Audio Spectrogram Transformer

This report compares the execution speed of the AST ONNX model on the two AWS platforms: C5 CPU and g4dn GPU

1. Overview of AST model

The architecture of Audio Spectrogram Transformer (AST) model is shown in Figure 1. It includes three main parts as follows:

- **Linear Projection**: The 2D audio spectrogram is split into a sequence of 16×16 patches with overlap, and then linearly projected to a sequence of 1-D patch embeddings. Each patch embedding is added with a learnable positional embedding. An additional classification token is prepended to the sequence.
- **Transformer encoder [1]**: The output from the linear projection is input to a Transformer. They use the frequency/time masking mechanism to train the transformer encoder.
- **Linear decoding layer**: This layer with sigmoid activation maps the audio spectrogram representation to labels for classification.

2. Verify the recognition rate in the paper

In this section, I show the recognition rate of AST model on the ESC 50 dataset. The dataset consists of a total of 2,000 recordings each of 5-second durations, separated into 50 classes.

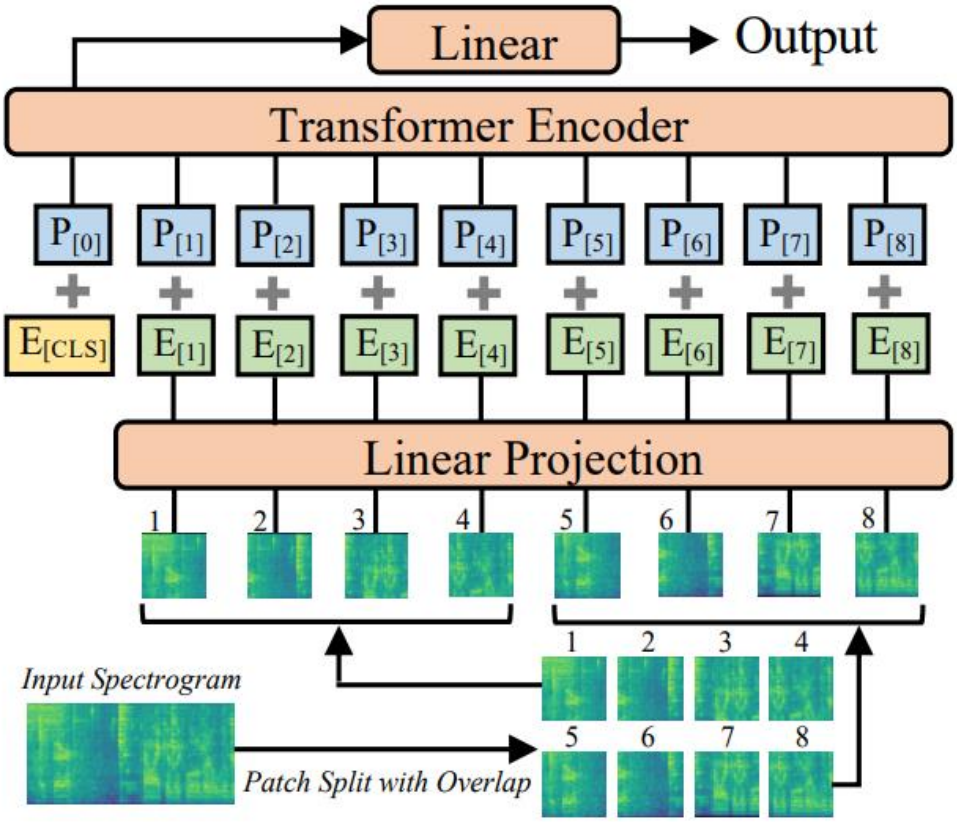

Figure 1. The architecture of the AST model.

