Genetic Neural Optimizer

Release 1.0

Raúl Reguillo Carmona

Sep 19, 2019

CONTENTS

Python Module Index 7

Index 9

class genetic_optimizer.**GeneticNeuralOptimizer**(*model*, *mutation_prob=0.5*, *mutation_rate=None*, *elite=2*, *genetic_train=True*, *stop_condition=None*, *epochs=-1*, *fitness_function=<function GeneticNeuralOptimizer.<lambda>>*, *mutables=1*, *original_mutation_prob=0.5*, *pop_dist='uniform'*, *mutation_dist=<built-in method uniform of mtrand.RandomState object>*, *min_delta=1.0*, *regularization_metric='val_acc'*, *iterations=None*)

GeneticNeuralOptimizer class.

Train a Neural Network using a Genetic Algorithm.

GA(*population*, *iterations*, *x*, *y*, *x_test*, *y_test*)

Parameters

- **population** – initial population. List of numpy vectors
- **iterations** – integer. Number of generations
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)
- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns tuple (best individual found as numpy vector, max value found, history of bests)

best_of_generation(*fitness*, *population*, *x*, *y*)

Parameters

- **fitness** – fitness function
- **eq** – equation to maximize
- **population** – list of individuals (numpy array)
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)

Returns tuple of numpy vector best individual and best value

fit(*population*, *x_train*, *y_train*, *x_test*, *y_test*)

Parameters

- **population** – initial population. List of numpy vectors
- **x_train** – train dataset (numpy array)
- **y_train** – labels for train dataset (numpy vector)
- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns best chromosome found (numpy vector), best value found for that chromsome (float), training history

(list of fitness values)

fitness(*individual*, *x*, *y*)

Parameters

- **individual** – list of individuals
- **x_train** – train dataset (numpy array)
- **y_train** – labels for train dataset (numpy vector)

Returns best chromosome found (numpy vector), best value found for that chromsome (float), training history

(list of fitness values), keras.history for the backpropagation training

transform_weights(*individual*)

Parameters **individual** – chromosome (numpy vector)

Returns reshaped chromosome into Neural Network weight arrangement (list of numpy vectors)

weights_to_vector()

Returns list of numpy vectors

weights_to_vector_alt()

Returns list of numpy vectors

class genetic_optimizer.**GeneticNeuralWeightOptimizer**(*model*, *mutation_prob=0.5*, *mutation_rate=None*, *elite=2*, *stop_condition=None*, *epochs=-1*, *fitness_function=<function GeneticNeuralWeightOptimizer.<lambda>>*, *mutables=1*, *mutation_dist=<built-in method uniform of mtrand.RandomState object>*, *pop_dist='uniform'*, *iterations=None*)

GeneticNeuralWeightOptimizer class

Initialize Neural Network weights using genetic algorithm.

GA(*population*, *iterations*, *x*, *y*, *x_test*, *y_test*)

Parameters

- **population** – initial population. List of numpy vectors
- **iterations** – integer. Number of generations
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)
- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns tuple (best individual found as numpy vector, max value found, history of bests)

best_of_generation(*fitness*, *population*, *x*, *y*)

Parameters

- **fitness** – fitness function
- **population** – list of individuals (numpy array)
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)

Returns tuple of numpy vector best individual and best value

fit (*population*, *x_train*, *y_train*, *x_test*, *y_test*)

Parameters

- **population** – list of individuals
- **x_train** – train dataset (numpy array)
- **y_train** – labels for train dataset (numpy vector)
- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns best chromosome found (numpy vector), best value found for that chromsome (float), training history

(list of fitness values), keras.history for the backpropagation training

fitness (*individual*, *x*, *y*)

Parameters

- **individual** – individual chromosome to mute (numpy vector)
- **x** – dataset to evaluate on the model
- **y** – labels for each point of the dataset x

Returns fitness value (float)

transform_weights (*individual*)

Parameters **individual** – chromosome (numpy vector)

Returns reshaped chromosome into Neural Network weight arrangement (list of numpy vectors)

weights_to_vector ()

Transform neural network weights to a chromosome :return: list of numpy vectors

weights_to_vector_alt ()

Returns list of numpy vectors

class genetic_optimizer.**GeneticOptimizer**

GeneticOptimizer Abstract class, define the basic functions of the rest of Genetic Optimizer classes

- fit function: train using Genetic Algorithm
- GA function

Operations needed durante Genetic Algorithms

- fitness function: evaluating a chromosome
- generate_population
- random_selection
- reproduce

- mutate

generate_population(*individuals*, *distribution='uniform'*)

Returns numpy array with

mutate(*x*, *dist=<built-in method uniform of mtrand.RandomState object>*)

Parameters

- **x** – individual chromosome to mute (numpy vector)
- **dist** – probability distribution used (default is Uniform)

Returns mutated x (numpy vector)

original_weights_shape()

