

NNI进阶项目Training a classifier优化文档

一、运行环境

系统: windows10

环境: Anaconda / Visual Studio Code

PyTorch

导入包: 运用pip工具安装numpy, torch, torchvision三个关键包。

```
import argparse
import functools
import logging
import os
import pprint

import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torchvision
import torchvision.models as models
import torchvision.transforms as transforms

import nni
from utils import AverageMeterGroup, accuracy, prepare_logger, reset_seed
```

二、文件说明

主函数文件

- def train(model, loader, criterion, optimizer, scheduler, args, epoch, device) 训练函数
- def test(model, loader, criterion, args, epoch, device) 验证函数
- def main(args): 主要进行三种优化器的选择

```
if args['optimizer'] == 'adam':
    optimizer = optim.Adam(model.parameters(), lr=args['initial_lr'],
weight_decay=args['weight_decay'])
else:
    if args['optimizer'] == 'sgd':
        optimizer_cls = optim.SGD
    elif args['optimizer'] == 'rmsprop':
        optimizer_cls = optim.RMSprop
    optimizer = optimizer_cls(model.parameters(), lr=args['initial_lr'],
momentum=args['momentum'], weight_decay=args['weight_decay'])
```

yml配置文件

```
authorName: SanDro #作者名
experimentName: SanDro #项目名
```

```

trialConcurrency: 1 #同时运行的最大尝试数
maxExecDuration: 24h #最长持续时间
maxTrialNum: 10 #最大尝试次数
trainingServicePlatform: local #本地训练, 可选local, remote, pai
searchSpacePath: search_space.json #搜索空间文件
useAnnotation: false #是否允许注释方式配置搜索空间, 可选true, false
tuner: #调节器选项
    #SMAC (SMAC should be installed through nnictl)
    builtinTunerName: TPE
    classArgs: #调节器算法参数
        optimize_mode: maximize
trial: #尝试选项
    command: python main.py
    codeDir: .
    gpuNum: 0

```

json搜索空间文件

```

{
  "initial_lr":{"_type":"choice", "_value":[0.1, 0.01, 0.001, 0.0001]},
  "ending_lr":{"_type":"choice", "_value":[0]},
  "weight_decay":{"_type":"choice", "_value":[5e-4, 4e-4, 3e-4, 2e-4]},
  "cutout":{"_type":"choice", "_value":[0]},
  "batch_size":{"_type":"choice", "_value":[128]},
  "epochs":{"_type":"choice", "_value":[10]},
  "momentum":{"_type":"choice", "_value":[0.9, 1.0]},
  "num_workers":{"_type":"choice", "_value":[2]},
  "seed":{"_type":"choice", "_value":[42, 50, 37]},
  "grad_clip":{"_type":"choice", "_value":[0]},
  "log_frequency":{"_type":"choice", "_value":[20]},
  "optimizer":{"_type":"choice", "_value":["sgd","rmsprop","adam"]},
  "model":{"_type":"choice", "_value":
["resnet18","resnet50","vgg16","vgg16_bn","densenet121","squeezenet1_1"]}
}

```

三、运行结果

Experiment

Name	SanDro	Start time	6/11/2020, 6:54:48 PM	Log directory	C:\Users\61457\nni\experiments\tKM60ynz
ID	tKM60ynz	End time	6/11/2020, 8:58:44 PM	Training platform	local

Status

Status: **DONE**

Duration: 0 / 1h (Max duration: 2h 20min)

Trial numbers: 0 / 10 (Max trial number: 10)

Best metric: N/A

Spent	Remaining	Concurrency
1h 20min	39min	1

Running: 0, Succeeded: 3, Stopped: 0, Failed: 7

Search space

```

1 {
2   "initial_lr": {
3     "_type": "choice",
4     "_value": [
5       0.1,
6       0.01,
7       0.001,
8       0.0001
9     ]
10  },
11  "ending_lr": {
12    "_type": "choice",
13    "_value": [
14      0
15    ]
16  },
17  "weight_decay": {
18    "_type": "choice",
19    "_value": [

```

Config

```

1 {
2   "revision": 497,
3   "execDuration": 4840,
4   "nextSequenceId": 11,
5   "params": {
6     "authorName": "SanDro",
7     "trialConcurrency": 1,
8     "maxExecDuration": 7200,
9     "maxTrialNum": 10,
10    "tuner": {
11      "builtinTunerName": "TPE",
12      "classArgs": {
13        "optimize_mode": "maximize"
14      },
15      "checkpointDir": "C:\Users\61457\nni\experiments\tKM60ynz\checkpoints"
16    },
17   "versionCheck": true,
18   "clusterMetadata": [
19     {

```