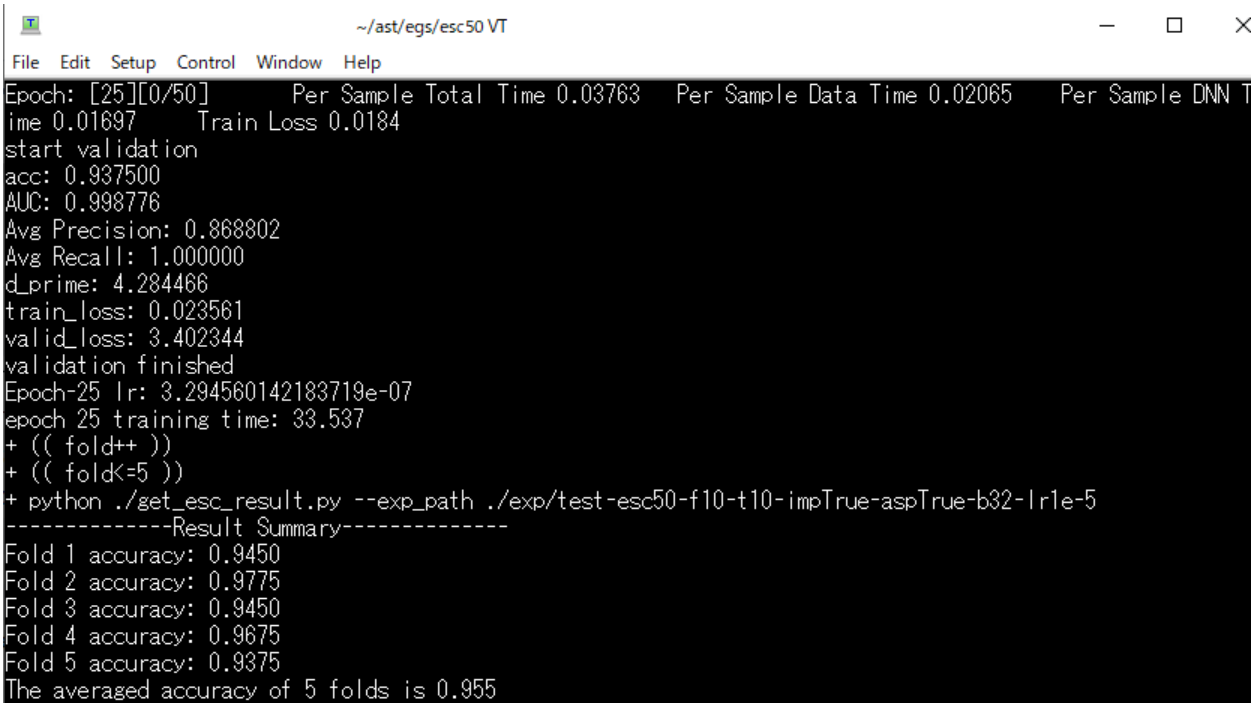

Figure 2. Experiment of AST model on ESC 50 dataset.

How to redo the experiment for the ESC 50 dataset:

- ❖ Download source code [AST source code]
- ❖ cd ast/egs/esc50 and run file run_esc50.sh. I changed some parameters in this file to make the program runnable. You can refer to run_esc50.sh to see what I changed.
  - Batch size : 48 (too big) → 16 or 32.
  - Memory leak when loading data: setup num_workers in dataloader less than the number of cpu cores.

Figure 2 shows the retraning result of AST model over 5-folds of the ESC 50 dataset after 25 epochs. The averaged accuracy of 5 folds is 0.955. This result is slightly lower than the result reported in the paper [1]: 0.9575.

3. Convert .pth model to .onnx model

Open Neural Network Exchange (ONNX) is an intermediary machine learning framework used to convert between different machine learning frameworks. ONNX is widely supported and can be found in many frameworks, tools, and hardware. Enabling interoperability between different frameworks and streamlining the path from research to production helps increase the speed of innovation in the AI community. ONNX helps to solve the challenge of hardware dependency related to AI models and enables deploying same AI models to several HW accelerated targets [ONNX introduction]

How to connvert the pytorch .pth AST model to a .onnx model. You can find more details in convert_onnx.py. To run this file, you should copy it into this folder: /ast/src. You should notice some points as follows:

- ❖ DataParallel: because data loader in AST model is run parallelly, we need to add ".module" to get the trained model.
- ❖ Set the model to inference mode: audio_model.eval()
- ❖ In AST model, they use "autocast" for GPU mode, so the input should be set to float16

4. Comparing the execution speed of .onnx model on CPU and GPU

In this experiment, I create two notebook instances by Sagemaker Amazon Web Services: one instance runs on is ml.c5.xlarge and another is ml.g4dn.xlarge. The differences between two instances are shown in Table 1.

