

## 1 初始

yacs 库，用于为一个系统构建配置文件

安装

```
1 pip install yacs
```

导入

```
1 from yacs.config import CfgNode as CN
```

创建配置节点。需要创造 CN() 这个作为容器来装载我们的参数，这个容器可以嵌套

create config node

```
1 from yacs.config import CfgNode as CN
2
3 __C = CN()
4 __C.name = 'test'
5 __C.model = CN() # 嵌套使用
6 __C.model.backbone = 'resnet'
7 __C.model.depth = 18
8
9 print(__C)
10
11 #OUTPUT:
12 #      name: test
13 #      model:
14 #          backbone: resnet
15 #          depth: 18
```

## 2 内置函数

使用上文中的 \_\_C 作为已经创建的配置文件。

### 2.1 clone()

返回一个复制配置文件，因此默认值不会被更改。

克隆

```
1 def get_cfg_defaults():
2     return __C.clone()
```

## 2.2 clear()

清空你的配置文件，你将得到 None 作为结果。

清除

```
1 print(__C.clear())
2
3 #OUTPUT:
4 #      None
```

## 2.3 merge from file()

对于不同的实验，你有不同的超参设置，所以你可以使用 yaml 文件来管理不同的配置文件，然后使用 merge\_from\_file() 这个方法，这个会比较每个实验特有的配置信息和默认参数的区别，会将默认参数与特定参数不同的部分，用特定参数覆盖。

加载不同的配置文件

```
1 __C.merge_from_file("./test_config.yaml")
```

注意点：

你需要合并的 yaml 文件中，不能有 default 参数中不存在的参数，不然会报错，但是可以比 default 中设定的参数少，比如 default 文件中有 name 参数，这是不需要特定改动的，你可以在 yaml 中不设置 name 这个 key。

补充

```
1 from yacs.config import CfgNode as CN
2 # default cfgs
3 __C = CN()
4 __C.name = 'test'
5 __C.model = CN()
6 __C.model.backbone = 'resnet'
7 __C.model.depth = 18
8
9 # yaml cfgs
10 # 不报错的情况1：参数和default中一样多，并且层级关系一致
11 name: test
12 model:
13   backbone: resnet
14   depth: 18
15
16 # 不报错的情况2：参数可以比default中少，以下例子就不包含name和model.
17   backbone
18 model:
19   depth: 34
```

```

19
20     # 报错的情况1: 以下多了model.batch_normalization这个额外的key, 这在
        default中是不存在的
21     name: test
22     model:
23     backbone: resnet
24     depth: 29
25     batch_normalization: True
26
27     # 报错的情况2: 关键词不一致, 这里的关键词是na_me, 而default中是name
28     na_me: test

```

## 2.4 merge from list()

可以用 list 来传递参数

列表传递参数

```

1     from yacs.config import CfgNode as CN
2     __C = CN()
3     __C.name = 'test'
4     __C.model = CN()
5     __C.model.backbone = 'resnet'
6     __C.model.depth = 18
7     print(__C)
8
9     #OUTPUT:
10    #     model:
11    #     backbone: resnet
12    #     depth: 18
13    #     name: test
14
15    opts = ["name", 'test_name', "model.backbone", "vgg"]
16    __C.merge_from_list(opts)
17    print(__C)
18
19    #OUTPUT:
20    #     model:
21    #     backbone: vgg
22    #     depth: 18
23    #     name: test_name

```

## 2.5 freeze()

冻结配置后, 不能修改配置。

### 冻结参数

```
1 from yacs.config import CfgNode as CN
2
3 __C = CN()
4 __C.name = 'test'
5 __C.model = CN()
6 __C.model.backbone = 'resnet'
7 __C.model.depth = 18
8
9 # freeze the config
10 __C.freeze()
11 # try to change the name's value, raise an error
12 __C.name = 'test2' # error
```

## 2.6 defrost()

解冻配置，可以修改配置。

### 解冻参数

```
1 from yacs.config import CfgNode as CN
2
3 __C = CN()
4 __C.name = 'test'
5 __C.model = CN()
6 __C.model.backbone = 'resnet'
7 __C.model.depth = 18
8
9 # freeze the config
10 __C.freeze()
11 # try to change the name's value, raise an error
12 __C.name = 'test2' # error
13
14 __C.defrost() # not freeze cfgs, after this operation you can change
               # the value
15 __C.name = 'test2' # work
```