Preliminary report #1 on Semester task WS23

NVIDIA NeMo model and runtime on minicomputers

Here is the description of the work I have done during my first month on the task

1) I have chosen one of the test sets of LibriSpeech ASR corpus, namely test-clean.tar.gz and test-other.tar.gz and written a python-code to unzip files and collect them into a set of samples to feed the model later

2) I have chosen one of the suggested STT En Citrinet 1024 model as a top one listed in the suggested models to use with NeMo-library

3) With nemo-toolkit I converted the model to the ONNX-model to later measure any changes in the model performance

4) Machines: one of the hardest part in my work was trying to create an environment to run my notebook for performance measuring on a variety of existing set of machines. Later in the report I will list all machines, actions taken, problems encountered and outcomes reached

5) Ready to use 3 Jupyter notebooks:

   a. *ASR Librispeech Wav Audios Preparation* - unzipping a dataset in a *Librispeech* format and converting audio samples to a `.wav`-format suitable for NeMo-models

   b. *ASR NeMo-ONNX Conversion* - converting an already downloaded `.nemo`-file to a `.onnx`-file with model to use in ONNX-runtime

   c. *ASR Librispeech Wav Audios Traversal* - one cell with gathering audio samples from unzipped dataset of wav-s, one cell to feed these sample to an ONNX-runtime, the other cell to measure same samples in NeMo-runtime (test-other sample has been chosen)

Installing NeMo and ONNX-environments to use Citrinet model

NeMo library provides a mechanism to convert demo-model to onnx-model. I have used it successfully, although the inferencing would only produce word-piece-tokens and their probabilities. Therefore, I still needed to use some kind of preprocessing and decoding code before and after inferencing, using the classes from NeMo packages. Maintainers in NeMo-repository and forums suggest to use not the overall `nemo_toolkit[all]` package with model-training features, but the smaller one `nemo_toolkit[asr]` containing preprocessing and decoding classes.

Nevertheless, in order to find out the difference between original and converted models on machine I have tried to install all environments (ONNX, NeMo, NeMo-ASR).

For now I have got mixed results on trying to run initial/converted models on a range of devices (both minicomputers and regular PCs). I am listing all the test-cases I have tried and my conclusions in a chronological order:

Mainly I followed the guide from the official NeMo guide, sometimes referring to issue-reports on specific device requirements in the NeMo GitHub repository and posts in NVidia Developers Forum

1. ***Raspberry Pi 3 Model B*** *- No success*

Multiple errors during installation, `pip` cannot find the proper version if wheels to install needed libraries multiple incompatibility issues

2. ***Apple Silicon M1*** *- No success*

Created a condo environment, installed all packages successfully, but the code produces errors. I didn't proceeded to try, since these are not our target devices anyway

3. ***Intel Core i7*** *- Success*

Model path	Model size (Mb)	Peak size of memory blocks	Model load time (s)	Inference time (s)	Dataset samples count	Dataset length (s)	RTF (s)	WER	MER	WIL	WIP	CER
models/stt_en_citrinet_1024.onnx	572.260998	586784744	0.3548078536987 3047	945.037310123 4436	2939	19229.5701249999 73	0.0491	0.0708022085092 5625	0.0702812387400201	0.1248508314283 0666	0.8751491685716 933	0.0286774356755 8055
models/stt_en_citrinet_1024.nemo	574.24896	25801311	4.8938279151916 5	5390.99065589 9048	2939	19229.5701249999 73	0.2803	0.0704010087308 7136	0.0698829910299 94444	0.1241176191639 7124	0.8758823808360288	0.0284171316698 31865

4. ***Intel NUC 12 wshi3000*** *- Success (Not full table - I am going to run notebooks with more metrics again)*

Model name	Samples count	Processing time (s)	WER
stt_en_citrinet_1024.onnx	2939	3391.3835377693176	0.0708786275146629
stt_en_citrinet_1024.nemo	2939	2918.6985421180725	0.07049653248762967

5. ***Raspberry Pi 4 Model B*** *- Partial Success*

ONNX model produce results, but NeMo model stops the notebook from running once

Model path	Model size (Mb)	Peak size of memory blocks	Model load time (s)	Inference time (s)	Dataset samples count	Data set length (s)	RTF (s)	WER	MER	WIL	WIP	CER
stt_en_citrinet_1024.onnx	572.260998	598059374	14.515744447708 13	422.552756547927 86	100	716.945000000 0003	0.5894	0.074116305587 22919	0.073363431151 24154	0.128104564172 58087	0.871895435827 4191	0.023384049831 512304

the inferencing starts - here is a quick snippet on a 100 samples, I am going to run all other samples later to gather more accurate *RTFs, WERs etc*.

6. ***Nvidia Jetson Nano Kit*** *- No success*

Unfortunately, the whole process of NeMo environment installation is very laborious, I didn't succeeded to use NeMo-toolkit on it. As mentioned in the maintainer's post they don't support Jetson Nano, but I have tried to install `.[asr]` part only and faced incompatibility issues as well

7. ***Nvidia Jetson Xavier NX*** *- No success*

The situation is similar to Jetson Nano. Possibly, the right way of implementing ASR on Jetson-devices is not python code, rather ready-to-use containers

Further work

In my further work I will try to further explore Nvidia NeMo model, decide how to not use nemo-library classes for preprocessing and decoding of inferencing results when using ONNX-model, try out suggested alternatives for running models on arm-based SBCs.