An example of packaging deep learning model "AIGenMols" for MolAICal

Qifeng Bai, Shuoyan Tan
Email: molaical@yeah.net
Homepage: https://molaical.github.io
Lanzhou University
Lanzhou, Gansu 730000, P. R. China

1. Introduction

Sometimes, the "AIGenMols" have not good compatibility in Linux operating system. In this case, it needs to generate the binary "AIGenMols" again. In this tutorial, the ORGAN (https://github.com/gablg1/ORGAN) is chosen for the installation of “AIGenMols”. Besides, you can build the "AIGenMols" from your trained deep learning model.

2. Materials

2.1. Software requirement

1) MolAICal: https://molaical.github.io

2) Anaconda: https://www.anaconda.com

You can choose the free version of individual edition of Anaconda. The version of Anaconda should be chosen based on Pyhon 3.x rather than Python 2.x, etc.

2.2. Example files

1) All the necessary tutorial files are downloaded from:

https://github.com/MolAICal/specialtopic/tree/master/012-AIGenMols

3. Procedure

3.1. Install Anconda (https://www.anaconda.com/distribution).

Make sure installation completely and environment variables are set rightly.

3.2. Download “spec-list.txt” and create an independent environment

```
#> conda create --name AIGen --file spec-list.txt
#> conda activate AIGen
```

Note: If some errors happen like: *CondaHTTPError: HTTP 404 NOT FOUND for url <https://conda.anaconda.org/conda-forge/linux-64/xxxx-xxxx>*, you can delete corresponding soft package in “spec-list.txt”. Then repeat above steps. The missing soft package can be installed without version number via similar below step:

```
#> conda install -c conda-forge xxxx
```

3.3. Install the necessary libraries

```
#> conda install -c conda-forge openblas
#> conda install -c conda-forge openssl
#> pip install editdistance==0.3.1
```

3.4. Finally, produce binary “deep learning model”.

Decompress “organ.zip”

```
#> unzip organ.zip
#> cd organ
#> pyinstaller --add-data="organ/NP_score.pkl.gz:organ" --add-
```

data="organ/SA_score.pkl.gz:organ" --add-data="organ/data/FDA-H.csv:organ/data" --add-data="organ/checkpoints/FDA-H/FDA-H_99.ckpt.data-00000-of-00001:organ/checkpoints/FDA-H" --add-data="organ/checkpoints/FDA-H/FDA-H_99.ckpt.index:organ/checkpoints/FDA-H" --add-data="organ/checkpoints/FDA-H/FDA-H_99.ckpt.meta:organ/checkpoints/FDA-H" --add-data="organ/data/FDA1884.csv:organ/data" --add-data="organ/checkpoints/FDA1884/FDA1884_119.ckpt.data-00000-of-00001:organ/checkpoints/FDA1884" --add-data="organ/checkpoints/FDA1884/FDA1884_119.ckpt.index:organ/checkpoints/FDA1884" --add-data="organ/checkpoints/FDA1884/FDA1884_119.ckpt.meta:organ/checkpoints/FDA1884" --add-data="organ/data/zinc.csv:organ/data" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.data-00000-of-00001:organ/checkpoints/ZINC" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.index:organ/checkpoints/ZINC" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.meta:organ/checkpoints/ZINC" -F main.py

Open folder “organ/dist”
#> cd dist

You will see file named “main”, rename it as “AIGenMols”
#> mv main AIGenMols
#> chmod +x AIGenMols

The deep binary deep learning model named “AIGenMols” is produced completely. Put it into MolAICal software folder directly.

Analysis of install process:

--add-data="organ/data/zinc.csv:organ/data" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.data-00000-of-00001:organ/checkpoints/ZINC" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.index:organ/checkpoints/ZINC" --add-data="organ/checkpoints/ZINC/ZINC_99.ckpt.meta:organ/checkpoints/ZINC" -F main.py

---

You can replace above data by trained deep learning model.

Notice:

1). When you install Anaconda, the install path of Anaconda should have no blank space for packaging “AIGenMols”. For example, /home/test install/anaconda. The blank space between the characters “test” and “install” may lead to the failure of the packaging process.

2). The command usage of “AIGenMols” should be like as below:

AIGenMols -f test.txt -s FDAFrag -n 100
AIGenMols -f test.txt -s ZINCMol -n 100

-f: represents file name which contains generated results of molecules with SMILES format

-s: represents selection of deep learning model
-n: represents the number of generated molecules

This is just an example for packaging “AIGenMols”. You can use the similar way for your trained model. We will update deep learning model.