ALGORITMO DE APRENDIZAJE PARA UNA RED NEURONAL PROFUNDA BASADO EN UNA METAHEURÍSTICA DE OPTIMIZACIÓN GLOBAL DE GRAN ESCALA

Versión: DNF_v1
Reporte: DNF - Tesis Maestría 2019

Universidad
del Cauca

Autores: Ing. Julián Fernando Muñoz Ordóñez
Ph.D. Carlos Alberto Cobos Lozada
Ph.D. Martha Eliana Mendoza Becerra

Version:
Fecha: 11/27/2019

Summary:

El aprendizaje profundo se basa en el incremento del número de capas ocultas en las arquitecturas de redes neuronales, este incremento proporciona a la red una mayor capacidad de abstracción sobre el conjunto de datos que se entrena, permitiendo mejorar considerablemente las tasas de precisión y error con respecto a los algoritmos tradicionales de aprendizaje máquina. Sin embargo, incrementar el número de capas ocultas conlleva a que los procesos de entrenamiento sufran estancamientos debido a diferentes problemas en la alta dimensionalidad: puntos de silla, múltiples óptimos locales y puntos de platea. El algoritmo por excelencia para el entrenamiento de redes neuronales profundas es conocido como RMSProp, este algoritmo funciona con información brindada por el gradiente descendente estocástico, sin embargo, la Neuroevolución: técnicas relacionadas con algoritmos evolutivos para entrenamiento de redes neuronales, han brindado nuevas posibilidades para resolver estos tipos de problemas. El presente trabajo de investigación se enfoca en definir cuál de tres (3) de los mejores algoritmos metaheurísticos mono-objetivo especializados en resolver problemas de alta dimensionalidad: Coevolución cooperativa basada en evolución diferencial (DECC-G), Muestreo múltiple de descendientes (MOS) e Hibridación iterativa de evolución diferencial (DE) con búsqueda local con reinicio (IHDELS) consiguen mejorar los resultados del error cuadrático medio logrados por el RMSProp y un simple algoritmo evolutivo de evaluaciones limitadas (LEEA) sobre conjuntos de datos de regresión y clasificación. Los resultados muestran que las metaheurísticas logran ser competitivas frente al RMSProp y logran vencer al algoritmo LEEA en conjuntos de datos de regresión. Por otra parte, en esta investigación se plantea un nuevo reto en la optimización de redes neuronales convolucionales profundas sobre conjuntos de datos de imágenes con un grado mayor de complejidad, durante esta etapa los resultados que se obtienen consisten en la construcción de una nueva propuesta memética inmersa en un nuevo framework de Neuroevolución (DNF) que logra vencer al RMSProp optimizando arquitecturas de redes neuronales convolucionales. Generando así un nuevo algoritmo y conocimiento en el área de la Neuroevolución aplicada sobre el aprendizaje profundo.

Tabla de Contenidos

I Introduction

I.1 Description

I.2 Description of DNF_v1

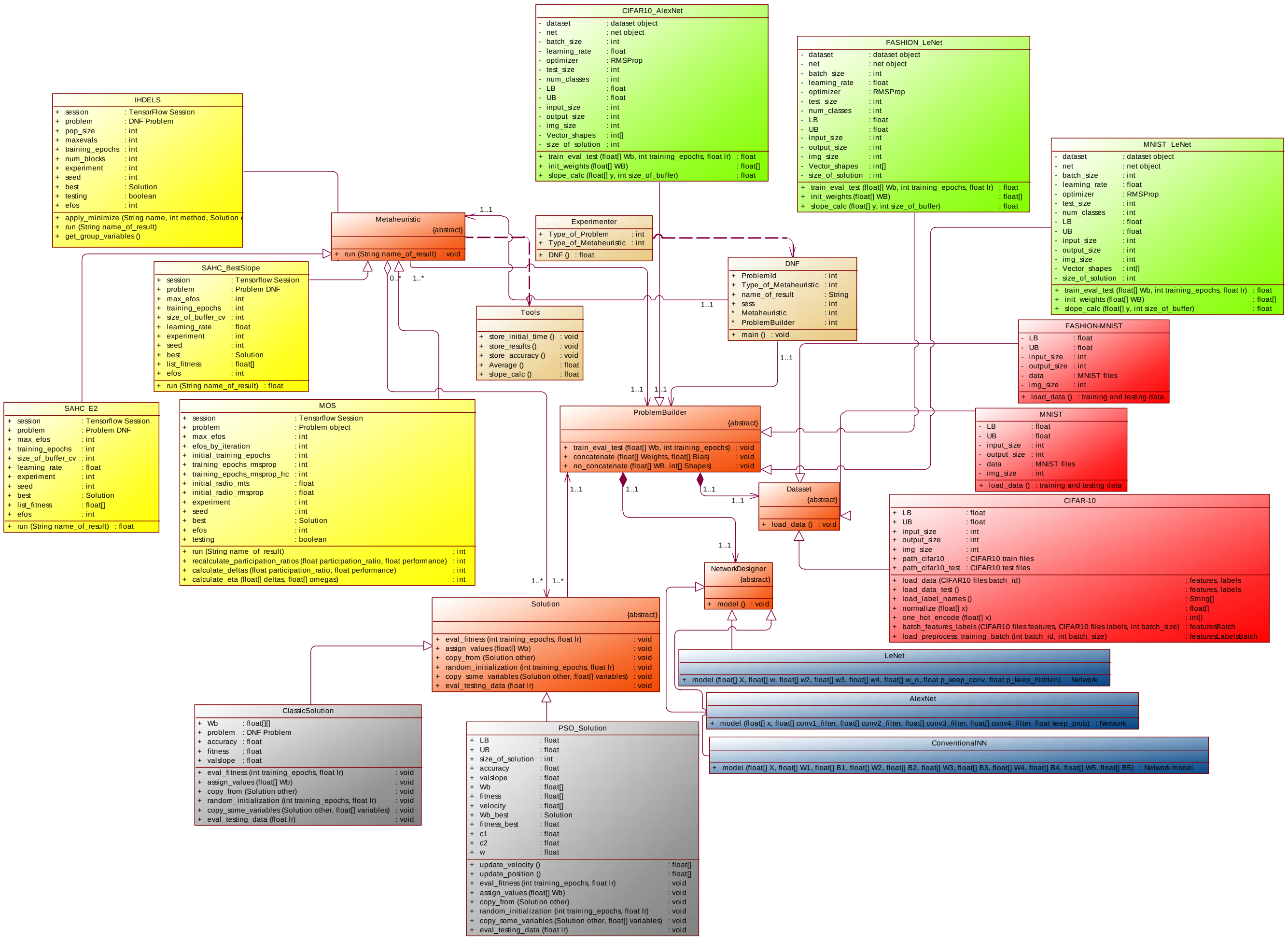
Name	DNF_v1
Code	DNF_v1
Object Language	Python
Comment	Framework para neuroevolución escrito en Python utilizando la librería TensorFlow
Author	Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
Version	1.0

II Short model description

II.1 List of diagrams

Name	Code
DNF_v1	DNF_v1

II.2 Diagram DNF_v1


CIFAR10_AlexNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
FASHION_LeNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
MNIST_LeNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
IHDELS
+ session : TensorFlow Session
+ problem : DNF Problem
+ pop_size : int
+ maxevals : int
+ training_epochs : int
+ num_blocks : int
+ experiment : int
+ seed : int
+ best : Solution
+ testing : boolean
+ efos : int
+ apply_minimize (String name, int method, Solution
+ run (String name_of_result)
+ get_group_variables ()
Metaheuristic
{abstract}
+ run (String name_of_result) : void
1..1
0..*
1..*
Experimenter
+ Type_of_Problem : int
+ Type_of_Metaheuristic : int
+ DNF () : float
DNF
+ ProblemId : int
+ Type_of_Metaheuristic : int
+ name_of_result : String
+ sess : int
* Metaheuristic : int
* ProblemBuilder : int
+ main () : void
1..1
1..1
Tools
+ store_initial_time () : void
+ store_results () : void
+ store_accuracy () : void
+ Average () : float
+ slope_calc () : float
SAHC_BestSlope
+ session : Tensorflow Session
+ problem : Problem DNF
+ max_efos : int
+ training_epochs : int
+ size_of_buffer_cv : int
+ learning_rate : float
+ experiment : int
+ seed : int
+ best : Solution
+ list_fitness : float[]
+ efos : int
+ run (String name_of_result) : float
FASHION-MNIST
- LB : float
- UB : float
- input_size : int
- output_size : int
- data : MNIST files
- img_size : int
+ load_data () : training and testing data
SAHC_E2
+ session : Tensorflow Session
+ problem : Problem DNF
+ max_efos : int
+ training_epochs : int
+ size_of_buffer_cv : int
+ learning_rate : float
+ experiment : int
+ seed : int
+ best : Solution
+ list_fitness : float[]
+ efos : int
+ run (String name_of_result) : float
MOS
+ session : Tensorflow Session
+ problem : Problem object
+ max_efos : int
+ efos_by_iteration : int
+ initial_training_epochs : int
+ training_epochs_rmsprop : int
+ training_epochs_rmsprop_hc : int
+ initial_radio_mts : float
+ initial_radio_rmsprop : float
+ experiment : int
+ seed : int
+ best : Solution
+ efos : int
+ testing : boolean
+ run (String name_of_result) : int
+ recalculate_participation_ratios (float participation_ratio, float performance) : int
+ calculate_deltas (float participation_ratio, float performance) : int
+ calculate_eta (float[] deltas, float[] omegas) : int
1..1
1..1
ProblemBuilder
{abstract}
+ train_eval_test (float[] Wb, int training_epochs) : void
+ concatenate (float[] Weights, float[] Bias) : void
+ no_concatenate (float[] WB, int[] Shapes) : void
MNIST
- LB : float
- UB : float
- input_size : int
- output_size : int
- data : MNIST files
- img_size : int
+ load_data () : training and testing data
1..1
1..1
1..1
1..1
Dataset
{abstract}
+ load_data () : void
CIFAR-10
+ LB : float
+ UB : float
+ input_size : int
+ output_size : int
+ img_size : int
+ path_cifar10 : CIFAR10 train files
+ path_cifar10_test : CIFAR10 test files
+ load_data (CIFAR10 files batch_id) : features, labels
+ load_data_test () : features, labels
+ load_label_names () : String[]
+ normalize (float[] x) : float[]
+ one_hot_encode (float[] x) : int[]
+ batch_features_labels (CIFAR10 files features, CIFAR10 files labels, int batch_size) : featuresBatch
+ load_preprocess_training_batch (int batch_id, int batch_size) : featuresLabelsBatch
1..1
NetworkDesigner
{abstract}
+ model () : void
1..*
1..*
Solution
{abstract}
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void
LeNet
+ model (float[] X, float[] w, float[] w2, float[] w3, float[] w4, float[] w_o, float p_keep_conv, float p_keep_hidden) : Network
AlexNet
+ model (float[] x, float[] conv1_filter, float[] conv2_filter, float[] conv3_filter, float[] conv4_filter, float keep_prob) : Network
ConventionalNN
+ model (float[] X, float[] W1, float[] B1, float[] W2, float[] B2, float[] W3, float[] B3, float[] W4, float[] B4, float[] W5, float[] B5) : Network model
ClassicSolution
+ Wb : float[][]
+ problem : DNF Problem
+ accuracy : float
+ fitness : float
+ valslope : float
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void
PSO_Solution
+ LB : float
+ UB : float
+ size_of_solution : int
+ accuracy : float
+ valslope : float
+ Wb : float[]
+ fitness : float[]
+ velocity : float[]
+ Wb_best : Solution
+ fitness_best : float
+ c1 : float
+ c2 : float
+ w : float
+ update_velocity () : float[]
+ update_position () : float[]
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void

