TÀI LIỆU HƯỚNG DẪN SỬ DỤNG

HƯỚNG DẪN HUẤN LUYỆN MÔ HÌNH ÂM THANH

SINH VIÊN THỰC HIỆN :

1. NGUYỄN HOÀNG QUYÊN **MSSV:** 1712712
2. TRẦN NGỌC QUANG **MSSV:** 1712706

EMAIL: 1712712@student.hcmus.edu.vn
1712706@student.hcmus.edu.vn
GVHD: TS. NGÔ HUY BIÊN

Khoa Công nghệ thông tin
Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

1. GIỚI THIỆU

Tài liệu này sẽ trình bày các bước tiến hành huấn luyện mô hình DeepSpeech 2. Mã nguồn mô hình huấn luyện và dữ liệu giọng nói Tiếng Việt được lưu ở thư mục VASR/SOURCE/1_Model_Data trong đĩa CD đính kèm.

Môi trường thực hiện trong tài liệu là hệ điều hành Ubuntu 16.04 và đã được cài đặt sẵn Python 3.6, gói cài đặt module mặc định là pip3

2. CHUẨN BỊ

2.1. Cài đặt các thư viện

- Cài đặt các thư viện bên dưới để có thể biên dịch mã nguồn:

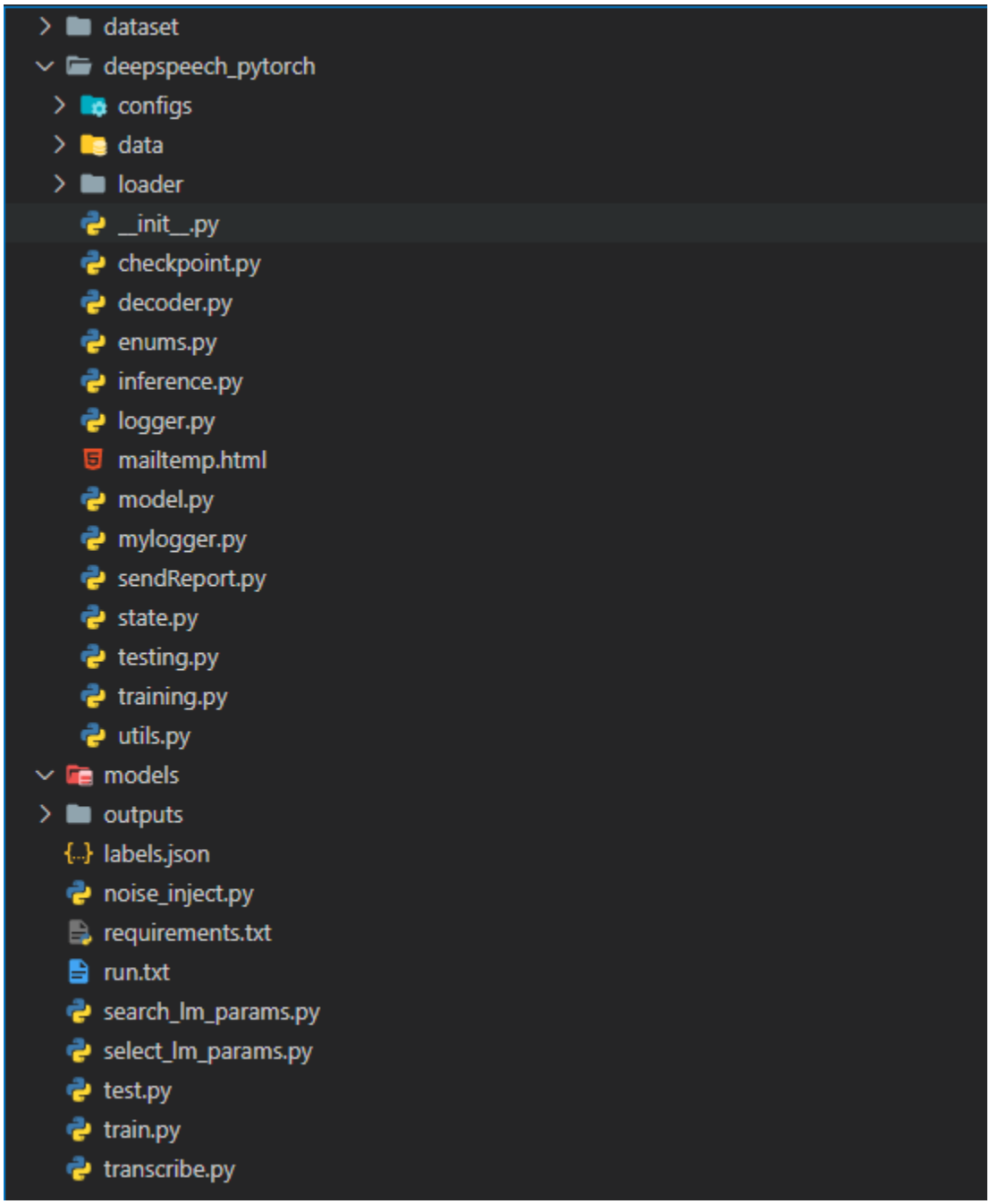
```
scipy
numpy
soundfile
python-levenshtein
torch
torchelastic
visdom
wget
librosa
numba==0.43.0
llvmlite==0.32.1
tqdm
matplotlib
flask
sox
sklearn
soundfile
pytest
hydra-core==1.0.0rc1
google-cloud-storage
jupyter
```

