Implémentation de l'algorithme de rétropagation sur un réseau neuronal

Gaspard Coulet - Félix Guyard - Areski Himeur - Lucas Picasarri-Arrieta - Jean Maurice Raboude

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 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

3.2 FonctionActivation Class Reference

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

Public Types

- enum **EnumFonctionActivation** {
  IDENTITY, **BINARYSTEP**, **SIGMOID**, **TAN**,
  SIN, **RELU**, **LRELU**, **PRELU**,
  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

3.2.1 Detailed Description

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

3.2.2 Constructor & Destructor Documentation

3.2.2.1 FonctionActivation::FonctionActivation ( )

Constructeur par défauts FonctionActivation

Parameters

fonctionActivation	Fonction activation à utiliser parmis l'enumération

3.2.2.2 FonctionActivation::FonctionActivation (EnumFonctionActivation *fonctionActivation*)

Constructeur FonctionActivation

Parameters

fonctionActivation	Fonction activation à utiliser parmis l'enumération

3.2.3 Member Function Documentation

3.2.3.1 double FonctionActivation::getValeurActivation (double *x*, double *k* = 0) const

getValeurActivation

Parameters

x	Valeur de la variable
k	Constante de changement

Returns

Valeur d'activation

3.2.3.2 double FonctionActivation::getValeurDerivee (double *x*, double *k* = 0) const

getValeurDerive

Parameters

x	Valeur de la variable
k	Constante de changement

Returns

Valeur de la derivée

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

FonctionActivation::setFonctionActivation

Parameters

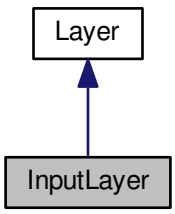
fonctionActivation	Fonction activation à utiliser parmis l'enumération

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

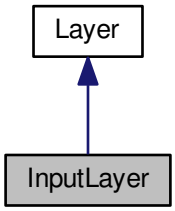
- Neurone/FonctionActivation.h
- Neurone/FonctionActivation.cpp

3.3 InputLayer Class Reference

Inheritance diagram for InputLayer:



Collaboration diagram for InputLayer:



Public Member Functions

- InputLayer (int, FonctionActivation::EnumFonctionActivation fct)
- std::vector< double > fire (std::vector< double >, double)

Additional Inherited Members

3.3.1 Constructor & Destructor Documentation

3.3.1.1 InputLayer::InputLayer (int *taille,* FonctionActivation::EnumFonctionActivation *fct*)

Constructeur par taille InputLayer::InputLayer

Parameters

taille	Nombre de neurones dans le layer

3.3.2 Member Function Documentation

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

Méthode de propagation en avant InputLayer::fire

Parameters

input	Vecteur en entrées
k	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

3.4 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

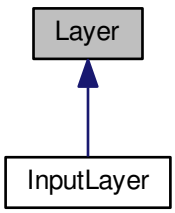
- **vertical_distance_between_layers**
- **horizontal_distance_between_neurons**
- **neuron_radius**
- **number_of_neurons_in_widest_layer**
- **previous_layer**
- **y**
- **neurons**

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

- Archi.py

3.5 Layer Class Reference

Inheritance diagram for Layer:



Public Types

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

Public Member Functions

- Layer ()
- Layer (TypeLayer, int, int, FonctionActivation::EnumFonctionActivation)
- Neurone * getNeurone (int index)
- virtual std::vector< double > fire (std::vector< double >, double)
- 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**

3.5.1 Constructor & Destructor Documentation

3.5.1.1 Layer::Layer ( )

Constructeur par défauts Layer::Layer

3.5.1.2 Layer::Layer (TypeLayer *type*, int *nbneur*, int *nbinput*, FonctionActivation::EnumFonctionActivation *fct*)

Constructeur avec arguments Layer::Layer

Parameters

type	Type de layer
nbneur	Nombre de neurone
nbinput	Nombre d'entrée par neurone

3.5.2 Member Function Documentation

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

Propagation en avant Layer::fire

Parameters

input	Vecteur en entrées
k	Coefficient de sigmoid

Returns

Vecteur des valeurs d'activations

Reimplemented in InputLayer.