II.2.1 List of classes in diagram

Name	Code	Generate	Visibility	Abstract	Classifier Type
AlexNet	AlexNet	X	public		Class
CIFAR-10	Cifar10	X	public		Class
CIFAR10_AlexNet	Cifar10AlexNet	X	public		Class
ClassicSolution	ClassicSolution	X	public		Class

ConventionalNN	ConventionalNN	X	public		Class
Dataset	Dataset	X	public	X	Class
DNF	Dnf	X	public		Class
Experimenter	Experimenter	X	public		Class
FASHION-MNIST	FashionMnist	X	public		Class
FASHION_LeNet	FashionLeNet	X	public		Class
IHDELS	Ihdels	X	public		Class
LeNet	LeNet	X	public		Class
Metaheuristic	Metaheuristic	X	public	X	Class
MNIST	Mnist	X	public		Class
MNIST_LeNet	MnistLeNet	X	public		Class
MOS	Mos	X	public		Class
NetworkDesigner	NetworkDesigner	X	public	X	Class
ProblemBuilder	ProblemBuilder	X	public	X	Class
PSO_Solution	PsoSolution	X	public		Class
SAHC_BestSlope	SahcBestSlope	X	public		Class
SAHC_E2	SahcE2	X	public		Class
Solution	Solution	X	public	X	Class
Tools	Tools	X	public		Class

II.2.2 List of associations in diagram

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	Problem Builder			1..1	1..1
Association_2	association2	Network Designer	Problem Builder			1..1	1..1
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1
Association_6	association6	Metaheuristic	DNF			1..1	1..1
Association_7	association7	Problem Builder	DNF			1..1	1..1
Association_8	association8	Problem Builder	Solution			1..*	1..1

II.2.3 List of generalizations in diagram

Name	Code	Parent Object	Child Object
Generalization_1	Generalization_1	Dataset	MNIST
Generalization_2	Generalization_2	Dataset	CIFAR-10
Generalization_3	Generalization_3	Dataset	FASHION-MNIST
Generalization_4	Generalization_4	NetworkDesigner	LeNet
Generalization_5	Generalization_5	NetworkDesigner	AlexNet
Generalization_6	Generalization_6	Metaheuristic	SAHC_BestSlope
Generalization_7	Generalization_7	Metaheuristic	MOS
Generalization_9	Generalization_9	NetworkDesigner	ConventionalNN
Generalization_10	Generalization_10	ProblemBuilder	MNIST_LeNet
Generalization_11	Generalization_11	ProblemBuilder	CIFAR10_AlexNet
Generalization_12	Generalization_12	Metaheuristic	IHDELS
Generalization_13	Generalization_13	ProblemBuilder	FASHION_LeNet
Generalization_14	Generalization_14	Solution	PSO_Solution
Generalization_15	Generalization_15	Solution	ClassicSolution
Generalization_16	Generalization_16	Metaheuristic	SAHC_E2

II.2.4 List of dependencies in diagram

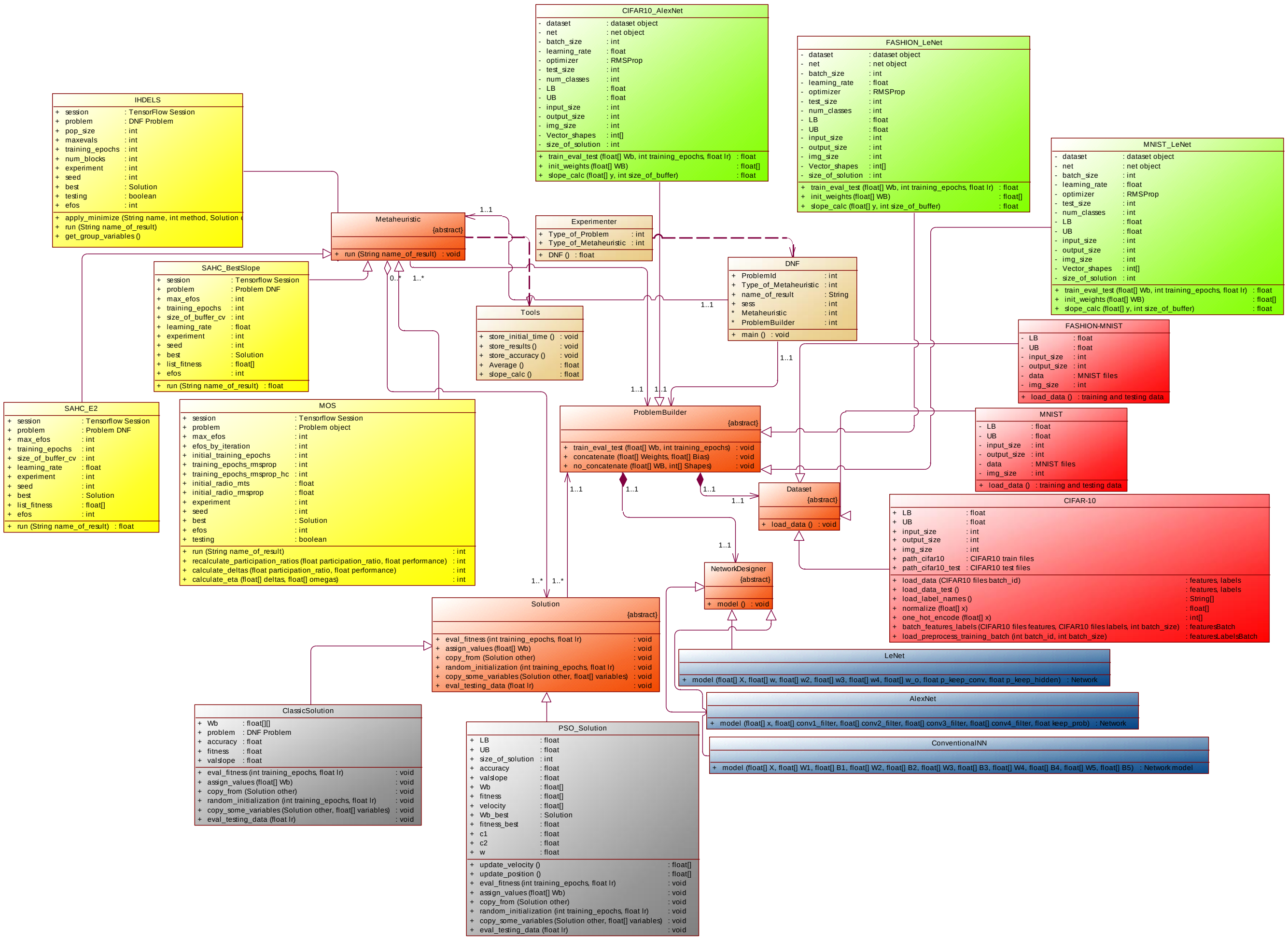
Name	Code	Influent Object	Dependent Object
Dependency_1	Dependency_1	Tools	Metaheuristic
Dependency_2	Dependency_2	DNF	Experimenter

