Convolutional Encoder Model for NTM
Gehring et al. , 2017
Facebook AI Research

Presenter : Maghav Kumar

Outline

- Introduction
- Why CNNs over RNNs
- Previous Work
  - RNNs
- Non-Recurrent Encoders
- Convolutional Encoder
- Related Work
- Datasets
- Architecture
- Results

Introduction

- End-to-End approach to Machine Translation (Sutskever et al., 2014).
- Most successful approach to date has been bi-directional RNN.
- RNNs usually parameterized as LSTMs(Hochreiter et al. 1997) or GRUs (Cho et al., 2014).
- Several attempts made in past but not competitive to recurrent alternatives (Cho et al., 2014a).

Attractive Properties of CNN’s over RNNs for NMT

- CNNs operate over a fixed-size of input sequence, enabling simultaneous computation of all features for a source sentence.
- RNNs maintain a hidden state of the entire past that prevents parallel computation within a sequence.
- Succession of convolutional layers provides a shorter path to capture relationships between elements of a sequence compared to RNNs.

Attractive Properties of CNN’s over RNNs for NMT

- A CNN would also <u>ease</u> learning as the resulting tree-structure applies a fixed number of non-linearities compared to an RNN for which the number of non-linearities vary depending on the time-step.
- Since processing is bottom-up, all words undergo the same number of transformations, whereas for RNNs the first word is over-processed and the last word is transformed only once.

Recurrent Neural Nets for NMT

- General Architecture follows Encoder-Decoder approach with soft attention (Bahdanau et al., 2015).
- Consider you have a source sentence X of m words

$$X = (x_1 , x_2 ,....., x_m)$$

- An encoder will output a sequence of states Z where

$$Z = (z_1 , z_2 ,....., z_m)$$

- A decoder is present which is an RNN that computes a new hidden state s_{i+1} based on the previous state s_i , an embedding g_i of the previous target language word y_i, as well as a conditional input c_i derived from Z.

Recurrent Neural Nets for NMT

$$d_i = W_d h_i + b_d + g_i,$$

$$a_{ij} = \frac{\exp\left(d_i^T z_j\right)}{\sum_{t=1}^{m} \exp\left(d_i^T z_t\right)}, \quad c_i = \sum_{j=1}^{m} a_{ij} z_j$$

Recurrent Neural Nets for NMT

- Usually LSTMs are used for all decoder networks.
- For which each state s_i comprises of a cell vector and a hidden vector h_i which is an output at each time step.
- The conditional input c_i is concatenated with g_i and is an input to the LSTM.
- Then the model computes a distribution over V possible target words y_{i+1} using (where W_0 is the weight and b_0 is bias.

$$p(\, y_{i+1} \mid y_1, \ldots\ldots, y_i, X) = \text{softmax}(W_0 h_{i+1} + b_0)$$

Non-Recurrent Encoders

1) Pooling Encoders
2) Convolutional Encoders

Pooling Encoders

- Initial work simply averages the embeddings of k consecutive words (Ranzato et al., 2015).
- This does not convey positional information though.
- FIX : Add position embeddings to encode the absolute position of each word in a sentence.

$$e_j = w_j + l_j$$

e_j - Source Embedding ; w_j - Word Embedding ; l_j - positional embedding

Pooling Encoders

- The pooled representations z_j are computed using the embeddings.

$$z_j = \sum^{k/2}_{-k/2} e_{j+t}$$

- The conditional input is a weighted sum of the embeddings e_j using the attention values denoted by a_{ij}.

$$c_i = \sum^{m}_{j=1} a_{ij}\, e_j$$

Convolutional Encoders

- Novel Approach: Use a convolutional kernel.
- Encoder output z_j contains information about a fixed-size context depending on kernel width k.
- Stacking 5 convolutions with k=3 results in an input field of 11 words. Hence each output would depend on these 11 words and the non-linearities allow the encoder to exploit the full input field.