- Các thư viện được liệt kê đầy đủ kèm theo phiên bản phù hợp trong tập tin ***requirements.txt***. Để tiến hành cài đặt, ta mở terminal và thực hiện dòng lệnh:

pip install -r requirements.txt

2.2. Cấu trúc thư mục

- Cấu trúc thư mục mặc định của mô hình được mô tả như hình dưới:



- Trong đó:
  + **dataset** là thư mục chứa dữ liệu giọng nói và văn bản.
  + **deepspeech_pytorch** là thư mục chưa các tập tin cấu hình, tập tin thực thi cho mã nguồn, tập tin mẫu cho việc gửi email thông báo quá trình, …

+ **models** là mặc định chứa các tập tin mô hình, checkpoint của mô hình trong quá trình huấn luyện, có thể khôi phục, tiếp tục huấn luyện nếu bị ngắt quãng, gián đoạn.
+ **outputs** là thư mục chứa kết quả trong quá trình huấn luyện

3. THAM SỐ HUẤN LUYỆN

Tài liệu này chỉ nêu lên một số tham số thực sự cần thiết và có ảnh hưởng đến quá trình huấn luyện để mang lại kết quả tốt nhất.

3.1 Các tham số huấn luyện (cần phải truyền vào)

Những tham số này có thể được cấu hình mặc định trong tập tin train_config.py, class DataConfig. Tuy nhiên, dễ sử dụng một cách linh hoạt, ta cần truyền vào bằng tham số dòng lệnh:

- data.train_manifest : đường dẫn đến bộ dữ liệu huấn luyên train.csv (mặc định "./dataset/vi_train.csv").
- data.val_manifest : đường dẫn bộ dữ liệu đánh giá dev.csv (mặc định "./dataset/vi_test.csv").
- data.batch_size: chọn batchsize phù hợp với cấu hình của thiết bị (mặc định 32).
- training.epoch: số epoch cần train (mặc định 50).
- checkpointing.checkpoint: cho phép lưu lại các checkpoint (mặc định true)
- checkpointing.load_auto_checkpoint: cho phép tự động khôi phục checkpoint từ lần training trước đó (mặc định true)
- data.num_workers: số worker được sử dụng để training (mặc định 0)

3.2 Các tham số huấn luyện (không bắt buộc)

(Các tham số này nếu không được gán giá trị, sẽ lấy giá trị mặc định được gán trong file **model/util/flags.py**, tài liệu chỉ trình bày những tham số quan trọng có khả năng tăng giảm độ chính xác của mô hình cao)

Các tham số này nếu không được gán giá trị sẽ lấy giá trị được cấu hình sẵn trong tập tin **training_config.py**

- **labels_path**: đường dẫn tập tin từ điển các kí tự Tiếng Việt (mặc định **labels.json**)
- **rnn_type**: Loại mạng nơ-ron được sử dụng
- **hidden_size**: mặc định **1024**.
- **hidden_layers**: số lớp của mạng nơ-ron (mặc định **5**)
- **sample_rate**: số mẫu trong một khoảng thời gian nhất định (mặc định **16000**)
- **window_size** : kích thước của sổ context (mặc định **0.2**)
- **save_folder**: thư mục lưu mô hình và các checkpoint (mặc định **models**)
- **save_n_recent_models**: số lượng checkpoint liên tiếp được lưu xuống bộ nhớ (mặc định **5**)
- **best_val_model_name**: tên tập tin mô hình cuối cùng được lưu xuống bộ nhớ (mặc định **deepspeech_final.pth**)