III Full model description

III.1 List of diagrams

Name	Code
DNF_v1	DNF_v1

III.2 Diagram DNF_v1


CIFAR10_AlexNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
FASHION_LeNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
MNIST_LeNet
- dataset : dataset object
- net : net object
- batch_size : int
- learning_rate : float
- optimizer : RMSProp
- test_size : int
- num_classes : int
- LB : float
- UB : float
- input_size : int
- output_size : int
- img_size : int
- Vector_shapes : int[]
- size_of_solution : int
+ train_eval_test (float[] Wb, int training_epochs, float lr) : float
+ init_weights (float[] WB) : float[]
+ slope_calc (float[] y, int size_of_buffer) : float
IHDELS
+ session : TensorFlow Session
+ problem : DNF Problem
+ pop_size : int
+ maxevals : int
+ training_epochs : int
+ num_blocks : int
+ experiment : int
+ seed : int
+ best : Solution
+ testing : boolean
+ efos : int
+ apply_minimize (String name, int method, Solution c
+ run (String name_of_result)
+ get_group_variables ()
Metaheuristic
{abstract}
+ run (String name_of_result) : void
1..1
0..*
1..*
Experimenter
+ Type_of_Problem : int
+ Type_of_Metaheuristic : int
+ DNF () : float
DNF
+ ProblemId : int
+ Type_of_Metaheuristic : int
+ name_of_result : String
+ sess : int
* Metaheuristic : int
* ProblemBuilder : int
+ main () : void
Tools
+ store_initial_time () : void
+ store_results () : void
+ store_accuracy () : void
+ Average () : float
+ slope_calc () : float
SAHC_BestSlope
+ session : Tensorflow Session
+ problem : Problem DNF
+ max_efos : int
+ training_epochs : int
+ size_of_buffer_cv : int
+ learning_rate : float
+ experiment : int
+ seed : int
+ best : Solution
+ list_fitness : float[]
+ efos : int
+ run (String name_of_result) : float
SAHC_E2
+ session : Tensorflow Session
+ problem : Problem DNF
+ max_efos : int
+ training_epochs : int
+ size_of_buffer_cv : int
+ learning_rate : float
+ experiment : int
+ seed : int
+ best : Solution
+ list_fitness : float[]
+ efos : int
+ run (String name_of_result) : float
MOS
+ session : Tensorflow Session
+ problem : Problem object
+ max_efos : int
+ efos_by_iteration : int
+ initial_training_epochs : int
+ training_epochs_rmsprop : int
+ training_epochs_rmsprop_hc : int
+ initial_radio_mts : float
+ initial_radio_rmsprop : float
+ experiment : int
+ seed : int
+ best : Solution
+ efos : int
+ testing : boolean
+ run (String name_of_result) : int
+ recalculate_participation_ratios (float participation_ratio, float performance) : int
+ calculate_deltas (float participation_ratio, float performance) : int
+ calculate_eta (float[] deltas, float[] omegas) : int
1..1
1..1
1..1
1..1
ProblemBuilder
{abstract}
+ train_eval_test (float[] Wb, int training_epochs) : void
+ concatenate (float[] Weights, float[] Bias) : void
+ no_concatenate (float[] WB, int[] Shapes) : void
1..1
1..1
1..1
1..1
1..1
FASHION-MNIST
- LB : float
- UB : float
- input_size : int
- output_size : int
- data : MNIST files
- img_size : int
+ load_data () : training and testing data
MNIST
- LB : float
- UB : float
- input_size : int
- output_size : int
- data : MNIST files
- img_size : int
+ load_data () : training and testing data
Dataset
{abstract}
+ load_data () : void
CIFAR-10
+ LB : float
+ UB : float
+ input_size : int
+ output_size : int
+ img_size : int
+ path_cifar10 : CIFAR10 train files
+ path_cifar10_test : CIFAR10 test files
+ load_data (CIFAR10 files batch_id) : features, labels
+ load_data_test () : features, labels
+ load_label_names () : String[]
+ normalize (float[] x) : float[]
+ one_hot_encode (float[] x) : int[]
+ batch_features_labels (CIFAR10 files features, CIFAR10 files labels, int batch_size) : featuresBatch
+ load_preprocess_training_batch (int batch_id, int batch_size) : featuresLabelsBatch
1..1
NetworkDesigner
{abstract}
+ model () : void
1..*
1..*
Solution
{abstract}
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void
LeNet
+ model (float[] X, float[] w, float[] w2, float[] w3, float[] w4, float[] w_o, float p_keep_conv, float p_keep_hidden) : Network
AlexNet
+ model (float[] x, float[] conv1_filter, float[] conv2_filter, float[] conv3_filter, float[] conv4_filter, float keep_prob) : Network
ConventionalNN
+ model (float[] X, float[] W1, float[] B1, float[] W2, float[] B2, float[] W3, float[] B3, float[] W4, float[] B4, float[] W5, float[] B5) : Network model
ClassicSolution
+ Wb : float[][]
+ problem : DNF Problem
+ accuracy : float
+ fitness : float
+ valslope : float
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void
PSO_Solution
+ LB : float
+ UB : float
+ size_of_solution : int
+ accuracy : float
+ valslope : float
+ Wb : float[]
+ fitness : float[]
+ velocity : float[]
+ Wb_best : Solution
+ fitness_best : float
+ c1 : float
+ c2 : float
+ w : float
+ update_velocity () : float[]
+ update_position () : float[]
+ eval_fitness (int training_epochs, float lr) : void
+ assign_values (float[] Wb) : void
+ copy_from (Solution other) : void
+ random_initialization (int training_epochs, float lr) : void
+ copy_some_variables (Solution other, float[] variables) : void
+ eval_testing_data (float lr) : void

III.2.1 List of classes in diagram

Name	*Code*	*Generate*	*Visibility*	*Abstract*	*Classifier Type*
AlexNet	AlexNet	X	public		Class
CIFAR-10	Cifar10	X	public		Class
CIFAR10_AlexNet	Cifar10AlexNet	X	public		Class
ClassicSolution	ClassicSolution	X	public		Class
ConventionalNN	ConventionalNN	X	public		Class
Dataset	Dataset	X	public	X	Class
DNF	Dnf	X	public		Class

Experimenter	Experimenter	X	public		Class
FASHION-MNIST	FashionMnist	X	public		Class
FASHION_LeNet	FashionLeNet	X	public		Class
IHDELS	Ihdels	X	public		Class
LeNet	LeNet	X	public		Class
Metaheuristic	Metaheuristic	X	public	X	Class
MNIST	Mnist	X	public		Class
MNIST_LeNet	MnistLeNet	X	public		Class
MOS	Mos	X	public		Class
NetworkDesigner	NetworkDesigner	X	public	X	Class
ProblemBuilder	ProblemBuilder	X	public	X	Class
PSO_Solution	PsoSolution	X	public		Class
SAHC_BestSlope	SahcBestSlope	X	public		Class
SAHC_E2	SahcE2	X	public		Class
Solution	Solution	X	public	X	Class
Tools	Tools	X	public		Class

III.2.1.1 Class AlexNet

III.2.1.1.1 Card of class AlexNet

Name	AlexNet
Code	AlexNet
Comment	Modelo de la red neuronal convolucional profunda AlexNet
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.1.2 Header of the class AlexNet

```
/***************************************************************************
 * Module:  AlexNet.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class AlexNet
 ***************************************************************************/
```

III.2.1.1.3 Imports of the class AlexNet

```
import py.util.*;
```

III.2.1.1.4 List of generalizations of the class AlexNet

Name	Code	Parent Object	Child Object
Generalization_5	Generalization_5	NetworkDesigner	AlexNet

III.2.1.1.5 List of inherited methods of the class AlexNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	void	public	X			Network	

							Designer	

III.2.1.1.6 List of diagrams containing the class AlexNet

Name	Code
DNF_v1	DNF_v1

III.2.1.1.7 List of operations of the class AlexNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	Network	public				AlexNet	

III.2.1.2 Class CIFAR-10

III.2.1.2.1 Card of class CIFAR-10

Name	CIFAR-10
Code	Cifar10
Comment	Carga el conjunto de datos CIFAR-10
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.2.2 Header of the class CIFAR-10

```
/*********************************************************************
 * Module:  Cifar10.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Cifar10
 *********************************************************************/
```

III.2.1.2.3 Imports of the class CIFAR-10

```
import py.util.*;
```

III.2.1.2.4 List of generalizations of the class CIFAR-10

Name	Code	Parent Object	Child Object
Generalization_2	Generalization_2	Dataset	CIFAR-10

III.2.1.2.5 List of inherited methods of the class CIFAR-10

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_d	loadDat	void	public	X			Dataset	

ata	a							

III.2.1.2.6 List of diagrams containing the class CIFAR-10

Name	Code
DNF_v1	DNF_v1

III.2.1.2.7 List of attributes of the class CIFAR-10

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
LB	lb	float	public		CIFAR-10	
UB	ub	float	public		CIFAR-10	
input_size	inputSize	int	public		CIFAR-10	
output_size	outputSize	int	public		CIFAR-10	
img_size	imgSize	int	public		CIFAR-10	
path_cifar10	pathCifar10	CIFAR10 train files	public		CIFAR-10	
path_cifar10_test	pathCifar10Test	CIFAR10 test files	public		CIFAR-10	

III.2.1.2.8 List of operations of the class CIFAR-10

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_data	loadData	features, labels	public				CIFAR-10	
load_data_test	loadDataTest	features, labels	public				CIFAR-10	
load_label_names	loadLabelNames	String[]	public				CIFAR-10	
normalize	normalize	float[]	public				CIFAR-10	
one_hot_encode	oneHotEncode	int[]	public				CIFAR-10	
batch_features_labels	batchFeaturesLabels	features Batch	public				CIFAR-10	
load_preprocess_training_batch	loadPreprocessTrainingBatch	features LabelsBatch	public				CIFAR-10	

III.2.1.3 Class CIFAR10_AlexNet

III.2.1.3.1 Card of class CIFAR10_AlexNet

Name	CIFAR10_AlexNet
Code	Cifar10AlexNet
Comment	Implementa el problema: Red neuronal convolucional profunda AlexNet para resolver el problema CIFAR10
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.3.2 Header of the class CIFAR10_AlexNet

```
/***********************************************************************
 * Module:  Cifar10AlexNet.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Cifar10AlexNet
 ***********************************************************************/
```