Table 1. The comparison between c5.xlarge CPU and g4dn.xlarge GPU Sagemaker notebooks. [Price]

	ml.c5.xlarge	**ml.g4dn.xlarge**
Number of virtual CPUs	4	4
Memory (GiB)	8	16
Number of GPUs	0	1
On-Demand Price	$0.204 /hr	$0.7364/hr

The execution time when using the trained AST ONNX model to predict a batch of 5 audio files is shown in Figure 3. Because the instaces may be not stable, I run it three times on each instance and measure the execution time each time. I found an interesting point that the execution time on the c5 CPU instance is faster than on the g4dn GPU instance.

- CPU ml.c5.xlarge: Total execution time 3.54~3.79 seconds
- GPU ml.g4dn.xlarge: Total execution time 4.39~4.40 seconds

```
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on C5 CPU
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 3.79949(seconds)
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on C5 CPU
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 3.540787(seconds)
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on C5 CPU
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 3.548499(seconds)
```

(a) C5 CPU

```
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on g4dn.xlarge
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 4.397527(seconds)
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on g4dn.xlarge
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 4.403037(seconds)
sh-4.2$ python run_onnx.py
Start testing speed of Onnx AST model on g4dn.xlarge
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 4.40962(seconds)
```

(b) G4DN GPU

Figure 3. The execution time when running AST ONNX model on different instances. I use the onnx model to predict a batch of 5 audio files. The model is executed three times on each instance.

5. Comparing the execution time between pth model and onnx model

In this section, I compare the execution time between pth model and onnx model. The source code is run_pth.py. To run this file, you should copy it to the ast/src folder. The execution time of the pth model on C5 CPU and g4dn GPU is shown in Figure 4.

```
number of patches=600
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 5.18769(seconds)
sh-4.2$ python run_pth.py
Start testing speed of .pth AST model on c5.xlarge CPU
---------------AST Model Summary---------------
ImageNet pretraining: False, AudioSet pretraining: False
frequncey stride=10, time stride=10
number of patches=600
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 5.208363(seconds)
sh-4.2$ python run_pth.py
Start testing speed of .pth AST model on c5.xlarge CPU
---------------AST Model Summary---------------
ImageNet pretraining: False, AudioSet pretraining: False
frequncey stride=10, time stride=10
number of patches=600
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-103415-A-2: pig
Total execution time: 5.220908(seconds)
```

(a) C5 CPU

```
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 1.632019(seconds)
sh-4.2$ python run_pth.py
Start testing speed of .pth AST model on Tesla T4 g4dn.xlarge GPU
---------------AST Model Summary---------------
ImageNet pretraining: False, AudioSet pretraining: False
frequncey stride=10, time stride=10
number of patches=600
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 1.602754(seconds)
sh-4.2$ python run_pth.py
Start testing speed of .pth AST model on Tesla T4 g4dn.xlarge GPU
---------------AST Model Summary---------------
ImageNet pretraining: False, AudioSet pretraining: False
frequncey stride=10, time stride=10
number of patches=600
The recognition result of file 5-103415-A-2: pig
The recognition result of file 5-248341-A-6: hen
The recognition result of file 5-194899-A-3: cow
The recognition result of file 5-169983-A-5: cat
The recognition result of file 5-208030-A-0: dog
Total execution time: 1.608619(seconds)
```

(b) G4DN GPU

Figure 4. The execution time when running AST pth model on different instances. I use the pth model to predict a batch of 5 audio files. The model is executed three times on each instance.

The execution time of the pth model on C5 CPU is 5.18~5.22 seconds for 5 audio files while this time on g4dn GPU is 1.6~1.63 seconds. The summarization of execution is shown in Table 2.

Table 2. The comparison of exectuion time between onnx model and original pth model.

	ml.c5.xlarge (seconds)	**ml.g4dn.xlarge (seconds)**
pth model	5.18~5.22	**1.6~1.63**
Onnx model	3.54~3.79	4.39~4.40

Referrence

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pretraining of deep bidirectional transformers for language understanding,” in NAACL-HLT, 2019

[2] Gong, Yuan, Yu-An Chung, and James Glass. “Ast: Audio spectrogram transformer.” *arXiv preprint arXiv:2104.01778* (2021).