
dpipe

Release 0.1

Ariel Iporre

Feb 21, 2023

CONTENTS:

1	dpipe	1
1.1	dpipe.factories	1
1.2	dpipe.datasets	3
1.3	dpipe.utils	4
1.4	Module contents	5
2	Indices and tables	7
	Python Module Index	9
	Index	11

1.1 dpipe.factories

class `dpipe.factories.AugmentedDataset` (*dataset*, *gen_object=None*, *length=None*, *training=True*)

Bases: `object`

Augments `tf.data.Dataset` to handle custom configurations

Parameters

- **dataset** (`tf.data.Dataset`) – Instance of a `tf.data.Dataset`
- **length** (*int*, *optional*) – length of the dataset, defaults to `None`
- **training** (*bool*, *optional*) – defines training/validation flag. If `True` then the augmented dataset handles training configurations, and if `False` the augmented dataset handles validation configurations, defaults to `True`

batch (*batch_size*)

Make dataset batches of specific batch size

Parameters **batch_size** (*int*) – size of the batches

build ()

Creates an augmented dataset that contains the arguments to be used in the method `tf.keras.model.fit()`

cache (*filename=""*)

Defines the cache file to store previously loaded samples

Parameters **filename** (*str*, *optional*) – File name of the file where the loaded samples are stored. The second access will be loaded from the cache, defaults to ''

enumerate (*start=0*)

As the build-in enumerate function creates an index next to the sample.

Parameters **start** (*int*, *optional*) – start count of the enumeration, defaults to 0

filter (*filter_fcn*)

Applies a filter function to all the samples of dataset. Applies lazily.

Parameters **filter_fcn** – function reference

map (*map_func*, *num_parallel_calls=None*)

Maps every sample in the dataset by a map function.

Parameters **map_func** – function reference

parallelize_extraction (*cycle_length=4, block_length=16, num_parallel_calls=- 1, read_fcn=None*)

Generates a parallel consuming of items in the list of the original object given the reading function. For example reading files, or images.

Parameters

- **cycle_length** – defaults to 4, read TF docs for more details.
- **block_length** – defaults to 16, read TF docs for more details.
- **num_parallel_calls** – defaults to Autotune, read TF docs for more details.
- **read_fcn** – defaults to None, function to process one item in the dataset $f(x)$ or $f(x,y)$ options.

@return:

prefetch (*buffer_size*)

Preloads samples on the tensor flow session i.e. memory to be processed.abs(\$0)

Parameters **buffer_size** (*int*) – size of the preloaded samples. If batch is specified then it loads a *buffer_size* of batches. For example, *buffer_size=2* with batches of 100 will load 200 samples to the memory.

recompute_length ()

Recompute the length of the datatase.

This may take long since all the samples must be accessed.

repeat (*count=None*)

Creates a concatenated repeated dataset_builder

Parameters **count** (*int*) – Number of repretitions

shuffle (*buffer_size, seed=None, reshuffle_each_iteration=None*)

class dpipe.factories.**GeneratorBase** (*obj, getitem_fcn=None, itemlist_name=None, length=None*)

Bases: object

Wraps an object with a method *getitem* to make it an iterable class

Parameters

- **obj** (*object*) – Instance of object that access data
- **getitem_fcn** (*str*) – Name of the method in the object to access data
- **length** (*int*) – length of the dataset, if None then infers from *len()* function, defaults to None

send (*ignored_arg*)

throw (*type=None, value=None, traceback=None*)

Raise a *StopIteration*

dpipe.factories.**from_function** (*read_fcn, list, training=True, undetermined_shape=None*)

dpipe.factories.**from_object** (*obj, getitem_fcn=None, itemlist_name=None, training=True, undetermined_shape=None*)

Creates a *tf.data.Dataset* object with configuration parameters for fitting

Parameters

- **obj** – Object instance of the data with '*getitem_fcn*' function to access dataset

- **getitem_fcn** (*str*, *optional*) – getitem_fcn Name of the method to access data . getitem_fcn can have any name defined for the in the class 'obj'. If not specified infers '__getitem__' as name of the access function
- **itemlist_name** (*str*, *optional*) – name of the list containing samples on the object, if None name is "list"
- **training** (*bool*, *optional*) – Specify training/validation flag
- **undetermined_shape** (*iterable*, *optional*) – defines positions in the shape vector where dimensions are undetermined

Returns An object `tf.data.Dataset` from the obj dataset

1.2 dpipe.datasets

`dpipe.datasets.make_dataset` (*x_type*, *y_type*, *x_path=None*, *y_path=None*, *x_size=None*, *y_size=None*, *training=True*, *video_frames=None*, *video_cropping=None*, *one_hot_encoding=False*)

