OPENSEQ2SEQ: A DEEP LEARNING TOOLKIT FOR SPEECH RECOGNITION, SPEECH SYNTHESIS, AND NLP

Boris Ginsburg, Vitaly Lavrukhin, Igor Gitman, Oleksii Kuchaiev, Jason Li, Vahid Noroozi, Ravi Teja Gadde, Chip Nguyen

18/10/2018
SUMMARY

OpenSeq2Seq is a TensorFlow toolkit based on sequence-to-sequence paradigm

- Modular design
- Mixed precision training optimized for Volta GPU
- Scalable multi-GPU training with Horovod

State-of-the art models

- Automatic Speech Recognition: DeepSpeech2, Wave2Letter
- Language Translation: GNMT, Transformer, ConvS2S
- Text-To-Speech: Tacotron2
WHY WE BUILT OPENSEQ2SEQ

The project started in January 2018, when we found that

- Adding float16 (“mixed precision”) support to TF models is difficult:
  - TF approach based on “custom getters” practically didn’t work for RNN, consumed large amount of memory, and was error-prone

- Multi-GPU training scales badly:
  - “native” TF tower-based scaling is inferior to Horovod (doesn’t use NCCL, hard to implement multi-node training)

- There were no state-of-the-art ASR models (DeepSpeech2, Wave2Letter) in TensorFlow

Need new high-level modular toolkit to solve these problems
OpenSeq2Seq: Modular Design

**Model**
- **Implements**: parsing parameters, multi-GPU details, creating data_layer, optimizer
- **Abstract**: graph creation, metrics logging, eval, infer

**DataLayer**
- **Implements**: parsing/setting parameters
- **Abstract**: tf.data object creation

**Encoder**
- **Implements**: parsing/setting parameters
  - accepts Encoder output
- **Abstract**: graph creation

**Decoder**
- **Implements**: parsing/setting parameters
  - accepts Decoder output
- **Abstract**: graph creation

**Loss**
- **Implements**: parsing/setting parameters
  - accepts Decoder output
- **Abstract**: graph creation

**Text2Text**
- **Implements**: metrics logging, eval, infer

**Speech2Text**
- **Implements**: metrics logging, eval, infer

**DeepSpeech2Encoder**
- **Implements**: graph creation

**RnnDecoderWithAttention**
- **Implements**: graph creation

**CTCLoss**
- **Implements**: graph creation

```python
def train(train_model, eval_model=None)
def eval(eval_model)
def infer(infer_model)
```
OpenSeq2Seq: Mixed Precision Training - 1

- Loss scaling with automatic back-off
- LARC (Layer-wise Automatic Rate Control) to speed-up and stabilize initial stage of training
def l2_regularizer(scale):
    def l2(weights):
        return scale * nn.l2_loss(weights)
    return l2
Regularization wrapper

```python
def mp_regularizer_wrapper(regularizer):
    def func_wrapper(weights):
        if weights.dtype.base_dtype == tf.float16:
            tf.add_to_collection('REGULARIZATION_FUNCTIONS',
                (weights, regularizer))
                # disabling the inner regularizer
            return None
        else:
            return regularizer(weights)
    return func_wrapper

reg_funcs = tf.get_collection('REGULARIZATION_FUNCTIONS')

reg = tf.contrib.layers.apply_regularization(
    reg_funcs[var.name],[fp32_var]),

fp32_grad += tf.gradients(reg, fp32_var)[0]
```
OpenSeq2Seq: Multi-GPU training and inference

- Tower-based approach
  - Building a separate graph for each GPU
  - Data feeding with separate data layer per each GPU
  - No NCCL

- Horovod-based approach
  - Multi-GPU training is straightforward: optimizer wrapper and broadcast hook
  - It relies on NCCL for fast inter-GPUs communications
OpenSeq2Seq: Available models

Available models (with checkpoints and training recipes):

machine translation: GNMT, Transformer, ConvS2S

speech recognition: DeepSpeech2, Wave2Letter

text-to-speech: Tacotron-2

image classification: ResNets, AlexNet

Many different encoders / decoders / losses to combine and play with!
Machine Translation: ConvS2S

- BLEU= 25 on News2014 EN-DE (better than GNMT)
- Convolutional encoder
- Convolutional decoder
  - Don’t need inputs from previous steps ➔ parallel training
  - Easy control on maximum dependency length
- Gated Linear Units (GLU): \( h(X) = (XW + b) \otimes \sigma(XV + c) \)
- Separate attention for each decoder layer
- Position embedding
- Weight Normalization