4. CẤU HÌNH EMAIL

Để thuận tiện cho việc theo dõi kết quả quá trình huấn luyện, ta cấu hình email vào tập tin **sendReport.py**. Hệ thống sẽ tự động gửi báo cáo, kết quả huấn luyện đến email nhận được cấu hình sẵn.

```
6   USERNAME = 'tendangnhap@gmail.com'
7   FROM_EMAIL = 'tenmailgui@zohomail.com'
8   MY_PASSWORD = 'mypassword'
9   TO_EMAIL = 'email1@gmail.com,email2@gmail.com,email3@gmail.com'
10  TEMPLATE_PATH = "work/Source/deepspeech.pytorch/deepspeech_pytorch/mailtemp.html"
11  def parse_template(file_name):
12      with open(file_name, 'r', encoding='utf-8') as msg_template:
13          msg_template_content = msg_template.read()
14      return Template(msg_template_content)
15
16  def sendReport(epoch,traintime, loss,wer,cer, lr, note):
17        today = date.today()
18        datee = today.strftime("%d/%m/%Y")
19        message_template = parse_template(TEMPLATE_PATH)
20        smtp_server = smtplib.SMTP_SSL('smtp.zoho.com', 465)
21        smtp_server.login(USERNAME, MY_PASSWORD)
22        multipart_msg = MIMEMultipart()
23        message = message_template.substitute(
24              EPOCH_STT=epoch,
25              traintime =traintime,
26              loss = loss,wer=wer     ,cer=cer,lr=lr,
27              note = note
28              )
29        multipart_msg['From']=FROM_EMAIL
30        multipart_msg['To']= TO_EMAIL
31        multipart_msg['Subject']= str(datee)+ " REPORT SUMMARY EPOCH : "+str(epoch)
32        multipart_msg.attach(MIMEText(message, 'html'))
33        smtp_server.send_message(multipart_msg)
34        del multipart_msg
35        smtp_server.quit()
```

- Trong đó:

+ USERNAME: tên đăng nhập vào dịch vụ email
+ FROM_EMAIL: tên email gửi
+ MY_PASSWORD: mật khẩu đăng nhập email
+ TO_EMAIL: danh sách email người nhận
+ TEMPLATE_PATH: đường dẫn đến tập tin mẫu email

Chú ý: Nhóm sử dụng dịch vụ email của Zoho mail, thông tin SMTP của Zoho mail như sau

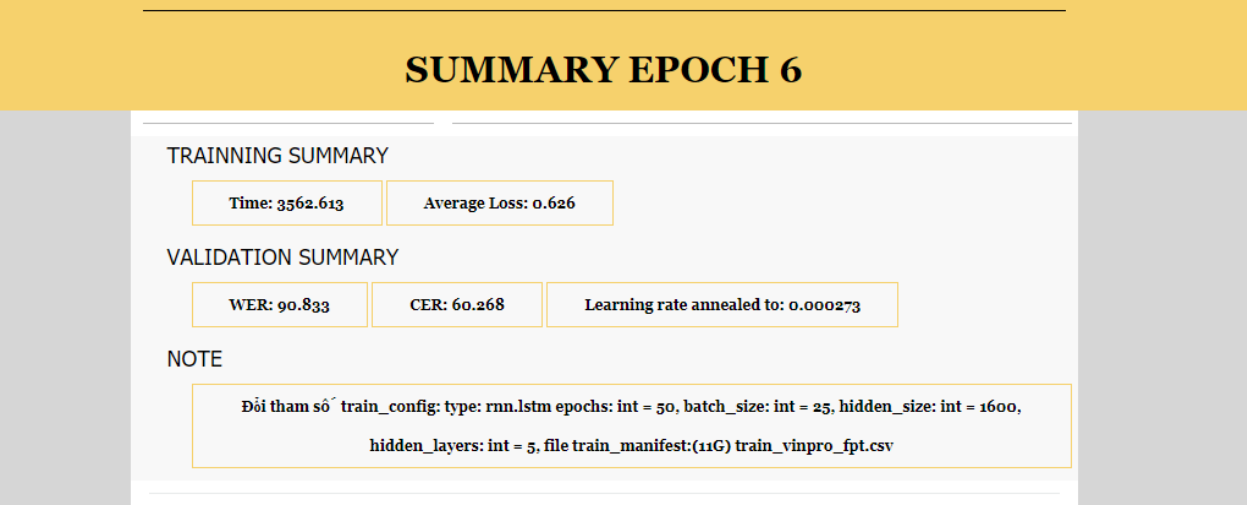
+ SMTP Server: smtp.zoho.com
+ SMTP Port: 465

