A Hierarchical Neural Autoencoder for Paragraphs and Documents

1 Idea

This work attempts to use a neural autoencoder to build hierarchical paragraph representations using sentence embeddings and decode the latent representation back into the original paragraph. This is an LSTM based model and different levels of LSTM are used to encode compositionality of token-to-token and sentence-to-sentence relations.

2 Method

- 3 different autoencoder models are experimented with. The first is a simple version that treats all the document tokens as a single sequences, just like a Seq2Seq model [1]. The second is a hierarchical autoencoder and the third a hierarchical autoencoder with an attention mechanism.
- The construction of a hierarchical autoencoder is simple. We assume that we have word embeddings per token. The final hidden state obtained after operating an LSTM over a sequence of tokens in a sentence is assumed to be the sentence embedding. Similarly, the hidden state obtained after operating an LSTM over a sequence of sentence embeddings is the paragraph embedding. The decoding is done in a similar manner. First the sentences embeddings are obtained by unrolling the paragraph vectors, and then word vectors are obtained by unrolling sentence embeddings.
- End of document e_D and end of sentence e_S tokens are treated as word embeddings, to signify the end of a sequence.

- Sequences are predicted using a softmax function, over the space of the vocabulary.
- Attention is computed by allowing, at each decoder step, to peek at every hidden state generated during the encoding phase. Each input sentence is characterized by a strength indicator v_i, which is a weighted combination of the hidden state at the last decoder step, and the encoder hidden state of sentence i. v_i is normalized to create the attention weight a_i of sentence i. The attention vector is the weighted sum of a_i and the encoder hidden state of each source sentence i.
- This attention vector m_t is added as another parameter to the decoder LSTM, in addition to current input e_t^s and previous hidden state of the decoder $h_{t-1}^s(dec)$.
- Training done using stochastic gradient descent with mini-batches.
- Model tested on a hotel reviews corpus and Wikipedia, using the ROUGE and BLEU metrics.
- 1000 dimensional word embeddings used with LSTMs of size 1000. 4 layers of encoding and decoding LSTMs are used.
- Input documents are reversed, similar to the original Seq2Seq paper.
- At most 1.5 times the number of input words are allowed to be generated by the decoder. Unclear how this is controlled.
- ROUGE and BLEU metrics don't evaluate coherence, so a custom grid evaluation metric is used. It measures the degree of preservation of the text order.

3 Observations

- Unclear whether a softmax to predict an end-of-sentence token or a binary classifier to predict end-of-sentence is actually used in the model.
- As expected, Hierarchical + Attention > Hierarchical > Vanilla Seq2Seq, in terms of evaluation metrics.

- Also as expected, performance is better on the hotel reviews corpus, because the format is more consistent.

References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.