Jonas Gehring et al. “Convolutional Sequence to Sequence Learning”
ConvS2S: Mixed Precision Speed-up, 1 GPU

<table>
<thead>
<tr>
<th>Batch/GPU</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>2.4x</td>
</tr>
<tr>
<td>256</td>
<td>2.6x</td>
</tr>
</tbody>
</table>

GPU: Single Quadro GV100 with 32GB Memory

Driver version: 384.146, CUDA version: 9.0, cuDNN version: 7.0

TensorFlow: 18.06, Horovod: 0.13.10
ConvS2S: Multi-GPU scalability

<table>
<thead>
<tr>
<th>Number of GPUs</th>
<th>w/o Horovod</th>
<th>w/ Horovod</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2.9</td>
<td>3.7</td>
</tr>
<tr>
<td>8</td>
<td>6.3</td>
<td>7.3</td>
</tr>
<tr>
<td>16</td>
<td>9.0</td>
<td>14.0</td>
</tr>
</tbody>
</table>
Machine Translation: Transformer

- BLEU = 28.0 on News2014 EN-DE (better than ConvS2S)
- No recurrent layers
- No convolutional layers
- Attention based
- SentencePiece tokenization (byte-pair-encoding)
- Layer Normalization
- Mixed precision training is 1.9x faster (than float32)

Transformer: Multi-GPU scalability

![Graph showing Multi-GPU scalability with and without Horovod](chart.png)
Speech Recognition: Wave2Letter

Wave2Letter V2:

- Gated Linear Units (GLU)
- Weight Normalization
- Gradient clipping
- ASG loss

Wave2Letter+: Architecture modifications

- BatchNorm instead of Weight Norm
- Clipped ReLU instead of GLU
- Standard CTC loss
- LARC instead of Gradient clipping

- Dilated Convolution:
  - Increase a context for the predictions

- Strided Convolution:
  - Decreases the length in time by 2x
  - Significant reduction in training time

<table>
<thead>
<tr>
<th>Model</th>
<th>WER, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base (B)</td>
<td>10.53</td>
</tr>
<tr>
<td>B + Dilated Conv</td>
<td>10.18</td>
</tr>
<tr>
<td>B + Strided Conv</td>
<td>8.59</td>
</tr>
</tbody>
</table>

Trained for 50 epochs
Speech Recognition: Wave2Letter+

Stride: 2

Dilation: 2

Conv Kw = 11 Filters: 256
Conv Kw = 13 Filters: 384
Conv Kw = 17 Filters: 512
Conv Kw = 21 Filters: 640
Conv Kw = 25 Filters: 768
Conv Kw = 29 Filters: 896
Conv Kw = 1 Filters: 1024
Conv Kw = 1 Filters: 29

CTC
Wave2Letter+: Accuracy

<table>
<thead>
<tr>
<th>LibriSpeech Dataset</th>
<th>WER, % Greedy Decoding</th>
<th>WER, % Beam Search with LM*</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev-clean</td>
<td>6.67</td>
<td>4.77</td>
</tr>
<tr>
<td>test-clean</td>
<td>6.58</td>
<td>4.92</td>
</tr>
<tr>
<td>dev-other</td>
<td>18.68</td>
<td>13.88</td>
</tr>
<tr>
<td>test-other</td>
<td>19.61</td>
<td>15.01</td>
</tr>
</tbody>
</table>

Training Configuration:
- 8 GPUs (NGC), 64 batch size
- Mixed Precision
- 200 Epochs, 44 Hours

* Language model is a standard 4-gram OpenSLR LM
### Wave2Letter+: Mixed Precision Speed-Up, 1 GPU

<table>
<thead>
<tr>
<th>Batch/GPU</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>160</td>
<td>3.22x</td>
</tr>
<tr>
<td>192</td>
<td>3.18x</td>
</tr>
<tr>
<td>352</td>
<td>3.58x</td>
</tr>
</tbody>
</table>

**Specifications:**
- GPU: Single Quadro GV100-32GB
- Driver Version: 384.146
- CUDA Version: 9.0
- cuDNN Version: 7.0
- TensorFlow Version: 1.9
CONCLUSIONS

OpenSeq2Seq is TensorFlow-based toolkit that

• supports distributed mixed precision training out of the box
• contains implementations of many SoTA models

More models and features to come!

Contributions are welcome 😊

GitHub: https://github.com/NVIDIA/OpenSeq2Seq
 Docs: https://nvidia.github.io/OpenSeq2Seq