Infer.java
project

GOBIERNO DE ESPAÑA · MINISTERIO DE ECONOMIA Y COMPETITIVIDAD

UNIÓN EUROPEA
Fondo Europeo de Desarrollo Regional
"Una manera de hacer Europa"

Reference:	**TIN2015-74368-JIN**
Project full title:	**INFER.JAVA: UN LENGUAJE DE PROGRAMACION. PROBABILISTICO PARA EL DESARROLLO DE APLICACIONES**
Deliverable no..:	**PT3**
Title of the deliverable:	**Design and development of a compiler for INFER.java**

Start date:	01/03/2017
End date:	
Author(s):	Andrés R. Masegosa, Rafael Cabañas de Paz
Participant(s):	Andrés R. Masegosa, Rafael Cabañas de Paz
Objective:	To design and to develop a compiler for INFER.java
Version:	0.1
Total number of pages:	
Start date of project:	1st March 2017. Duration: 6 months

Abstract:

**Keywords list:** compiler, architecture, probability distributions, Edward, Tensorflow

1 Features

The main features of InferPy are listed below.

- The models that can be defined in Inferpy are those that can be defined using Edward, whose probability distribuions are mainly inherited from TensorFlow Distribuitons package.

- Edward's drawback is that for the model definition, the user has to manage complex multidimensional arrays called tensors. By contrast, in InferPy all the parameters in a model can be defined using the standard Python types (compatibility with Numpy is available as well).

- InferPy directly relies on top of Edward's inference engine and includes all the inference algorithms included in this package. As Edward's inference engine relies on TensorFlow computing engine, InferPy also relies on it too.

- InferPy seamlessly process data contained in a numpy array, Tensorflow's tensor, Tensorflow's Dataset (tf.Data API), Pandas' DataFrame or Apache Spark's DataFrame.

- InferPy also includes novel distributed statistical inference algorithms by combining Tensorflow computing engines.

2 Architecture

Given the previous considerations, we might summarize the InferPy architecture as follows.
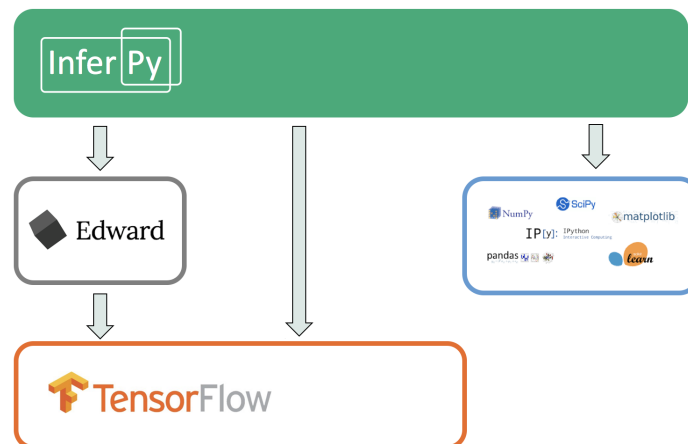
Note that InferPy can be seen as an upper layer for working with probabilistic distributions defined over tensors. Most of the interaction is done with Edward: the definitions of the distributions, the inference. However, InferPy also interacts directly with Tensorflow in some operations that are hidden to the user, e.g. the manipulation of the tensors representing the parameters of the distributions.

An additional advantage of using Edward and Tensorflow as inference engine, is that all the paralelisation details are hidden to the user. Moreover, the same code will run either in CPUs or GPUs.

For some less important task, InferPy might also interact with other third-party software. For example, reading data is done with Pandas or the visualization tasks are leveraged to MatPlotLib.