Thông tin SMTP của Gmail

+ SMTP Server: smtp.gmail.com

+ SMTP Port: 587

Kết quả sau khi huấn luyện một epoch:



5. TIẾN HÀNH HUẤN LUYỆN

- Mở terminal tại thư mục chứa mã nguồn

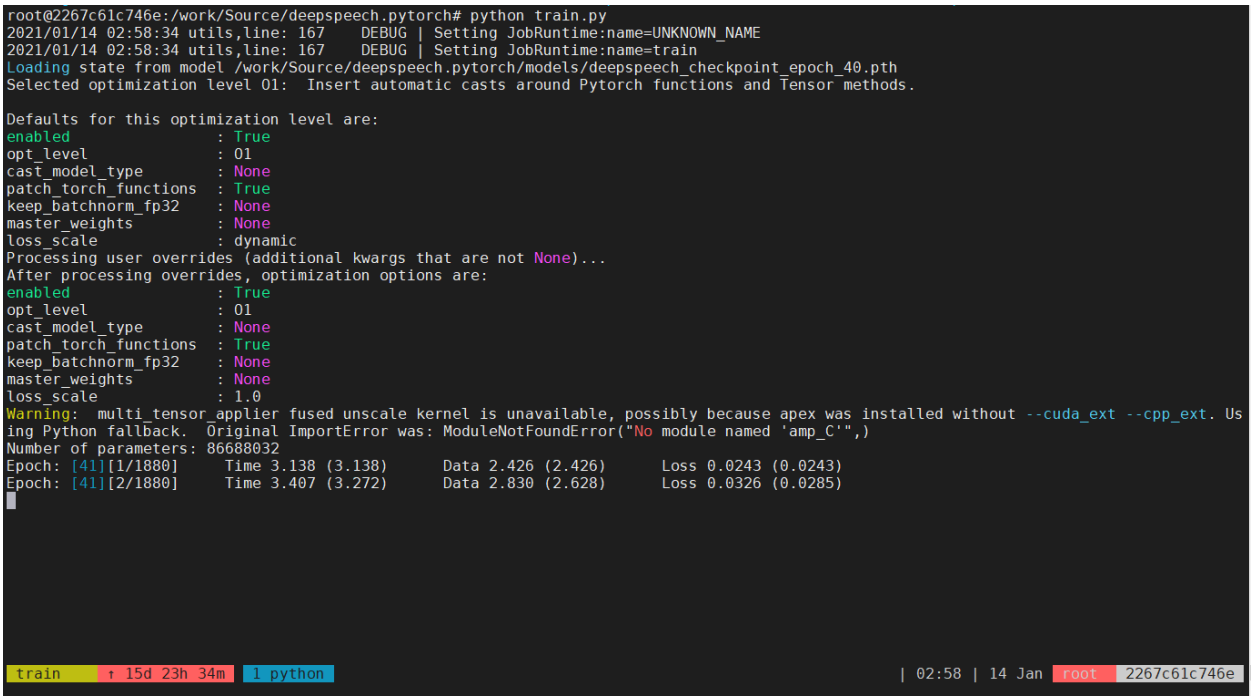
+ Nếu cấu hình tất cả các tham số đầy đủ trong tập tin training_config.py, ta gõ câu lệnh:

python train.py

+ Nếu muốn thay đổi các tham số huấn luyện, ta gõ câu lệnh như sau

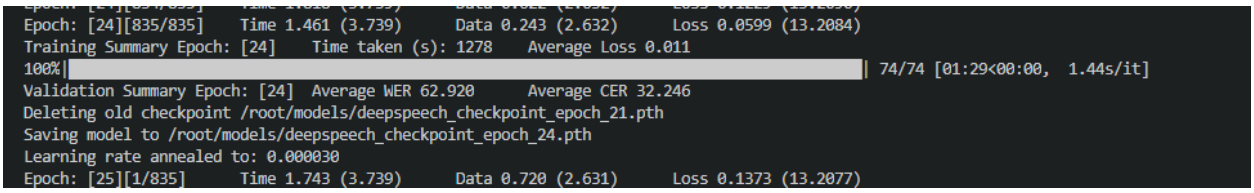
python train.py data.train_manifest="YOUR_PATH" data.val_manifest="YOUR_PATH" data.batch_size=NUMBER training.epochs=NUMBER checkpointing.checkpoint=BOOLEAN checkpointing.load_auto_checkpoint= BOOLEAN data.num_workers=NUMBER

Kết quả thực hiện:

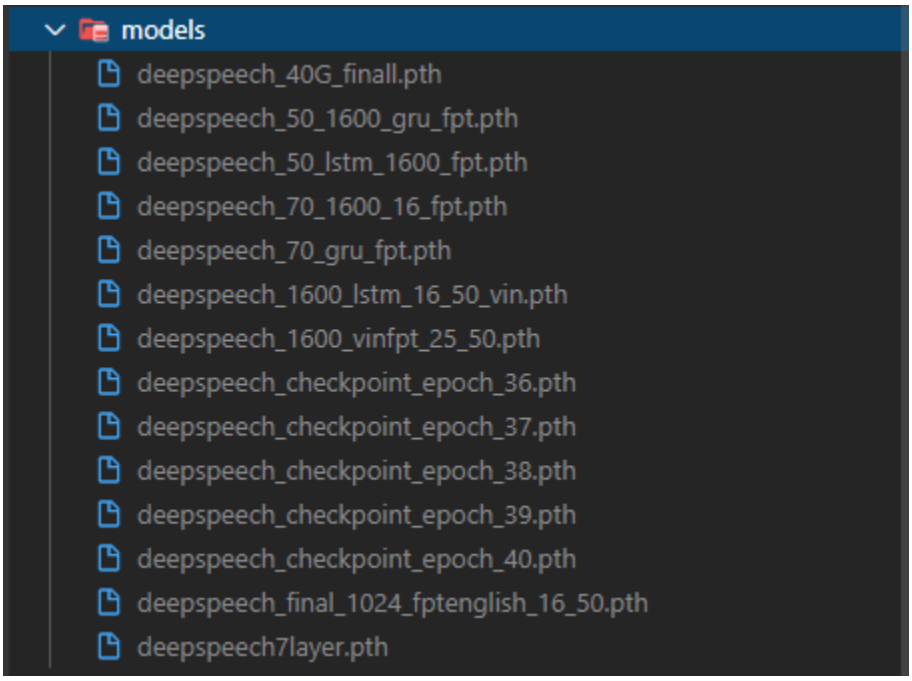


6. KẾT QUẢ

- Sau khi kết thúc mỗi epoch, màn hình cửa sổ dòng lệnh sẽ hiển thị như sau:



