectée : tous les neurones sont reliés d'une couche à une autre.

**Auteur**

Adrien

**Version**

1.0

**Date**

avril 2019

Classe permettant la création d'un neurone. Cette classe hérite de la classe [CoucheConnectée](#).

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

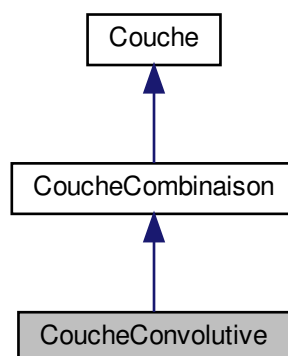
— `src/deeplearn/archi/CoucheConnectee.hpp`

### 3.18 Référence de la classe CoucheConvulsive

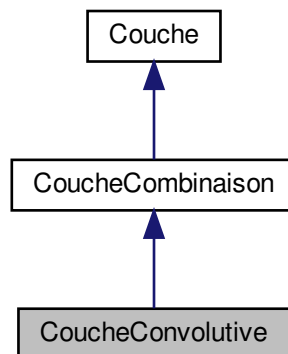
Création d'une couche convulsive.

```
#include <CoucheConvulsive.hpp>
```

Graphe d'héritage de CoucheConvulsive :



Graphe de collaboration de CoucheConvulsive :



### Fonctions membres publiques

- **CoucheConvulsive** (**DimTenseur** din, **DimTenseur** dout, std : :string no, **Tenseur** par, int l\_fil, int h\_fil, int nb\_fil)  
*Constructeur d'une couche convulsive à partir de la longueur du filtre, de la hauteur du filtre et du nombre de filtres.*
- **Tenseur propagation** (**Tenseur** t)  
*Méthode permettant la propagation d'une couche à une autre.*
- void **setL\_Filtre** (int l\_fil)  
*Changer la longueur de chaque filtre.*
- void **setH\_Filtre** (int h\_fil)
- void **setParams** (int nb\_filtres)  
*Changer la largeur de chaque filtre.*

#### 3.18.1 Description détaillée

Création d'une couche convulsive.

Auteur

Adrien

Version

1.0

Date

avril 2019

Classe permettant la création d'une couche de combinaison de "type" convulsive.

Définition à la ligne 17 du fichier CoucheConvulsive.hpp.

### 3.18.2 Documentation des constructeurs et destructeur

#### 3.18.2.1 CoucheConvulsive()

```
CoucheConvulsive::CoucheConvulsive (
    DimTenseur din,
    DimTenseur dout,
    std::string no,
    Tenseur par,
    int l_fil,
    int h_fil,
    int nb_fil )
```

Constructeur d'une couche convolutive à partir de la longueur du filtre, de la hauteur du filtre et du nombre de filtres.

Constructeur d'une couche convolutive à partir de la taille du tenseur d'entrée, de la longueur du filtre, de la hauteur du filtre et du nombre de \* filtres

Définition à la ligne 3 du fichier CoucheConvulsive.cpp.

### 3.18.3 Documentation des fonctions membres

#### 3.18.3.1 propagation()

```
Tenseur CoucheConvulsive::propagation (
    Tenseur t ) [virtual]
```

Méthode permettant la propagation d'une couche a une autre.

##### Paramètres

<i>t</i>	le tenseur d'entree
----------	---------------------

##### Renvoie

la sortie de la couche

Réimplémentée à partir de [Couche](#).

Définition à la ligne 9 du fichier CoucheConvulsive.cpp.

### 3.18.3.2 setParams()

```
void CoucheConvulsive::setParams (
    int nb_filtres )
```

Changer la largeur de chaque filtre.

Changer le nombre de filtres.

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

- src/deeplearn/archi/CoucheConvulsive.hpp
- src/deeplearn/archi/CoucheConvulsive.cpp

## 3.19 Référence de la classe DimTenseur

Classe stockant les dimensions d'un tenseur.

```
#include <DimTenseur.hpp>
```

### Fonctions membres publiques

- [DimTenseur](#) ()  
*Constructeur standard.*
- [DimTenseur](#) (int ord, std::vector< int > di)  
*Constructeur avec d'obtenir la dimension souhaitée.*
- int [getOrdre](#) ()  
*Méthode pour obtenir l'ordre du tenseur.*
- std::vector< int > [getDim](#) ()  
*Méthode pour obtenir la liste des dimensions du tenseur.*

### 3.19.1 Description détaillée

Classe stockant les dimensions d'un tenseur.

#### Auteur

Adrien

#### Version

1.0

#### Date

avril 2019

Classe qui gère la dimension d'un tenseur

Définition à la ligne 18 du fichier DimTenseur.hpp.

### 3.19.2 Documentation des fonctions membres

### 3.19.2.1 getDim()

```
DimTenseur DimTenseur::getDim ( )
```

Méthode pour obtenir la liste des dimensions du tenseur.

#### Renvoie

la liste des dimensions du tenseur

Définition à la ligne 26 du fichier DimTenseur.cpp.

### 3.19.2.2 getOrdre()

```
int DimTenseur::getOrdre ( )
```

Méthode pour obtenir l'ordre du tenseur.

#### Renvoie

l'ordre du tenseur

Définition à la ligne 19 du fichier DimTenseur.cpp.

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

- src/deeplearn/archi/DimTenseur.hpp
- src/deeplearn/archi/DimTenseur.cpp

## 3.20 Référence de la classe Donnee

Gère une des donnees necessaires a l'apprentissage.

```
#include <Donnee.hpp>
```

### Fonctions membres publiques

- [Donnee](#) ([Tenseur](#) entree, [Tenseur](#) sortie)  
*Constructeur de [Donnee](#) avec une entree et une sortie.*
- [Tenseur](#) getEntree ()  
*Retourne l'entree de la donnee.*
- [Tenseur](#) getSortie ()  
*Retourne la sortie de la donnee.*
- void [setEntree](#) ([Tenseur](#) e)  
*Met a jour l'entree de la donnee.*
- void [setSortie](#) ([Tenseur](#) s)  
*Met a jour la sortie de la donnee.*



### 3.20.1 Description détaillée

Gère une des donnees necessaires a l'apprentissage.

**Auteur**

Marion

**Version**

1.0

**Date**

avril 2019

Module permettant le stockage en mémoire vive d'une donnee necessaire a l'apprentissage. La donnee est vu comme un couple Entree/Sortie.

Définition à la ligne 17 du fichier Donnee.hpp.

### 3.20.2 Documentation des fonctions membres

#### 3.20.2.1 getEntree()

```
Tenseur Donnee::getEntree ( )
```

Retourne l'entree de la donnee.

**Renvoie**

l'entree. \*

#### 3.20.2.2 getSortie()

```
Tenseur Donnee::getSortie ( )
```

Retourne la sortie de la donnee.

**Renvoie**

la sortie. \* \*

#### 3.20.2.3 setEntree()

```
void Donnee::setEntree (
    Tenseur e )
```

Met a jour l'entree de la donnee.

## Paramètres

<code>e</code>	le tenseur d'entree
----------------	---------------------

3.20.2.4 `setSortie()`

```
void Donnee::setSortie (
    Tenseur s )
```

Met a jour la sortie de la donnee.

## Paramètres

<code>s</code>	le tenseur de sortie
----------------	----------------------

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

— `src/deeplearn/train/Donnee.hpp`

3.21 Référence de la classe `Donnees`

Gère les donnees necessaire a l'apprentissage.

```
#include <Donnees.hpp>
```

## Fonctions membres publiques

- `Donnees ()`  
*Constructeur des donnees d'apprentissage.*
- `Donnees (std::vector< Donnee > vd)`  
*Constructeur des donnees d'apprentissage a partir d'un vecteur de donnees unitaires.*
- `void ajouterDonnees (Donnees d)`  
*Ajoute un ensemble de donnees.*
- `void ajouterDonnee (Donnee d)`  
*Ajoute une donnee a l'ensemble des donnees.*
- `void melanger ()`  
*Melange les donnees.*
- `Donnee getDonnee (int num)`  
*Retourne la donnee numero num parmi l'ensemble des donnees.*
- `std::vector< Donnees > getDonnees ()`  
*Retourne l'ensemble des donnees.*
- `int getNbDonnees ()`  
*Retourne le nombre de donnees.*

### 3.21.1 Description détaillée

Gère les donnees necessaire a l'apprentissage.

**Auteur**

Marion

**Version**

1.0

**Date**

avril 2019

Module permettant le stockage en memoire vive des donnees necessaire a l'apprentissage.

Définition à la ligne 18 du fichier Donnees.hpp.

### 3.21.2 Documentation des fonctions membres

#### 3.21.2.1 ajouterDonnee()

```
void Donnees::ajouterDonnee (
    Donnee d )
```

Ajoute une donnee a l'ensemble des donnees.

