K-CHAIN Documentation

Nurali Virani

Mar 31, 2019

CONTENTS:

DARPA ASKE TA1

CHAPTER

ONE

MODULE: KCHAIN

This module consists of kChainModel class to create, fit, append, and update K-CHAIN models in TensorFlow

class kChain.**kChainModel**(*debug=False*)

__init__(*debug=False*)
Initialize object of type K-CHAIN model.

Parameters **debug** (*bool*) – various print statements throughout the code execution will be executed to help in debugging.

_createEqnModel(*inputVar*, *outputVar*, *mdlName*, *eqMdl*)
Build a K-CHAIN model using input and output variables from the KG and the physics equation.

Parameters

- **inputVar** (*JSON array*) – array of JSON variable objects with name, type, and value fields
- **outputVar** (*JSON array*) – array of JSON variable objects with name, type, and value fields
- **mdlName** (*string*) – Name to assign to the final model (E.g.: 'Newtons2ndLaw')
- **eqMdl** (*string*) – Equation relating inputs to output (E.g.: "c = a * b")

Returns Computational graph of the physics equation metagraphLoc (string): Location on disk where computational model was stored

Return type mdl (TensorFlow Graph)

_createNNModel(*inputVar*, *outputVar*, *mdlName*)
Build a K-CHAIN model as a neural network using input and output variables from the KG.

Parameters

- **inputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset) and type fields
- **outputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset) and type fields
- **mdlName** (*string*) – Name to assign to the final model (E.g.: 'Newtons2ndLaw')

Returns

computational graph of the neural network metagraphLoc (string):

Location on disk where computational model is stored

Return type mdl (TensorFlow Graph)

_getVarType(*typeStr*)

Obtain tensorflow datatypes for variable type information from KG

Parameters **typeStr** (*string*) – String denoting type of variable with possible values of bool, integer, float, and double (default).

Returns datatype in TensorFlow (e.g. tf.bool)

_makePyFile(*stringfun*)

Write the formatted code into a python module for conversion to tensorflow graph

Parameters **stringfun** (*string*) – formatted python code as string to be written in python file

build(*inputVar*, *outputVar*, *mdlName*, *dataLoc=None*, *eqMdl=None*)

Build a K-CHAIN model using input and output variables from the KG.

Parameters

- **inputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset), type, and value fields
- **outputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset), type, and value fields
- **mdlName** (*string*) – Name to assign to the final model (E.g.: 'Newtons2ndLaw')
- **dataLoc** (*string*) – Location of dataset as .csv with Row 1 - Variables names, Row 2 - Units, Row 3 onwards - data (default = None)
- **eqMdl** (*string*) –

evaluate(*inputVar*, *outputVar*, *mdlName*)

Evaluates a model with given inputs to compute output values

Parameters

- **inputVar** (*JSON array*) – array of JSON variable objects with name, type, and value fields
- **outputVar** (*JSON array*) – array of JSON variable objects with name, type, and value fields
- **mdlName** (*string*) – Name to model to use (E.g.: 'Newtons2ndLaw')

Returns array of JSON variable objects with name, type, and value fields. The resulting output of the computation is assigned to the value field of the JSON object.

Return type outputVar (JSON array)

fitModel(*dataset*, *inputVar*, *outputVar*, *mdlName*)

Fit a K-CHAIN model using input and output variables from the KG and the corresponding dataset.

Parameters

- **dataset** (*Pandas Dataframe*) – dataset with inputs and outputs
- **inputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset) and type fields
- **outputVar** (*JSON array*) – array of JSON variable objects with name (as in dataset) and type fields
- **mdlName** (*string*) – Name to assign to the final model (E.g.: 'Newtons2ndLaw')

Returns Location on disk where computational model and trained parameters are stored

Return type metagraphLoc (string)

`getDataset`(*dataLoc=None*)

Create Pandas DataFrame from identified csv.

Parameters **`dataLoc`** (*`string`*) – Location of dataset as .csv with Row 1 - Variables names, Row 2 - Units, Row 3 onwards - data (default = None)

Returns DataFrame with values read from csv file

Return type df (Pandas DataFrame)

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX