evolutionary_keras

Release 1.0.0

Stefano Carrazza, Juan Cruz-Martinez, Roy Stegeman

Feb 10, 2020

CONTENTS:

This is the documentation for the latest release of the `evolutionary_keras` python module. This documentation includes a quick how-to use guide, provides several examples and collects changelogs of new releases.

CHAPTER ONE

WHAT IS `EVOLUTIONARY_KERAS`

Keras is one of the most widely used Machine Learning frameworks available in the market. It is a high-level API written in Python and that can run on mulitple backends. Their goal is to be able to build and test new model as fast as possible.

Keras models are trained through the usage of optimizers, all of which are Gradient Descent based. This module deals with that shortcoming of Keras implementing several Genetic Algorithms on top of Keras while keeping the main philosophy of the project: it must be easy to prototype.

CHAPTER

TWO

INSTALLING EVOLUTIONARY_KERAS

evolutionary_keras is available in PyPI, conda-forge.

```
pip install evolutionary_keras
```

Furthermore, the code is available under GPL3.0 in github: N3PDF/evolutionary_keras.

2.1 How to use

In order to use the capabilities of evolutionary_keras in a project is necessary to use the model-classes provided. These classes inherit from the Keras model class and transparently defer to them whenever a Gradient Descent algorithm is used.

As an example, let us consider a project in which we have some neural network constructed with an input layer input_layer and an output layer output_layer. The Keras model would usually be constructed as:

```
from keras.models import Model
my_model = Model(input_layer, output_layer)
```

Using evolutionary_keras is as easy as doing:

```
from evolutionary_keras.models import EvolModel
my_model = EvolModel(input_layer, output_layer)
```

From that point onwards my_model behaves exactly as a normal Keras model implementing the same methods and attributes as well as allowing the usage of Evolutionary *Optimizers*. For instance, the example belows utilizes the Nodal Genetic Algorithm (NGA):

```
my_model.compile("nga")
```

Which will use the default parameters of the *Nodal Genetic Algorithm (NGA)*. Subsequent calls to methods such as my_model.fit will use the NGA algorithm to train.

For a more fine-grained usage we can also import the optimizer and instantiate it ourselves:

```
from evolutionary_keras.optimizers import NGA
my_nga = NGA(population_size = 42, mutation_rate = 0.2)
my_model.compile(my_nga)
```

2.2 Optimizers

This page lists all evolutionary strategies currently implemented

2.2.1 Nodal Genetic Algorithm (NGA)

Implementation of the Nodal Genetic Algorithm as implemented by the NNPDF collaboration and which is presented in the NNPDF3.0 release paper.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search