**Paramètres**

<i>d</i>	la donnee a ajouter
----------	---------------------

#### 3.21.2.2 ajouterDonnees()

```
void Donnees::ajouterDonnees (
    Donnees d )
```

Ajoute un ensemble de donnees.

**Paramètres**

<i>d</i>	l'ensemble des donnees
----------	------------------------

### 3.21.2.3 getDonnee()

```
Donnee Donnees::getDonnee (
    int num )
```

Retourne la donnee numero num parmi l'ensemble des donnees.

#### Paramètres

<i>num</i>	le numero de la donnee
------------	------------------------

#### Renvoie

la donnee en position "num"

### 3.21.2.4 getDonnees()

```
vector< Donnees > Donnees::getDonnees ( )
```

Retourne l'ensemble des donnees.

#### Renvoie

le vecteur des donnees

### 3.21.2.5 getNbDonnees()

```
int Donnees::getNbDonnees ( )
```

Retourne le nombre de donnees.

#### Renvoie

le nombre de donnees

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

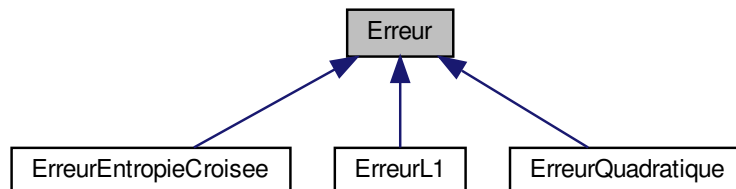
— src/deeplearn/train/Donnees.hpp

## 3.22 Référence de la classe Erreur

Gère le calcul de l'erreur commise lors d'un apprentissage.

```
#include <Erreur.hpp>
```

Graphe d'héritage de Erreur :



### Fonctions membres publiques

- virtual **Tenseur eval** (**Tenseur** sortieRN, **Tenseur** label)  
*Methode virtuelle pour evaluer l'erreur effectuee entre la sortie et la prediction.*
- virtual **Tenseur derivee** (**Tenseur** t)  
*Methode virtuelle pour avoir la derivee d'un tenseur donnee en entree.*

#### 3.22.1 Description détaillée

Gère le calcul de l'erreur commise lors d'un apprentissage.

**Auteur**

Marion

**Version**

1.0

**Date**

avril 2019

Module permettant le traitement de l'erreur commise lors d'un apprentissage. Il permet également la récupération des données liées aux erreurs.

Définition à la ligne 18 du fichier Erreur.hpp.

#### 3.22.2 Documentation des fonctions membres

##### 3.22.2.1 derivee()

```
void Erreur::derivee (  
    Tenseur t ) [virtual]
```

Méthode virtuelle pour avoir la derivee d'un tenseur donnee en entree.

## Paramètres

<i>t</i>	le tenseur pour lequel on veut la derivee
----------	---

## 3.22.2.2 eval()

```
Tenseur Erreur::eval (
    Tenseur sortieRN,
    Tenseur prediction ) [virtual]
```

Methode virtuelle pour evaluer l'erreur effectuee entre la sortie et la prediction.

## Paramètres

<i>sortieRN</i>	le tenseur de sortie du reseau de neurones.
<i>label</i>	la sortie souhaitée du reseau de neurones.

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

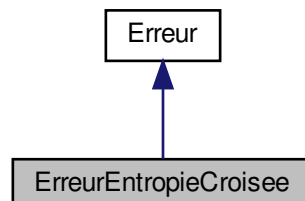
— src/deeplearn/train/Erreur.hpp

## 3.23 Référence de la classe ErreurEntropieCroisee

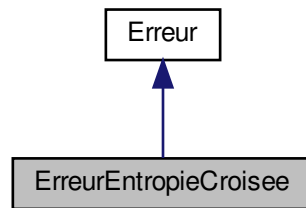
Gère le calcul de l'erreur commise suivant l'entropie croisee lors d'un apprentissage.

```
#include <ErreurEntropieCroisee.hpp>
```

Graphe d'héritage de ErreurEntropieCroisee :



Graphe de collaboration de ErreurEntropieCroisee :



### Fonctions membres publiques

- [ErreurEntropieCroisee](#) ()  
*Constructeur de l'erreur d'entropie croisee.*

#### 3.23.1 Description détaillée

Gère le calcul de l'erreur commise suivant l'entropie croisee lors d'un apprentissage.

##### Auteur

Marion

##### Version

1.0

##### Date

avril 2019

Module permettant le traitement de l'erreur commise suivant l'entropie croisee lors d'un apprentissage. Il permet également la récupération des données liées aux erreurs. Il herite de la classe [Erreur](#).

Définition à la ligne 19 du fichier ErreurEntropieCroisee.hpp.

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

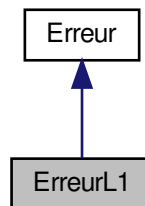
- src/deeplearn/train/ErreurEntropieCroisee.hpp

### 3.24 Référence de la classe ErreurL1

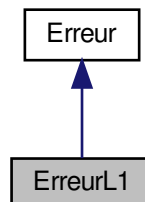
Gère le calcul de l'erreur commise lors d'un apprentissage suivant la norme L1.

```
#include <ErreurL1.hpp>
```

Graphe d'héritage de ErreurL1 :



Graphe de collaboration de ErreurL1 :



#### Fonctions membres publiques

- [ErreurL1](#) ()  
*Constructeur de l'erreur L1.*

#### 3.24.1 Description détaillée

Gère le calcul de l'erreur commise lors d'un apprentissage suivant la norme L1.

Auteur

Marion



## Version

1.0

## Date

avril 2019

Module permettant le traitement de l'erreur commise lors d'un apprentissage suivant la norme L1. Il permet également la récupération des données liées aux erreurs. Il herite de la classe [Erreur](#).

Définition à la ligne 19 du fichier ErreurL1.hpp.

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

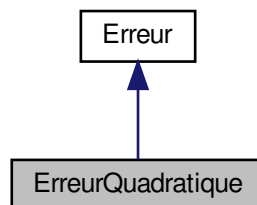
— src/deeplearn/train/ErreurL1.hpp

### 3.25 Référence de la classe ErreurQuadratique

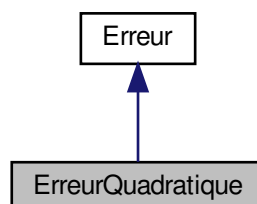
Gère le calcul de l'erreur commise de façon quadratique lors d'un apprentissage.

```
#include <ErreurQuadratique.hpp>
```

Graphe d'héritage de ErreurQuadratique :



Graphe de collaboration de ErreurQuadratique :



## Fonctions membres publiques

- [ErreurQuadratique](#) ()  
*Constructeur de l'erreur quadratique.*

### 3.25.1 Description détaillée

Gère le calcul de l'erreur commise de façon quadratique lors d'un apprentissage.

#### Auteur

Marion

#### Version

1.0

#### Date

avril 2019

Module permettant le traitement de l'erreur quadratique commise lors d'un apprentissage. Il permet également la récupération des données liées aux erreurs. Il hérite de la classe [Erreur](#).

Définition à la ligne 19 du fichier ErreurQuadratique.hpp.

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

- `src/deeplearn/train/ErreurQuadratique.hpp`

## 3.26 Référence du modèle de la classe Graphe< Type >

Gestion du type [Graphe](#).

```
#include <Graphe.hpp>
```

## Fonctions membres publiques

- [Graphe](#) ()  
*Constructeur de graphe vide.*
- [Graphe](#) (Type noeuds[])  
*Constructeur de graphe à partir de noeuds isolés.*
- void [ajouterNoeud](#) (Type noeud)  
*Ajout d'un noeud.*
- void [ajouterArc](#) (Type noeud\_init, Type noeud\_final)  
*Ajout d'un arc.*
- void [supprimerNoeud](#) (Type noeud)  
*Suppression d'un noeud.*
- void [supprimerArc](#) (Type noeud\_init, Type noeud\_final)  
*Suppression d'un arc.*
- bool [contientCycle](#) ()  
*Teste si le graphe contient un cycle.*
- bool [estConnexe](#) ()  
*Teste si le graphe est connexe.*

### 3.26.1 Description détaillée

```
template<class Type>
class Graphe< Type >
```

Gestion du type [Graphe](#).

## Template Parameters

<i>Type</i>	Type de noeud du graphe.
-------------	--------------------------

## Auteur

David

## Version

1.0

## Date

avril 2019

Module permettant l'utilisation d'un graphe orienté générique. Les noeuds du graphe portent cette généricité et peuvent donc être de n'importe quel type de données.

Définition à la ligne 19 du fichier Graphe.hpp.