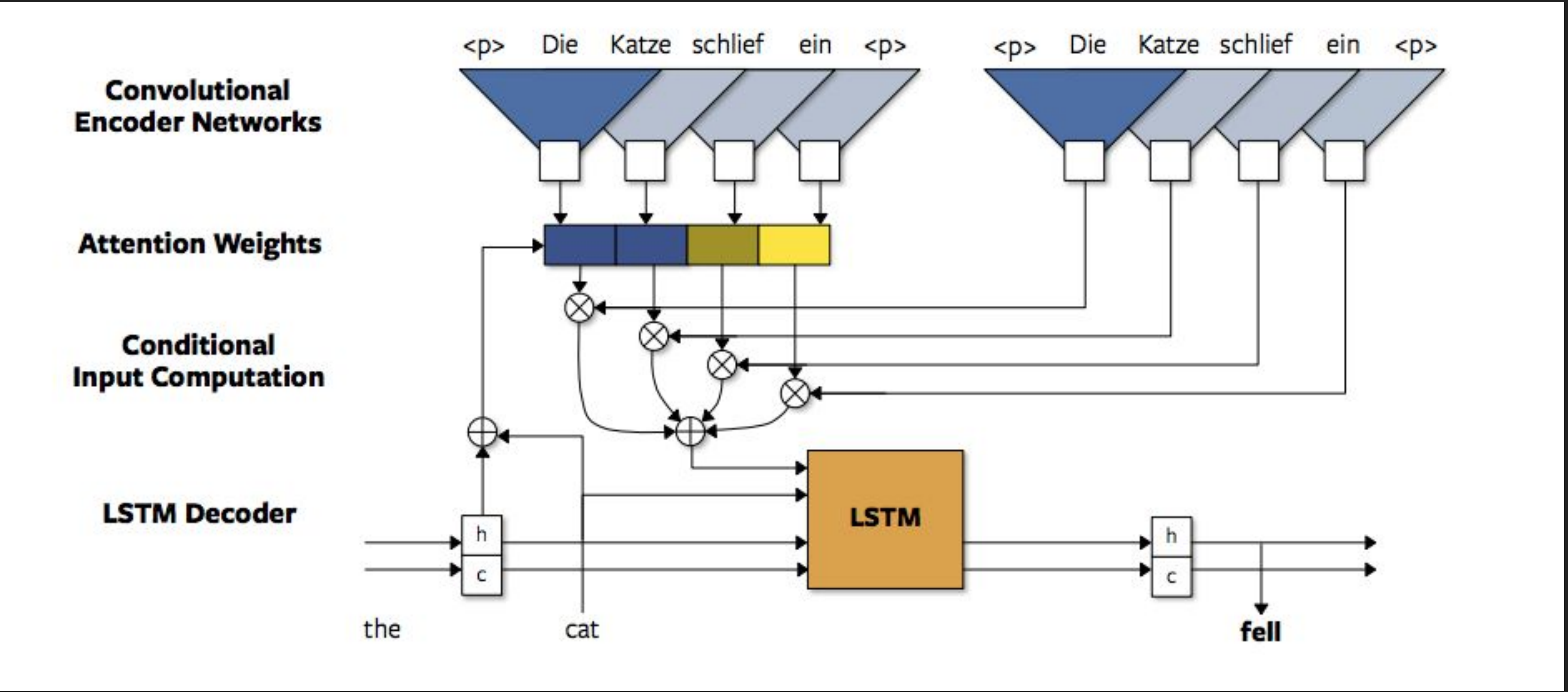
Convolutional Encoders

- The convolutional encoder also uses position embeddings
- Final encoder has 2 stacked convolutional networks:
    - First CNN produces encoder output z_j to compute attention scores a_i
    - Second CNN outputs are used to calculate the conditional input c_i

$$z_j = CNN_1(e)_j$$

$$c_i = \sum^{m}_{j=1} a_{ij}\, CNN_2(e)_j$$

Model Architecture



Related Work to Convolutional Approaches for NMT

1) Kalchbrenner et al., 2016- Convolutional translational models without an explicit attention mechanism but not state-of-the-art accuracy.
2) Lamb and Xie, 2016 - also proposed multi-layer CNN to generate a fixed-size encoder representation, but not enough quantitative evaluation in terms of BLEU.
3) Pham et al., 2016 - Convolutional architectures have been successful in language modeling but failed to outperform LSTMs.

Datasets

1) IWSLT’ 14 German-English
2) WMT’ 16 English-Romanian
3) WMT’ 15 English-German
4) WMT’ 14 English-French

Initial Results (on IWSLT' 14 German-English)

System/Encoder	BLEU wrd+pos	BLEU wrd	PPL wrd+pos
Phrase-based	–	28.4	–
LSTM	27.4	27.3	10.8
BiLSTM	29.7	29.8	9.9
Pooling	26.1	19.7	11.0
Convolutional	29.9	20.1	9.1
Deep Convolutional 6/3	30.4	25.2	8.9

Detailed Results

WMT'16 English-Romanian	Encoder	Vocabulary	BLEU
(Sennrich et al., 2016a)	BiGRU	BPE 90K	28.1
Single-layer decoder	BiLSTM	80K	27.5
	Convolutional	80K	27.1
	Deep Convolutional 8/4	80K	27.8

WMT'15 English-German	Encoder	Vocabulary	BLEU
(Jean et al., 2015) RNNsearch-LV	BiGRU	500K	22.4
(Chung et al., 2016) BPE-Char	BiGRU	Char 500	23.9
(Yang et al., 2016) RNNSearch + UNK replace	BiLSTM	50K	24.3
+ *recurrent attention*	BiLSTM	50K	25.0
Single-layer decoder	BiLSTM	80K	23.5
	Deep Convolutional 15/5	80K	23.6
Two-layer decoder	Two-layer BiLSTM	80K	24.1
	Deep Convolutional 15/5	80K	24.2

WMT'14 English-French (12M)	Encoder	Vocabulary	BLEU
(Bahdanau et al., 2015) RNNsearch	BiGRU	30K	28.5
(Luong et al., 2015b) Single LSTM	6-layer LSTM	40K	32.7
(Jean et al., 2014) RNNsearch-LV	BiGRU	500K	34.6
(Zhou et al., 2016) Deep-Att	Deep BiLSTM	30K	35.9
Single-layer decoder	BiLSTM	30K	34.3
	Deep Convolutional 8/4	30K	34.6
Two-layer decoder	2-layer BiLSTM	30K	35.3
	Deep Convolutional 20/5	30K	35.7

Training Time

Encoder	Words/s	BLEU
BiLSTM	139.7	22.4
Deep Conv. 6/3	187.9	23.1

(a) IWSLT'14 German-English generation speed on *tst2013* with beam size 10.

Training Time

Encoder	Words/s	BLEU
2-layer BiLSTM	109.9	23.6
Deep Conv. 8/4	231.1	23.7
Deep Conv. 15/5	203.3	24.0

(b) WMT'15 English-German generation speed on *newstest2015* with beam size 5.