
preeminence_utils Documentation

Release 0.1

Tushar Pawar

Dec 07, 2017

Contents:

1	Mongo Utils	1
2	Tensorflow Utils	3
3	Indices and tables	5
	Python Module Index	7

CHAPTER 1

Mongo Utils

This is mongo utils Centralised utility class for all operations to be performed on a mongo database.

```
class mongo_utils.MongoUtils (address='127.0.0.1', port=27017, db_name='', collection_name='')
```

```
get_list_from_db (filter_condition=None)
```

Connect to mongo db and fetch data.

Parameters **filter_condition** – Filter condition

Returns Return collection as a list

```
insert_record (new_record=None)
```

Insert a record in table :param new_record: New record to be inserted. :return:

```
update_record (filter_condition=None, new_value=None)
```

Connect to mongo db and update data.

Parameters

- **filter_condition** – Filter condition
- **new_value** – new value of the selected document

Returns Return collection as a list

CHAPTER 2

Tensorflow Utils

This is tensorflow utils

```
from preeminence_utils import tf_utils
model = tf_utils.Model()
```

Helper functions for using while making a neural network using tensorflow

class `tf_utils.Model`

get_latest_checkpoint (*checkpoint_path='./model_weights/'*)

Get the name of the latest checkpoint in the checkpoints directory in order to load the latest weights to continue training for that point.

Parameters `checkpoint_path` – Custom directory where checkpoints are saved

Returns

graph_info ()

Get ops in the graph

Returns Graph ops

init ()

Initialise a new model and return its graph. This function will spawn a new graph and return it. You'll have to set it to default graph in order to add ops to it. Sample: `model = tf_utils.Model() model_graph = model.init().as_default()`

Returns Returns a new graph.

next_batch (*data, batch_start, batch_size*)

Get next batch from the training data This should be a generator function :/

Parameters

- **data** – Training data
- **batch_start** – batch start index

- **batch_size** – size of the batch

Returns

restore_weights (*checkpoint_path*='./model_weights/', *s3*=False, *s3_path*=None)

Restore weights from the checkpoint path to the latest checkpoint to resume training from that point.

Parameters

- **s3** – Flag to decide to download checkpoints from S3 or not
- **s3_path** – Name of subfolder in preeminence-models bucket
- **checkpoint_path** – Custom directory where checkpoints are saved

Returns

save_weights (*checkpoint_path*=None, *checkpoint_number*=None, *s3*=False, *s3_path*=None)

Save the current weights of the model to disk at the checkpoint path.

Parameters

- **s3** – Flag to decide to upload weights to S3 or not
- **s3_path** – Subfolder in S3 bucket
- **checkpoint_path** – Custom directory where checkpoints are saved
- **checkpoint_number** – Custom number to append at the end of checkpoint

Returns

session ()

Create and return a new session for training.

Returns New session object

train (*ops*, *x*, *y*, *x_data*, *y_data*, *num_epochs*=1, *batch_size*=1)

Training function. Executes the graph on a given dataset.

Parameters

- **ops** – Graph ops to be calculated and returned. Must be [optimiser_op, loss_op]
- **x** – placeholder tensor for x
- **y** – placeholder tensor for y
- **x_data** – Training data to be fed into x
- **y_data** – Training data to be fed into y
- **num_epochs** – Number of epochs to be executed
- **batch_size** – Size of a batch to be fed at a time

Returns Nothing

visualise (*logdir*='./logs')

Save graph summary in the logdir to be visualised by tensorboard. Summaries for individual ops to be added.

Parameters **logdir** – Destination for storing graph logs. ./logs by default

Returns Nothing

CHAPTER 3

Indices and tables

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

m

`mongo_utils`, 1

t

`tf_utils`, 3

G

`get_latest_checkpoint()` (`tf_utils.Model` method), 3
`get_list_from_db()` (`mongo_utils.MongoUtils` method), 1
`graph_info()` (`tf_utils.Model` method), 3

I

`init()` (`tf_utils.Model` method), 3
`insert_record()` (`mongo_utils.MongoUtils` method), 1

M

`Model` (class in `tf_utils`), 3
`mongo_utils` (module), 1
`MongoUtils` (class in `mongo_utils`), 1

N

`next_batch()` (`tf_utils.Model` method), 3

R

`restore_weights()` (`tf_utils.Model` method), 4

S

`save_weights()` (`tf_utils.Model` method), 4
`session()` (`tf_utils.Model` method), 4

T

`tf_utils` (module), 3
`train()` (`tf_utils.Model` method), 4

U

`update_record()` (`mongo_utils.MongoUtils` method), 1

V

`visualise()` (`tf_utils.Model` method), 4