### 3.26.2 Documentation des fonctions membres

#### 3.26.2.1 ajouterArc()

```
template<class Type>
void Graphe< Type >::ajouterArc (
    Type noeud_init,
    Type noeud_final )
```

Ajout d'un arc.

## Paramètres

<i>noeud_init</i>	le noeud de départ de l'arc.
<i>noeud_final</i>	le noeud d'arrivé de l'arc.

#### 3.26.2.2 ajouterNoeud()

```
template<class Type>
void Graphe< Type >::ajouterNoeud (
    Type noeud )
```

Ajout d'un noeud.

**Paramètres**

<i>noeud</i>	un noeud du graphe.
--------------	---------------------

**3.26.2.3   contientCycle()**

```
template<class Type>
bool Graphe< Type >::contientCycle ( )
```

Teste si le graphe contient un cycle.

**Renvoie**

un booléen.

**3.26.2.4   estConnexe()**

```
template<class Type>
bool Graphe< Type >::estConnexe ( )
```

Teste si le graphe est connexe.

**Renvoie**

un booléen.

**3.26.2.5   supprimerArc()**

```
template<class Type>
void Graphe< Type >::supprimerArc (
    Type noeud_init,
    Type noeud_final )
```

Suppression d'un arc.

**Paramètres**

<i>noeud_init</i>	le noeud de départ de l'arc.
<i>noeud_final</i>	le noeud d'arrivé de l'arc.

## 3.26.2.6 supprimerNoeud()

```
template<class Type>
void Graphe< Type >::supprimerNoeud (
    Type noeud )
```

Suppression d'un noeud.

## Paramètres

<i>noeud</i>	un noeud du graphe.
--------------	---------------------

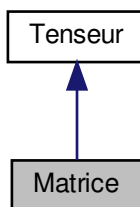
La documentation de cette classe a été générée à partir du fichier suivant :  
— src/deeplearn/archi/Graphe.hpp

## 3.27 Référence de la classe Matrice

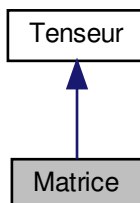
Classe qui crée une matrice.

```
#include <Matrice.hpp>
```

Graphe d'héritage de Matrice :



Graphe de collaboration de Matrice :



## Fonctions membres publiques

- [Matrice](#) (void \*valeur, int l, int c)  
*Constructeur d'une matrice de taille lxc.*
- [Matrice produitMatriciel](#) ([Matrice](#) m1)  
*Méthode qui calcule le produit matriciel entre 2 matrices.*

### 3.27.1 Description détaillée

Classe qui crée une matrice.

#### Auteur

Adrien

#### Version

1.0

#### Date

avril 2019

Classe qui crée un tenseur d'ordre 2 (= matrice).

Définition à la ligne 17 du fichier Matrice.hpp.

### 3.27.2 Documentation des fonctions membres

#### 3.27.2.1 produitMatriciel()

```
Matrice Matrice::produitMatriciel (  
    Matrice m1 )
```

Méthode qui calcule le produit matriciel entre 2 matrices.

#### Paramètres

<i>une</i>	matrice
------------	---------

#### Renvoie

une matrice

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

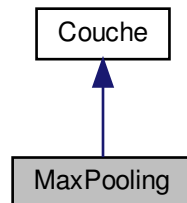
- src/deeplearn/archi/Matrice.hpp
- src/deeplearn/archi/Matrice.cpp

## 3.28 Référence de la classe MaxPooling

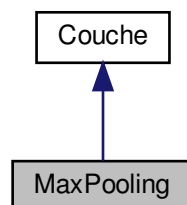
Classe gérant une couche de [MaxPooling](#).

```
#include <MaxPooling.hpp>
```

Graphe d'héritage de MaxPooling :



Graphe de collaboration de MaxPooling :



### Fonctions membres publiques

- `MaxPooling` (`DimTenseur` din, `DimTenseur` dout, std : :string no, int pl\_x, int pl\_y)  
*Constructeur afin d'obtenir une image de taille pool\_x par pool\_y.*
- `MaxPooling` (`DimTenseur` din, `DimTenseur` dout, std : :string no, int pl)  
*Constructeur afin d'obtenir une image de taille pool par pool.*
- `Tenseur propagation` (`Tenseur` t)  
*Méthode permettant la propagation d'une couche à une autre.*
- void `setPoolX` (int pl\_x)  
*Méthode pour fixer le nombre de pixels en dimension x.*
- void `setPoolY` (int pl\_y)  
*Méthode pour fixer le nombre de pixels en dimension y.*
- int `getPoolX` () const  
*Méthode pour obtenir le nombre de pixels fixé pour les images en dimension x.*
- int `getPoolY` () const  
*Méthode pour obtenir le nombre de pixels fixé pour les images en dimension y.*

### 3.28.1 Description détaillée

Classe gérant une couche de [MaxPooling](#).

**Auteur**

Adrien

**Version**

1.0

**Date**

avril 2019

Classe qui va effectuer l'opération de redimensionnement des images par la technique du [MaxPooling](#).

Définition à la ligne 18 du fichier MaxPooling.hpp.

### 3.28.2 Documentation des fonctions membres

#### 3.28.2.1 `getPoolX()`

```
int MaxPooling::getPoolX ( ) const
```

Méthode pour obtenir le nombre de pixels fixé pour les images en dimension x.

**Renvoie**

Le nombre de pixels en dimension x

Définition à la ligne 38 du fichier MaxPooling.cpp.

#### 3.28.2.2 `getPoolY()`

```
int MaxPooling::getPoolY ( ) const
```

Méthode pour obtenir le nombre de pixels fixé pour les images en dimension y.

**Renvoie**

Le nombre de pixels en dimension y

Définition à la ligne 45 du fichier MaxPooling.cpp.

#### 3.28.2.3 `propagation()`

```
Tenseur MaxPooling::propagation (
    Tenseur t ) [virtual]
```

Méthode permettant la propagation d'une couche à une autre.



## Paramètres

<i>t</i>	le tenseur
----------	------------

## Renvoie

Le tenseur à l'étape d'après

Réimplémentée à partir de [Couche](#).

Définition à la ligne 17 du fichier MaxPooling.cpp.

## 3.28.2.4 setPoolX()

```
void MaxPooling::setPoolX (
    int pl_x )
```

Méthode pour fixer le nombre de pixels en dimension x.

## Paramètres

<i>dim</i> ↔ <i>In</i>	La dimension du tenseur d'entrée
---------------------------	----------------------------------

Définition à la ligne 24 du fichier MaxPooling.cpp.

## 3.28.2.5 setPoolY()

```
void MaxPooling::setPoolY (
    int pl_y )
```

Méthode pour fixer le nombre de pixels en dimension y.

## Paramètres

<i>dim</i> ↔ <i>In</i>	La dimension du tenseur de sortie
---------------------------	-----------------------------------

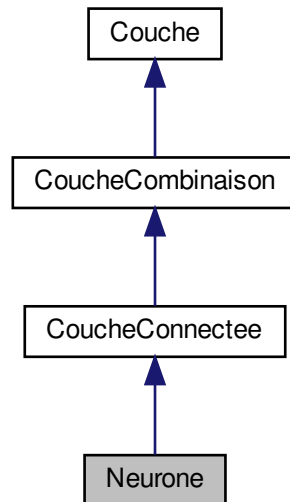
Définition à la ligne 31 du fichier MaxPooling.cpp.

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

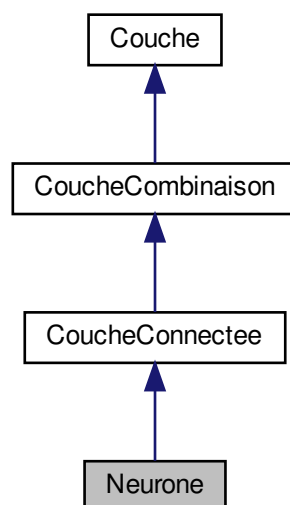
- src/deeplearn/archi/MaxPooling.hpp
- src/deeplearn/archi/MaxPooling.cpp

### 3.29 Référence de la classe Neurone

Graphe d'héritage de Neurone :



Graphe de collaboration de Neurone :



## Fonctions membres publiques

- [Neurone](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no, [Tenseur](#) par)  
*Constructeur d'un neurone.*

### 3.29.1 Description détaillée

Définition à la ligne 17 du fichier Neurone.hpp.

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

- src/deeplearn/archi/Neurone.hpp
- src/deeplearn/archi/Neurone.cpp

## 3.30 Référence de la classe Optimisateur

Gère l'optimisation d'un apprentissage.

```
#include <Optimisateur.hpp>
```

## Fonctions membres publiques

- [Optimisateur](#) ([ReseauNeurones](#) reseauN)  
*Constructeur d'un otpimisateur a partir d'un reseau de neurones.*
- void [minimiser](#) ([Erreur](#) err)  
*Minimise une erreur donnee.*

### 3.30.1 Description détaillée

Gère l'optimisation d'un apprentissage.

