Sequence-to-Sequence Model

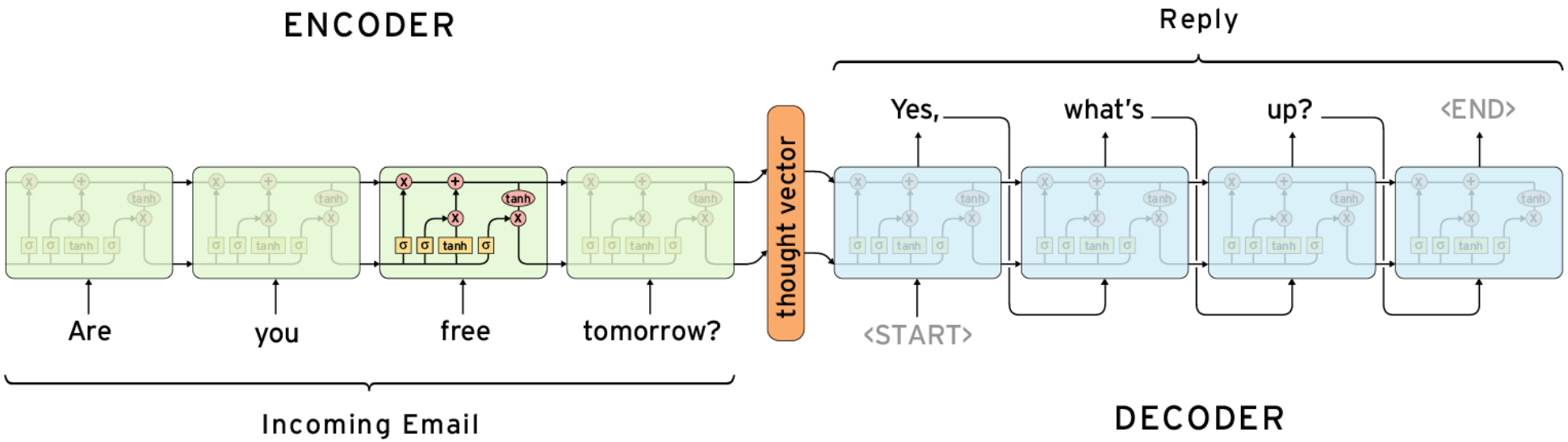
Hanock Kwak

2016-11-10 (Thu)

BI Lab

Sequence-to-Sequence Model

- A basic sequence-to-sequence model consists of two recurrent neural networks (RNNs): an **encoder** that processes the input and a **decoder** that generates the output.



Setup

- First, download following repository
  - **git clone https://github.com/nicolas-ivanov/tf_seq2seq_chatbot.git**

```
mlc@mlc-Z97X-UD5H:/opt2/hnkwak/ml$ git clone https://github.com/nicolas-ivanov/tf_seq2seq_chatbot.git
Cloning into 'tf_seq2seq_chatbot'...
remote: Counting objects: 245, done.
remote: Total 245 (delta 0), reused 0 (delta 0), pack-reused 245
Receiving objects: 100% (245/245), 18.70 MiB | 3.94 MiB/s, done.
Resolving deltas: 100% (124/124), done.
Checking connectivity... done.
```

- Then, go into the downloaded directory
  - **cd tf_seq2seq_chatbot**

```
mlc@mlc-Z97X-UD5H:/opt2/hnkwak/ml$ cd tf_seq2seq_chatbot
mlc@mlc-Z97X-UD5H:/opt2/hnkwak/ml/tf_seq2seq_chatbot$
```

Setup

- Run setup.sh
  - **bash setup.sh**
  - If permission error arises, then run as sudo **“sudo bash setup.sh”**
- Train a seq2seq model on a small (17 MB) corpus of movie subtitles
  - **python train.py**
  - **sudo python train.py** (if permission error occurs)

