

Neural Network & Backpropagation

Documentation

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Archi.DrawNN Class Reference	7
4.2	FonctionActivation Class Reference	7
4.2.1	Detailed Description	8
4.2.2	Constructor & Destructor Documentation	8
4.2.2.1	FonctionActivation()	8
4.2.2.2	FonctionActivation(EnumFonctionActivation fonctionActivation)	8
4.2.3	Member Function Documentation	8
4.2.3.1	getValeurActivation(double sum, double k=0) const	8
4.2.3.2	getValeurDerivee(double sum, double k=0) const	8
4.2.3.3	setFonctionActivation(EnumFonctionActivation fonctionActivation)	9
4.3	InputLayer Class Reference	9
4.3.1	Constructor & Destructor Documentation	10
4.3.1.1	InputLayer(int)	10
4.3.2	Member Function Documentation	10
4.3.2.1	fire(std::vector< double >, double)	10

4.4	Archi.Layer Class Reference	11
4.5	Layer Class Reference	11
4.5.1	Constructor & Destructor Documentation	12
4.5.1.1	Layer()	12
4.5.1.2	Layer(TypeLayer, int, int)	12
4.5.2	Member Function Documentation	12
4.5.2.1	fire(std::vector< double >, double)	12
4.5.2.2	getInput()	13
4.5.2.3	getNbNeurones()	13
4.5.2.4	getNeurone(int index)	13
4.5.2.5	printWeight()	13
4.6	Archi.NeuralNetwork Class Reference	14
4.7	Archi.Neuron Class Reference	14
4.8	Neurone Class Reference	15
4.8.1	Constructor & Destructor Documentation	16
4.8.1.1	Neurone()	16
4.8.1.2	Neurone(int taille)	16
4.8.1.3	Neurone(int taille, std::vector< double > *x)	16
4.8.2	Member Function Documentation	16
4.8.2.1	aleaWeights()	16
4.8.2.2	derive_activate(double sum, double k) const	16
4.8.2.3	fire(std::vector< double > x, double k) const	17
4.8.2.4	fw_activate(double sum, double k) const	17
4.8.2.5	fw_sum(std::vector< double > x) const	17
4.8.2.6	getNbPoids()	18
4.8.2.7	getWeight()	18
4.8.2.8	learn(std::vector< double > x, double o, double k, double mu)	18
4.8.2.9	printWeight()	18
4.9	NeuroneB Class Reference	19
4.9.1	Constructor & Destructor Documentation	20

4.9.1.1	NeuroneB(int taille)	20
4.9.1.2	NeuroneB(int taille, std::vector< double > *x, double biais_w)	20
4.9.2	Member Function Documentation	20
4.9.2.1	aleaWeights()	20
4.9.2.2	fw_sum(std::vector< double > x) const	20
4.9.2.3	getNbPoids()	21
4.9.2.4	learn(std::vector< double > x, double o, double k, double mu)	21
4.9.2.5	printWeight()	21
4.10	Option Class Reference	21
4.10.1	Constructor & Destructor Documentation	22
4.10.1.1	Option()	22
4.10.2	Member Function Documentation	22
4.10.2.1	getOptionID() const	22
4.10.2.2	getOptionInt() const	22
4.10.2.3	getOptionRac() const	22
4.10.2.4	getOptionType() const	22
4.10.2.5	print(std::ostream &os) const	22
4.10.2.6	setOptionInt(std::string &intitul)	23
4.10.2.7	setOptionRac(std::string &rac)	23
4.10.2.8	setOptionType(std::string &t)	23
4.11	OptionTab Class Reference	23
4.11.1	Constructor & Destructor Documentation	24
4.11.1.1	OptionTab()	24
4.11.2	Member Function Documentation	24
4.11.2.1	addOption(const Option &opt)	24
4.11.2.2	getArgument(const std::string &opt) const	24
4.11.2.3	getIntitul(const std::string &opt) const	24
4.11.2.4	getOptionID(const std::string &opt) const	24
4.11.2.5	getRaccour(const std::string &opt) const	25
4.11.2.6	printOptions() const	25