#### Auteur

Marion

#### Version

1.0

#### Date

avril 2019

Module permettant le traitement de l'optimisation de l'erreur lors d'un apprentissage.

Définition à la ligne 19 du fichier Optimisateur.hpp.

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

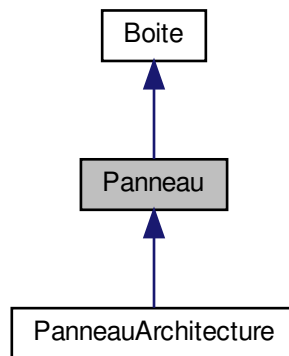
- src/deeplearn/train/Optimisateur.hpp

### 3.31 Référence de la classe Panneau

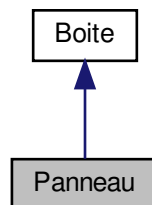
Gestion du type [Panneau](#).

```
#include <Panneau.hpp>
```

Graphe d'héritage de Panneau :



Graphe de collaboration de Panneau :



#### Fonctions membres publiques

- [BoiteArchitecture](#) ()  
*Constructeur d'une boite d'architecture.*
- [BoiteArchitecture](#) ([ReseauNeurones](#) rn)  
*Constructeur prenant en paramètre un reseau de neurones.*
- void [afficher](#) ([ReseauNeurones](#) rn)  
*Méthode permettant l'affichage d'un reseau de neurones.*
- [Panneau](#) ()  
*Constructeur du panneau vide.*
- void [sauvegarderRN](#) ()  
*Méthode permettant la sauvegarde d'un Reseau de Neurones.*
- void [sauvegarderRN](#) (string nomFichier)
- [ReseauNeurones](#) [getReseauNeurones](#) ()  
*getteur permettant d'accéder au Reseau de Neurones en attribut de la classe*

### 3.31.1 Description détaillée

Gestion du type [Panneau](#).

**Auteur**

Samra

**Version**

1.0

**Date**

avril 2019

Définition à la ligne 20 du fichier BoiteArchitecture.hpp.

### 3.31.2 Documentation des fonctions membres

#### 3.31.2.1 afficher()

```
void Panneau::afficher (
    ReseauNeurones rn )
```

Méthode permettant l'affichage d'un reseau de neurones.

**Paramètres**

<i>rn</i>	le réseau de neurones
-----------	-----------------------

#### 3.31.2.2 BoiteArchitecture()

```
Panneau::BoiteArchitecture (
    ReseauNeurones rn )
```

Constucteur prenant en paramètre un reseau de neurones.

**Paramètres**

<i>rn</i>	un réseau de neurones préalablement défini
-----------	--

### 3.31.2.3 getReseauNeurones()

```
ReseauNeurones Panneau::getReseauNeurones ( )
```

getteur permettant d'accéder au Reseau de Neurones en attribut de la classe

#### Renvoie

le Reseau de neurones en attribut de la classe [Panneau](#)

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

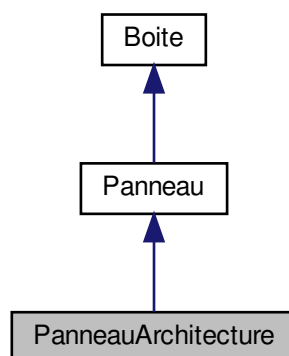
- src/ihm/BoiteArchitecture.hpp
- src/ihm/Panneau.hpp

## 3.32 Référence de la classe PanneauArchitecture

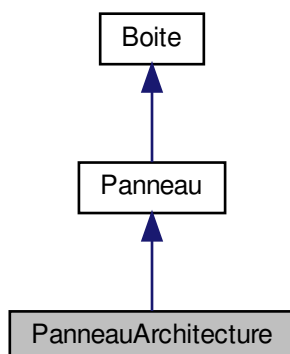
???

```
#include <PanneauArchitecture.hpp>
```

Graphe d'héritage de PanneauArchitecture :



Graphe de collaboration de PanneauArchitecture :



### Fonctions membres publiques

- [PanneauArchitecture](#) ()  
*Constructeur du panneau vide.*
- void [ajouterCouche](#) ()  
*Méthode permettant l'ajout d'une couche.*
- void [ajouterRN](#) ()  
*Méthode permettant l'ajout d'un Réseau de Neurones.*
- void [ajouterLiaison](#) ()  
*Méthode permettant d'ajouter une liaison.*
- void [supprimerCouche](#) ()  
*Méthode permettant la suppression d'une couche.*
- void [supprimerLiaison](#) ()  
*Méthode permettant de supprimer une liaison.*
- void [selectCouche](#) ([Couche](#))  
*Méthode mettant à jour la ccouche sélectionnée.*
- void [selectLiaison](#) ([Couche](#), [Couche](#))  
*Méthode mettant à jour la liaison sélectionnée.*
- [Couche](#) [getCoucheSelectionnee](#) ()  
*Méthode permettant d'accéder à la couche sélectionnée.*
- pair< [Couche](#), [Couche](#) > [getLiaison](#) ()  
*Méthode permettant d'accéder à la liaisonn sélectionnée.*

#### 3.32.1 Description détaillée

???

Auteur

???

Version

1.0

Date

avril 2019

Regroupe les différentes boîtes ( choix couche, RN ,connexion ) dans un même [Panneau](#)

Définition à la ligne 22 du fichier PanneauArchitecture.hpp.

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

— src/min/PanneauArchitecture.hpp

### 3.33 Référence de la classe ParametresApprentissage

Gère les parametres d'apprentissage du réseau de neurones.

```
#include <ParametresApprentissage.hpp>
```

#### Fonctions membres publiques

- [ParametresApprentissage](#) ()  
*Constructeur des parametres d'apprentissage.*
- int [getNbEpoques](#) ()  
*Retourne le nombre d'epoques de l'apprentissage.*
- double [getTauxApprentissage](#) ()  
*Retourne le taux d'apprentissage du reseau.*
- void [setNbEpoques](#) (int ep)  
*Met a jour le nombre d'epoques de l'apprentissage.*
- void [setTauxApprentissage](#) (double ta)  
*Met a jour le taux d'apprentissage du reseau.*

#### 3.33.1 Description détaillée

Gère les parametres d'apprentissage du réseau de neurones.

##### Auteur

Marion

##### Version

1.0

##### Date

avril 2019

Module permettant le choix des parametres de l'apprentissage. Il permet également leur récupération.

Définition à la ligne 16 du fichier ParametresApprentissage.hpp.

#### 3.33.2 Documentation des fonctions membres

##### 3.33.2.1 [getNbEpoques\(\)](#)

```
int ParametresApprentissage::getNbEpoques ( )
```

Retourne le nombre d'epoques de l'apprentissage.

##### Renvoie

le nombre d'epoques



### 3.33.2.2 getTauxApprentissage()

```
double ParametresApprentissage::getTauxApprentissage ( )
```

Retourne le taux d'apprentissage du reseau.

Renvoie

le tauc d'apprentissage \*

### 3.33.2.3 setNbEpoques()

```
void ParametresApprentissage::setNbEpoques (
    int ep )
```

Met a jour le nombre d'epoques de l'apprentissage.

Paramètres

<i>ep</i>	le nombre d'epoques de l'apprentissage
-----------	--

### 3.33.2.4 setTauxApprentissage()

```
void ParametresApprentissage::setTauxApprentissage (
    double ta )
```

Met a jour le taux d'apprentissage du reseau.

Paramètres

<i>ta</i>	le taux d'apprentissage
-----------	-------------------------

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

— src/deeplearn/train/ParametresApprentissage.hpp

## 3.34 Référence de la classe Pretraitement

Gère le pretraitement des donnees necessaires a l'apprentissage.

```
#include <Pretraitement.hpp>
```

## Fonctions membres publiques statiques

- static [Donnees chargerDonnees](#) (std : :string nomDossier)  
*Charge les donnees du dossier.*
- static [ReseauNeurones chargerRN](#) (std : :string nomFichier)  
*Charge le reseau de neurones contenu dans un fichier.*
- static [Tenseur imageToTenseur](#) (std : :string nomFichier)  
*Transforme l'image d'entree en un tenseur.*
- static [Tenseur csvToTenseur](#) (std : :string nomFichier)  
*Transforme un fichier csv en tenseur.*
- static void [normaliser](#) ([Tenseur](#) &t, double minNorm, double maxNorm, double minValeur, double maxValeur)  
*Normalise le tenseur passe en entree en fonction des autres parametres.*
- static void [denormaliser](#) ([Tenseur](#) &t, double minNorm, double maxNorm, double minValeur, double maxValeur)  
*Denormalise le tenseur passe en entree en fonction des autres parametres.*

### 3.34.1 Description détaillée

Gère le pretraitement des donnees necessaires a l'apprentissage.

