[How to Train My Own Coqui model?]

목차

1. GCP 환경 구축
2. Corqui 세팅
3. 모델 학습시키기

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GCP 환경 구축

> 인스턴스 만들기

1. GCP 입장
2. 인스턴스 만들기 (이름은 꼭 구분 가능하도록)
3. 리전, 영역 설정
4. GPU : T4
5. 머신 유형 : n1-standard-4
6. 부팅 디스크 : Ubuntu, 100GB
7. 방화벽 옵션 세개 다 체크
8. 고급옵션-네트워킹-네트워크 태그-jupyter

> SSH 연결

1. sudo passwd 👉🏻 비밀번호는 편의상 1234로 두번 입력
2. su - 👉🏻 비밀번호 입력
3. wget <https://repo.anaconda.com/archive/Anaconda3-2022.10-Linux-x86_64.sh>
4. bash [Anaconda3-2022.10-Linux-x86\_64.sh](https://repo.anaconda.com/archive/Anaconda3-2022.10-Linux-x86_64.sh)
5. source ~/.bashrc
6. 뭔가 저장했던 내용이 날라가있다면 처음부터 다시 실행하고 conda activate 입력해보기
7. jupyter notebook --generate-config
8. vi /root/.jupyter/jupyter\_notebook\_config.py 👉🏻 a 눌러서 편집모드 전환 👉🏻  
   c = get\_config()  
   c.NotebookApp.ip='\*'  
   c.NotebookApp.open\_browser=False  
   c.NotebookApp.port = 8888 입력 👉🏻 esc, :wq 입력하여 저장 및 종료
9. jupyter notebook --no-browser --port=8888 --allow-root 으로 주피터 실행해보기  
   👉🏻 인스턴스의 외부IP주소:8888로 접속, 토큰은 아까까지 있었던 cmd 창에서 확인
10. 다시 cmd창으로 와서 ctr+c로 주피터 종료
11. curl https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install\_gpu\_driver.py --output install\_gpu\_driver.py
12. sudo python3 install\_gpu\_driver.py
13. sudo apt-get install nvidia-driver-515
14. conda install pytorch==1.13.0 torchvision==0.14.0 torchaudio==0.13.0 pytorch-cuda=11.7 -c pytorch -c nvidia
15. 이제 다시 주피터를 실행
16. import torch  
    torch.\_\_version\_\_으로 torch 버전이 1.13.0인지 확인

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Corqui 세팅

1. jupyter notebook 파일 생성
2. import torch  
   device = torch.device("cuda:0") 로 GPU 사용 (device 실행해서 GPU 연결됐는지 확인)
3. jupyter notebook의 root 폴더에 TTS-0.11.1.zip 파일 다운받아 업로드
4. !unzip TTS-0.11.1.zip 으로 압축 해제
5. %cd /root/TTS-0.11.1  
   !pip install -e .  
   !pip install tensorboard 로 requirements 모두 설치

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

모델 학습시키기

1. jupyter notebook의 root 폴더에 LJSpeech\_mini100.zip 파일 다운받아 업로드
2. !unzip LJSpeech\_mini100.zip 으로 압축 해제
3. 그 다음, 폴더 이름을 LJSpeech-1.1로 바꾸고, 그 안에 wavs 폴더를 만들어 음성 파일을 모두 옮기고, 무슨 txt파일 이름은 metadata.csv로 바꾸기  
   아래와 같은 구조로 만들면 됨!  
   Root  
   |--------TTS-0.11.1  
   |--------LJSpeech-1.1  
    |--------wavs  
    | |--------wav 파일들..  
    |--------metadata.csv
4. %cd /root/TTS-0.11.1 로 경로 변경
5. TTS-0.11.1 - recipes - ljspeech - vits\_tts - train\_vits.py 파일 열기
6. 14번째 줄을 formatter="ljspeech", meta\_file\_train="metadata.csv", path="/root/LJSpeech-1.1/"로 수정 및 저장 (아래에서 Epoch도 변경 가능)
7. 다시 주피터 파일로 돌아와서 %run recipes/ljspeech/vits\_tts/train\_vits.py 실행
8. %cd /root/TTS-0.11.1/recipes/ljspeech/vits\_tts/vits\_ljspeech-February-23-2024\_06+06AM-0000000 (단, 저 vist\_ljspeech… 이 폴더는 직접 확인하고 이름 수정이 필요함!)
9. !tts --text "Text for TTS, Text for TTS, Text for TTS, Text for TTS" --config\_path config.json --model\_path best\_model.pth 실행해서 output 생성 가능  
   “Text for TTS… 👉🏻 원하는 텍스트로 변경 가능

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

모델 이어서 학습시키기

1. train\_vits.py 수정하기  
   previous = '/root/TTS-0.11.1/recipes/ljspeech/vits\_tts/vits\_ljspeech-February-26-2024\_07+26AM-0000000'  
   op = '/root/TTS-0.11.1/recipes/ljspeech/vits\_tts/moretrainedmodel'  
   trainer = Trainer(  
   TrainerArgs(continue\_path=previous, skip\_train\_epoch=False),  
   config,  
   output\_path = op,  
   model=model,  
   train\_samples=train\_samples,  
   eval\_samples=eval\_samples,  
   )
2. 수정하면 이런 모습이 됩니다!!  
   
3. 다만, previous 경로는 기존에 훈련했던 결과가 저장된 폴더로 바꿔주어야 함!
4. op는 output\_path 지정해준건데, 그냥 결과는 덮어씌워지는듯..
5. 저 상태로 다시 %run recipes/ljspeech/vits\_tts/train\_vits.py하면 됨!!