Lab10 Sequence to Sequence

COMP4901K and MATH 4824B

Fall 2018

1 GRU

What is the GRU (Gated Recurrent Unit)?

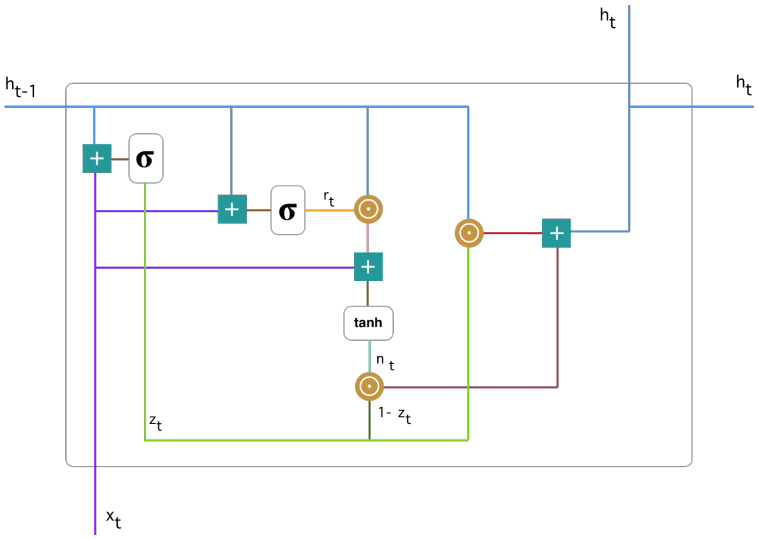

Figure 1: Illustration of the GRU. Image Source: `https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be`

The GRU (Gated Recurrent Unit) is a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho et al. The details are as follows:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{1}$$
$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \tag{2}$$
$$n_t = \tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{(t-1)} + b_{hn})) \tag{3}$$
$$h_t = (1 - z_t)n_t + z_t h_{(t-1)} \tag{4}$$

where h_t is the hidden state at time t, x_t is the input at time t, $h_{(t-1)}$ is the hidden state of the previous layer at time $t-1$ or the initial hidden state at tiem 0, and r_t, z_t, n_t are the reset, update, and new gates, respectively σ is the sigmoid function.

Why GRU?

- The GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit. It just exposes the full hidden content without any control.
- GRU is relatively new, and from my perspective, the performance is on par with LSTM, but computationally more efficient (less complex structure as pointed out).

2 Sequence to Sequence

Sequence to Sequence (Seq2Seq) is about training models to convert sequences from one domain (e.g. sentences in French) to sequences in another domain (e.g. the same sentences translated to English), introduced in 2014 by Sutskever et al.

Typically, a Seq2Seq model consists of 2 parts, the encoder part and the decoder part.
The encoder encodes the information of one sequence to one vector or multiple vectors, and then the decoder decodes the information provided by the encoder into the target sequence.

Simple Seq2Seq

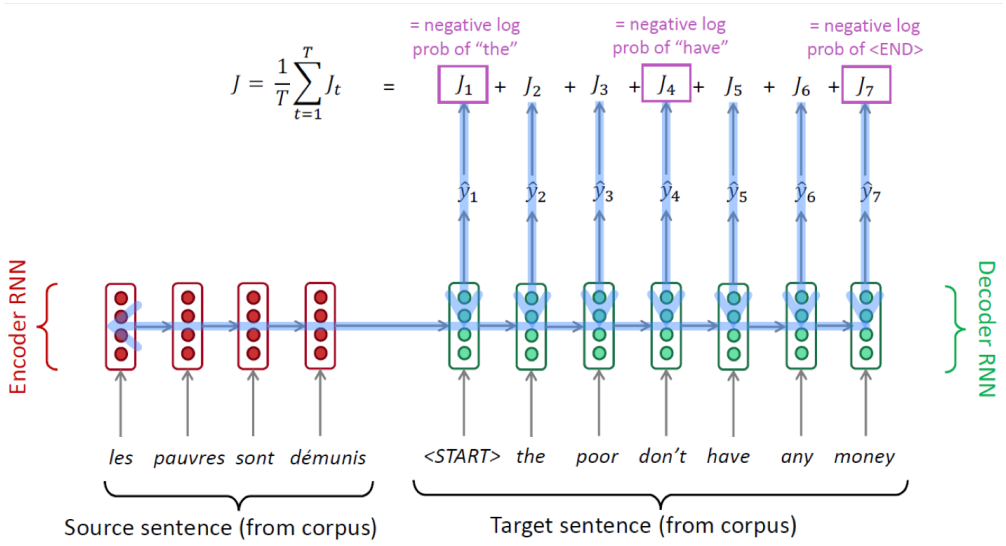

Figure 2: Illustration of the Seq2Seq.

The simple version Seq2Seq2 uses one encoder RNN and one decoder RNN. And the information from the encoder is only the last hidden state of the encoder RNN. In this lab, we use GRU to encode and decode sequence information.