#### Auteur

Marion

#### Version

1.0

#### Date

avril 2019

Module permettant le pretraitement des donnees necessaires avant l'apprentissage.

Définition à la ligne 18 du fichier Pretraitement.hpp.

### 3.34.2 Documentation des fonctions membres

#### 3.34.2.1 chargerDonnees()

```
static Donnees Pretraitement::chargerDonnees (
    std::string nomDossier ) [static]
```

Charge les donnees du dossier.

#### Paramètres

<i>nomDossier</i>	le nom du dossier dans lequel se trouve les donnees a recuperer
-------------------	---

## 3.34.2.2 chargerRN()

```
static ReseauNeurones Pretraitement::chargerRN (
    std::string nomFichier ) [static]
```

Charge le reseau de neurones contenu dans un fichier.

## Paramètres

<i>nomFichier</i>	le nom du fichier dans lequel se trouve le reseau de neurones a recuperer
-------------------	---

## 3.34.2.3 csvToTenseur()

```
static Tenseur Pretraitement::csvToTenseur (
    std::string nomFichier ) [static]
```

Transforme un fichier csv en tenseur.

## Paramètres

<i>nomFichier</i>	le nom du fichier csv a transformer en tenseur
-------------------	--

## 3.34.2.4 denormaliser()

```
static void Pretraitement::denormaliser (
    Tenseur & t,
    double minNorm,
    double maxNorm,
    double minValeur,
    double maxValeur ) [static]
```

Denormalise le tenseur passe en entree en fonction des autres parametres.

## Paramètres

<i>t</i>	le tenseur a normalise
<i>minNorm</i>	la norme minimale
<i>maxNorm</i>	la norme maximale
<i>minValeur</i>	la valeur minimale
<i>maxValeur</i>	la valeur maximale

#### 3.34.2.5 imageToTenseur()

```
static Tenseur Pretraitement::imageToTenseur (
    std::string nomFichier ) [static]
```

Transforme l'image d'entree en un tenseur.

##### Paramètres

<i>nomFichier</i>	le nom du fichier image a transformer en tenseur
-------------------	--

#### 3.34.2.6 normaliser()

```
static void Pretraitement::normaliser (
    Tenseur & t,
    double minNorm,
    double maxNorm,
    double minValeur,
    double maxValeur ) [static]
```

Normalise le tenseur passe en entree en fonction des autres parametres.

##### Paramètres

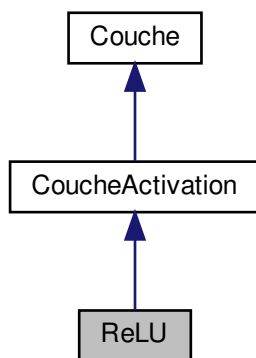
<i>t</i>	le tenseur a normalise
<i>minNorm</i>	la norme minimale
<i>maxNorm</i>	la norme maximale
<i>minValeur</i>	la valeur minimale
<i>maxValeur</i>	la valeur maximale

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

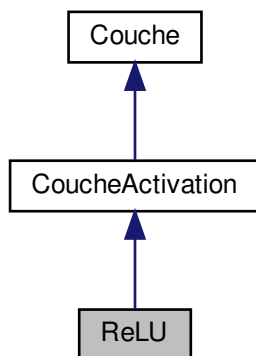
— src/pretraitement/Pretraitement.hpp

### 3.35 Référence de la classe ReLU

Graphe d'héritage de ReLU :



Graphe de collaboration de ReLU :



#### Fonctions membres publiques

- [ReLU](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no)  
*Constructeur de la fonction [ReLU](#).*
- [Tenseur propagation](#) ([Tenseur](#) t)  
*Méthode permettant la propagation d'une couche à une autre.*

#### 3.35.1 Description détaillée

Définition à la ligne 17 du fichier ReLU.hpp.

### 3.35.2 Documentation des fonctions membres

#### 3.35.2.1 propagation()

```
Tenseur ReLU::propagation (
    Tenseur t ) [virtual]
```

Méthode permettant la propagation d'une couche à une autre.

##### Paramètres

<i>t</i>	le tenseur d'entree
----------	---------------------

##### Renvoie

la sortie de la fonction [ReLU](#)

Réimplémentée à partir de [Couche](#).

Définition à la ligne 8 du fichier ReLu.cpp.

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

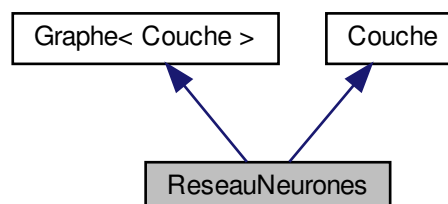
- `src/deeplearn/archi/ReLU.hpp`
- `src/deeplearn/archi/ReLu.cpp`

### 3.36 Référence de la classe ReseauNeurones

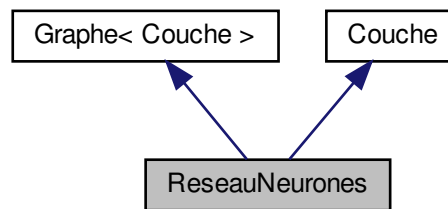
Classe liée à la création du réseau de neurones.

```
#include <ReseauNeurones.hpp>
```

Graphe d'héritage de ReseauNeurones :



Graphe de collaboration de ReseauNeurones :



### Fonctions membres publiques

- `ReseauNeurones ()`  
*Constructeur du réseau vide.*
- `ReseauNeurones (Couche,...)`  
*Constructeur du réseau à partir de couches déjà créées.*
- `Tenseur propagation (Tenseur t)`  
*Méthode permettant la propagation dans un réseau de neurone.*
- `void ajouterCoucheInitiale (Couche c)`  
*Ajout d'une la couche initiale.*
- `void ajouterCoucheFinale (Couche c)`  
*Ajout d'une couche finale.*
- `void supprimerCoucheInitiale (Couche c)`  
*Suppression d'une couche initiale.*
- `void supprimerCoucheFinale (Couche c)`  
*Suppression d'une couche finale.*
- `std::string getNom ()`  
*Méthode pour obtenir le nom du réseau de neurones.*

#### 3.36.1 Description détaillée

Classe liée à la création du réseau de neurones.

##### Auteur

Adrien

##### Version

1.0

##### Date

avril 2019

Cette classe permet de créer un réseau de neurones : on peut ajouter/supprimer des couches/noeuds/arcs.

Définition à la ligne 21 du fichier `ReseauNeurones.hpp`.

### 3.36.2 Documentation des fonctions membres

#### 3.36.2.1 ajouterCoucheFinale()

```
void ReseauNeurones::ajouterCoucheFinale (
    Couche c )
```

Ajout d'une couche finale.

##### Paramètres

<i>c</i>	une couche déjà créée.
----------	------------------------

#### 3.36.2.2 ajouterCoucheInitiale()

```
void ReseauNeurones::ajouterCoucheInitiale (
    Couche c )
```

Ajout d'une la couche initiale.

##### Paramètres

<i>c</i>	une couche déjà créée.
----------	------------------------

#### 3.36.2.3 getNom()

```
string ReseauNeurones::getNom ( )
```

Méthode pour obtenir le nom du reseau de neurones.

##### Renvoie

Le nom du RN

#### 3.36.2.4 propagation()

```
Tenseur ReseauNeurones::propagation (
    Tenseur t ) [virtual]
```

Méthode permettant la propagation dans un reseau de neurone.



## Paramètres

<i>t</i>	l'entree du reseau de neurone
----------	-------------------------------

## Renvoie

la sortie du reseau de neurone

Réimplémentée à partir de [Couche](#).

## 3.36.2.5 supprimerCoucheFinale()

```
void ReseauNeurones::supprimerCoucheFinale (
    Couche c )
```

Suppression d'une couche finale.

## Paramètres

<i>c</i>	une couche présente dans le rése
----------	----------------------------------

## 3.36.2.6 supprimerCoucheInitiale()

```
void ReseauNeurones::supprimerCoucheInitiale (
    Couche c )
```

Suppression d'une couche initiale.