Train Process

```
mlc@mlc-Z97X-UD5H:/opt2/hnkwak/ml/tf_seq2seq_chatbot$ sudo python train.py
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcublas.so locally
I tensorflow/stream_executor/dso_loader.cc:102] Couldn't open CUDA library libcudnn.so. LD_LIBRARY_PATH:
I tensorflow/stream_executor/cuda/cuda_dnn.cc:2259] Unable to load cuDNN DSO
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcufft.so locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcurand.so locally
Preparing dialog data in /var/lib/tf_seq2seq_chatbot/data
Creating vocabulary /var/lib/tf_seq2seq_chatbot/data/vocab20000.in from data /var/lib/tf_seq2seq_chatbot/data/chat.in
  processing line 100000
  processing line 200000
  processing line 300000
  processing line 400000
Tokenizing data in /var/lib/tf_seq2seq_chatbot/data/chat.in
  tokenizing line 100000
  tokenizing line 200000
  tokenizing line 300000
  tokenizing line 400000
Tokenizing data in /var/lib/tf_seq2seq_chatbot/data/chat_test.in
I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:925] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
I tensorflow/core/common_runtime/gpu/gpu_init.cc:118] Found device 0 with properties:
name: GeForce GTX TITAN X
major: 5 minor: 2 memoryClockRate (GHz) 1.076
pciBusID 0000:01:00.0
Total memory: 12.00GiB
Free memory: 11.87GiB
I tensorflow/core/common_runtime/gpu/gpu_init.cc:138] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_init.cc:148] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:868] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX TITAN X, pci bus id: 0000:01:00.0)
Creating 1 layers of 128 units.
Created model with fresh parameters.
Reading development and training data (limit: 0).
  reading data line 100000
  reading data line 200000
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 6248 get requests, put_count=2777 evicted_count=1000 eviction_rate=0.360101 and unsatisfied allocation rate=0.731594
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:256] Raising pool_size_limit_ from 100 to 110
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=4013 evicted_count=4000 eviction_rate=0.996761 and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=2019 evicted_count=2000 eviction_rate=0.990589 and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 6248 get requests, put_count=6013 evicted_count=4000 eviction_rate=0.665225 and unsatisfied allocation rate=0.682298
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:256] Raising pool_size_limit_ from 309 to 339
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 1843 get requests, put_count=4952 evicted_count=4000 eviction_rate=0.807754 and unsatisfied allocation rate=0.503527
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:256] Raising pool_size_limit_ from 409 to 449
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=1065 evicted_count=1000 eviction_rate=0.938967 and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=1105 evicted_count=1000 eviction_rate=0.904977 and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=1169 evicted_count=1000 eviction_rate=0.855432 and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:244] PoolAllocator: After 0 get requests, put_count=1272 evicted_count=1000 eviction_rate=0.786164 and unsatisfied allocation rate=0
global step 100 learning rate 0.5000 step-time 0.13 perplexity 1020.24
  eval: bucket 0 perplexity 2242.59
  eval: bucket 1 perplexity 1552.19
```

Neural Chatbot

- Finally, chat with neural chatbot!
  - **python chat.py**
  - **sudo python chat.py** (if permission error occurs)

```
mlc@mlc-Z97X-UD5H:/opt2/hnkwak/ml/tf_seq2seq_chatbot$ sudo python chat.py
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcublas.so locally
I tensorflow/stream_executor/dso_loader.cc:102] Couldn't open CUDA library libcudnn.so. LD_LIBRARY_PATH:
I tensorflow/stream_executor/cuda/cuda_dnn.cc:2259] Unable to load cuDNN DSO
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcufft.so locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcurand.so locally
I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:925] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
I tensorflow/core/common_runtime/gpu/gpu_init.cc:118] Found device 0 with properties:
name: GeForce GTX TITAN X
major: 5 minor: 2 memoryClockRate (GHz) 1.076
pciBusID 0000:01:00.0
Total memory: 12.00GiB
Free memory: 11.87GiB
I tensorflow/core/common_runtime/gpu/gpu_init.cc:138] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_init.cc:148] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:868] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX TITAN X, pci bus id: 0000:01:00.0)
Reading model parameters from /var/lib/tf_seq2seq_chatbot/nn_models/model.ckpt-1800
> hello my friend
what ?
>
```

Key Functions

- The typical measure is average per-word **perplexity**

$$e^{-\frac{1}{N}\sum_{i=1}^{N}\ln p_{\text{target}_i}} = e^{\text{loss}}$$

- The session can be saved and loaded.

```
# Save checkpoint and zero timer and loss.
checkpoint_path = os.path.join(FLAGS.model_dir, "model.ckpt")
model.saver.save(sess, checkpoint_path, global_step=model.global_step)
```

- The word IDs will be embedded into a dense representation.

Key Functions

- Sampled softmax loss

```
def sampled_loss(inputs, labels):
  labels = tf.reshape(labels, [-1, 1])
  return tf.nn.sampled_softmax_loss(w_t, b, inputs, labels, num_samples,
          self.target_vocab_size)
```

- Various cells (LSTM, GRU) and stacking multi-layer

```
# Create the internal multi-layer cell for our RNN.
single_cell = tf.nn.rnn_cell.GRUCell(size)
if use_lstm:
  single_cell = tf.nn.rnn_cell.BasicLSTMCell(size)
cell = single_cell
if num_layers > 1:
  cell = tf.nn.rnn_cell.MultiRNNCell([single_cell] * num_layers)
```

Key Functions

- Seq2seq library

```
# The seq2seq function: we use embedding for the input and attention.
def seq2seq_f(encoder_inputs, decoder_inputs, do_decode):
  return tf.nn.seq2seq.embedding_attention_seq2seq(
      encoder_inputs, decoder_inputs, cell,
      num_encoder_symbols=source_vocab_size,
      num_decoder_symbols=target_vocab_size,
      embedding_size=size,
      output_projection=output_projection,
      feed_previous=do_decode)
```

Key Functions

- Buckets is a method to efficiently handle sentences of different lengths.

```
buckets = [(5, 10), (10, 15), (20, 25), (40, 50)]
```

  - If the input is an English sentence with 3 tokens, and the corresponding output is a French sentence with 6 tokens, then they will be put in the first bucket and padded to length 5 for encoder inputs, and length 10 for decoder inputs.

we	are	the	one			
hello	my	friend				
the	cat	is	jumping	into	trash	can