DOCTORAL PROGRAM IN ENGINEERING SCIENCES AT ITESO

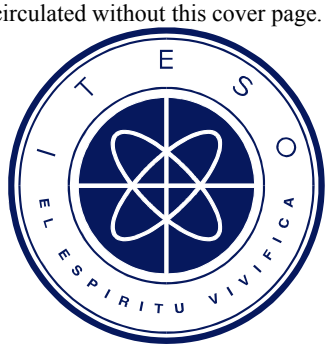
ADAPTATIVE DISCOVERING ALGORITHM BASED ON NEURAL NETWORKS

Edgar-D. Ramirez-de-L.
Maria-del-P. Pozos-P.
Ivan Villalon-Turrubiates

Algorithm 2 - readSource.doc

May 15, 2018
Tlaquepaque, Mexico 45604

Doctoral Program in Engineering Sciences
ITESO (*Instituto Tecnológico y de Estudios Superiores de Occidente*)



ADAPTATIVE DISCOVERING ALGORITHM BASED ON NEURAL NETWORKS

Edgar-D. Ramirez-de-L.
Maria-del-P. Pozos-P.
Ivan Villalon-Turrubiates

May 15, 2018

Doctoral Program in Engineering Sciences
ITESO (*Instituto Tecnológico y de Estudios Superiores de Occidente*)

Tlaquepaque, Mexico 45604
Tel +52 33 3669 3598
E-mail: dci@iteso.mx

Keywords *neural networks, deep learning, machine learning, classifiers, python*

Abstract

We present the Algorithm 2 (`readSource`) which is part of the Adaptative Discovering Algorithm based on Neural networks (ADAN algorithm).

Algorithm 2 readSource

Require: $args \neq \varnothing \wedge readArg('dataSource', args) \neq \varnothing$

1: $dataFrame \leftarrow loadData(readArg('dataSource', args))$
2: $label \leftarrow readArg('label', args)$
3: $randomizedDataFrame \leftarrow chooseRandomFeatures(args, dataFrame)$
4: $randomizedDataFrame \leftarrow concat(randomizedDataFrame,$ pop(dataFrame, label), 'columns')
5: $(dfTraining, trainY), (dfTesting, testY), dfPredict, expected \leftarrow chooseRandomData(args, randomizedDataFrame)$
6: **return** $(dfTraining, trainY), (dfTesting, testY), dfPredict, expected$

For Algorithm 2 please consider the next:

1. The function `loadData(dataSource)` in line 1, will depend from the programming language and/or from the library to achieve this. In our case, like we said at the beginning of this document, we use Keras for this. In the file `data.py`[1] you will see the implementation for the function `loadData(dataSource)`. We don't include here because is a trivial process.

2. `concat(dataFrame, obj, concatBy)`, in line 4, like its name suggests, is a function that will concat the `dataFrame` with `obj` in the axis specified by `concatBy`.

[1] adan/data.py, The Authors. May 7, 2018, https://github.com/EDario333/adan/blob/0.1.4/algorithm/data.py

In the same line, you can see the `pop(dataFrame, ‘serieName’)` function, like its name suggests, it will pop from the `dataFrame` the serie (or the y axis) specified by `‘serieName’`.

All this give us the next implementation, in Python: `randomized_df = pd.concat([randomized_df, df.pop(args.label)], 'columns')`