## Paramètres

<i>c</i>	une couche présente dans le réseau.
----------	-------------------------------------

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

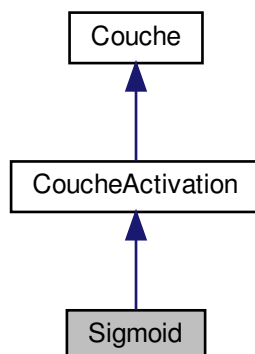
— src/deeplearn/archi/ReseauNeurones.hpp

## 3.37 Référence de la classe Sigmoid

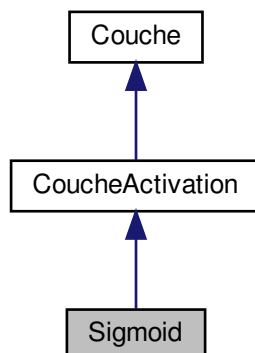
Création de la fonction [ReLU](#).

```
#include <ReLU.hpp>
```

Graphe d'héritage de Sigmoid :



Graphe de collaboration de Sigmoid :



### Fonctions membres publiques

- [Sigmoid](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no)  
*Constructeur d'une fonction sigmoid.*
- [Tenseur propagation](#) ([Tenseur](#) t)  
*Méthode permettant la propagation d'une couche à une autre.*

#### 3.37.1 Description détaillée

Création de la fonction [ReLU](#).

Création d'une couche [Sigmoid](#).

**Auteur**

Adrien

**Version**

1.0

**Date**

avril 2019

Classe permettant la création d'une couche de type [ReLU](#) (= Rectified Linear Unit). C'est à dire  $f(x)=\max(0,x)$

**Auteur**

Adrien

**Version**

1.0

**Date**

avril 2019

Classe permettant la création d'une fonction sigmoid.

Définition à la ligne 17 du fichier Sigmoid.hpp.

### 3.37.2 Documentation des fonctions membres

#### 3.37.2.1 propagation()

```
Tenseur Sigmoid::propagation (  
    Tenseur t ) [virtual]
```

Méthode permettant la propagation d'une couche à une autre.

**Paramètres**

<i>t</i>	le tenseur d'entree
----------	---------------------

**Renvoie**

la sortie de la fonction sigmoid =  $1./1-e^{(-t)}$

Réimplémentée à partir de [Couche](#).

Définition à la ligne 8 du fichier Sigmoid.cpp.

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

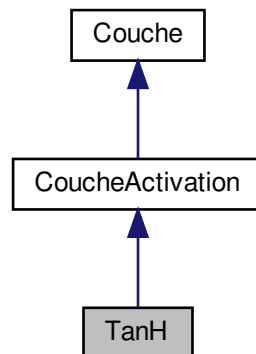
- src/deeplearn/archi/Sigmoid.hpp
- src/deeplearn/archi/Sigmoid.cpp

### 3.38 Référence de la classe TanH

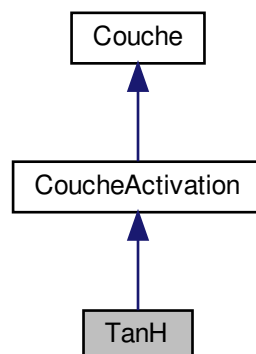
Création d'une fonction tangente hyperbolique.

```
#include <TanH.hpp>
```

Graphe d'héritage de TanH :



Graphe de collaboration de TanH :



#### Fonctions membres publiques

- [TanH](#) ([DimTenseur](#) din, [DimTenseur](#) dout, std : :string no)  
*Constructeur d'une fonction tangente hyperbolique.*
- [Tenseur propagation](#) ([Tenseur](#) t)  
*Méthode permettant la propagation d'une couche à une autre.*

### 3.38.1 Description détaillée

Création d'une fonction tangente hyperbolique.

**Auteur**

Adrien

**Version**

1.0

**Date**

avril 2019

Classe permettant la création d'une fonction tangente hyperbolique Cette classe hérite de la classe [Couche](#)↗  
[Activation](#).

Définition à la ligne 17 du fichier TanH.hpp.

### 3.38.2 Documentation des fonctions membres

#### 3.38.2.1 propagation()

```
Tenseur TanH::propagation (  
    Tenseur t ) [virtual]
```

Méthode permettant la propagation d'une couche à une autre.

**Paramètres**

$t$	le tenseur d'entree
-----	---------------------

**Renvoie**

la sortie de la fonction tanH

Réimplémentée à partir de [Couche](#).

Définition à la ligne 7 du fichier TanH.cpp.

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

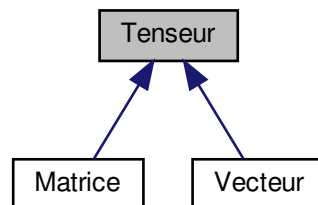
- src/deeplearn/archi/TanH.hpp
- src/deeplearn/archi/TanH.cpp

### 3.39 Référence de la classe Tenseur

Classe liée à la manipulation de tenseurs.

```
#include <Tenseur.hpp>
```

Graphe d'héritage de Tenseur :



#### Fonctions membres publiques

- [Tenseur](#) (int dim)  
*Constructeur d'un tenseur dont la taille est fixée grâce à des entiers.*
- [Tenseur](#) (void \*val, [DimTenseur](#) di)  
*Constructeur d'un tenseur dont la taille est fixée grâce à un objet [DimTenseur](#).*
- void [initValeurGaussienne](#) ()  
*Initialisation du tenseur selon une loi gaussienne.*
- void [initValeurNulle](#) ()  
*Initialisation du tenseur avec des valeurs nulles.*
- void [initValeurUnif](#) ()  
*Initialisation du tenseur selon une loi uniforme.*
- void [setValeur](#) (void \*vl)  
*Méthode pour fixer la valeur du [Tenseur](#).*
- void [setDim](#) ([DimTenseur](#) di)  
*Méthode pour fixer la dimension du [Tenseur](#).*
- void \* [getValeur](#) ()  
*Méthode pour obtenir la valeur du [Tenseur](#).*
- [Tenseur produitTermeATerme](#) ([Tenseur](#) t1)  
*Méthode qui calcule le produit terme à terme entre deux tenseurs.*
- [DimTenseur](#) [getDim](#) ()  
*Méthode pour obtenir la dimension du [Tenseur](#).*

#### 3.39.1 Description détaillée

Classe liée à la manipulation de tenseurs.

Auteur

Adrien

Version

1.0

Date

avril 2019

Cette classe permet de créer un tenseur de la taille souhaitée. On peut initialiser un tenseur de trois façons différentes (uniforme,nulle,gaussienne).

Définition à la ligne 17 du fichier Tenseur.hpp.

### 3.39.2 Documentation des constructeurs et destructeur

#### 3.39.2.1 Tenseur()

```
Tenseur::Tenseur (
    int dim )
```

Constructeur d'un tenseur dont la taille est fixée grâce à des entiers.

##### Paramètres

<i>dim</i>	suite de dimensions
------------	---------------------

### 3.39.3 Documentation des fonctions membres

#### 3.39.3.1 getDim()

```
DimTenseur Tenseur::getDim ( )
```

Méthode pour obtenir la dimension du [Tenseur](#).

##### Renvoie

La dimension du [Tenseur](#)

Définition à la ligne 51 du fichier Tenseur.cpp.

#### 3.39.3.2 getValeur()

```
int Tenseur::getValeur ( )
```

Méthode pour obtenir la valeur du [Tenseur](#).

##### Renvoie

La valeur du [Tenseur](#)

Définition à la ligne 44 du fichier Tenseur.cpp.

#### 3.39.3.3 produitTermeATerme()

```
Tenseur Tenseur::produitTermeATerme (
    Tenseur t1 )
```

Méthode qui calcule le produit terme à terme entre deux tenseurs.

**Paramètres**

<i>un</i>	tenseur
-----------	---------

**Renvoie**

un tenseur

**3.39.3.4 setDim()**

```
void Tenseur::setDim (
    DimTenseur di )
```

Méthode pour fixer la dimension du [Tenseur](#).

**Paramètres**

<i>vl</i>	La dimension du tenseur
-----------	-------------------------

Définition à la ligne 37 du fichier Tenseur.cpp.

**3.39.3.5 setValeur()**

```
void Tenseur::setValeur (
    void * vl )
```

Méthode pour fixer la valeur du [Tenseur](#).

**Paramètres**

<i>vl</i>	La valeur du tenseur
-----------	----------------------

Définition à la ligne 30 du fichier Tenseur.cpp.

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

- src/deeplearn/archi/Tenseur.hpp
- src/deeplearn/archi/Tenseur.cpp

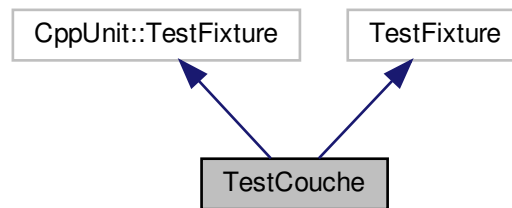
**3.40 Référence de la classe TestCouche**

Test des méthodes de la classe [Couche](#).