3.5.2.2 std::vector< double > Layer::getInput ( )

Layer::getInput

Returns

Retourne le vecteur d'entrées

3.5.2.3 int Layer::getNbNeurones ( )

Layer::getNbNeurones

Returns

Nombre de neurones

3.5.2.4 Neurone * Layer::getNeurone (int *index*)

Layer::getNeurone

Parameters

index	index

Returns

Neurone

3.5.2.5 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

3.6 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

3.7 Archi.Neuron Class Reference

Public Member Functions

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

Public Attributes

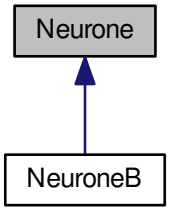
- **x**
- **y**

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

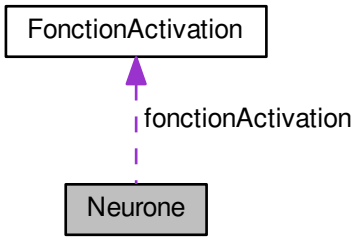
- Archi.py

3.8 Neurone Class Reference

Inheritance diagram for Neurone:



Collaboration diagram for Neurone:



Public Member Functions

- Neurone ()
- Neurone (int taille, FonctionActivation::EnumFonctionActivation fct)
- Neurone (int taille, std::vector< double > *x, FonctionActivation::EnumFonctionActivation fct)
- virtual double fw_sum (std::vector< double > x) const
- double fw_activate (double sum, double k) const
- double derive_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 Attributes

- std::vector< double > * **w**
- int **n**
- FonctionActivation **fonctionActivation**

3.8.1 Constructor & Destructor Documentation

3.8.1.1 Neurone::Neurone ( )

Constructeur par defauts Neurone::Neurone

3.8.1.2 Neurone::Neurone (int *n*, FonctionActivation::EnumFonctionActivation *fct*)

Constructeur par nombre d'entrées Neurone::Neurone

Parameters

n	Nombre d'entrées
n	Fonction d'Activation

3.8.1.3 Neurone::Neurone (int *n*, std::vector< double > * *x*, FonctionActivation::EnumFonctionActivation *fct*)

Constructeur par arguments Neurone::Neurone

Parameters

taille	Nombre d'entrées
x	Vecteur de poids

3.8.2 Member Function Documentation

3.8.2.1 void Neurone::aleaWeights ( ) [virtual]

Méthode de génération de poids aléatoire selon une loi normale centrée réduite Neurone::aleaWeights

Reimplemented in NeuroneB.

3.8.2.2 double Neurone::derive_activate (double *sum*, double *k*) const

Méthode de dérivation Neurone::derive_activate

Parameters

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

Returns

Valeur de dérivation

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

Méthode de propagation en avant Neurone::fire

Parameters

in	Vecteur d'entrées
k	Coefficient de sigmoid k

Returns

Valeur d'activation

3.8.2.4 double Neurone::fw_activate (double *sum*, double *k*) const

Méthode d'activation Neurone::fw_activate

Parameters

sum	Valeur de la somme du vecteur (cf Neurone::fw_sum)
k	Valeur du biais

Returns

Valeur d'activation

3.8.2.5 double Neurone::fw_sum (std::vector< double > *x*) const [virtual]

Méthode de somme des valeurs du vecteur Neurone::fw_sum

Parameters

x	Vecteur

Returns

Somme

Reimplemented in NeuroneB.

3.8.2.6 int Neurone::getNbPoids ( ) [virtual]

Neurone::getNbPoids

Returns

Nombre de poids

Reimplemented in NeuroneB.

