
easycore

Feb 29, 2020

Contents

1	Tutorials	1
1.1	Installation	1
1.2	Light weight config tools	1
2	API Documentation	5
2.1	easycore.common	5
3	Indices and tables	7
	Python Module Index	9
	Index	11

1.1 Installation

1.1.1 Requirements

- Python 3.6

1.1.2 Build easycore from Source

```
pip install 'git+https://github.com/YuxinZhaozyx/easycore.git'

# or build it from local
git clone https://github.com/YuxinZhaozyx/easycore.git
cd easycore
pip install -e .
```

1.2 Light weight config tools

easycore make it easy to load config from local yaml file, save config and control the config in runtime.

1.2.1 Load config from local yaml file

An example of yaml file is shown bellow:

```
MODEL:
  IN_FEATRES: ["res3", "res4", "res5"]
  INPUT_SIZE: (224, 224)
  NUM_CLASSES: 100
NAME: YuxinZhaozyx
```

You can load the yaml file in the follow way:

```
from easycore.common.config import CfgNode as CN

cfg = CN.open('example.yaml')

# or
with open('example.yaml', 'r', encoding='utf-8') as f:
    cfg = CN.open(f)
```

1.2.2 Get an empty config

```
cfg = CN()
```

1.2.3 Get a config from from python dict

```
init_dict = {
    "MODEL": {
        "IN_FEATURES": ["res3", "res4", "res5"],
        "INPUT_SIZE": (224, 224),
        "NUM_CLASSES": 100,
    },
    "NAME": "YuxinZhaozyx",
}
cfg = CN(init_dict)
```

1.2.4 Use config

```
# get value from config
# the config has been automatically transform into python data type.
in_features = cfg.MODEL.IN_FEATURES # list
input_size = cfg.MODEL.INPUT_SIZE   # tuple
num_classes = cfg.MODEL.NUM_CLASSES  # int
name = cfg.NAME                      # str

# add new value to config
cfg.LICENSE = 'MIT'

# add a new CfgNode to config
cfg.SOLVER = CN()
cfg.SOLVER.LEARNING_RATE = 0.001
cfg.SOLVER.BATCH_SIZE = 128
```

1.2.5 Merge two config

```
cfg_a = CN()
cfg_a.key1 = 1
cfg_a.key2 = 2

cfg_b = CN()
```

(continues on next page)

(continued from previous page)

```
cfg_b.key2 = 3
cfg_c.key3 = 4

# merge two config
cfg_a.merge(cfg_b)  # now cfg_a.key2 is 3
```

1.2.6 Copy a config

```
cfg_copy = cfg.copy()  # get a deepcopy of cfg
```

1.2.7 Save config to yaml file

```
cfg.save("example-save.yaml")

# or
with open("example-save.yaml", 'w', encoding='utf-8') as f:
    cfg.save(f)
```

1.2.8 API Documentation

- `easycore.common.config`

2.1 easycore.common

2.1.1 easycore.common.config

class `easycore.common.config.CfgNode` (*init_dict: dict = None, copy=True*)

Bases: `dict`

Config Node

__init__ (*init_dict: dict = None, copy=True*)

Parameters

- **init_dict** (*dict*) – a possibly-nested dictionary to initialize the CfgNode.
- **copy** (*bool*) – if this option is set to False, the CfgNode instance will share the value with the *init_dict*, otherwise the contents of *init_dict* will be deepcopied.

freeze (*frozen: bool = True*)

freeze or unfreeze the CfgNode and all of its children

Parameters **frozen** (*bool*) – freeze or unfreeze the config

is_frozen ()

get the state of the config.

Returns *bool* – whether the config tree is frozen.

copy ()

deepcopy this CfgNode

Returns *CfgNode*

merge (*cfg*)

merge another CfgNode into this CfgNode, the another CfgNode will override this CfgNode.

Parameters **cfg** (*CfgNode*) –

save (*save_path*, *encoding*='utf-8')
 save the CfgNode into a yaml file

Parameters *save_path* –

classmethod open (*file*, *encoding*='utf-8')
 load a CfgNode from file.

Parameters

- **file** (*io.IOBase* or *str*) – file object or path to the yaml file.
- **encoding** (*str*) –

Returns *CfgNode*

classmethod load (*yaml_str: str*)
 load a CfgNode from a string of yaml format

Parameters *yaml_str* (*str*) –

Returns *CfgNode*

classmethod dump (*cfg*, *stream=None*, *encoding=None*, ***kwargs*)
 dump CfgNode into yaml str or yaml file

Note: if *stream* option is set to non-None object, the CfgNode will be dumped into stream and return None, if *stream* option is not given or set to None, return a string instead.

Parameters

- **cfg** (*CfgNode*) –
- **stream** (*io.IOBase* or *None*) – if set to a file object, the CfgNode will be dumped into stream and return None, if set to None, return a string instead.
- **encoding** (*str* or *None*) –
- ****kwargs** – options of the yaml dumper.

Some useful options: ["allow_unicode", "line_break", "explicit_start", "explicit_end", "version", "tags"].

See more details at https://github.com/yaml/pyyaml/blob/2f463cf5b0e98a52bc20e348d1e69761bf263b86/lib3/yaml/__init__.py#L252

Returns *None* or *str*

CHAPTER 3

Indices and tables

- `genindex`
- `search`

e

`easycore.common`, 6

`easycore.common.config`, 5

Symbols

`__init__()` (*easycore.common.config.CfgNode* method), 5

C

`CfgNode` (class in *easycore.common.config*), 5

`copy()` (*easycore.common.config.CfgNode* method), 5

D

`dump()` (*easycore.common.config.CfgNode* class method), 6

E

`easycore.common` (module), 6

`easycore.common.config` (module), 5

F

`freeze()` (*easycore.common.config.CfgNode* method), 5

I

`is_frozen()` (*easycore.common.config.CfgNode* method), 5

L

`load()` (*easycore.common.config.CfgNode* class method), 6

M

`merge()` (*easycore.common.config.CfgNode* method), 5

O

`open()` (*easycore.common.config.CfgNode* class method), 6

S

`save()` (*easycore.common.config.CfgNode* method), 5