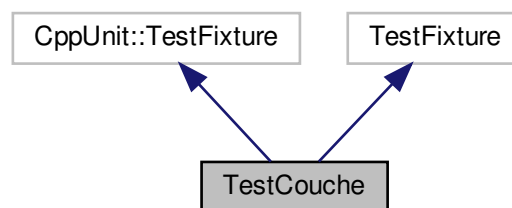
```
#include <TestCouche.hpp>
```



Graphe d'héritage de TestCouche :



Graphe de collaboration de TestCouche :



## Fonctions membres publiques

- void **setUp** ()
- void **tearDown** ()
- void **testGetDimInput** ()
- void **testGetDimOutput** ()
- void **testSetDimInput** ()
- void **testSetDimOutput** ()
- void **testPropagation** ()
- void **testDerivee** ()
- void **setUp** ()  
*Initialiser les variables.*
- void **tearDown** ()  
*Supprimer les variables.*
- void **testGetDimInput** ()  
*Vérifier que la propagation d'une couche à une autre se déroule normalement.*
- void **testGetDimOutput** ()
- void **testSetDimInput** ()
- void **testSetDimOutput** ()
- void **testPropagation** ()
- void **testDerivee** ()  
*Vérifier que la dérivation d'une couche s'effectue correctement.*

### 3.40.1 Description détaillée

Test des méthodes de la classe [Couche](#).

Auteur

Adrien

Version

1.0

Date

avril 2019

Définition à la ligne 6 du fichier TestCouche.cpp.

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

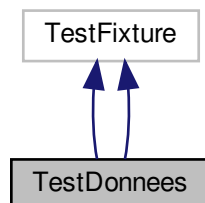
- src/deeplearn/archi/TestCouche.cpp
- src/deeplearn/archi/TestCouche.hpp

## 3.41 Référence de la classe TestDonnees

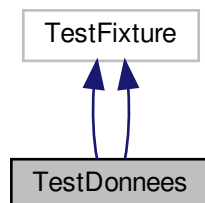
Test des méthodes de la classe [Donnees](#).

```
#include <TestDonnees.hpp>
```

Graphe d'héritage de TestDonnees :



Graphe de collaboration de TestDonnees :



### Fonctions membres publiques

- void `setUp()`  
*Initialiser les variables.*
- void `tearDown()`  
*Supprimer les variables.*
- void `testAjouterDonnees()`  
*Vérifier que les données s'ajoutent bien.*
- void `testMelanger()`  
*Vérifier que les données se mélangent bien.*

#### 3.41.1 Description détaillée

Test des méthodes de la classe `Donnees`.

##### Auteur

Adrien

##### Version

1.0

##### Date

avril 2019

Définition à la ligne 5 du fichier `TestDonnees.cpp`.

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

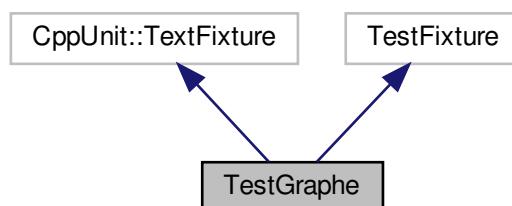
- `src/deeplearn/train/TestDonnees.cpp`
- `src/deeplearn/train/TestDonnees.hpp`

## 3.42 Référence de la classe TestGraphe

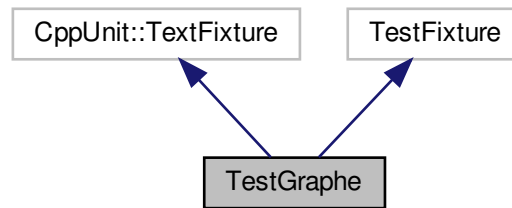
Test des méthodes de la classe `Graphe`.

```
#include <TestGraphe.hpp>
```

Graphe d'héritage de `TestGraphe` :



Graphe de collaboration de TestGraphe :



### Fonctions membres publiques

- void **setUp** ()
- void **tearDown** ()
- void **testAjouterNoeud** ()
- void **testAjouterArc** ()
- void **testSupprimerNoeud** ()
- void **testSupprimerArc** ()
- void **testContientCycle** ()
- void **testEstConnexe** ()

#### 3.42.1 Description détaillée

Test des methodes de la classe [Graphe](#).

##### Auteur

Coralie

##### Version

1.0

##### Date

avril 2019

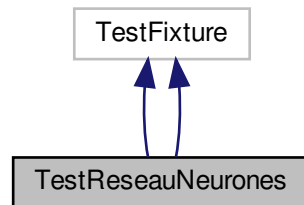
Définition à la ligne 5 du fichier `TestGraphe.cpp`.

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

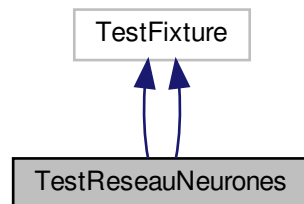
- `src/deeplearn/archi/TestGraphe.cpp`
- `src/deeplearn/archi/TestGraphe.hpp`

### 3.43 Référence de la classe TestReseauNeurones

Graphe d'héritage de TestReseauNeurones :



Graphe de collaboration de TestReseauNeurones :



#### Fonctions membres publiques

- void **setUp** ()
- void **tearDown** ()
- void **testGetCoucheInitiale** ()
- void **testGetCoucheFinale** ()
- void **testAjouterCoucheInitiale** ()
- void **testAjouterCoucheFinale** ()
- void **testSupprimerCoucheInitiale** ()
- void **testSupprimerCoucheFinale** ()
- void **setUp** ()
- void **tearDown** ()
- void **testGetCoucheInitiale** ()
- void **testGetCoucheFinale** ()
- void **testAjouterCoucheInitiale** ()
- void **testAjouterCoucheFinale** ()
- void **testSupprimerCoucheInitiale** ()
- void **testSupprimerCoucheFinale** ()

#### 3.43.1 Description détaillée

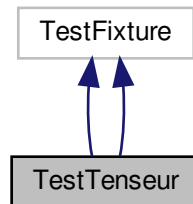
Définition à la ligne 6 du fichier `TestReseauNeurones.cpp`.

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

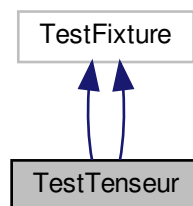
- `src/deeplearn/archi/TestReseauNeurones.cpp`
- `src/deeplearn/archi/TestReseauNeurones.hpp`

### 3.44 Référence de la classe TestTenseur

Graphe d'héritage de TestTenseur :



Graphe de collaboration de TestTenseur :



#### Fonctions membres publiques

- void **setUp** ()
- void **tearDown** ()
- void **testGetValeur** ()
- void **testInitValeurGaussienne** ()
- void **testInitValeurUnif** (int valeur)
- void **testInitValeurNulle** ()
- void **setUp** ()
- void **tearDown** ()
- void **testGetValeur** ()
- void **testInitValeurGaussienne** ()
- void **testInitValeurUnif** (int valeur)
- void **testInitValeurNulle** ()

#### 3.44.1 Description détaillée

Définition à la ligne 6 du fichier TestTenseur.cpp.

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

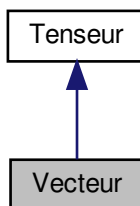
- src/deeplearn/archi/TestTenseur.cpp
- src/deeplearn/archi/TestTenseur.hpp

## 3.45 Référence de la classe Vecteur

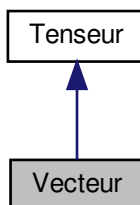
Classe qui crée un vecteur.

```
#include <Vecteur.hpp>
```

Graphe d'héritage de Vecteur :



Graphe de collaboration de Vecteur :



### Fonctions membres publiques

- [Vecteur](#) (void \*vl, int l)  
*Constructeur d'un vecteur de longueur l.*
- double [produitScalaire](#) ([Vecteur](#) v1)  
*Méthode qui calcule le produit scalaire entre 2 vecteurs.*

### 3.45.1 Description détaillée

Classe qui crée un vecteur.

Auteur

Adrien

**Version**

1.0

**Date**

avril 2019

Classe qui crée un tenseur d'ordre 1 (= vecteur)

Définition à la ligne 17 du fichier Vecteur.hpp.

### 3.45.2 Documentation des fonctions membres

#### 3.45.2.1 produitScalaire()

```
double Vecteur::produitScalaire (
    Vecteur v1 )
```

Méthode qui calcule le produit scalaire entre 2 vecteurs.

