wav2letter++ : A Fast Open-Source Speech Recognition Framework

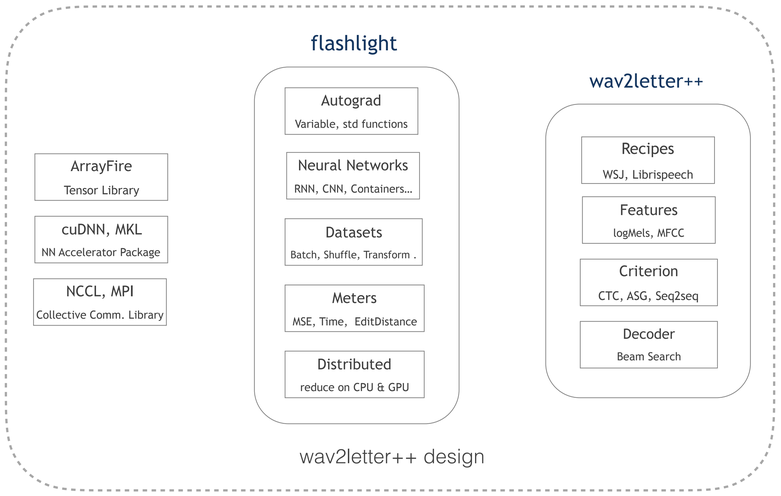
Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, Ronan Collobert

INTRODUCTION

- wav2letter++ is a fast, open-source deep learning speech recognition framework.
  - Written entirely in **C++** and backed by the efficient **ArrayFire** tensor library
  - **Scales linearly to 64 GPUs** for models with 100+ million parameters.
  - **Over 2x faster** in some cases than other optimized frameworks for training end-to-end neural networks for speech recognition.

DESIGN

- The design of wav2letter++ is motivated by three requirements:
  - It must efficiently train models on datasets containing **thousands of hours of speech**,
  - make expressing and incorporating **new network architectures, loss functions, and other operations** easy, and
  - **make the path from model research to deployment straightforward**, requiring as little new code as possible while maintaining the flexibility needed for research.



wav2letter++ design

ArrayFire Tensor Library

- ArrayFire [1] is a highly-optimized tensor library that supports CPU, GPU, and OpenCL backends.
- Uses just-in-time (JIT) code generation to combine series of simple operations into a single kernel call.
- Less verbose and relies on fewer C++ idiosyncrasies.

Flashlight Machine Learning Library

- A standalone machine learning library that:
  - extends ArrayFire with autograd, NN modules, distributed training, etc. to support neural network training.
  - extends the core ArrayFire CUDA back-end with more efficient cuDNN operations including convolutions and RNN operations.
- wav2letter++ library is built on top of flashlight.

```
Variable forward(const Variable& x) {
    auto hidden = matmul(weights[0], x);
    hidden = max(hidden, 0); // ReLU
    return matmul(weights[1], hidden);
}
Variable criterion(const Variable& yhat, const Variable& y) {
    auto probs = sigmoid(yhat);
    return -(y * log(probs) + (1 - y) * log(1 - probs));
}
for (const auto& xy : trainSet) {
    criterion(forward(xy[0]), xy[1]).backward();
    for (auto& w : weights) {
       w -= lr * w.grad();
       w.zeroGrad(); // Set gradient to zero
    }
}
```

Example: one layer MLP trained with binary cross-entropy and SGD, using autograd

Data Preparation and Feature Extraction

- wav2letter++ supports multiple audio file formats (e.g. wav, flac... / mono, stereo / int, float) and several feature types including raw audio, a linearly scaled power spectrum, log-Mels (MFSC) and MFCCs.
- Data loading computes features on the fly prior to each network evaluation.
- To make this efficient while training models, we load the audio and compute the features asynchronously and in parallel with inference.