4.12	Reseau Class Reference	25
4.12.1	Constructor & Destructor Documentation	25
4.12.1.1	Reseau(int, std::vector< int >, double, double)	25
4.12.2	Member Function Documentation	26
4.12.2.1	backPropagation(std::vector< double > output)	26
4.12.2.2	fire_all(std::vector< double > input)	26
4.12.2.3	learn(std::vector< std::vector< std::vector< double > > > jeuxTest)	26
4.12.2.4	printWeight()	26
4.13	Settings Class Reference	27
4.13.1	Constructor & Destructor Documentation	27
4.13.1.1	Settings(char const *fileName)	27
4.13.2	Member Function Documentation	27
4.13.2.1	getArchi()	27
4.13.2.2	getDifferentOutputs()	27
4.13.2.3	getNbrHiddenLayers()	28
4.14	Type Class Reference	28
4.14.1	Constructor & Destructor Documentation	28
4.14.1.1	Type()	28
4.14.1.2	Type(std::string type)	28
4.14.2	Member Function Documentation	28
4.14.2.1	affiche() const	28
4.14.2.2	getType() const	29
4.14.2.3	setType(std::string type)	29
5	File Documentation	31
5.1	main.cpp File Reference	31
5.1.1	Detailed Description	32
5.1.2	Function Documentation	32
5.1.2.1	main(int argc, char **argv)	32
Index		33

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Archi.DrawNN	7
FonctionActivation	7
Archi.Layer	11
Layer	11
InputLayer	9
Archi.NeuralNetwork	14
Archi.Neuron	14
Neurone	15
NeuroneB	19
Option	21
OptionTab	23
Reseau	25
Settings	27
Type	28

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Archi.DrawNN	7
FonctionActivation	7
InputLayer	9
Archi.Layer	11
Layer	11
Archi.NeuralNetwork	14
Archi.Neuron	14
Neurone	15
NeuroneB	19
Option	21
OptionTab	23
Reseau	25
Settings	27
Type	28

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

main.cpp	Programme de test	31
Common/ fonctions.h	??
Layer/ InputLayer.h	??
Layer/ Layer.h	??
Neurone/ FonctionActivation.h	??
Neurone/ Neurone.h	??
Neurone/ NeuroneB.h	??
Option/ Option.h	??
Option/ OptionTab.h	??
Option/ Type.h	??
Reseau/ Reseau.h	??
Settings/ Settings.h	??
testfiles/ fonctions.h	??

Chapter 4

Class Documentation

4.1 Archi.DrawNN Class Reference

Public Member Functions

- def **__init__** (self, neural_network)
- def **draw** (self)

Public Attributes

- **neural_network**

The documentation for this class was generated from the following file:

- Archi.py

4.2 FonctionActivation Class Reference

```
#include <FonctionActivation.h>
```

Public Types

- enum **EnumFonctionActivation** {
 IDENTITY, **BINARYSTEP**, **SIGMOID**, **TAN**,
 SIN, **ELU** }

Public Member Functions

- [FonctionActivation](#) ()
- [FonctionActivation](#) (EnumFonctionActivation fonctionActivation)
- void [setFonctionActivation](#) (EnumFonctionActivation fonctionActivation)
- double [getValeurActivation](#) (double sum, double k=0) const
- double [getValeurDerivee](#) (double sum, double k=0) const

4.2.1 Detailed Description

L'objectif de cette classe est d'être appelée pour l'activation et donc changer plus facilement la fonction d'activation pour le test.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `FonctionActivation::FonctionActivation ()`

Constructeur par défauts [FonctionActivation](#)

Parameters

<i>fonctionActivation</i>	Fonction activation à utiliser parmi l'énumération
---------------------------	--

4.2.2.2 `FonctionActivation::FonctionActivation (EnumFonctionActivation fonctionActivation)`

Constructeur [FonctionActivation](#)

Parameters

<i>fonctionActivation</i>	Fonction activation à utiliser parmi l'énumération
---------------------------	--

4.2.3 Member Function Documentation

4.2.3.1 `double FonctionActivation::getValeurActivation (double x, double k = 0) const`

`getValeurActivation`

Parameters

<i>x</i>	Valeur de la variable
<i>k</i>	Constante de changement

Returns

Valeur d'activation

4.2.3.2 `double FonctionActivation::getValeurDerivee (double x, double k = 0) const`

`getValeurDerivee`

Parameters

x	Valeur de la variable
k	Constante de changement

Returns

Valeur de la dérivée

4.2.3.3 void FonctionActivation::setFonctionActivation (EnumFonctionActivation *fonctionActivation*)

[FonctionActivation::setFonctionActivation](#)

Parameters

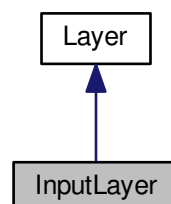
<i>fonctionActivation</i>	Fonction activation à utiliser parmi l'énumération
---------------------------	--

The documentation for this class was generated from the following files:

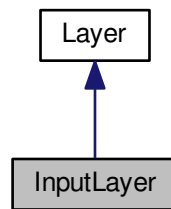
- Neurone/FonctionActivation.h
- Neurone/FonctionActivation.cpp

4.3 InputLayer Class Reference

Inheritance diagram for InputLayer:



Collaboration diagram for InputLayer:



Public Member Functions

- [InputLayer](#) (int)
- `std::vector< double >` [fire](#) (`std::vector< double >`, double)

Additional Inherited Members

4.3.1 Constructor & Destructor Documentation

4.3.1.1 `InputLayer::InputLayer (int taille)`

Constructeur par taille [InputLayer::InputLayer](#)

Parameters

<i>taille</i>	Nombre de neurones dans le layer
---------------	----------------------------------

4.3.2 Member Function Documentation

4.3.2.1 `std::vector< double > InputLayer::fire (std::vector< double > input, double k)` [virtual]

Méthode de propagation en avant [InputLayer::fire](#)

Parameters

<i>input</i>	Vecteur en entrées
<i>k</i>	Coefficient de sigmoid

Returns

Valeur d'activation

Reimplemented from [Layer](#).

The documentation for this class was generated from the following files:

- Layer/InputLayer.h
- Layer/InputLayer.cpp

4.4 Archi.Layer Class Reference

Public Member Functions

- def **__init__** (self, network, number_of_neurons, number_of_neurons_in_widest_layer)
- def **draw** (self, layerType=0)

Public Attributes

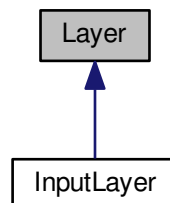
- **vertical_distance_between_layers**
- **horizontal_distance_between_neurons**
- **neuron_radius**
- **number_of_neurons_in_widest_layer**
- **previous_layer**
- **y**
- **neurons**

The documentation for this class was generated from the following file:

- Archi.py

4.5 Layer Class Reference

Inheritance diagram for Layer:



Public Types

- enum **TypeLayer** { **INPUT**, **OUTPUT**, **HIDDEN** }

Public Member Functions

- [Layer](#) ()
- [Neurone](#) * [getNeurone](#) (int index)
- virtual std::vector< double > [fire](#) (std::vector< double >, double)
- [Layer](#) (TypeLayer, int, int)
- int [getNbNeurones](#) ()
- std::vector< double > [getInput](#) ()
- std::vector< double > [getOutput](#) ()
- void [printWeight](#) ()

Protected Attributes

- double **k**
- int **nbNeurone**
- std::vector< [Neurone](#) * > **membres**
- std::vector< double > **input**
- std::vector< double > **output**
- TypeLayer **type**

4.5.1 Constructor & Destructor Documentation

4.5.1.1 [Layer::Layer](#) ()

Constructeur par défauts [Layer::Layer](#)

4.5.1.2 [Layer::Layer](#) (TypeLayer *type*, int *nbneur*, int *nbinput*)

Constructeur avec arguments [Layer::Layer](#)

Parameters

<i>type</i>	Type de layer
<i>nbneur</i>	Nombre de neurone
<i>nbinput</i>	Nombre d'entrée par neurone

4.5.2 Member Function Documentation

4.5.2.1 std::vector< double > [Layer::fire](#) (std::vector< double > *input*, double *k*) [virtual]

Propagation en avant [Layer::fire](#)

Parameters

<i>input</i>	Vecteur en entrées
<i>k</i>	Coefficient de sigmoid

Returns

Vecteur des valeurs d'activations

Reimplemented in [InputLayer](#).

4.5.2.2 `std::vector< double > Layer::getInput ()`

[Layer::getInput](#)

Returns

Retourne le vecteur d'entrées

4.5.2.3 `int Layer::getNbNeurones ()`

[Layer::getNbNeurones](#)

Returns

Nombre de neurones

4.5.2.4 `Neurone * Layer::getNeurone (int index)`

[Layer::getNeurone](#)

Parameters

<i>index</i>	index
--------------	-------

Returns

[Neurone](#)

4.5.2.5 `void Layer::printWeight ()`

Affiche les poids [Layer::printWeight](#)

The documentation for this class was generated from the following files:

- Layer/Layer.h
- Layer/Layer.cpp

4.6 Archi.NeuralNetwork Class Reference

Public Member Functions

- def **__init__** (self, number_of_neurons_in_widest_layer)
- def **add_layer** (self, number_of_neurons)
- def **draw** (self)

Public Attributes

- **number_of_neurons_in_widest_layer**
- **layers**
- **layertype**

The documentation for this class was generated from the following file:

- Archi.py

4.7 Archi.Neuron Class Reference

Public Member Functions

- def **__init__** (self, x, y)
- def **draw** (self, neuron_radius)

Public Attributes

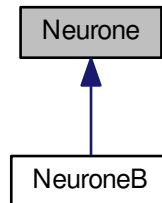
- **x**
- **y**

The documentation for this class was generated from the following file:

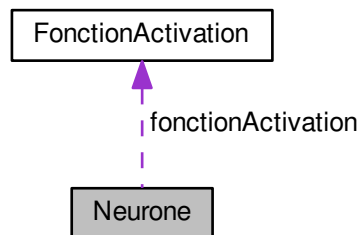
- Archi.py

4.8 Neurone Class Reference

Inheritance diagram for Neurone:



Collaboration diagram for Neurone:



Public Member Functions

- `Neurone` ()
- `Neurone` (int taille)
- `Neurone` (int taille, std::vector< double > *x)
- virtual double `fw_sum` (std::vector< double > x) const
- double `fw_activate` (double sum, double k) const
- virtual double `fire` (std::vector< double > x, double k) const
- virtual void `learn` (std::vector< double > x, double o, double k, double mu)
- virtual void `printWeight` ()
- virtual void `aleaWeights` ()
- std::vector< double > * `getWeight` ()
- virtual int `getNbPoids` ()

Protected Member Functions

- double `derive_activate` (double sum, double k) const

Protected Attributes

- `std::vector< double > * w`
- `int n`
- [FonctionActivation](#) `fonctionActivation`

4.8.1 Constructor & Destructor Documentation

4.8.1.1 `Neurone::Neurone ()`

Constructeur par defaults [Neurone::Neurone](#)

4.8.1.2 `Neurone::Neurone (int n)`

Constructeur par nombre d'entrées [Neurone::Neurone](#)

Parameters

<i>n</i>	Nombre d'entrées
----------	------------------

4.8.1.3 `Neurone::Neurone (int n, std::vector< double > * x)`

Constructeur par arguments [Neurone::Neurone](#)

Parameters

<i>taille</i>	Nombre d'entrées
<i>x</i>	Vecteur de poids

4.8.2 Member Function Documentation

4.8.2.1 `void Neurone::aleaWeights ()` `[virtual]`

Méthode de génération de poids aléatoire selon une loi normale centrée réduite [Neurone::aleaWeights](#)

Reimplemented in [NeuroneB](#).

4.8.2.2 `double Neurone::derive_activate (double sum, double k) const` `[protected]`

Méthode de dérivation [Neurone::derive_activate](#)

Parameters

<i>sum</i>	Valeur de la somme du vecteur (cf Neurone::fw_sum)
<i>k</i>	Valeur du coefficient de sigmoid k

Returns

Valeur de dérivation

4.8.2.3 `double Neurone::fire (std::vector< double > in, double k) const` [virtual]

Méthode de propagation en avant [Neurone::fire](#)

Parameters

<i>in</i>	Vecteur d'entrées
<i>k</i>	Coefficient de sigmoid k

Returns

Valeur d'activation

4.8.2.4 `double Neurone::fw_activate (double sum, double k) const`

Méthode d'activation [Neurone::fw_activate](#)

Parameters

<i>sum</i>	Valeur de la somme du vecteur (cf Neurone::fw_sum)
<i>k</i>	Valeur du biais

Returns

Valeur d'activation

4.8.2.5 `double Neurone::fw_sum (std::vector< double > x) const` [virtual]

Méthode de somme des valeurs du vecteur [Neurone::fw_sum](#)

Parameters

<i>x</i>	Vecteur
----------	---------

Returns

Somme

Reimplemented in [NeuroneB](#).

4.8.2.6 `int Neurone::getNbPoids ()` [virtual]

[Neurone::getNbPoids](#)

Returns

Nombre de poids

Reimplemented in [NeuroneB](#).

4.8.2.7 `std::vector< double > * Neurone::getWeight ()`

[Neurone::getWeight](#)

Returns

Vecteur de poids

4.8.2.8 `void Neurone::learn (std::vector< double > x, double o, double k, double mu)` [virtual]

Méthode d'apprentissage (cf [Neurone::fw_sum](#), cf [Neurone::fw_activate](#), cf [Neurone::derive_activate](#)) [Neurone↔::learn](#)

Parameters

<i>x</i>	Vecteur de poids
<i>o</i>	Valeur attendue
<i>k</i>	Valeur du coefficient de sigmoid k
<i>mu</i>	Taux d'apprentissage Learning rate

Reimplemented in [NeuroneB](#).

4.8.2.9 `void Neurone::printWeight ()` [virtual]

Affichage des poids [Neurone::printWeight](#)

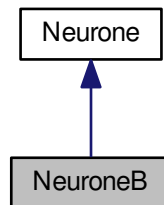
Reimplemented in [NeuroneB](#).

The documentation for this class was generated from the following files:

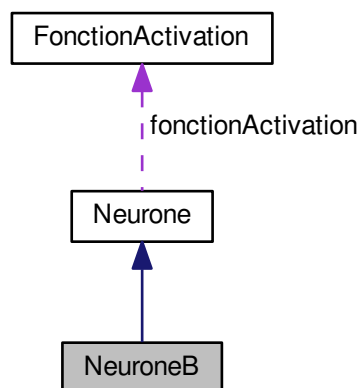
- Neurone/Neurone.h
- Neurone/Neurone.cpp

4.9 NeuroneB Class Reference

Inheritance diagram for NeuroneB:



Collaboration diagram for NeuroneB:



Public Member Functions

- `NeuroneB` (int taille)
- `NeuroneB` (int taille, std::vector< double > *x, double biais_w)
- void `printWeight` ()
- void `aleaWeights` ()
- void `learn` (std::vector< double > x, double o, double k, double mu)
- double `fw_sum` (std::vector< double > x) const
- int `getNbPoids` ()

Additional Inherited Members

4.9.1 Constructor & Destructor Documentation

4.9.1.1 `NeuroneB::NeuroneB (int taille)`

Constructeur du [Neurone](#) avec biais [NeuroneB::NeuroneB](#)

Parameters

<i>taille</i>	Taille du neurone
---------------	-------------------

4.9.1.2 `NeuroneB::NeuroneB (int taille, std::vector< double > * x, double biais_w)`

Constructeur du [Neurone](#) avec biais [NeuroneB::NeuroneB](#)

Parameters

<i>taille</i>	Taille du neurone
<i>x</i>	Vecteur de poids
<i>biais_w</i>	Valeur du biais

4.9.2 Member Function Documentation

4.9.2.1 `void NeuroneB::aleaWeights () [virtual]`

Génération aléatoire des poids [NeuroneB::aleaWeights](#)

Reimplemented from [Neurone](#).

4.9.2.2 `double NeuroneB::fw_sum (std::vector< double > x) const [virtual]`

Méthode de somme du vecteur de poids [NeuroneB::fw_sum](#)

Parameters

<i>x</i>	Vecteur de poids
----------	------------------

Returns

Somme des poids

Reimplemented from [Neurone](#).

4.9.2.3 `int NeuroneB::getNbPoids () [virtual]`

[NeuroneB::getNbPoids](#)

Returns

Nombre de poids

Reimplemented from [Neurone](#).

4.9.2.4 `void NeuroneB::learn (std::vector< double > x, double o, double k, double mu) [virtual]`

Méthode d'apprentissage (cf [Neurone::fw_sum](#), cf [Neurone::fw_activate](#), cf [Neurone::derive_activate](#)) [NeuroneB::learn](#)

Parameters

<i>x</i>	Vecteur de poids
<i>o</i>	Valeur attendue
<i>k</i>	Valeur du biais
<i>mu</i>	Taux d'apprentissage Learning rate

Reimplemented from [Neurone](#).

4.9.2.5 `void NeuroneB::printWeight () [virtual]`

Affichage des poids [NeuroneB::printWeight](#)

Reimplemented from [Neurone](#).

The documentation for this class was generated from the following files:

- Neurone/NeuroneB.h
- Neurone/NeuroneB.cpp

4.10 Option Class Reference

Public Member Functions

- [Option](#) ()
- **Option** (int id, const std::string &intitul, const std::string &m_rac, const std::string &type, const std::string &m_description)
- int [getOptionID](#) () const
- std::string [getOptionType](#) () const
- std::string [getOptionInt](#) () const
- std::string [getOptionRac](#) () const
- void [setOptionType](#) (std::string &t)
- void [setOptionInt](#) (std::string &intitul)
- void [setOptionRac](#) (std::string &rac)
- void [print](#) (std::ostream &os) const

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Option::Option ()

Constructeur d'option vide [Option::Option](#)

