malware项目代码文档

malware项目是通过二进制文件流，判别一个软件是否为恶意软件，该文档的研究方向是致力于使用convolution neural network(cnn)的模型解决这个问题，而cnn的处理数据需满足structure data，并且关注于局部信息，所以在理论上cnn是适合于这个问题.

数据处理

我们通过将二进制流每 8 位转化为一个 8 bit数据，范围为[0,255],然后将其整合为二维图像数据．

抽取二进制数据

给定训练集(train.csv)和测试集(test.csv),代码为copy_file.py.根据两个csv文件中的malware ID去指定的文件夹中搜索文件，然后储存到文件中，文件的目录结构为：

```
.
├── test
│   ├── malware
│   └── normal
├── train
│   ├── malware
│   └── normal
```

代码运行的格式:

```
"""
positional arguments:
  source_root   path of source file
  des_root      path of store destination

optional arguments:
  -h, --help    show this help message and exit
  --train, -t   bool value,represent whether train or test.
  --server, -s  bool value,run in server or local
  """
python copy_file.py /macml-data/ /home/lili/Tao_Zhang/malware_2017_9_7/train --server --train
```

将二进制数据转化为图片

将抽取出来所有的二进制文件都转化为图片

utils.py

```
"""
将一个文件转化为图片
args:
    source_path:源文件的地址
    des_path:图片的储存地址
    size:图片的大小
"""
def transform_to_img(source_path,des_path,size):
```

transfer_img.py

将train和test文件夹中的所有文件都转化为图片，数据的储存目录格式为:

```
img
├── test
│   ├── malware
│   └── normal
├── train
```

```
│   ├── malware
│   └── normal
```

```
"""
positional arguments:
  source_dir   path of source file
  des_dir      path of store image file

optional arguments:
  -h, --help   show this help message and exit
  --size SIZE  size of image
"""
python transfer_img.py /home/lili/Tao_Zhang/malware_2017_9_7 /home/lili/Tao_Zhang/malware_2017_9_7/img
```

上述就是所有的数据处理的代码，所有的文件都储存在img文件，img文件夹的格式就如上述所示．

模型构建

测试系统环境：ubuntu14.04LTS
测试代码环境：pytorch0.2.0+GPU(需要设置config文件中的slef.cuda=True,否则默认使用cpu训练)

目录结构：

```
├── config.py
├── data
│   ├── dataset.py
│   ├── __init__.py
├── main.py
├── models
│   ├── BasicModel.py
│   ├── __init__.py
│   └── SPL
├── README.md
├── test.ipynb
└── utils
```

config.py

该文件为模型的配置文件，其中涉及到超参的取值，包括learning-rate等，其中与self-paced learning相关的参数并没有在base-line的模型中使用到.

models

pytorch构建的cnn模型

data

处理数据的代码，将img目录下的图像文件，整合读入到一起输入到模型中

main.py

主文件，

```
python main.py#就可以运行
```