The network includes

- 1 layer of Embedding, which is shared by the encoder and the decoder for convenience.
- the encoder part (assuming N layer(s) of Bidirectional GRU)
  - $N-1$ layer(s) of Bidirectional GRU, which return(s) sequences only.
  - 1 layer of Bidirectional GRU, which returns sequences and the last state.
  - Dropout layers between GRU layers if applicable.
- the decoder part (assuming M layer(s) of Bidirectional GRU)
  - 1 layer of Bidirectional GRU, whose input is the ground truth and initial hidden state is the encoder's last state.

- $M-1$ layer(s) of Bidirectional GRU.
- Dropout layers between GRU layers if applicable.
- 1 layer of Dense layer with softmax activation wrapped by TimeDistributed.

The loss function is the **sparse_categorical_crossentropy**.
The Adam optimizer is chosen.

Seq2Seq with Attention

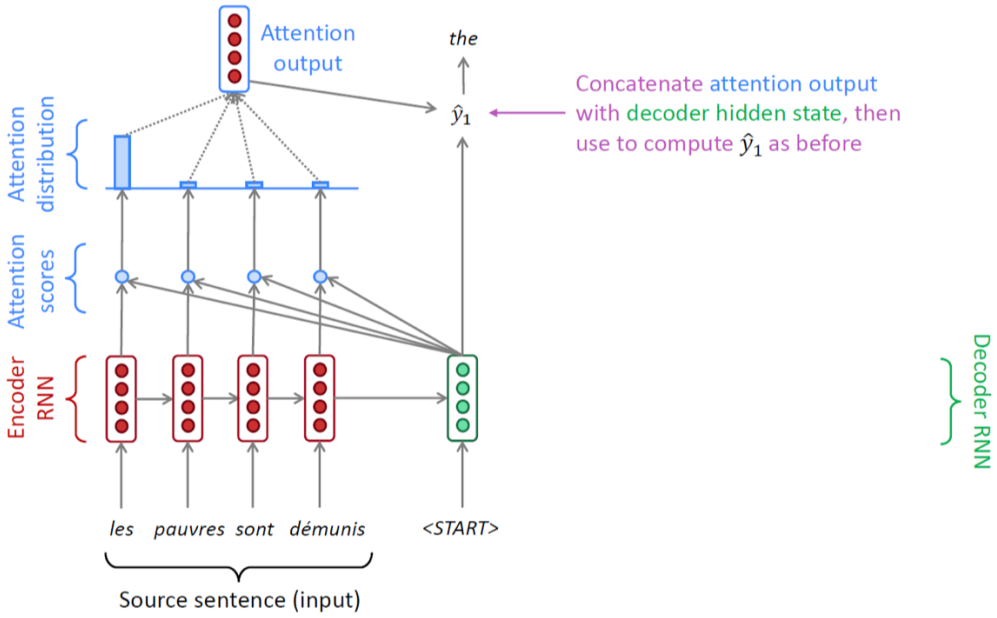

Figure 3: Illustration of the Seq2Seq with Attention.

Encoding of the source sentence needs to capture all information about the source sentence. But just using the last hidden state causes information bottleneck!
Core idea: in each step of the decoder, focus on a particular part of the source sequence.

Compared with the Simple version, the encoder remains unchanged, but the decoder needs more information of the source sequence.
There are many ways to fuse information. One way is to concatenate attention with the current decoder hidden state before sending to the decoder RNN. And a faster and easier way is to concatenate attention with the output of decoder RNN before making predictions. To make this lab easier, we choose the latter method.
The decoder with attention includes

- 1 layer of Bidirectional GRU, whose input is the ground truth and initial hidden state is the encoder's last state.

- $M-1$ layer(s) of Bidirectional GRU.
- Dropout layers between GRU layers if applicable.
- **Necessary layers (e.g. Dense layers) and computation operations (e.g. dot-product).**
- **The concatenate operation and a Dense layer to fuse the attention with the output of decoder RNN.**
- 1 layer of Dense layer with softmax activation wrapped by TimeDistributed.

3 Applications

Neural Machine Translation

In this lab, we will use the Seq2Seq model to train a tranlstion system from French to English.
Just set the parameter of **load_data** as *translation*. This function will load sentence pairs of the Tatoeba project.

Dialogue System

We also use the Seq2Seq model to train a dialogue system.
Just set the parameter of **load_data** as *dialogue*. This function will load sentence pairs of the Cornell Movie–Dialogs Corpus.

4 Submission

You need to submit two files, program outputs (**20** epoch) on the **two** datasets and your python script.

You must use the attention mechanism in your Seq2Seq models. But you can choose the dot-product attention, the multiplicative attention or the additive attention as long as your results are better than those of simple Seq2Seq models.

After you finished the assignments, make sure you include the header information in the beginning of your code # *author: Your_name* # *student_id: Your_student_id*