Models

- We support several end-to-end sequence models with loss functions including Connectionist Temporal Classification (CTC) [6], wav2letter's AutoSegmentationCriterion (ASG) criterion [2], and sequence-to-sequence models with attention (seq2seq).
- Adding a new sequence criteria is particularly easy; ASG and CTC are already efficiently implemented in C++.
- Since the flashlight library we use provides dynamic graph construction and automatic differentiation, building new layers or other primitive operations requires little effort.

Training and Scale

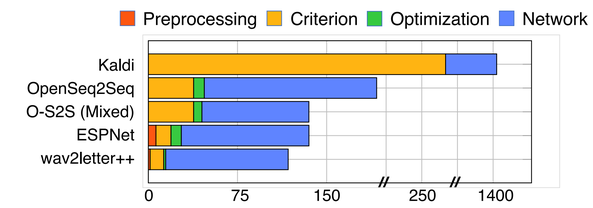
- Flexibility for the user to experiment with different features, architectures. and optimization parameters. Hackable to the core.
- Training can be run in three modes:
  - `train` (flat-start training)
  - `continue` (continuing with a checkpoint state)
  - `fork` (for e.g. transfer learning)
- We scale wav2letter++ to larger datasets with data-parallel, synchronous and asynchronous SGD and provide a simple framework with which to create custom distributed optimization schemes.

Decoding

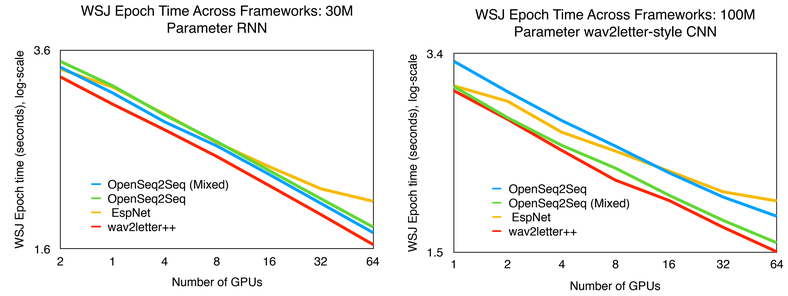
- The wav2letter++ decoder is performance-optimized beam-search decoder which:
  - supports any type of language model which exposes the interface required by our decoder including n-gram LMs and any other stateless parametric LM.
  - supports online decoding, where emissions are streamed into the decoder.

BENCHMARKS

Training Performance by Component



Distributed Training Scales Linearly



Distributed training performance on the WSJ Dataset.

Decoding Speed and Throughput

Name	WER (%)	Time/sample (ms)	Memory (GB)
ESPNet	7.20	1548	-
OpenSeq2Seq	5.00	1700	7.8
OpenSeq2Seq	4.92	9500	26.6
wav2letter++	5.00	10	3.9
wav2letter++	4.91	140	5.5

Decoding performance on Librispeech *dev-clean*.

REFERENCES

1. Pavan Yalamanchili, Umar Arshad, Zakiuddin Mohammed, Pradeep Garigipati, Peter Entschev, Brian Kloppenborg, James Malcolm, and John Melonakos , “ArrayFire - A high performance software library for parallel computing with an easy-to-use API,” 2015
2. Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, “Wav2letter: an end-to-end convnet-based speech recognition system,” CoRR, vol. abs/1609.03193, 2016
3. Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” in IEEE 2011 workshop on automatic speech recognition and understanding IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.
4. Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “Espnet: End-to-end speech processing toolkit,” arXiv preprint arXiv: 1804.00015, 2018.
5. Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Carl Case, and Paulius Micikevicius, “Openseq2seq: extensible toolkit for distributed and mixed precision training of sequence-to-sequence models,” arXiv preprint arXiv: 1805.10387, 2018
6. Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd inter-national conference on Machine learning. ACM, 2006, pp. 369–376.

wav2letter++ - https://github.com/facebookresearch/wav2letter

flashlight - https://github.com/facebookresearch/flashlight