Create custom dataset from a path list

Parameters

- **x_type** – Defines the type of the input data to the model. It can be: label, video or image. The proper reading is generated accordingly.
- **y_type** – Defines the type of the target data to the model, idem as above
- **x_path** – Path to the dataset of inputs, the path is expected to contain images or videos sorted in a way that the name of the containing parent is the label, if label is relevant. For example `cat/image1.png` and `dog/image2.png`. All the files are indexed as individual samples.
- **y_path** – Path to the dataset of targets, idem as above.
- **x_size** – Size of the image or video for the input to the model
- **y_size** – Size of the image or video for the target to the model
- **training** (*bool*, *optional*) – Specify training/validation flag
- **video_frames** – number of frames of the output video if data type is video
- **video_cropping** – video cropping method creates a crop of the video with a length defined by video frames. Working modes are single and multi. single where the video will be just from the first frame to the number of video_frames defined; or the multi where the video is cropped sequences of clips with the number of frames defined by video_frames.
- **one_hot_encoding** – Activate one hot encoding for the label input

Returns Created dataset `tf.data.Dataset` with the pairs input and target (x,y)

1.3 dpipe.utils

`dpipe.utils.create_label_dict` (*paths*, *one_hot_encoding=False*)

Creates a label dictionary from list of paths

Parameters *paths* (*list*) – Path from where the label dictionary is created. The parent folder is considered as label for the images.

Returns Dictionary of labels sorted alphabetically

Return type dict

`dpipe.utils.get_parent_path` (*path*)

`dpipe.utils.get_read_fcn` (*data_type*, *label_dict=None*)

`dpipe.utils.get_single_value` (*value*, *counter=0*)

Recursively tracks a single element of value. It assumes all elements as of the same kind

Parameters *value* – Value get a single value from.

`dpipe.utils.get_tf_dtype` (*value*)

Obtains tensorflow datatype

The available transformations are:

- tf.float16: 16-bit half-precision floating-point.
- tf.float32: 32-bit single-precision floating-point.
- tf.float64: 64-bit double-precision floating-point.
- tf.complex64: 64-bit single-precision complex.
- tf.complex128: 128-bit double-precision complex.
- tf.int8: 8-bit signed integer.
- tf.uint8: 8-bit unsigned integer.
- tf.uint16: 16-bit unsigned integer.
- tf.uint32: 32-bit unsigned integer.
- tf.uint64: 64-bit unsigned integer.
- tf.int16: 16-bit signed integer.
- tf.int32: 32-bit signed integer.
- tf.int64: 64-bit signed integer.
- tf.bool: Boolean.
- tf.string: String.

Parameters *value* – Value to identify class from.

`dpipe.utils.get_tf_shape` (*value*)

Obtains the shape of an variable

Parameters *value* – input value can be `numpy.ndarray`, numeric or string class. It supports list of numerics or string but not nested lists.

`dpipe.utils.get_video_length` (*path*)

Reads the number of frames from the metadata of a video file

Parameters `path` – path to the video file .

Returns Number of frames extracted with ffprobe.

Return type float64

`dpipe.utils.is_iterable` (*value*)

Verifies the value is an is_iterable

Parameters `value` – value to identify if iterable or not.

`dpipe.utils.is_supported_format` (*filename*)

1.4 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

d

- `dpipe`, 5
- `dpipe.datasets`, 3
- `dpipe.factories`, 1
- `dpipe.utils`, 4

A

AugmentedDataset (class in *dpipe.factories*), 1

B

batch() (*dpipe.factories.AugmentedDataset* method), 1

build() (*dpipe.factories.AugmentedDataset* method), 1

C

cache() (*dpipe.factories.AugmentedDataset* method), 1

create_label_dict() (in module *dpipe.utils*), 4

D

dpipe

module, 5

dpipe.datasets

module, 3

dpipe.factories

module, 1

dpipe.utils

module, 4

E

enumerate() (*dpipe.factories.AugmentedDataset* method), 1

F

filter() (*dpipe.factories.AugmentedDataset* method), 1

from_function() (in module *dpipe.factories*), 2

from_object() (in module *dpipe.factories*), 2

G

GeneratorBase (class in *dpipe.factories*), 2

get_parent_path() (in module *dpipe.utils*), 4

get_read_fcn() (in module *dpipe.utils*), 4

get_single_value() (in module *dpipe.utils*), 4

get_tf_dtype() (in module *dpipe.utils*), 4

get_tf_shape() (in module *dpipe.utils*), 4

get_video_length() (in module *dpipe.utils*), 4

I

is_iterable() (in module *dpipe.utils*), 5

is_supported_format() (in module *dpipe.utils*), 5

M

make_dataset() (in module *dpipe.datasets*), 3

map() (*dpipe.factories.AugmentedDataset* method), 1

module

dpipe, 5

dpipe.datasets, 3

dpipe.factories, 1

dpipe.utils, 4

P

parallelize_extraction()

(*dpipe.factories.AugmentedDataset* method), 1

prefetch() (*dpipe.factories.AugmentedDataset* method), 2

R

recompute_length()

(*dpipe.factories.AugmentedDataset* method), 2

repeat() (*dpipe.factories.AugmentedDataset* method), 2

S

send() (*dpipe.factories.GeneratorBase* method), 2

shuffle() (*dpipe.factories.AugmentedDataset* method), 2

T

throw() (*dpipe.factories.GeneratorBase* method), 2