EasyASR: A Distributed Machine Learning Platform for End-to-end Automatic Speech Recognition

Chengyu Wang,[1] **Mengli Cheng,**[1] **Xu Hu,**[2*] **Jun Huang**[1†]
[1] Alibaba Group [2] ByteDance Inc.
{chengyu.wcy, mengli.cml}@alibaba-inc.com, huxu.hx@bytedance.com, huangjun.hj@alibaba-inc.com

Abstract

We present EasyASR, a distributed machine learning platform for training and serving large-scale Automatic Speech Recognition (ASR) models, as well as collecting and processing audio data at scale. Our platform is built upon the Machine Learning Platform for AI of Alibaba Cloud. Its main functionality is to support efficient learning and inference for end-to-end ASR models on distributed GPU clusters. It allows users to learn ASR models with either pre-defined or user-customized network architectures via simple user interface. On EasyASR, we have produced state-of-the-art results over several public datasets for Mandarin speech recognition.

Introduction

As a fundamental task in speech and language processing, Automatic Speech Recognition (ASR) aims to generate transcripts from human speech. Recently, the successful application of deep neural networks has pushed the accuracy of end-to-end ASR models to a new level, but brings significant challenges for building large-scale, robust ASR systems, especially for industrial applications. Major bottlenecks are twofold: i) abundant labeled training data for learning large, accurate ASR models; and ii) an efficient distributed, computing framework for model training and serving at scale.

In this demo, we present EasyASR, a distributed machine learning platform to address both challenges. EasyASR is built upon the Machine Learning Platform for AI (PAI) of Alibaba Cloud [1], which provides an ultra-scale, deep learning framework on distributed GPU clusters. Our platform supports the complete process of training, evaluating and serving ASR models. Additionally, it is integrated with the functionalities i) to extract high-quality audio aligned with transcripts from massive video data and ii) to expand existing ASR training sets with various augmentation policies. We have designed easy-to-use PAI components that enable users to build or run ASR models within only a few lines of command, which hides complicated techniques from starters. We also provide add-on configurations with the PAI commands to allow advanced users to customize network architectures for their own models. On EasyASR, we achieve state-of-the-art performance for Mandarin speech recognition over multiple public datasets.

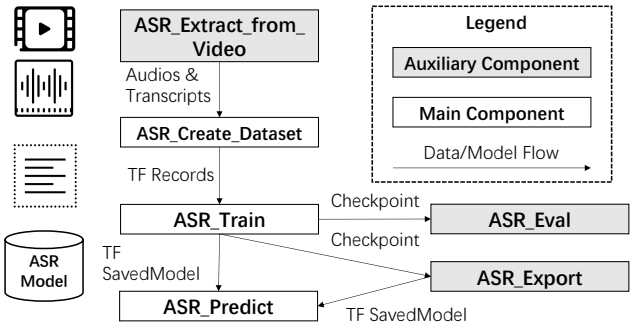

Figure 1: An overview of PAI's components in EasyASR.

Platform Description

In this section, we introduce the EasyASR platform in detail.

Function Design. On the EasyASR platform, each module is encapsulated as a PAI component, with the overall framework illustrated in Figure 1. Among the three main components, `ASR_Create_Dataset` extracts acoustic features from raw waves and generate audio-transcript pairs in the TFRecord format. Users also have the option to enlarge their training sets by various augmentation policies (Park et al. 2019) by passing optional parameters to this component. In `ASR_Train`, ASR models can be trained from scratch or fine-tuned given training sets, evaluation sets, model configurations and pre-trained model checkpoints (if available) as inputs. EasyASR supports various popular ASR model architectures such as Wav2Letter (Collobert, Puhrsch, and Synnaeve 2016) and Speech Transformer (Zhao et al. 2019). After training, the component automatically exports the selected checkpoint to the designated path as a TF SavedModel. The model can be used in the `ASR_Predict` component for fast inference.

Apart from the three components, EasyASR integrates the technique of extracting wave-transcript pairs from massive video data into the `ASR_Extract_from_Video` component to support weakly supervised training of ASR mod-

*The work was conducted when X. Hu was afflicated with Alibaba Group.

†J. Huang is the corresponding author.

[1]https://www.alibabacloud.com/product/machine-learning/

els. Interested readers may refer to (Cheng et al. 2020) for details. If the performance of the exported model is not satisfactory. Users can call `ASR_Eval` and `ASR_Export` to evaluate and export checkpoints of their own choice, instead of running the fully automatic process provided by `ASR_Train`.

Note that, EasyASR has already equipped its own model zoo trained by our team, containing a number of pre-trained ASR models with high accuracy. Hence, a simple call to `ASR_Predict` can fulfill basic requirements of common ASR applications. The remaining components are designed for domain-specific or other special scenarios.

System Design. EasyASR is not a *machine learning library* but rather a *machine learning platform for ASR applications*. Hence, various system optimization techniques are designed for large-scale model training. For example, the training procedures in EasyASR are implemented based on the PAISoar framework [2], which significantly speeds up the training process distributed across multiple workers and GPUs. The TensorFlow framework we use has been largely optimized to support faster mixed-precision training and have improved communication, memory allocation and I/O mechanisms.

User Interface. Despite its sophisticated system and model design, EasyASR is truly EASY to use. On our user interface [3], only a simple PAI command is required to call the components you need to use. For example, the following command can be used to i) train an ASR model from scratch (based on the model structure and other settings specified in `model_config`) and ii) to export the model to `model_export_dir` for the inference purpose:

```
PAI -name ASR_Train
-Dfinetune=false
-Dconfig='your_path/model_config'
-Dexport='your_path/model_export_dir'
-Dcluster='{"worker": {"count": 4,
    "cpu": 2000, "gpu": 800,
    "memory": 100000}}';
```

Here, four separate workers in the PAI cluster are employed to train the model based on data and model parallelism, each using 20 CPUs, 8 GPUs and 100GB memory.

Specifically, the configuration file `model_config` provides all details on training parameters, settings and the model structure. Take our transformer model as an example, a clip of the configuration file (in JSON) is as follows:

```
"encoder": TransformerEncoder,
"encoder_params": {
    "encoder_layers": 12,
    "num_heads": 8,...
  },
"decoder": JointCTCAttenDecoder,
"decoder_params": {
    "attn_decoder": TransformerDecoder,
    "attn_decoder_params": {
      "hidden_layers": 6,
      "num_heads": 8,...
    },
    "ctc_decoder": CTCDecoder,
    "ctc_decoder_params": {...},
  },
"loss": MultiTaskCTCEntropyLoss,
"loss_params": {
    "seq_loss_params": {...},
    "ctc_loss_params": {...},
    "lambda_value": 0.30,
}
```

As seen, the model uses both Connectionist Temporal Classification (CTC) and the transformer decoder to generate transcripts. By modifying the configuration file, advanced users have the liberty to customize their models.

Performance. Based on the improved transformer model described above and the weakly supervised training technique, we have produced state-of-the-art results on Mandarin speech recognition on six public datasets. The experimental results are reported in (Cheng et al. 2020).

Related Work and Discussion

Previously, various deep learning frameworks have been released for training and evaluating ASR models, such as Kaldi [4], OpenSeq2Seq [5], ESPNet (Watanabe et al. 2018) and wav2letter++ (Pratap et al. 2018). Our work is different from these frameworks as we integrate our ASR library with the PAI platform for efficient distributed learning. We provide easy-to-use PAI components on the platform for users with no re-development needed, and user-customized configurations and modules for advanced developers at the same time.

Conclusion

In this demo, we present EasyASR, a distributed machine learning platform for learning and serving large-scale, end-to-end ASR models. A simple user interface is created for users to learn ASR models with either pre-defined or user-customized network architectures based on PAI commands. On EasyASR, we produce state-of-the-art results for Mandarin speech recognition. In the future, we will continue to develop our platform to support more state-of-the-art ASR models and make our platform publicly available.

References

Cheng, M.; Wang, C.; Hu, X.; Huang, J.; and Wang, X. 2020. Weakly Supervised Construction of ASR Systems with Massive Video Data. *arXiv preprint* 2008.01300.

Collobert, R.; Puhrsch, C.; and Synnaeve, G. 2016. Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. *arXiv preprint* 1609.03193.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech*, 2613–2617.

[2] https://developpaper.com/how-to-improve-the-speed-of-deep-learning-training-100-times-here-comes-paisoar/

[3] https://datastudio.dw.alibaba-inc.com/

[4] https://github.com/kaldi-asr/kaldi

[5] https://github.com/NVIDIA/OpenSeq2Seq

Pratap, V.; Hannun, A.; Xu, Q.; Cai, J.; Kahn, J.; Synnaeve, G.; Liptchinsky, V.; and Collobert, R. 2018. wav2letter++: The Fastest Open-source Speech Recognition System. *arXiv preprint* 1812.07625.

Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Soplin, N. E. Y.; Heymann, J.; Wiesner, M.; Chen, N.; Renduchintala, A.; and Ochiai, T. 2018. ESPnet: End-to-End Speech Processing Toolkit. In *Interspeech*, 2207–2211.

Zhao, Y.; Li, J.; Wang, X.; and Li, Y. 2019. The Speechtransformer for Large-scale Mandarin Chinese Speech Recognition. In *ICASSP*, 7095–7099.