Returns shape of each layer in the neural network (list of numpy vectors)

random_selection(*population*, *distribution=None*)

Parameters

- **population** – list of numpy array
- **distribution** – probability distribution used during selection

Returns tuple of two more promising individuals (numpy vectors)

static reproduce(*x*, *y*)

Parameters

- **x** – numpy vector of genes
- **y** – numpy vector of genes

Returns random crossover of x with y as numpy vector

class genetic_optimizer.**GeneticRegularizator**(*model*, *mutation_prob=0.5*, *mutation_rate=None*, *elite=2*, *genetic_train=True*, *stop_condition=None*, *epochs=-1*, *fitness_function=<function GeneticRegularizator.<lambda>>*, *mutables=1*, *original_mutation_prob=0.5*, *pop_dist='uniform'*, *mutation_dist=<built-in method uniform of mtrand.RandomState object>*, *callbacks=[]*, *iterations=None*)

GeneticRegularizator class

Using during a backpropagation training, will regularize the whole training if

- If callback is provided and then, the training process is interrupted, or
- When the training process with backpropagation is finished

GA(*population*, *iterations*, *x*, *y*, *x_test*, *y_test*)

Parameters

- **population** – initial population. List of numpy vectors
- **iterations** – integer. Number of generations
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)

- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns tuple (best individual found as numpy vector, max value found, history of bests)

best_of_generation(*fitness*, *population*, *x*, *y*)

Parameters

- **fitness** – fitness function
- **eq** – equation to maximize
- **population** – list of individuals (numpy array)
- **x** – train dataset (numpy array)
- **y** – labels for train dataset (numpy vector)

Returns tuple of numpy vector best individual and best value

fit(*population*, *x_train*, *y_train*, *x_test*, *y_test*)

Parameters

- **population** – initial population. List of numpy vectors
- **x_train** – train dataset (numpy array)
- **y_train** – labels for train dataset (numpy vector)
- **x_test** – test dataset (numpy array)
- **y_test** – labels for test dataset (numpy vector)

Returns best chromosome found (numpy vector), best value found for that chromsome (float), training history

(list of fitness values)

fitness(*individual*, *x*, *y*)

Parameters

- **individual** – individual chromosome to mute (numpy vector)
- **x** – dataset to evaluate on the model
- **y** – labels for each point of the dataset x

Returns fitness value (float)

transform_weights(*individual*)

Parameters **individual** – chromosome (numpy vector)

Returns reshaped chromosome into Neural Network weight arrangement (list of numpy vectors)

weights_to_vector()

Returns list of numpy vectors

weights_to_vector_alt()

Returns list of numpy vectors

PYTHON MODULE INDEX

g

genetic_optimizer, ??

INDEX

B

best_of_generation() (*genetic_optimizer.GeneticNeuralOptimizer method*), 1
best_of_generation() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 2
best_of_generation() (*genetic_optimizer.GeneticRegularizator method*), 5

F

fit() (*genetic_optimizer.GeneticNeuralOptimizer method*), 1
fit() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 3
fit() (*genetic_optimizer.GeneticRegularizator method*), 5
fitness() (*genetic_optimizer.GeneticNeuralOptimizer method*), 2
fitness() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 3
fitness() (*genetic_optimizer.GeneticRegularizator method*), 5

G

GA() (*genetic_optimizer.GeneticNeuralOptimizer method*), 1
GA() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 2
GA() (*genetic_optimizer.GeneticRegularizator method*), 4
generate_population() (*genetic_optimizer.GeneticOptimizer method*), 4
genetic_optimizer (*module*), 1
GeneticNeuralOptimizer (*class in genetic_optimizer*), 1
GeneticNeuralWeightOptimizer (*class in genetic_optimizer*), 2
GeneticOptimizer (*class in genetic_optimizer*), 3
GeneticRegularizator (*class in genetic_optimizer*), 4

M

mutate() (*genetic_optimizer.GeneticOptimizer method*), 4

O

original_weights_shape() (*genetic_optimizer.GeneticOptimizer method*), 4

R

random_selection() (*genetic_optimizer.GeneticOptimizer method*), 4
reproduce() (*genetic_optimizer.GeneticOptimizer static method*), 4

T

transform_weights() (*genetic_optimizer.GeneticNeuralOptimizer method*), 2
transform_weights() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 3
transform_weights() (*genetic_optimizer.GeneticRegularizator method*), 5

W

weights_to_vector() (*genetic_optimizer.GeneticNeuralOptimizer method*), 2
weights_to_vector() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 3
weights_to_vector() (*genetic_optimizer.GeneticRegularizator method*), 5
weights_to_vector_alt() (*genetic_optimizer.GeneticNeuralOptimizer method*), 2

weights_to_vector_alt() (*genetic_optimizer.GeneticNeuralWeightOptimizer method*), 3
weights_to_vector_alt() (*genetic_optimizer.GeneticRegularizator method*), 5