III.2.1.3.3 Imports of the class CIFAR10_AlexNet

```
import py.util.*;
```

III.2.1.3.4 List of generalizations of the class CIFAR10_AlexNet

Name	Code	Parent Object	Child Object
Generalization_11	Generalization_11	ProblemBuilder	CIFAR10_AlexNet

III.2.1.3.5 List of inherited methods of the class CIFAR10_AlexNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
concatenate	concatenate	void	public	X			ProblemBuilder	
no_concatenate	noConcatenate	void	public	X			ProblemBuilder	
train_eval_test	trainEvalTest	void	public	X			ProblemBuilder	

III.2.1.3.6 List of inherited associations of the class CIFAR10_AlexNet

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	ProblemBuilder			1..1	1..1

Association_2	association2	Network Designer	Problem Builder			1..1	1..1

III.2.1.3.7 List of diagrams containing the class CIFAR10_AlexNet

Name	Code
DNF_v1	DNF_v1

III.2.1.3.8 List of attributes of the class CIFAR10_AlexNet

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
dataset	dataset	dataset object	private		CIFAR10_AlexNet	
net	net	net object	private		CIFAR10_AlexNet	
batch_size	batchSize	int	private		CIFAR10_AlexNet	
learning_rate	learningRate	float	private		CIFAR10_AlexNet	
optimizer	optimizer	RMSProp	private		CIFAR10_AlexNet	
test_size	testSize	int	private		CIFAR10_AlexNet	
num_classes	numClasses	int	private		CIFAR10_AlexNet	
LB	lb	float	private		CIFAR10_AlexNet	
UB	ub	float	private		CIFAR10_AlexNet	
input_size	inputSize	int	private		CIFAR10_AlexNet	
output_size	outputSize	int	private		CIFAR10_AlexNet	
img_size	imgSize	int	private		CIFAR10_AlexNet	
Vector_shapes	vectorShapes	int	private		CIFAR10_AlexNet	
size_of_solution	sizeOfSolution	int	private		CIFAR10_AlexNet	

III.2.1.3.9 List of operations of the class CIFAR10_AlexNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
train_e	trainEva	float	public				CIFAR1	

val_test	lTest						0_AlexNet	
init_weights	initWeights	float[]	public				CIFAR10_AlexNet	
slope_calc	slopeCalc	float	public				CIFAR10_AlexNet	

III.2.1.4 Class ClassicSolution

III.2.1.4.1 Card of class ClassicSolution

Name	ClassicSolution
Code	ClassicSolution
Comment	Posee los métodos para inicializar, evaluar, entrenar, probar una solución con una representación clásica de soluciones en los algoritmos evolutivos
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.4.2 Header of the class ClassicSolution

```
/***********************************************************************
 * Module:  ClassicSolution.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class ClassicSolution
 ***********************************************************************/
```

III.2.1.4.3 Imports of the class ClassicSolution

```
import py.util.*;
```

III.2.1.4.4 List of generalizations of the class ClassicSolution

Name	Code	Parent Object	Child Object
Generalization_15	Generalization_15	Solution	ClassicSolution

III.2.1.4.5 List of inherited methods of the class ClassicSolution

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
assign_values	assignValues	void	public	X			Solution	
copy_from	copyFrom	void	public	X			Solution	
copy_some_v	CopySomeVaria	void	public	X			Solution	

ariables	bles							
eval_fitness	eval_fitness	void	public	X			Solution	
eval_testing_data	eval_testing_data	void	public	X			Solution	
random_initialization	randomInitialization	void	public	X			Solution	

III.2.1.4.6 List of inherited associations of the class ClassicSolution

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_8	association8	Problem Builder	Solution			1..*	1..1

III.2.1.4.7 List of diagrams containing the class ClassicSolution

Name	Code
DNF_v1	DNF_v1

III.2.1.4.8 List of attributes of the class ClassicSolution

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
Wb	wb	float[]	public		ClassicSolution	
problem	problem	DNF Problem	public		ClassicSolution	
accuracy	accuracy	float	public		ClassicSolution	
fitness	fitness	float	public		ClassicSolution	
valslope	valslope	float	public		ClassicSolution	

III.2.1.4.9 List of operations of the class ClassicSolution

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
eval_fitness	eval_fitness	void	public	X			ClassicSolution	
assign_values	assignValues	void	public	X			ClassicSolution	
copy_fr	copyFro	void	public	X			ClassicS	

om	m						olution	
random_initialization	randomInitialization	void	public	X			ClassicSolution	
copy_some_variables	CopySomeVariables	void	public	X			ClassicSolution	
eval_testing_data	eval_testing_data	void	public	X			ClassicSolution	

III.2.1.5 Class ConventionalNN

III.2.1.5.1 Card of class ConventionalNN

Name	ConventionalNN
Code	ConventionalNN
Comment	Modelo de red neuronal convencional
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.5.2 Header of the class ConventionalNN

```
/***********************************************************************
 * Module:  ConventionalNN.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class ConventionalNN
 ***********************************************************************/
```

III.2.1.5.3 Imports of the class ConventionalNN

```
import py.util.*;
```

III.2.1.5.4 List of generalizations of the class ConventionalNN

Name	Code	Parent Object	Child Object
Generalization_9	Generalization_9	NetworkDesigner	ConventionalNN

III.2.1.5.5 List of inherited methods of the class ConventionalNN

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	void	public	X			NetworkDesigner	

III.2.1.5.6 List of diagrams containing the class ConventionalNN

Name	Code
DNF_v1	DNF_v1

III.2.1.5.7 List of operations of the class ConventionalNN

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	Network model	public				ConventionalNN	

III.2.1.6 Class Dataset

III.2.1.6.1 Card of class Dataset

Name	Dataset
Code	Dataset
Comment	Clase abstracta que permite cargar el conjunto de datos, pueden ser conjuntos de datos para algoritmos de regresión y clasificación
Generate	Yes
Visibility	public
Abstract	Yes
Classifier Type	Class

III.2.1.6.2 Header of the class Dataset

```
/***********************************************************************
 * Module:  Dataset.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Dataset
 ***********************************************************************/
```

III.2.1.6.3 Imports of the class Dataset

```
import py.util.*;
```

III.2.1.6.4 List of associations of the class Dataset

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	Problem Builder			1..1	1..1

III.2.1.6.5 List of specializations of the class Dataset

Name	Code	Parent Object	Child Object
Generalization_1	Generalization_1	Dataset	MNIST
Generalization_2	Generalization_2	Dataset	CIFAR-10

Generalization_3	Generalization_3	Dataset	FASHION-MNIST

III.2.1.6.6 List of diagrams containing the class Dataset

Name	Code
DNF_v1	DNF_v1

III.2.1.6.7 List of operations of the class Dataset

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_data	loadData	void	public	X			Dataset	

III.2.1.7 Class DNF

III.2.1.7.1 Card of class DNF

Name	DNF
Code	Dnf
Comment	Script que evalúa y ejecuta un experimento que une: una arquitectura de red neuronal profunda, una metaheurística y un conjunto de datos para obtener el objeto
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.7.2 Header of the class DNF

```
/***********************************************************************
 * Module:  Dnf.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Dnf
 ***********************************************************************/
```

III.2.1.7.3 Imports of the class DNF

```
import py.util.*;
```

III.2.1.7.4 List of associations of the class DNF

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_6	association6	Metaheuristic	DNF			1..1	1..1
Association_7	association7	Problem Builder	DNF			1..1	1..1

III.2.1.7.5 List of dependent objects of the class DNF

Name	Code	Influent Object	Dependent Object
Dependency_2	Dependency_2	DNF	Experimenter

III.2.1.7.6 List of navigable associations of the class DNF

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_6	association6	Metaheuristic	DNF			1..1	1..1
Association_7	association7	Problem Builder	DNF			1..1	1..1

III.2.1.7.7 List of diagrams containing the class DNF

Name	Code
DNF_v1	DNF_v1

III.2.1.7.8 List of attributes of the class DNF

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
ProblemId	problemId	int	public		DNF	
Type_of_Metaheuristic	typeOfMetaheuristic	int	public		DNF	
name_of_result	nameOfResult	String	public		DNF	
sess	sess	int	public		DNF	
Metaheuristic	metaheuristic	int	package		DNF	
ProblemBuilder	problemBuilder	int	package		DNF	

III.2.1.7.9 List of operations of the class DNF

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
main	main	void	public				DNF	

III.2.1.8 Class Experimenter

III.2.1.8.1 Card of class Experimenter

Name	Experimenter

Code	Experimenter
Comment	Script que ejecuta 30 veces un algoritmo que se compone de: tipo de metaheurística, tipo de problema: arquitectura de red neuronal y conjunto de datos
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.8.2 Header of the class Experimenter

```
/***********************************************************************
 * Module:  Experimenter.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Experimenter
 ***********************************************************************/
```

III.2.1.8.3 Imports of the class Experimenter

```
import py.util.*;
```

III.2.1.8.4 List of influent objects of the class Experimenter

Name	Code	Influent Object	Dependent Object
Dependency_2	Dependency_2	DNF	Experimenter

III.2.1.8.5 List of diagrams containing the class Experimenter

Name	Code
DNF_v1	DNF_v1

III.2.1.8.6 List of attributes of the class Experimenter

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
Type_of_Problem	typeOfProblem	int	public		Experimenter	
Type_of_Metaheuristic	typeOfMetaheuristic	int	public		Experimenter	

III.2.1.8.7 List of operations of the class Experimenter

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
DNF	dnf	float	public		X		Experimenter	

III.2.1.9 Class FASHION-MNIST

III.2.1.9.1 Card of class FASHION-MNIST

Name	FASHION-MNIST
Code	FashionMnist
Comment	Carga de los datos FASHION-MNIST: prendas de vestir
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.9.2 Header of the class FASHION-MNIST

```
/***********************************************************************
 * Module:  FashionMnist.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class FashionMnist
 ***********************************************************************/
```

III.2.1.9.3 Imports of the class FASHION-MNIST

```
import py.util.*;
```

III.2.1.9.4 List of generalizations of the class FASHION-MNIST

Name	Code	Parent Object	Child Object
Generalization_3	Generalization_3	Dataset	FASHION-MNIST

III.2.1.9.5 List of inherited methods of the class FASHION-MNIST

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_data	loadData	void	public	X			Dataset	

III.2.1.9.6 List of diagrams containing the class FASHION-MNIST

Name	Code
DNF_v1	DNF_v1

III.2.1.9.7 List of attributes of the class FASHION-MNIST

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
LB	lb	float	private		FASHION-MNIST	
UB	ub	float	private		FASHION-MNIST	

input_size	inputSize	int	private		FASHION-MNIST	
output_size	outputSize	int	private		FASHION-MNIST	
data	data	MNIST files	private		FASHION-MNIST	
img_size	imgSize	int	private		FASHION-MNIST	

III.2.1.9.8 List of operations of the class FASHION-MNIST

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_data	loadData	training and testing data	public				FASHION-MNIST	

III.2.1.10 Class FASHION_LeNet

III.2.1.10.1 Card of class FASHION_LeNet

Name	FASHION_LeNet
Code	FashionLeNet
Comment	Implementa el problema: Red neuronal convolucional profunda LeNet para resolver el problema FASHION-MNIST
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.10.2 Header of the class FASHION_LeNet

```
/***********************************************************************
 * Module:  FashionLeNet.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class FashionLeNet
 ***********************************************************************/
```

III.2.1.10.3 Imports of the class FASHION_LeNet

```
import py.util.*;
```

III.2.1.10.4 List of generalizations of the class FASHION_LeNet

Name	Code	Parent Object	Child Object
Generalization_13	Generalization_13	ProblemBuilder	FASHION_LeNet

III.2.1.10.5 List of inherited methods of the class FASHION_LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
concatenate	concatenate	void	public	X			Problem Builder	
no_concatenate	noConcatenate	void	public	X			Problem Builder	
train_eval_test	trainEvalTest	void	public	X			Problem Builder	

III.2.1.10.6 List of inherited associations of the class FASHION_LeNet

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	Problem Builder			1..1	1..1
Association_2	association2	Network Designer	Problem Builder			1..1	1..1

III.2.1.10.7 List of diagrams containing the class FASHION_LeNet

Name	Code
DNF_v1	DNF_v1

III.2.1.10.8 List of attributes of the class FASHION_LeNet

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
dataset	dataset	dataset object	private		FASHION_LeNet	
net	net	net object	private		FASHION_LeNet	
batch_size	batchSize	int	private		FASHION_LeNet	
learning_rate	learningRate	float	private		FASHION_LeNet	
optimizer	optimizer	RMSProp	private		FASHION_LeNet	
test_size	testSize	int	private		FASHION_LeNet	
num_classes	numClasses	int	private		FASHION_LeNet	
LB	lb	float	private		FASHION_LeNet	
UB	ub	float	private		FASHION_	

					LeNet	
input_size	inputSize	int	private		FASHION_LeNet	
output_size	outputSize	int	private		FASHION_LeNet	
img_size	imgSize	int	private		FASHION_LeNet	
Vector_shapes	vectorShapes	int	private		FASHION_LeNet	
size_of_solution	sizeOfSolution	int	private		FASHION_LeNet	

III.2.1.10.9 List of operations of the class FASHION_LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
train_eval_test	trainEvalTest	float	public				FASHION_LeNet	
init_weights	initWeights	float[]	public				FASHION_LeNet	
slope_calc	slopeCalc	float	public				FASHION_LeNet	

III.2.1.11 Class IHDELS

III.2.1.11.1 Card of class IHDELS

Name	IHDELS
Code	Ihdels
Comment	Iterative hybridization of DE with local search
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.11.2 Header of the class IHDELS

```
/***************************************************************************
 * Module:  Ihdels.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Ihdels
 ***************************************************************************/
```

III.2.1.11.3 Imports of the class IHDELS

import py.util.*;

III.2.1.11.4 List of generalizations of the class IHDELS

Name	Code	Parent Object	Child Object
Generalization_12	Generalization_12	Metaheuristic	IHDELS

III.2.1.11.5 List of inherited methods of the class IHDELS

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	void	public	X			Metaheuristic	

III.2.1.11.6 List of inherited associations of the class IHDELS

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1

III.2.1.11.7 List of diagrams containing the class IHDELS

Name	Code
DNF_v1	DNF_v1

III.2.1.11.8 List of attributes of the class IHDELS

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
session	session	TensorFlow Session	public		IHDELS	
problem	problem	DNF Problem	public		IHDELS	
pop_size	popSize	int	public		IHDELS	
maxevals	maxevals	int	public		IHDELS	
training_epochs	trainingEpochs	int	public		IHDELS	
num_blocks	numBlocks	int	public		IHDELS	
experiment	experiment	int	public		IHDELS	
seed	seed	int	public		IHDELS	
best	best	Solution	public		IHDELS	
testing	testing	boolean	public		IHDELS	

efos	efos	int	public		IHDELS	

III.2.1.11.9 List of operations of the class IHDELS

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
apply_minimize	applyMinimize	Solution	public				IHDELS	
run	run	Solution	public				IHDELS	
get_group_variables	getGroupVariables	int[]	public				IHDELS	

III.2.1.12 Class LeNet

III.2.1.12.1 Card of class LeNet

Name	LeNet
Code	LeNet
Comment	Modelo de la red neuronal convolucional LeNet
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.12.2 Header of the class LeNet

```
/***********************************************************************
 * Module:  LeNet.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class LeNet
 ***********************************************************************/
```

III.2.1.12.3 Imports of the class LeNet

```
import py.util.*;
```

III.2.1.12.4 List of generalizations of the class LeNet

Name	Code	Parent Object	Child Object
Generalization_4	Generalization_4	NetworkDesigner	LeNet

III.2.1.12.5 List of inherited methods of the class LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	void	public	X			Network	

							Designer	

III.2.1.12.6 List of diagrams containing the class LeNet

Name	Code
DNF_v1	DNF_v1

III.2.1.12.7 List of operations of the class LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	Network	public				LeNet	

III.2.1.13 Class Metaheuristic

III.2.1.13.1 Card of class Metaheuristic

Name	Metaheuristic
Code	Metaheuristic
Comment	Clase abstracta que permite la implementación run para cada metaheurística que se implemente en DNF
Generate	Yes
Visibility	public
Abstract	Yes
Classifier Type	Class

III.2.1.13.2 Header of the class Metaheuristic

```
/***********************************************************************
 * Module:  Metaheuristic.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Metaheuristic
 ***********************************************************************/
```

III.2.1.13.3 Imports of the class Metaheuristic

```
import py.util.*;
```

III.2.1.13.4 List of associations of the class Metaheuristic

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1
Associat	associatio	Metaheur	DNF			1..1	1..1

ion_6	n6	istic					

III.2.1.13.5 List of specializations of the class Metaheuristic

Name	Code	Parent Object	Child Object
Generalization_6	Generalization_6	Metaheuristic	SAHC_BestSlope
Generalization_7	Generalization_7	Metaheuristic	MOS
Generalization_12	Generalization_12	Metaheuristic	IHDELS
Generalization_16	Generalization_16	Metaheuristic	SAHC_E2

III.2.1.13.6 List of influent objects of the class Metaheuristic

Name	Code	Influent Object	Dependent Object
Dependency_1	Dependency_1	Tools	Metaheuristic

III.2.1.13.7 List of navigable associations of the class Metaheuristic

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1

III.2.1.13.8 List of diagrams containing the class Metaheuristic

Name	Code
DNF_v1	DNF_v1

III.2.1.13.9 List of operations of the class Metaheuristic

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	void	public	X			Metaheuristic	

III.2.1.14 Class MNIST

III.2.1.14.1 Card of class MNIST

Name	MNIST
Code	Mnist
Comment	Carga de los datos MNIST: escritura de dígitos a mano alzada
Generate	Yes
Visibility	public

Abstract	No
Classifier Type	Class

III.2.1.14.2 Header of the class MNIST

```
/***********************************************************************
 * Module:  Mnist.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Mnist
 ***********************************************************************/
```

III.2.1.14.3 Imports of the class MNIST

```
import py.util.*;
```

III.2.1.14.4 List of generalizations of the class MNIST

Name	Code	Parent Object	Child Object
Generalization_1	Generalization_1	Dataset	MNIST

III.2.1.14.5 List of inherited methods of the class MNIST

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
load_data	loadData	void	public	X			Dataset	

III.2.1.14.6 List of diagrams containing the class MNIST

Name	Code
DNF_v1	DNF_v1

III.2.1.14.7 List of attributes of the class MNIST

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
LB	lb	float	private		MNIST	
UB	ub	float	private		MNIST	
input_size	inputSize	int	private		MNIST	
output_size	outputSize	int	private		MNIST	
data	data	MNIST files	private		MNIST	
img_size	imgSize	int	private		MNIST	

III.2.1.14.8 List of operations of the class MNIST

Name	Code	Return	Visibilit	Abstract	Final	Static	Classifie	Read-

		Type	y				r	only
load_data	loadData	training and testing data	public				MNIST	

III.2.1.15 Class MNIST_LeNet

III.2.1.15.1 Card of class MNIST_LeNet

Name	MNIST_LeNet
Code	MnistLeNet
Comment	Implementa el problema: Red neuronal convolucional profunda LeNet para resolver el problema MNIST
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.15.2 Header of the class MNIST_LeNet

```
/***********************************************************************
 * Module:  MnistLeNet.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class MnistLeNet
 ***********************************************************************/
```