4.10.2 Member Function Documentation

4.10.2.1 int Option::getOptionID () const

Option::GetOptionId

Returns

ID de l'option

4.10.2.2 string Option::getOptionInt () const

Option::GetOptionInt

Returns

Intitulé de l'option

4.10.2.3 string Option::getOptionRac () const

Option::GetOptionRac

Returns

Raccourci de l'option

4.10.2.4 string Option::getOptionType () const

Option::GetOptionType

Returns

[Type](#) d'option

4.10.2.5 void Option::print (std::ostream & os) const

Méthode d'affichage en complément de << print

Parameters

<i>os</i>	Flux
-----------	------

4.10.2.6 void Option::setOptionInt (std::string & *intitul*)

Option::SetOptionInt

Parameters

<i>intitul</i>	Nouveau Intitulé
----------------	------------------

4.10.2.7 void Option::setOptionRac (std::string & *rac*)

Option::SetOptionRac

Parameters

<i>rac</i>	Nouveau Raccourci
------------	-------------------

4.10.2.8 void Option::setOptionType (std::string & *t*)

Option::SetOptionType

Parameters

<i>t</i>	Nouveau type d'option
----------	-----------------------

The documentation for this class was generated from the following files:

- Option/Option.h
- Option/Option.cpp

4.11 OptionTab Class Reference

Public Member Functions

- [OptionTab](#) ()
- void [addOption](#) (const [Option](#) &opt)
- void [printOptions](#) () const
- int [getOptionID](#) (const std::string &opt) const
- std::string [getArgument](#) (const std::string &opt) const
- std::string [getRaccour](#) (const std::string &opt) const
- std::string [getTypeOption](#) (const std::string &opt) const
- std::string [getIntitul](#) (const std::string &opt) const

4.11.1 Constructor & Destructor Documentation

4.11.1.1 OptionTab::OptionTab ()

Constructeur par défauts [OptionTab::OptionTab](#)

4.11.2 Member Function Documentation

4.11.2.1 void OptionTab::addOption (const Option & *opt*)

Ajoute une option OptionTab::AddOption

Parameters

<i>opt</i>	[description]
------------	---------------

4.11.2.2 std::string OptionTab::getArgument (const std::string & *opt*) const

OptionTab::GetArgument

Parameters

<i>opt</i>	Nom de l'option
------------	-----------------

Returns

Argument de cette option

4.11.2.3 std::string OptionTab::getIntitul (const std::string & *opt*) const

[OptionTab::getIntitul](#)

Parameters

<i>opt</i>	Nom de l'option
------------	-----------------

Returns

Intitulé de cette option

4.11.2.4 int OptionTab::getOptionID (const std::string & *opt*) const

OptionTab::GetOptionID

Parameters

<i>opt</i>	Nom de l'option
------------	-----------------

Returns

ID de cette option

4.11.2.5 `std::string OptionTab::getRaccour (const std::string & opt) const`

[OptionTab::getRaccour](#)

Parameters

<i>opt</i>	Nom de l'option
------------	-----------------

Returns

Raccourci de cette option

4.11.2.6 `void OptionTab::printOptions () const`

Affiche les options `OptionTab::PrintOptions`

The documentation for this class was generated from the following files:

- Option/OptionTab.h
- Option/OptionTab.cpp

4.12 Reseau Class Reference

Public Member Functions

- [Reseau](#) (int, std::vector< int >, double, double)
- std::vector< double > [fire_all](#) (std::vector< double > input)
- void [learn](#) (std::vector< std::vector< std::vector< double > > > jeuxTest)
- void [backPropagation](#) (std::vector< double > output)
- void [printWeight](#) ()

4.12.1 Constructor & Destructor Documentation

4.12.1.1 `Reseau::Reseau (int nbLayers, std::vector< int > layerInformation, double k, double eta)`

Constructeur [Reseau::Reseau](#)

Parameters

<i>nbLayers</i>	Nombre de couche
<i>layerInformation</i>	Vecteur descriptif de chaque layer
<i>k</i>	constante k dont depends la sigmoide
<i>eta</i>	coefficient d'evolution

4.12.2 Member Function Documentation

4.12.2.1 void Reseau::backPropagation (std::vector< double > *output*)

Méthode de propagation en arrière [Reseau::backPropagation](#)

Parameters

<i>output</i>	Sortie
<i>k</i>	Valeur du coefficient
<i>eta</i>	Valeur d'eta

4.12.2.2 std::vector< double > Reseau::fire_all (std::vector< double > *input*)

Fire général [Reseau::fire_all](#)

Parameters

<i>input</i>	Vecteur d'entrée
--------------	------------------

Returns

Valeur d'activation

4.12.2.3 void Reseau::learn (std::vector< std::vector< std::vector< double > > > *jeuxTest*)

Méthode d'apprentissage [Reseau::learn](#)