**Paramètres**

<i>un</i>	vecteur
-----------	---------

**Renvoie**

un réel

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

- src/deeplearn/archi/Vecteur.hpp
- src/deeplearn/archi/Vecteur.cpp



# Index

- afficher
  - Couche, [26](#)
  - Panneau, [59](#)
- ajouterArc
  - Graphe, [49](#)
- ajouterChoix
  - BoiteChoixMultiple, [18](#)
- ajouterCoucheFinale
  - ReseauNeurones, [70](#)
- ajouterCoucheInitiale
  - ReseauNeurones, [70](#)
- ajouterDonnee
  - Donnees, [41](#)
- ajouterDonnees
  - Donnees, [41](#)
- ajouterNoeud
  - Graphe, [49](#)
- Apprentissage, [5](#)
  - getDonnees, [6](#)
  - getErreur, [6](#)
  - getOptimisateur, [6](#)
  - getParam, [6](#)
  - getRN, [7](#)
  - setDonnees, [7](#)
  - setParam, [7](#)
- Boite, [8](#)
- BoiteArchitecture, [9](#)
  - Panneau, [59](#)
- BoiteChoixCouche, [9](#)
- BoiteChoixDonnées, [11](#)
- BoiteChoixDonnees, [10](#)
- BoiteChoixErreur, [12](#)
- BoiteChoixFichier, [14](#)
  - BoiteChoixFichier, [15](#)
  - getNomFichier, [16](#)
- BoiteChoixMultiple, [16](#)
  - ajouterChoix, [18](#)
  - BoiteChoixMultiple, [17](#)
  - getValeurSectionnee, [18](#)
  - supprimerChoix, [18](#)
- BoiteChoixReseauNeurones, [19](#)
- BoiteConnexionCouche, [20](#)
  - BoiteConnexionCouche, [21](#)
  - getCoucheFinale, [22](#)
  - getCoucheInitiale, [22](#)
  - setCoucheFinale, [22](#)
  - setCoucheInitiale, [22](#)
- BoiteParametrage, [23](#)
  - BoiteParametrage, [24](#)
  - getParametrage, [24](#)
- chargerDonnees
  - Pretraitement, [64](#)
- chargerRN
  - Pretraitement, [64](#)
- contientCycle
  - Graphe, [50](#)
- Couche, [25](#)
  - afficher, [26](#)
  - derivee, [26](#)
  - getDimInput, [26](#)
  - getDimOutput, [26](#)
  - getNom, [27](#)
  - propagation, [27](#)
  - setDimInput, [27](#)
  - setDimOutput, [28](#)
- CoucheActivation, [28](#)
- CoucheCombinaison, [30](#)
  - getParams, [31](#)
- CoucheConnectée, [33](#)
- CoucheConnectee, [32](#)
  - propagation, [33](#)
- CoucheConvulsive, [34](#)
  - CoucheConvulsive, [36](#)
  - propagation, [36](#)
  - setParams, [36](#)
- csvToTenseur
  - Pretraitement, [65](#)
- denormaliser
  - Pretraitement, [65](#)
- derivee
  - Couche, [26](#)
  - Erreur, [43](#)
- DimTenseur, [37](#)
  - getDim, [37](#)
  - getOrdre, [38](#)
- Donnee, [38](#)
  - getEntree, [39](#)
  - getSortie, [39](#)
  - setEntree, [39](#)
  - setSortie, [40](#)
- Donnees, [40](#)
  - ajouterDonnee, [41](#)
  - ajouterDonnees, [41](#)
  - getDonnee, [41](#)
  - getDonnees, [42](#)
  - getNbDonnees, [42](#)

- Erreur, [43](#)
  - derivee, [43](#)
  - eval, [44](#)
- ErreurEntropieCroisee, [44](#)
- ErreurL1, [46](#)
- ErreurQuadratique, [47](#)
- estConnexe
  - Graphe, [50](#)
- eval
  - Erreur, [44](#)
- getCoucheFinale
  - BoiteConnexionCouche, [22](#)
- getCoucheInitiale
  - BoiteConnexionCouche, [22](#)
- getDim
  - DimTenseur, [37](#)
  - Tenseur, [77](#)
- getDimInput
  - Couche, [26](#)
- getDimOutput
  - Couche, [26](#)
- getDonnee
  - Donnees, [41](#)
- getDonnees
  - Apprentissage, [6](#)
  - Donnees, [42](#)
- getEntree
  - Donnee, [39](#)
- getErreur
  - Apprentissage, [6](#)
- getNbDonnees
  - Donnees, [42](#)
- getNbEpoques
  - ParametresApprentissage, [62](#)
- getNom
  - Couche, [27](#)
  - ReseauNeurones, [70](#)
- getNomFichier
  - BoiteChoixFichier, [16](#)
- getOptimisateur
  - Apprentissage, [6](#)
- getOrdre
  - DimTenseur, [38](#)
- getParam
  - Apprentissage, [6](#)
- getParametrage
  - BoiteParametrage, [24](#)
- getParams
  - CoucheCombinaison, [31](#)
- getPoolX
  - MaxPooling, [54](#)
- getPoolY
  - MaxPooling, [54](#)
- getReseauNeurones
  - Panneau, [59](#)
- getRN
  - Apprentissage, [7](#)
- getSortie
  - Donnee, [39](#)
- getTauxApprentissage
  - ParametresApprentissage, [62](#)
- getValeur
  - Tenseur, [77](#)
- getValeurSectionnee
  - BoiteChoixMultiple, [18](#)
- Graphe
  - ajouterArc, [49](#)
  - ajouterNoeud, [49](#)
  - contientCycle, [50](#)
  - estConnexe, [50](#)
  - supprimerArc, [50](#)
  - supprimerNoeud, [50](#)
- Graphe< Type >, [48](#)
- imageToTenseur
  - Pretraitement, [65](#)
- Matrice, [51](#)
  - produitMatriciel, [52](#)
- MaxPooling, [53](#)
  - getPoolX, [54](#)
  - getPoolY, [54](#)
  - propagation, [54](#)
  - setPoolX, [55](#)
  - setPoolY, [55](#)
- Neurone, [56](#)
- normaliser
  - Pretraitement, [66](#)
- Optimisateur, [57](#)
- Panneau, [58](#)
  - afficher, [59](#)
  - BoiteArchitecture, [59](#)
  - getReseauNeurones, [59](#)
- PanneauArchitecture, [60](#)
- ParametresApprentissage, [62](#)
  - getNbEpoques, [62](#)
  - getTauxApprentissage, [62](#)
  - setNbEpoques, [63](#)
  - setTauxApprentissage, [63](#)
- Pretraitement, [63](#)
  - chargerDonnees, [64](#)
  - chargerRN, [64](#)
  - csvToTenseur, [65](#)
  - denormaliser, [65](#)
  - imageToTenseur, [65](#)
  - normaliser, [66](#)
- produitMatriciel
  - Matrice, [52](#)
- produitScalaire
  - Vecteur, [86](#)
- produitTermeATerme
  - Tenseur, [77](#)
- propagation
  - Couche, [27](#)

- CoucheConnectee, [33](#)
- CoucheConvulsive, [36](#)
- MaxPooling, [54](#)
- ReLU, [68](#)
- ReseauNeurones, [70](#)
- Sigmoid, [73](#)
- TanH, [75](#)
- ReLU, [67](#)
  - propagation, [68](#)
- ReseauNeurones, [68](#)
  - ajouterCoucheFinale, [70](#)
  - ajouterCoucheInitiale, [70](#)
  - getNom, [70](#)
  - propagation, [70](#)
  - supprimerCoucheFinale, [71](#)
  - supprimerCoucheInitiale, [71](#)
- setCoucheFinale
  - BoiteConnexionCouche, [22](#)
- setCoucheInitiale
  - BoiteConnexionCouche, [22](#)
- setDim
  - Tenseur, [78](#)
- setDimInput
  - Couche, [27](#)
- setDimOutput
  - Couche, [28](#)
- setDonnees
  - Apprentissage, [7](#)
- setEntree
  - Donnee, [39](#)
- setNbEpoques
  - ParametresApprentissage, [63](#)
- setParam
  - Apprentissage, [7](#)
- setParams
  - CoucheConvulsive, [36](#)
- setPoolX
  - MaxPooling, [55](#)
- setPoolY
  - MaxPooling, [55](#)
- setSortie
  - Donnee, [40](#)
- setTauxApprentissage
  - ParametresApprentissage, [63](#)
- setValeur
  - Tenseur, [78](#)
- Sigmoid, [71](#)
  - propagation, [73](#)
- supprimerArc
  - Graphe, [50](#)
- supprimerChoix
  - BoiteChoixMultiple, [18](#)
- supprimerCoucheFinale
  - ReseauNeurones, [71](#)
- supprimerCoucheInitiale
  - ReseauNeurones, [71](#)
- supprimerNoeud
  - Graphe, [50](#)
- TanH, [74](#)
  - propagation, [75](#)
- Tenseur, [76](#)
  - getDim, [77](#)
  - getValeur, [77](#)
  - produitTermeATerme, [77](#)
  - setDim, [78](#)
  - setValeur, [78](#)
  - Tenseur, [77](#)
- TestCouche, [78](#)
- TestDonnees, [80](#)
- TestGraphe, [81](#)
- TestReseauNeurones, [83](#)
- TestTenseur, [84](#)
- Vecteur, [85](#)
  - produitScalaire, [86](#)