III.2.1.15.3 Imports of the class MNIST_LeNet

```
import py.util.*;
```

III.2.1.15.4 List of generalizations of the class MNIST_LeNet

Name	Code	Parent Object	Child Object
Generalization_10	Generalization_10	ProblemBuilder	MNIST_LeNet

III.2.1.15.5 List of inherited methods of the class MNIST_LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
concatenate	concatenate	void	public	X			Problem Builder	
no_concatenate	noConcatenate	void	public	X			Problem Builder	
train_eval_test	trainEvalTest	void	public	X			Problem Builder	

III.2.1.15.6 List of inherited associations of the class MNIST_LeNet

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	Problem Builder			1..1	1..1
Association_2	association2	Network Designer	Problem Builder			1..1	1..1

III.2.1.15.7 List of diagrams containing the class MNIST_LeNet

Name	Code
DNF_v1	DNF_v1

III.2.1.15.8 List of attributes of the class MNIST_LeNet

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
dataset	dataset	dataset object	private		MNIST_LeNet	
net	net	net object	private		MNIST_LeNet	
batch_size	batchSize	int	private		MNIST_LeNet	
learning_rate	learningRate	float	private		MNIST_LeNet	
optimizer	optimizer	RMSProp	private		MNIST_LeNet	
test_size	testSize	int	private		MNIST_LeNet	
num_classes	numClasses	int	private		MNIST_LeNet	
LB	lb	float	private		MNIST_LeNet	
UB	ub	float	private		MNIST_LeNet	
input_size	inputSize	int	private		MNIST_LeNet	
output_size	outputSize	int	private		MNIST_LeNet	
img_size	imgSize	int	private		MNIST_LeNet	
Vector_shapes	vectorShapes	int	private		MNIST_LeNet	
size_of_solution	sizeOfSolution	int	private		MNIST_LeNet	

III.2.1.15.9 List of operations of the class MNIST_LeNet

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
train_eval_test	trainEvalTest	float	public				MNIST_LeNet	
init_weights	initWeights	float[]	public				MNIST_LeNet	
slope_calc	slopeCalc	float	public				MNIST_LeNet	

III.2.1.16 Class MOS

III.2.1.16.1 Card of class MOS

Name	MOS
Code	Mos
Comment	Implementación del algoritmo: Multiple Offspring Sampling
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.16.2 Header of the class MOS

```
/***********************************************************************
 * Module:  Mos.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Mos
 ***********************************************************************/
```

III.2.1.16.3 Imports of the class MOS

```
import py.util.*;
```

III.2.1.16.4 List of generalizations of the class MOS

Name	Code	Parent Object	Child Object
Generalization_7	Generalization_7	Metaheuristic	MOS

III.2.1.16.5 List of inherited methods of the class MOS

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	void	public	X			Metaheuristic	

III.2.1.16.6 List of inherited associations of the class MOS

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1

III.2.1.16.7 List of diagrams containing the class MOS

Name	Code
DNF_v1	DNF_v1

III.2.1.16.8 List of attributes of the class MOS

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
session	session	Tensorflow Session	public		MOS	
problem	problem	Problem object	public		MOS	
max_efos	maxEfos	int	public		MOS	
efos_by_iteration	efosByIteration	int	public		MOS	
initial_training_epochs	initialTrainingEpochs	int	public		MOS	
training_epochs_rmsprop	trainingEpochsRmsprop	int	public		MOS	
training_epochs_rmsprop_hc	trainingEpochsRmspropHc	int	public		MOS	
initial_radio_mts	initialRadioMts	float	public		MOS	
initial_radio_rmsprop	initialRadioRmsprop	float	public		MOS	
experiment	experiment	int	public		MOS	
seed	seed	int	public		MOS	
best	best	Solution	public		MOS	
efos	efos	int	public		MOS	
testing	testing	boolean	public		MOS	

III.2.1.16.9 List of operations of the class MOS

Name	Code	Return	Visibilit	Abstract	Final	Static	Classifie	Read-

		Type	y				r	only
run	run	int	public				MOS	
recalculate_participation_ratios	recalculateParticipationRatios	int	public				MOS	
calculate_deltas	calculateDeltas	int	public				MOS	
calculate_eta	calculateEta	int	public				MOS	

III.2.1.17 Class NetworkDesigner

III.2.1.17.1 Card of class NetworkDesigner

Name	NetworkDesigner
Code	NetworkDesigner
Comment	Clase abstracta que permite el diseño e implementación del modelo de la red neuronal
Generate	Yes
Visibility	public
Abstract	Yes
Classifier Type	Class

III.2.1.17.2 Header of the class NetworkDesigner

```
/***********************************************************************
 * Module:  NetworkDesigner.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class NetworkDesigner
 ***********************************************************************/
```

III.2.1.17.3 Imports of the class NetworkDesigner

import py.util.*;

III.2.1.17.4 List of associations of the class NetworkDesigner

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_2	association2	Network Designer	Problem Builder			1..1	1..1

III.2.1.17.5 List of specializations of the class NetworkDesigner

Name	Code	Parent Object	Child Object
Generalization_4	Generalization_4	NetworkDesigner	LeNet

Generalization_5	Generalization_5	NetworkDesigner	AlexNet
Generalization_9	Generalization_9	NetworkDesigner	ConventionalNN

III.2.1.17.6 List of diagrams containing the class NetworkDesigner

Name	Code
DNF_v1	DNF_v1

III.2.1.17.7 List of operations of the class NetworkDesigner

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
model	model	void	public	X			NetworkDesigner	

III.2.1.18 Class ProblemBuilder

III.2.1.18.1 Card of class ProblemBuilder

Name	ProblemBuilder
Code	ProblemBuilder
Comment	Clase abstracta que permite construir el problema a optimizar, el problema se compone del conjunto de datos y la arquitectura de la red neuronal
Generate	Yes
Visibility	public
Abstract	Yes
Classifier Type	Class

III.2.1.18.2 Header of the class ProblemBuilder

```
/***********************************************************************
 * Module:  ProblemBuilder.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class ProblemBuilder
 ***********************************************************************/
```

III.2.1.18.3 Imports of the class ProblemBuilder

```
import py.util.*;
```

III.2.1.18.4 List of associations of the class ProblemBuilder

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	ProblemBuilder			1..1	1..1
Associat	associatio	Network	Problem			1..1	1..1

ion_2	n2	Designer	Builder				
Association_5	association5	ProblemBuilder	Metaheuristic			1..*	1..1
Association_7	association7	ProblemBuilder	DNF			1..1	1..1
Association_8	association8	ProblemBuilder	Solution			1..*	1..1

III.2.1.18.5 List of specializations of the class ProblemBuilder

Name	Code	Parent Object	Child Object
Generalization_10	Generalization_10	ProblemBuilder	MNIST_LeNet
Generalization_11	Generalization_11	ProblemBuilder	CIFAR10_AlexNet
Generalization_13	Generalization_13	ProblemBuilder	FASHION_LeNet

III.2.1.18.6 List of navigable associations of the class ProblemBuilder

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_1	association1	Dataset	ProblemBuilder			1..1	1..1
Association_2	association2	NetworkDesigner	ProblemBuilder			1..1	1..1

III.2.1.18.7 List of diagrams containing the class ProblemBuilder

Name	Code
DNF_v1	DNF_v1

III.2.1.18.8 List of operations of the class ProblemBuilder

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
train_eval_test	trainEvalTest	void	public	X			ProblemBuilder	
concatenate	concatenate	void	public	X			ProblemBuilder	
no_concatenate	noConcatenate	void	public	X			ProblemBuilder	

III.2.1.19 Class PSO_Solution

III.2.1.19.1 Card of class PSO_Solution

Name	PSO_Solution

Code	PsoSolution
Comment	Diseño de la solución adaptada para el algoritmo Particle Swarm Optimization
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.19.2 Header of the class PSO_Solution

```
/***********************************************************************
 * Module:  PsoSolution.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class PsoSolution
 ***********************************************************************/
```

III.2.1.19.3 Imports of the class PSO_Solution

```
import py.util.*;
```

III.2.1.19.4 List of generalizations of the class PSO_Solution

Name	Code	Parent Object	Child Object
Generalization_14	Generalization_14	Solution	PSO_Solution

III.2.1.19.5 List of inherited methods of the class PSO_Solution

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
assign_values	assignValues	void	public	X			Solution	
copy_from	copyFrom	void	public	X			Solution	
copy_some_variables	CopySomeVariables	void	public	X			Solution	
eval_fitness	eval_fitness	void	public	X			Solution	
eval_testing_data	eval_testing_data	void	public	X			Solution	
random_initialization	randomInitialization	void	public	X			Solution	

III.2.1.19.6 List of inherited associations of the class PSO_Solution

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B

Association_8	association8	Problem Builder	Solution			1..*	1..1

III.2.1.19.7 List of diagrams containing the class PSO_Solution

Name	Code
DNF_v1	DNF_v1

III.2.1.19.8 List of attributes of the class PSO_Solution

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
LB	lb	float	public		PSO_Solution	
UB	ub	float	public		PSO_Solution	
size_of_solution	sizeOfSolution	int	public		PSO_Solution	
accuracy	accuracy	float	public		PSO_Solution	
valslope	valslope	float	public		PSO_Solution	
Wb	wb	float	public		PSO_Solution	
fitness	fitness	float	public		PSO_Solution	
velocity	velocity	float	public		PSO_Solution	
Wb_best	wbBest	Solution	public		PSO_Solution	
fitness_best	fitnessBest	float	public		PSO_Solution	
c1	c1	float	public		PSO_Solution	
c2	c2	float	public		PSO_Solution	
w	w	float	public		PSO_Solution	