Parameters

<i>jeuxTest</i>	Vecteur de vecteur de vecteur : décrivant en <code>jeuxTest[i][0]</code> le vecteur d'entrée et en <code>jeuxTest[i][1]</code> la sortie attendue
-----------------	---

4.12.2.4 void Reseau::printWeight ()

Affichage des poids [Reseau::printWeight](#)

The documentation for this class was generated from the following files:

- Reseau/Reseau.h
- Reseau/Reseau.cpp

4.13 Settings Class Reference

Public Member Functions

- [Settings](#) (char const *fileName)
- int [getNbrHiddenLayers](#) ()
- std::vector< int > * [getArchi](#) ()
- std::vector< double > * [getDifferentOutputs](#) ()

4.13.1 Constructor & Destructor Documentation

4.13.1.1 Settings::Settings (char const * *fileName*)

Constructeur settings::settings

Parameters

<i>fileName</i>	Nom du fichier
-----------------	----------------

Counter

Parser

strinsplit à partir du caractère ','

4.13.2 Member Function Documentation

4.13.2.1 std::vector< int > * Settings::getArchi ()

settings::getArchi

Returns

Architecture

4.13.2.2 std::vector< double > * Settings::getDifferentOutputs ()

settings::getDifferentOutputs

Returns

DifferentOutputs

4.13.2.3 int Settings::getNbrHiddenLayers ()

settings::getNbrHiddenLayers

Returns

Nombre de Layers "caché"

The documentation for this class was generated from the following files:

- Settings/Settings.h
- Settings/Settings.cpp

4.14 Type Class Reference

Public Member Functions

- [Type](#) ()
- [Type](#) (std::string type)
- void [affiche](#) () const
- void [setType](#) (std::string type)
- std::string [getType](#) () const

4.14.1 Constructor & Destructor Documentation

4.14.1.1 Type::Type ()

Constructeur par défaut [Type::Type](#)

4.14.1.2 Type::Type (std::string type)

Constructeur avec arguments [Type::Type](#)

Parameters

<i>type</i>	Nom du type
-------------	-------------

4.14.2 Member Function Documentation

4.14.2.1 void Type::affiche () const

Affichage du type [Type::affiche](#)

4.14.2.2 `std::string Type::getType () const`

[Type::getType](#)

Returns

[Type](#)

4.14.2.3 `void Type::setType (std::string type)`

[Type::setType](#)

Parameters

<i>type</i>	Nouveau type
-------------	--------------

The documentation for this class was generated from the following files:

- [Option/Type.h](#)
- [Option/Type.cpp](#)

Chapter 5

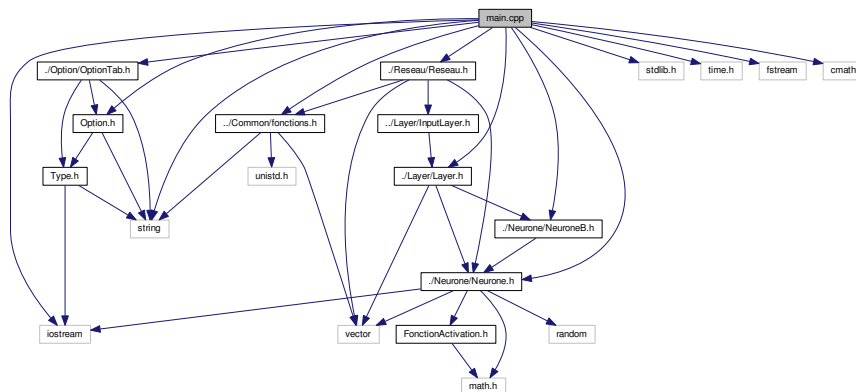
File Documentation

5.1 main.cpp File Reference

Programme de test.

```
#include <iostream>
#include "../Option/OptionTab.h"
#include "../Option/Option.h"
#include "../Neurone/Neurone.h"
#include "../Neurone/NeuroneB.h"
#include "../Layer/Layer.h"
#include "../Reseau/Reseau.h"
#include "../Common/fonctions.h"
#include <string>
#include <stdlib.h>
#include <time.h>
#include <fstream>
#include <cmath>
```

Include dependency graph for main.cpp:



Functions

- int [main](#) (int argc, char **argv)

5.1.1 Detailed Description

Programme de test.

