Generate sound effect/music with Meta's AudioCraft

AudioCraft is a framework for applying generative AI in the sound field. It comes with a pre-trained model called AudioGen, which generates audio samples based on given descriptions. The following code block provides instructions for creating a Conda environment and running the script.

```
module load Anaconda3/2022.05 GCCcore/11.3.0 FFmpeg/4.4.2

# Create conda environment
# conda create -n [env_name]
conda create -n audioCraft

# source activate [env_name]
source activate audioCraft

# Install required packages
conda install pip
pip3 install git+https://github.com/facebookresearch/audiocraft.git

# Now you are ready to play around with the model on a GPU node
srun --pty -p gpu --cpus-per-task=12 --gres=gpu:a100:1 --mem=100G bash
python3 audio_craft_example.py
```

audio_craft_example.py imports the necessary packages and loads the pre-trained model from our storage. It then sets parameters for the audio generation and provides three sample descriptions. The model generates audio based on these descriptions, which is saved to a file using loudness normalization.

```
## audio_craft.py

import torchaudio
from audiocraft.models import AudioGen
from audiocraft.data.audio import audio_write

model = AudioGen.get_pretrained(
```

```
    '/pfss/toolkit/audio_craft_audiogen_medium_1.5b/snapshots/3b776a70d1d682d75e01ed5c4924ea31d156a62c/'
 )
 model.set_generation_params(duration=5)  # generate 8 seconds.
 descriptions = [
   'The sound of nails on a chalkboard in a noisy classroom',
   'someone chew with their mouth open',
   'sound of a car alarm going off repeatedly'
 ]
 wav = model.generate(descriptions)  # generates 3 samples.


 for idx, one_wav in enumerate(wav):
 # Will save under {idx}.wav, with loudness normalization at -14 db LUFS.
 audio_write(f'{idx}', one_wav.cpu(), model.sample_rate, strategy="loudness", loudness_compressor=True)
```

AudioGen is an autoregressive transformer LM that synthesizes general audio conditioned on text (Text-to-Audio). Internally, AudioGen operates over discrete representations learned from the raw waveform using an EnCodec tokenizer.

AudioGen was presented at AudioGen: Textually Guided Audio Generation by *Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, Yossi Adi*.

AudioGen 1.5B is a variant of the original AudioGen model that follows MusicGen architecture. More specifically, it is trained over a 16kHz EnCodec tokenizer with 4 codebooks sampled at 50 Hz and a delay pattern between them. Having only 50 auto-regressive steps per second of audio, this AudioGen model allows faster generation while reaching similar performances to the original AudioGen model introduced in the paper.

Revision #5
Created 4 August 2023 08:06:21 by Don Chu
Updated 21 August 2023 02:11:11 by Loki Ng