3.8.2.7 std::vector< double > * Neurone::getWeight ( )

Neurone::getWeight

Returns

Vecteur de poids

3.8.2.8 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

x	Vecteur de poids
o	Valeur attendue
k	Valeur de l'hyperparamètre
mu	Taux d'apprentissage | Learning rate

Reimplemented in NeuroneB.

3.8.2.9 void Neurone::printWeight ( ) [virtual]

Affichage des poids Neurone::printWeight

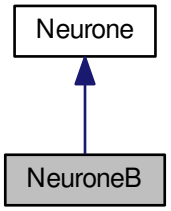
Reimplemented in NeuroneB.

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

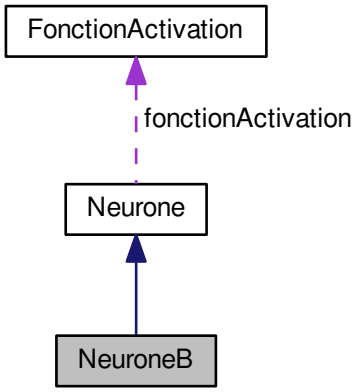
- Neurone/Neurone.h
- Neurone/Neurone.cpp

3.9 NeuroneB Class Reference

Inheritance diagram for NeuroneB:



Collaboration diagram for NeuroneB:



Public Member Functions

- NeuroneB (int taille, FonctionActivation::EnumFonctionActivation fct)
- 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

3.9.1 Constructor & Destructor Documentation

3.9.1.1 NeuroneB::NeuroneB (int *taille,* FonctionActivation::EnumFonctionActivation *fct*)

Constructeur du Neurone avec biais NeuroneB::NeuroneB

Parameters

taille	Taille du neurone

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

Constructeur du Neurone avec biais NeuroneB::NeuroneB

Parameters

taille	Taille du neurone
x	Vecteur de poids
biais_w	Valeur du biais

3.9.2 Member Function Documentation

3.9.2.1 void NeuroneB::aleaWeights ( ) [virtual]

Génération aléatoire des poids NeuroneB::aleaWeights

Reimplemented from Neurone.

3.9.2.2 double NeuroneB::fw_sum (std::vector< double > *x*) const [virtual]

Méthode de somme du vecteur de poids NeuroneB::fw_sum

Parameters

x	Vecteur de poids

Returns

Somme des poids

Reimplemented from Neurone.

3.9.2.3 int NeuroneB::getNbPoids ( ) [virtual]

NeuroneB::getNbPoids

Returns

Nombre de poids

Reimplemented from Neurone.

3.9.2.4 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) Neurone::learn

Parameters

x	Vecteur de poids
o	Valeur attendue
k	Valeur du biais
mu	Taux d'apprentissage | Learning rate

Reimplemented from Neurone.

3.9.2.5 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

3.10 Option Class Reference

Public Member Functions

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

3.10.1 Constructor & Destructor Documentation

3.10.1.1 Option::Option ( )

Constructeur d'option vide Option::Option

3.10.2 Member Function Documentation

3.10.2.1 int Option::getOptionID ( ) const

Option::GetOptionId

Returns

ID de l'option

3.10.2.2 string Option::getOptionInt ( ) const

Option::GetOptionInt

Returns

Intitulé de l'option

3.10.2.3 string Option::getOptionRac ( ) const

Option::GetOptionRac

Returns

Raccourci de l'option

3.10.2.4 Type::TypeEnum Option::getOptionType ( ) const

Option::GetOptionType

Returns

Type d'option

3.10.2.5 void Option::print (std::ostream & *os*) const

Méthode d'affichage en complément de << print

Parameters

os	Flux

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

Option::SetOptionInt

Parameters

intitul	Nouveau Intitulé

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

Option::SetOptionRac

Parameters

rac	Nouveau Raccourci

3.10.2.8 void Option::setOptionType (Type::TypeEnum *t*)

