NVIDIA Deep Speech

Piotr Ambroszczyk Łukasz Kondraciuk
Wojciech Przybyszewski Jan Tabaszewski

Our Team Programming Project is realized with the support of NVIDIA company, and we are going to implement speech recognition model from the following paper [1]. Most of the commercial speech recognition systems rely on complex algorithms (e.g. Hidden Markov Models) and use machine learning only as a minor part. Our program will use end-to-end deep learning method. Since `pytorch` framework is supported with CUDA, we decided to use this one.

There are a few problems we are going to face working on this project. Firstly, the efficiency of the model is strictly correlated with the size and number of training datasets. The network will have to process hours of utterances, so concurrency is crucial. As the fourth layer of the model is going to be bidirectional recurrent one and therefore model parallelism is not trivial. Secondly, before we start training our network using large data, we have to obtain that data. We will have to find utterances with transcriptions[1]; moreover, we have to pay attention to licences and copyright of that materials. Besides that, working with specially prepared, clear vocal sounds does not indicate success with processing words and statements of lower quality, and best model's learning parameters can vary in different languages. Last but not least, there are going to be a lot of implementation details such as sound supplying, storing, outputting transcription etc.

To solve the following problems, we plan to implement some optimizations. First, as `pytorch` framework supports usage of CUDA devices, we are going to train network using the power of multiple GPUs. Our goal is to obtain 2-way model parallelism based on independence of forward and backward iteration in the fourth layer[2] and realize 4-way data parallelism, which refers to partitioning the datasets across processes. Second, we have to find accurate training parameters such that the model will not overfit or underfit the data. Furthermore, we are going to use fixed precision numbers (FP16) to speed up system performance.

What we expect to achieve:

- Working in noisy environments

[1] Maybe audio-books?

[2] Processing within the fourth layer takes 40% of the total training time.

- Highly efficient performance[3]
- Real time speech recognition
- (hopefully) Polish language recognition

References

[1] Hannun et al. (2014) *Deep Speech: Scaling up end-to-end speech recognition*

[3] 16% error rate on the Switchboard Hub5'00 corpus.