III.2.1.19.9 List of operations of the class PSO_Solution

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
update_velocity	updateVelocity	float[]	public				PSO_Solution	

update_position	updatePosition	float[]	public				PSO_Solution	
eval_fitness	eval_fitness	void	public				PSO_Solution	
assign_values	assignValues	void	public				PSO_Solution	
copy_from	copyFrom	void	public				PSO_Solution	
random_initialization	randomInitialization	void	public				PSO_Solution	
copy_some_variables	CopySomeVariables	void	public				PSO_Solution	
eval_testing_data	eval_testing_data	void	public				PSO_Solution	

III.2.1.20 Class SAHC_BestSlope

III.2.1.20.1 Card of class SAHC_BestSlope

Name	SAHC_BestSlope
Code	SahcBestSlope
Comment	Ejecuta el enfoque basado en lanzar procesos de exploración teniendo en cuenta el último valor de pendiente al final del batch
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.20.2 Header of the class SAHC_BestSlope

```
/***********************************************************************
 * Module:  SahcBestSlope.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class SahcBestSlope
 ***********************************************************************/
```

III.2.1.20.3 Imports of the class SAHC_BestSlope

import py.util.*;

III.2.1.20.4 List of generalizations of the class SAHC_BestSlope

Name	Code	Parent Object	Child Object
Generalization_6	Generalization_6	Metaheuristic	SAHC_BestSlope

III.2.1.20.5 List of inherited methods of the class SAHC_BestSlope

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	void	public	X			Metaheuristic	

III.2.1.20.6 List of inherited associations of the class SAHC_BestSlope

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	ProblemBuilder	Metaheuristic			1..*	1..1

III.2.1.20.7 List of diagrams containing the class SAHC_BestSlope

Name	Code
DNF_v1	DNF_v1

III.2.1.20.8 List of attributes of the class SAHC_BestSlope

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
session	session	TensorflowSession	public		SAHC_BestSlope	
problem	problem	ProblemDNF	public		SAHC_BestSlope	
max_efos	maxEfos	int	public		SAHC_BestSlope	
training_epochs	trainingEpochs	int	public		SAHC_BestSlope	
size_of_buffer_cv	sizeOfBufferCv	int	public		SAHC_BestSlope	
learning_rate	learningRate	float	public		SAHC_BestSlope	
experiment	experiment	int	public		SAHC_BestSlope	
seed	seed	int	public		SAHC_BestSlope	
best	best	Solution	public		SAHC_BestSlope	
list_fitness	listFitness	float	public		SAHC_BestSlope	
efos	efos	int	public		SAHC_BestSlope	

III.2.1.20.9 List of operations of the class SAHC_BestSlope

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	float	public				SAHC_BestSlope	

III.2.1.21 Class SAHC_E2

III.2.1.21.1 Card of class SAHC_E2

Name	SAHC_E2
Code	SahcE2
Comment	Enfoque memético: RMSProp (Explotación) y SAHC (Exploración) basado en analizar el comportamiento de la pendiente en el entrenamiento con el fin de lanzar procesos de exploración inteligentes cuando el RMSProp quede atrapado
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.21.2 Header of the class SAHC_E2

```
/***********************************************************************
 * Module:  SahcE2.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class SahcE2
 ***********************************************************************/
```

III.2.1.21.3 Imports of the class SAHC_E2

import py.util.*;

III.2.1.21.4 List of generalizations of the class SAHC_E2

Name	Code	Parent Object	Child Object
Generalization_16	Generalization_16	Metaheuristic	SAHC_E2

III.2.1.21.5 List of inherited methods of the class SAHC_E2

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	void	public	X			Metaheuristic	

III.2.1.21.6 List of inherited associations of the class SAHC_E2

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	Problem Builder	Metaheuristic			1..*	1..1

III.2.1.21.7 List of diagrams containing the class SAHC_E2

Name	Code
DNF_v1	DNF_v1

III.2.1.21.8 List of attributes of the class SAHC_E2

Name	Code	Data Type	Visibility	Initial Value	Classifier	Read-only
session	session	Tensorflow Session	public		SAHC_E2	
problem	problem	Problem DNF	public		SAHC_E2	
max_efos	maxEfos	int	public		SAHC_E2	
training_epochs	trainingEpochs	int	public		SAHC_E2	
size_of_buffer_cv	sizeOfBufferCv	int	public		SAHC_E2	
learning_rate	learningRate	float	public		SAHC_E2	
experiment	experiment	int	public		SAHC_E2	
seed	seed	int	public		SAHC_E2	
best	best	Solution	public		SAHC_E2	
list_fitness	listFitness	float	public		SAHC_E2	
efos	efos	int	public		SAHC_E2	

III.2.1.21.9 List of operations of the class SAHC_E2

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
run	run	float	public				SAHC_E2	

III.2.1.22 Class Solution

III.2.1.22.1 Card of class Solution

Name	Solution
Code	Solution
Comment	Clase abstracta que implementa todos los métodos requeridos para crear una solución de representación clásica que permita lograr una sinergia entre las metaheurísticas y el aprendizaje profundo
Generate	Yes
Visibility	public
Abstract	Yes
Classifier Type	Class

III.2.1.22.2 Header of the class Solution

```
/***********************************************************************
 * Module:  Solution.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Solution
 ***********************************************************************/
```

III.2.1.22.3 Imports of the class Solution

```
import py.util.*;
```

III.2.1.22.4 List of associations of the class Solution

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_8	association8	Problem Builder	Solution			1..*	1..1

III.2.1.22.5 List of specializations of the class Solution

Name	Code	Parent Object	Child Object
Generalization_14	Generalization_14	Solution	PSO_Solution
Generalization_15	Generalization_15	Solution	ClassicSolution

III.2.1.22.6 List of objects using the class as type of the class Solution

Name	Code
apply_minimize	applyMinimize
best	best
best	best
best	best
best	best
current_best	currentBest
other	other
other	other
other	other

other	other
other	other
other	other
run	run
Wb_best	wbBest

III.2.1.22.7 List of navigable associations of the class Solution

Name	Code	Class B	Class A	Role A	Role B	Multiplicity A	Multiplicity B
Association_8	association8	Problem Builder	Solution			1..*	1..1

III.2.1.22.8 List of diagrams containing the class Solution

Name	Code
DNF_v1	DNF_v1

III.2.1.22.9 List of operations of the class Solution

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
eval_fitness	eval_fitness	void	public	X			Solution	
assign_values	assignValues	void	public	X			Solution	
copy_from	copyFrom	void	public	X			Solution	
random_initialization	randomInitialization	void	public	X			Solution	
copy_some_variables	CopySomeVariables	void	public	X			Solution	
eval_testing_data	eval_testing_data	void	public	X			Solution	

III.2.1.23 Class Tools

III.2.1.23.1 Card of class Tools

Name	Tools
Code	Tools
Comment	Módulo compuesto por 4 clases:

	Clase Log: encargada de generar reportes de los resultados de los entrenamiento y consumo de memoria RAM del PC Clase measure_memory: mide la cantidad de memoria RAM consumida durante cada iteración de la metaheurística Clase Operations: Contiene funciones para calcular promedios y cálculo de pendientes Clase PoolProb: Calcula valores de probabilidad para la ejecución de diferentes metaheurísticas en el algoritmo IHDELS
Generate	Yes
Visibility	public
Abstract	No
Classifier Type	Class

III.2.1.23.2 Header of the class Tools

```
/***********************************************************************
 * Module:  Tools.py
 * Author:  Julián Fernando Muñoz Ordóñez, Carlos Alberto Cobos Lozada, Martha Eliana Mendoza Becerra
 * Purpose: Defines the Class Tools
 ***********************************************************************/
```

III.2.1.23.3 Imports of the class Tools

```
import py.util.*;
```

III.2.1.23.4 List of dependent objects of the class Tools

Name	Code	Influent Object	Dependent Object
Dependency_1	Dependency_1	Tools	Metaheuristic

III.2.1.23.5 List of diagrams containing the class Tools

Name	Code
DNF_v1	DNF_v1

III.2.1.23.6 List of operations of the class Tools

Name	Code	Return Type	Visibility	Abstract	Final	Static	Classifier	Read-only
store_initial_time	storeInitialTime	void	public				Tools	
store_results	storeResults	void	public				Tools	
store_accuracy	storeAccuracy	void	public				Tools	
Average	average	float	public				Tools	
slope_c	slopeCal	float	public				Tools	

alc	c							

III.2.2List of associations in diagram

Name	*Code*	*Class B*	*Class A*	*Role A*	*Role B*	*Multiplicity A*	*Multiplicity B*
Association_1	association1	Dataset	ProblemBuilder			1..1	1..1
Association_2	association2	NetworkDesigner	ProblemBuilder			1..1	1..1
Association_3	association3	Solution	Metaheuristic			0..*	1..*
Association_5	association5	ProblemBuilder	Metaheuristic			1..*	1..1
Association_6	association6	Metaheuristic	DNF			1..1	1..1
Association_7	association7	ProblemBuilder	DNF			1..1	1..1
Association_8	association8	ProblemBuilder	Solution			1..*	1..1

III.2.2.1 Association Association_1

III.2.2.1.1 Card of association Association_1

Name	Association_1
Code	association1
Comment	
Class A	ProblemBuilder
Class B	Dataset
Role A	
Role B	
Multiplicity A	1..1
Multiplicity B	1..1