Option::SetOptionType

Parameters

t	Nouveau type d'option

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

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

3.11 OptionTab Class Reference

Public Member Functions

- OptionTab ()
- void addOption (const Option &opt)
- void printOptions () const
- int getOptionID (const std::string &opt) const
- Type::TypeEnum 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

3.11.1 Constructor & Destructor Documentation

3.11.1.1 OptionTab::OptionTab ( )

Constructeur par défauts OptionTab::OptionTab

3.11.2 Member Function Documentation

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

Ajoute une option OptionTab::AddOption

Parameters

opt	[description]

3.11.2.2 Type::TypeEnum OptionTab::getArgument (const std::string & *opt*) const

OptionTab::GetArgument

Parameters

opt	Nom de l'option

Returns

Argument de cette option

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

OptionTab::getIntitul

Parameters

opt	Nom de l'option

Returns

Intitulé de cette option

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

OptionTab::GetOptionID

Parameters

opt	Nom de l'option

Returns

ID de cette option

3.11.2.5 std::string OptionTab::getRaccour (const std::string & *opt*) const

OptionTab::getRaccour

Parameters

opt	Nom de l'option

Returns

Raccourci de cette option

3.11.2.6 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

3.12 Reseau Class Reference

Public Member Functions

- Reseau (int, std::vector< int >, double, double, FonctionActivation::EnumFonctionActivation fct)
- std::vector< double > fire_all (std::vector< double > input)
- void learn (std::vector< std::vector< std::vector< double > > > jeuxTest, unsigned int nbPasDescente↩ Gradient)
- void backPropagation (std::vector< double > erreurs)
- void printWeight ()

3.12.1 Constructor & Destructor Documentation

3.12.1.1 Reseau::Reseau (int *nbLayers*, std::vector< int > *layerInformation*, double *k*, double *eta*, FonctionActivation::EnumFonctionActivation *fct*)

Constructeur Reseau::Reseau

Parameters

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

3.12.2 Member Function Documentation

3.12.2.1 void Reseau::backPropagation (std::vector< double > *erreurs*)

Méthode de propagation en arriére Reseau::backPropagation

Parameters

output	Sortie
k	Valeur du coefficient
eta	Valeur d'eta

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

Fire général Reseau::fire_all

Parameters

input	Vecteur d'entrée

Returns

Valeur d'activation

3.12.2.3 void Reseau::learn (std::vector< std::vector< std::vector< double > > > *jeuxTest*, unsigned int *nbPasDescenteGradient*)

Méthode d'apprentissage Reseau::learn

Parameters

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

3.12.2.4 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

3.13 Settings Class Reference

Public Member Functions

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

3.13.1 Constructor & Destructor Documentation

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

Constructeur settings::settings

Parameters

fileName	Nom du fichier

Counter

Parser

strinsplit à partir du caractère ','

3.13.2 Member Function Documentation

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

settings::getArchi

Returns

Architecture

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

settings::getDifferentOutputs

Returns

DifferentOutputs

3.13.2.3 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

3.14 Type Class Reference

Public Types

- enum **TypeEnum** { **NONE**, **INT**, **FLOAT**, **STRING** }

Public Member Functions

- Type ()
- Type (Type::TypeEnum type)
- void print (std::ostream &os) const
- void setType (Type::TypeEnum type)
- Type::TypeEnum getType () const

3.14.1 Constructor & Destructor Documentation

3.14.1.1 Type::Type ( )

Constructeur par défaut Type::Type

3.14.1.2 Type::Type (Type::TypeEnum *type*)

Constructeur avec arguments Type::Type

Parameters

type	Nom du type

3.14.2 Member Function Documentation

3.14.2.1 Type::TypeEnum Type::getType ( ) const

Type::getType

Returns

Type

3.14.2.2 void Type::print (std::ostream & *os*) const

Affiche dans le flux une chaine de caractére décrivant le type Type::print

Parameters

os	Flux

3.14.2.3 void Type::setType (Type::TypeEnum *type*)

Type::setType

Parameters

type	Nouveau type

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

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

Index