5.1.2 Function Documentation

5.1.2.1 `int main (int argc, char ** argv)`

main

Parameters

<i>argc</i>	Nombre d'argument
<i>argv</i>	Tableaux des arguments

Returns

Code de sortie

Index

- addOption
 - OptionTab, [24](#)
- affiche
 - Type, [28](#)
- aleaWeights
 - Neurone, [16](#)
 - NeuroneB, [20](#)
- Archi.DrawNN, [7](#)
- Archi.Layer, [11](#)
- Archi.NeuralNetwork, [14](#)
- Archi.Neuron, [14](#)
- backPropagation
 - Reseau, [26](#)
- derive_activate
 - Neurone, [16](#)
- fire
 - InputLayer, [10](#)
 - Layer, [12](#)
 - Neurone, [17](#)
- fire_all
 - Reseau, [26](#)
- FonctionActivation, [7](#)
 - FonctionActivation, [8](#)
 - getValeurActivation, [8](#)
 - getValeurDerivee, [8](#)
 - setFonctionActivation, [9](#)
- fw_activate
 - Neurone, [17](#)
- fw_sum
 - Neurone, [17](#)
 - NeuroneB, [20](#)
- getArchi
 - Settings, [27](#)
- getArgument
 - OptionTab, [24](#)
- getDifferentOutputs
 - Settings, [27](#)
- getInput
 - Layer, [13](#)
- getIntitul
 - OptionTab, [24](#)
- getNbNeurones
 - Layer, [13](#)
- getNbPoids
 - Neurone, [17](#)
 - NeuroneB, [20](#)
- getNbrHiddenLayers
 - Settings, [27](#)
- getNeurone
 - Layer, [13](#)
- getOptionID
 - Option, [22](#)
 - OptionTab, [24](#)
- getOptionInt
 - Option, [22](#)
- getOptionRac
 - Option, [22](#)
- getOptionType
 - Option, [22](#)
- getRaccour
 - OptionTab, [25](#)
- getType
 - Type, [28](#)
- getValeurActivation
 - FonctionActivation, [8](#)
- getValeurDerivee
 - FonctionActivation, [8](#)
- getWeight
 - Neurone, [18](#)
- InputLayer, [9](#)
 - fire, [10](#)
 - InputLayer, [10](#)
- Layer, [11](#)
 - fire, [12](#)
 - getInput, [13](#)
 - getNbNeurones, [13](#)
 - getNeurone, [13](#)
 - Layer, [12](#)
 - printWeight, [13](#)
- learn
 - Neurone, [18](#)
 - NeuroneB, [21](#)
 - Reseau, [26](#)
- main
 - main.cpp, [32](#)
- main.cpp, [31](#)
 - main, [32](#)
- Neurone, [15](#)
 - aleaWeights, [16](#)
 - derive_activate, [16](#)
 - fire, [17](#)
 - fw_activate, [17](#)

- fw_sum, [17](#)
- getNbPoids, [17](#)
- getWeight, [18](#)
- learn, [18](#)
- Neurone, [16](#)
- printWeight, [18](#)
- NeuroneB, [19](#)
 - aleaWeights, [20](#)
 - fw_sum, [20](#)
 - getNbPoids, [20](#)
 - learn, [21](#)
 - NeuroneB, [20](#)
 - printWeight, [21](#)
- Option, [21](#)
 - getOptionID, [22](#)
 - getOptionInt, [22](#)
 - getOptionRac, [22](#)
 - getOptionType, [22](#)
 - Option, [22](#)
 - print, [22](#)
 - setOptionInt, [23](#)
 - setOptionRac, [23](#)
 - setOptionType, [23](#)
- OptionTab, [23](#)
 - addOption, [24](#)
 - getArgument, [24](#)
 - getIntitul, [24](#)
 - getOptionID, [24](#)
 - getRaccour, [25](#)
 - OptionTab, [24](#)
 - printOptions, [25](#)
- print
 - Option, [22](#)
- printOptions
 - OptionTab, [25](#)
- printWeight
 - Layer, [13](#)
 - Neurone, [18](#)
 - NeuroneB, [21](#)
 - Reseau, [26](#)
- Reseau, [25](#)
 - backPropagation, [26](#)
 - fire_all, [26](#)
 - learn, [26](#)
 - printWeight, [26](#)
 - Reseau, [25](#)
- setFonctionActivation
 - FonctionActivation, [9](#)
- setOptionInt
 - Option, [23](#)
- setOptionRac
 - Option, [23](#)
- setOptionType
 - Option, [23](#)
- setType
 - Type, [29](#)
- Settings, [27](#)
 - getArchi, [27](#)
 - getDifferentOutputs, [27](#)
 - getNbrHiddenLayers, [27](#)
 - Settings, [27](#)
- Type, [28](#)
 - affiche, [28](#)
 - getType, [28](#)
 - setType, [29](#)
 - Type, [28](#)