III.2.2.1.2 List of diagrams containing the association Association_1

Name	Code
DNF_v1	DNF_v1

III.2.2.2 Association Association_2

III.2.2.2.1 Card of association Association_2

Name	Association_2
Code	association2
Comment	
Class A	ProblemBuilder

Class B	NetworkDesigner
Role A	
Role B	
Multiplicity A	1..1
Multiplicity B	1..1

III.2.2.2.2 List of diagrams containing the association Association_2

Name	Code
DNF_v1	DNF_v1

III.2.2.3 Association Association_3

III.2.2.3.1 Card of association Association_3

Name	Association_3
Code	association3
Comment	
Class A	Metaheuristic
Class B	Solution
Role A	
Role B	
Multiplicity A	0..*
Multiplicity B	1..*

III.2.2.3.2 List of diagrams containing the association Association_3

Name	Code
DNF_v1	DNF_v1

III.2.2.4 Association Association_5

III.2.2.4.1 Card of association Association_5

Name	Association_5
Code	association5
Comment	
Class A	Metaheuristic
Class B	ProblemBuilder
Role A	
Role B	
Multiplicity A	1..*
Multiplicity B	1..1

III.2.2.4.2 List of diagrams containing the association Association_5

Name	Code
DNF_v1	DNF_v1

III.2.2.5 Association Association_6

III.2.2.5.1 Card of association Association_6

Name	Association_6
Code	association6
Comment	
Class A	DNF
Class B	Metaheuristic
Role A	
Role B	
Multiplicity A	1..1
Multiplicity B	1..1

III.2.2.5.2 List of diagrams containing the association Association_6

Name	Code
DNF_v1	DNF_v1

III.2.2.6 Association Association_7

III.2.2.6.1 Card of association Association_7

Name	Association_7
Code	association7
Comment	
Class A	DNF
Class B	ProblemBuilder
Role A	
Role B	
Multiplicity A	1..1
Multiplicity B	1..1

III.2.2.6.2 List of diagrams containing the association Association_7

Name	Code
DNF_v1	DNF_v1

III.2.2.7 Association Association_8

III.2.2.7.1 Card of association Association_8

Name	Association_8

Code	association8
Comment	
Class A	Solution
Class B	ProblemBuilder
Role A	
Role B	
Multiplicity A	1..*
Multiplicity B	1..1

III.2.2.7.2 List of diagrams containing the association Association_8

Name	Code
DNF_v1	DNF_v1

III.2.3List of generalizations in diagram

Name	*Code*	*Parent Object*	*Child Object*
Generalization_1	Generalization_1	Dataset	MNIST
Generalization_2	Generalization_2	Dataset	CIFAR-10
Generalization_3	Generalization_3	Dataset	FASHION-MNIST
Generalization_4	Generalization_4	NetworkDesigner	LeNet
Generalization_5	Generalization_5	NetworkDesigner	AlexNet
Generalization_6	Generalization_6	Metaheuristic	SAHC_BestSlope
Generalization_7	Generalization_7	Metaheuristic	MOS
Generalization_9	Generalization_9	NetworkDesigner	ConventionalNN
Generalization_10	Generalization_10	ProblemBuilder	MNIST_LeNet
Generalization_11	Generalization_11	ProblemBuilder	CIFAR10_AlexNet
Generalization_12	Generalization_12	Metaheuristic	IHDELS
Generalization_13	Generalization_13	ProblemBuilder	FASHION_LeNet
Generalization_14	Generalization_14	Solution	PSO_Solution
Generalization_15	Generalization_15	Solution	ClassicSolution
Generalization_16	Generalization_16	Metaheuristic	SAHC_E2

III.2.3.1 Generalization Generalization_1

III.2.3.1.1 Card of generalization Generalization_1

Name	Generalization_1
Code	Generalization_1
Comment	
Child Object	MNIST
Parent Object	Dataset

III.2.3.1.2 List of diagrams containing the generalization Generalization_1

Name	Code
DNF_v1	DNF_v1

III.2.3.2 Generalization Generalization_2

III.2.3.2.1 Card of generalization Generalization_2

Name	Generalization_2
Code	Generalization_2
Comment	
Child Object	CIFAR-10
Parent Object	Dataset

III.2.3.2.2 List of diagrams containing the generalization Generalization_2

Name	Code
DNF_v1	DNF_v1

III.2.3.3 Generalization Generalization_3

III.2.3.3.1 Card of generalization Generalization_3

Name	Generalization_3
Code	Generalization_3
Comment	
Child Object	FASHION-MNIST
Parent Object	Dataset

III.2.3.3.2 List of diagrams containing the generalization Generalization_3

Name	Code
DNF_v1	DNF_v1

III.2.3.4 Generalization Generalization_4

III.2.3.4.1 Card of generalization Generalization_4

Name	Generalization_4
Code	Generalization_4
Comment	
Child Object	LeNet
Parent Object	NetworkDesigner

III.2.3.4.2 List of diagrams containing the generalization Generalization_4

Name	Code
DNF_v1	DNF_v1

III.2.3.5 Generalization Generalization_5

III.2.3.5.1 Card of generalization Generalization_5

Name	Generalization_5
Code	Generalization_5
Comment	
Child Object	AlexNet
Parent Object	NetworkDesigner

III.2.3.5.2 List of diagrams containing the generalization Generalization_5

Name	Code
DNF_v1	DNF_v1

III.2.3.6 Generalization Generalization_6

III.2.3.6.1 Card of generalization Generalization_6

Name	Generalization_6
Code	Generalization_6
Comment	
Child Object	SAHC_BestSlope
Parent Object	Metaheuristic

III.2.3.6.2 List of diagrams containing the generalization Generalization_6

Name	Code
DNF_v1	DNF_v1

III.2.3.7 Generalization Generalization_7

III.2.3.7.1 Card of generalization Generalization_7

Name	Generalization_7
Code	Generalization_7
Comment	
Child Object	MOS
Parent Object	Metaheuristic

III.2.3.7.2 List of diagrams containing the generalization Generalization_7

Name	Code
DNF_v1	DNF_v1

III.2.3.8 Generalization Generalization_9

III.2.3.8.1 Card of generalization Generalization_9

Name	Generalization_9
Code	Generalization_9
Comment	
Child Object	ConventionalNN
Parent Object	NetworkDesigner

III.2.3.8.2 List of diagrams containing the generalization Generalization_9

Name	Code
DNF_v1	DNF_v1

III.2.3.9 Generalization Generalization_10

III.2.3.9.1 Card of generalization Generalization_10

Name	Generalization_10
Code	Generalization_10
Comment	
Child Object	MNIST_LeNet
Parent Object	ProblemBuilder

III.2.3.9.2 List of diagrams containing the generalization Generalization_10

Name	Code
DNF_v1	DNF_v1

III.2.3.10 Generalization Generalization_11

III.2.3.10.1 Card of generalization Generalization_11

Name	Generalization_11
Code	Generalization_11
Comment	
Child Object	CIFAR10_AlexNet
Parent Object	ProblemBuilder

III.2.3.10.2 List of diagrams containing the generalization Generalization_11

Name	Code
DNF_v1	DNF_v1

III.2.3.11 Generalization Generalization_12

III.2.3.11.1 Card of generalization Generalization_12

Name	Generalization_12
Code	Generalization_12
Comment	
Child Object	IHDELS
Parent Object	Metaheuristic

III.2.3.11.2 List of diagrams containing the generalization Generalization_12

Name	Code
DNF_v1	DNF_v1

III.2.3.12 Generalization Generalization_13

III.2.3.12.1 Card of generalization Generalization_13

Name	Generalization_13
Code	Generalization_13
Comment	
Child Object	FASHION_LeNet
Parent Object	ProblemBuilder

III.2.3.12.2 List of diagrams containing the generalization Generalization_13

Name	Code
DNF_v1	DNF_v1

III.2.3.13 Generalization Generalization_14

III.2.3.13.1 Card of generalization Generalization_14

Name	Generalization_14
Code	Generalization_14
Comment	
Child Object	PSO_Solution
Parent Object	Solution

III.2.3.13.2 List of diagrams containing the generalization Generalization_14

Name	Code
DNF_v1	DNF_v1

III.2.3.14 Generalization Generalization_15

III.2.3.14.1 Card of generalization Generalization_15

Name	Generalization_15
Code	Generalization_15
Comment	
Child Object	ClassicSolution
Parent Object	Solution

III.2.3.14.2 List of diagrams containing the generalization Generalization_15

Name	Code
DNF_v1	DNF_v1

III.2.3.15 Generalization Generalization_16

III.2.3.15.1 Card of generalization Generalization_16

Name	Generalization_16
Code	Generalization_16
Comment	
Child Object	SAHC_E2
Parent Object	Metaheuristic

III.2.3.15.2 List of diagrams containing the generalization Generalization_16

Name	Code
DNF_v1	DNF_v1

III.2.4List of dependencies in diagram

Name	*Code*	*Influent Object*	*Dependent Object*
Dependency_1	Dependency_1	Tools	Metaheuristic
Dependency_2	Dependency_2	DNF	Experimenter

III.2.4.1 Dependency Dependency_1

III.2.4.1.1 Card of dependency Dependency_1

Name	Dependency_1
Code	Dependency_1
Comment	
Dependent Object	Metaheuristic
Influent Object	Tools

III.2.4.1.2 List of diagrams containing the dependency Dependency_1

Name	Code
DNF_v1	DNF_v1

III.2.4.2 Dependency Dependency_2

III.2.4.2.1 Card of dependency Dependency_2

Name	Dependency_2
Code	Dependency_2
Comment	
Dependent Object	Experimenter
Influent Object	DNF

III.2.4.2.2 List of diagrams containing the dependency Dependency_2

Name	Code
DNF_v1	DNF_v1