Assignment 5

Design

Sentence Encoder

A bi-directional RNN is used for implementing the encoder. The outputs from either directions are concatenated and fed to the decoder layer.

Decoder

- **Input:** The embedding of the previous word (predicted or actual) and the last context are passed as input to the GRU.
- **Hidden input:** In the first step the output from the encoder is used as the hidden input. For subsequent timesteps, the hidden output from the previous timestep is used.
- **Attention weights:** The attention weights are derived using the general scoring mechanism using the formula:

  t(h_t).W.h_s

  where h_t represents the hidden tensor and h_s represents the encoder outputs.
- **Context vector:** A weighted sum of the attention weights and the encoder outputs is computed to generate the context vector.
- **Decoder output:** A linear expression is formed using the concatenation of the output of GRU and context vector, and is then fed through a softmax layer.

Training

The source words are fed into the encoder and its final hidden stage is fed into the decoder. Scheduled sampling is used in the decoder stage of training. A teacher ratio of 0.5 has been used. This ensures that in around 50% of the cases, actual target words are fed in to the RNN, and the previous word's predictions are fed into the RNN the remaining times.

Prediction

The source words are fed into the encoder and its final hidden stage is fed into the decoder. In the decoder stage, the predicted words from each timestep is fed into the GRU's in the subsequent timesteps. The analysis stops either when all the sentences' translations are over (which happens when one of the most likely words is EOS).

Output of NMT

seq_batch_wd_prob: A tensor representing each target vocab word's probability for every sequence/batch combination.
translated_sentence_wd_index: A tensor indicating sequence of word indexes in each sentence of the batch.

Metrics

WER: 0.96