

```
from yacs.config import CfgNode as CN

# Construct root
_D = CN()

# Training or evaluation
_D.mode = "train"

# Save parameters
_D.save = CN()
_D.save.experiment_batch_name = "deep_test_1"

# Dataset parameters
_D.dataset = CN()
_D.dataset.name = "Core50Dataset"

# Model parameters
_D.model = CN()
_D.model.name = "cnn"
_D.model.input_axis_norm = 3
_D.model.conv = CN()
_D.model.conv.filters = [5, 10, 15]
_D.model.conv.kernels = [2, 2, 2]
_D.model.conv.strides = [1, 1, 1]
_D.model.conv.l2 = 0
_D.model.conv.max_pool_sizes = [4, 4, 4]
_D.model.conv.batch_norms = [1, 1, 1]
_D.model.dense = CN()
_D.model.dense.hidden_sizes = [5, 5]
_D.model.dense.dropout = .5
_D.model.dense.batch_norms = [1, 1]
_D.model.output = CN()
_D.model.output.output_size = 2
_D.model.output.activation = "softmax"
_D.model.reload_path = ""

# Training parameters
_D.train = CN()
_D.train.learning_rate = .001
_D.train.epochs = 200
_D.train.batch_size = 32
_D.train.loss = "mse"
_D.train.metrics = ["acc"]
_D.train.verbose = 2

# Callback parameters
_D.callbacks = CN()
_D.callbacks.names = ["EarlyStopping", "FileMetricLogger"]
_D.callbacks.EarlyStopping = CN()
_D.callbacks.EarlyStopping.patience = 50
_D.callbacks.EarlyStopping.min_delta = .0001

# Evaluation parameters
_D.evaluate = CN()

# Misc parameters
_D.misc = CN()
_D.misc.default_duplicate = 1 # Duplicates experiments by this amount. Only activates if
all options are empty

# Construct list of configuration keys and their possible options
# If the key is in the list, the default is overwritten, unless its corresponding
value list is empty
all_options_dict = {}
```