- Mô hình và các checkpoint được lưu trong thư mục models



- Sau khi train xong tất cả epoch, kết quả hiển thị như sau:

```
Epoch: [70][817/835]    Time 0.780 (2.634)      Data 0.169 (1.766)      Loss 0.0615 (4.6511)
Epoch: [70][818/835]    Time 1.042 (2.633)      Data 0.174 (1.766)      Loss 0.1106 (4.6510)
Epoch: [70][819/835]    Time 0.839 (2.633)      Data 0.154 (1.766)      Loss 0.1017 (4.6510)
Epoch: [70][820/835]    Time 0.882 (2.633)      Data 0.211 (1.766)      Loss 0.2614 (4.6509)
Epoch: [70][821/835]    Time 0.868 (2.633)      Data 0.154 (1.766)      Loss 0.1066 (4.6508)
Epoch: [70][822/835]    Time 0.834 (2.633)      Data 0.185 (1.766)      Loss 0.1086 (4.6507)
Epoch: [70][823/835]    Time 1.067 (2.633)      Data 0.170 (1.766)      Loss 0.1040 (4.6507)
Epoch: [70][824/835]    Time 0.880 (2.633)      Data 0.167 (1.766)      Loss 0.1195 (4.6506)
Epoch: [70][825/835]    Time 1.049 (2.633)      Data 0.164 (1.766)      Loss 0.0831 (4.6505)
Epoch: [70][826/835]    Time 0.876 (2.633)      Data 0.148 (1.766)      Loss 0.0531 (4.6504)
Epoch: [70][827/835]    Time 0.955 (2.633)      Data 0.220 (1.766)      Loss 0.0685 (4.6503)
Epoch: [70][828/835]    Time 1.126 (2.633)      Data 0.257 (1.766)      Loss 0.1346 (4.6503)
Epoch: [70][829/835]    Time 1.090 (2.633)      Data 0.225 (1.766)      Loss 0.2194 (4.6502)
Epoch: [70][830/835]    Time 1.099 (2.633)      Data 0.222 (1.766)      Loss 0.2424 (4.6501)
Epoch: [70][831/835]    Time 0.606 (2.633)      Data 0.124 (1.766)      Loss 0.1056 (4.6500)
Epoch: [70][832/835]    Time 0.876 (2.633)      Data 0.194 (1.766)      Loss 0.2614 (4.6500)
Epoch: [70][833/835]    Time 1.043 (2.633)      Data 0.162 (1.765)      Loss 0.0371 (4.6499)
Epoch: [70][834/835]    Time 0.899 (2.633)      Data 0.188 (1.765)      Loss 0.5058 (4.6498)
Epoch: [70][835/835]    Time 0.954 (2.633)      Data 0.159 (1.765)      Loss 0.0937 (4.6497)
Training Summary Epoch: [70]    Time taken (s): 766     Average Loss 0.005
100%|                                                                                | 74/74 [00:47<00:00,  1.31it/s]
Validation Summary Epoch: [70]  Average WER 61.982      Average CER 31.303
Deleting old checkpoint /root/models/deepspeech_checkpoint_epoch_67.pth
Saving model to /root/models/deepspeech_checkpoint_epoch_70.pth
Learning rate annealed to: 0.000000
/usr/local/lib/python3.6/dist-packages/omegaconf/basecontainer.py:244: UserWarning: cfg.pretty() is deprecated and will be remove
d in a future version.
Use OmegaConf.to_yaml(cfg)

  category=UserWarning,
```

7. THỬ NGHIỆM

Tập tin **transcribe.py** cho phép chạy chuyển giọng nói thành văn bản. Tập tin cấu hình các tham số là **inference_config.py.**

Trong đó, các tham số cần truyền vào:

+ **audio_path**: đường dẫn tập tin âm thanh đầu vào
+ **model.path**: đường dẫn mô hình âm thanh đã được huấn luyện

Các tham số có thể được cấu hình sẵn trong class **LMConfig**

+ **lm_path:** đường dẫn đến mô hình ngôn ngữ
+ **decoder_type:** loại mô hình ngôn ngữ

Các tham số về mô hình ngôn ngữ (nếu có) được cấu hình trong
Để tiến hành thử nghiệm mô hình được huấn luyện, ta mở cửa sổ dòng lệnh terminal tại thư mục chứa mã nguồn, sau đó gõ lệnh:
python transcribe.py model.path="YOUR_PATH" audio_path="YOUR_PATH"
Kết quả:

```
root@2267c61c746e:/work/Source/deepspeech.pytorch# python transcribe.py model.model_path="/work/Source/deepspeech.pytorch/models/deepspeech_50_1600_gru_fpt.pth" audio_path="/dataset/wavtest/FPTOpenSpeechData_Set002_V0.1_010349.wav"
DEBUG :  {"output": [{"transcription": "cô có hai thứ trời phú một là nhan sắc hai là khả năng giao tiếp"}], "_meta": {"acoustic_model": {"path": "/work/Source/deepspeech.pytorch/models/deepspeech_50_1600_gru_fpt.pth"}, "language_model": {"path": "/work/languagemodel/ARPA_BINARY/final-1234.binary"}, "decoder": {"alpha": 2.0, "beta": 1.0, "type": "beam"}}}

 Output transcript :  cô có hai thứ trời phú một là nhan sắc hai là khả năng giao tiếp
```