How Much Attention Do You Need?

A Granular Analysis of Neural Machine Translation Architectures

Steven Walton
University of Oregon

21 Feb 2019

Overview

Questions:

- If attention is all you need, then how much?
- Where is the attention important?
- What type of attention do we need? Self? LSTM? Transformers?

Overview

Questions:

- If attention is all you need, then how much?
- Where is the attention important?
- What type of attention do we need? Self? LSTM? Transformers?

Answers:

- Source attention on lower encoder layers brings no additional benefits.
- Multiple source attention and residual feed-forward layers are key.
- Self-attention is more important for the source than for the target side.

- *Flexible Neural Machine Translation Architecture Combination*
    - *Neural Machine Translation (NMT)*
    - *Architecture Definition Language (ADL)*
    - *Layer Definitions*
    - *Standard Architectures*
- Related Work
- Experiments
- Conclusion

Neural Machine Translation (NMT)

- NMT is a sequence to sequence prediction task

$$X \mapsto Y$$

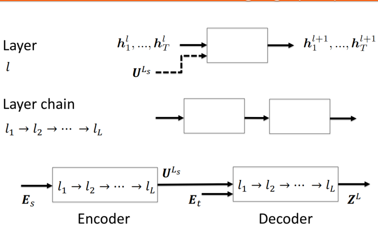
$$p(y_t | Y_{1:t-1}, X; \boldsymbol{\theta}) = \text{softmax}(\boldsymbol{W}_o \boldsymbol{z}^L + \boldsymbol{b}_o)$$

- $\boldsymbol{W}_o$ projects a model dependent hidden vector $\boldsymbol{z}^L$ of the L$^{\text{th}}$ decoder layer to the dimension of the target vocabulary $\boldsymbol{V}_{trg}$
- Training minimizes cross-entropy loss

Architecture Definition Language (ADL)

- ADL is a language used to describe network structures.
- This language lets us discuss a NMT in an easy to understand manner.
- E.G. *pos* $\rightarrow$ *repeat*(*n*, *res*(*cnn*(*glu*) $\rightarrow$ *dropout*))
  - Positional encoding $\rightarrow$ residual CNN with gated linear units $\rightarrow$ dropout

Architecture Definition Language (ADL)



Layer	Description
pos	Positional embeddings.
norm, *bnorm*	Layer and batch normalization.
repeat(n, l)	Layer repetition.
res(l)	Residual connection.
birnn, *rnn*	Recurrent neural network.
cnn	Convolutional neural network.
ff, *linear*, *ffl*	Feed-forward and linear layer.
[*mh*]_*dot_src_att*(h)	[Multi-head] dot source attention.
[*mh*]_*dot_self_att*(h)	[Multi-head] dot self-attention.
res_nd(l)	*res*(*norm*$\rightarrow l \rightarrow$*dropout*)
res_d(l)	*res*($l \rightarrow$*dropout*)
act, *dropout*	Other layers.
parallel, *highway*	

Architecture Definition Language (ADL)

Fixed Positional Embeddings

$$pos(\mathbf{h}_t) = dropout(\sqrt{d}\mathbf{h}_t + \mathbf{p}_t)$$

Gated Linear Unit (GLU)

$$glu([\mathbf{h}_A; \mathbf{h}_B]) = \mathbf{h}_A \otimes \sigma(\mathbf{h}_B)$$

Dot Product Attention

$$dot\_att(\mathbf{Q}, \mathbf{K}, \mathbf{V}, s) = softmax(\frac{\mathbf{QK}^T}{\sqrt{s}})\mathbf{V}$$

Layer Normalization

$$nrom(\mathbf{h}_t) = \frac{\mathbf{g}}{\sigma_t} \otimes (\mathbf{h}_t - \mu_t) + \mathbf{b}$$

$$\mu_t = \frac{1}{d}\mathbf{h}_{t,j} \quad \sigma_t = \sqrt{\frac{1}{d}(\mathbf{h}_{t,j} - \mu_j)^2}$$

Residual Layer

$$res(\mathbf{h}_t, l) = \mathbf{h}_t + l(\mathbf{h}_t$$

Architecture Definition Language (ADL)

Transformer

Encoder

$$t_{enc} = res\_nd(mh\_dot\_self\_attn) \rightarrow res\_nd(ffl)$$

Decoder

$$t_{dec} = res\_nd(mh\_dot\_self\_attn) \rightarrow res\_nd(mh\_dot\_src\_att) \rightarrow res\_nd(ffl)$$

RNN

$$rnn(\mathbf{h}_t) = f_{rnn\_o}(\mathbf{h}_t, \mathbf{s}_{t-1})$$
$$\mathbf{s}_t = f_{rnn\_h}(\mathbf{h}_t, \mathbf{s}_{t-1})$$

Convolution

$$cnn(\mathbf{H}, v, k) = v(\mathbf{W}[\mathbf{h}_{i-\lfloor k/2 \rfloor}; \ldots \mathbf{h}_{i+\lfloor k/2 \rfloor}] + \mathbf{b})$$

- Flexible Neural Machine Translation Architecture Combination
- *Related Work*
- Experiments
- Conclusion

Related Work

- Britz et al. (2017) explored hyperparams of RNN NMT models with different attention mechanisms.
- Schrimpf et al (2018) defined language for exploring architectures for RNNs.
- Zoph and Le (2016), Negrinho and Gordan (2017), and Liu et al. (2017) explored architectures for image classification.

- Flexible Neural Machine Translation Architecture Combination
- Related Work
- *Experiments*
- Conclusion

Experiments: Setup

- Used adapted version of SOCKEYE (Heiber et al. 2017)
- Used WMT and IWSLT datasets for different sets of text, which are on the order of 5 million training sentences.
- English $\rightarrow$ German and Latvian $\rightarrow$ English translation problems.
- Ran each experiment 3 times with different random seeds. values reported are the mean result and standard deviation of the BLEU and METEOR scores.

NMT Architectures defined by ADL

e.g. Transformer [Vaswani et al. 2017]

$$t_{\text{enc}} = \mathit{res\_nd}(\mathit{mh\_dot\_self\_att}) \rightarrow \mathit{res\_nd}(\mathit{ffl})$$

$$t_{\text{dec}} = \mathit{res\_nd}(\mathit{mh\_dot\_self\_att}) \rightarrow \mathit{res\_nd}(\mathit{mh\_dot\_src\_att}) \rightarrow \mathit{res\_nd}(\mathit{ffl})$$

$$\mathbf{U}^{L_s} = \mathit{pos} \rightarrow \mathit{repeat}(n, t_{\text{enc}}) \rightarrow \mathit{norm}$$

$$\mathbf{Z}^{L} = \mathit{pos} \rightarrow \mathit{repeat}(n, t_{\text{dec}}) \rightarrow \mathit{norm}$$

Usage in Sockeye, our NMT toolkit:

```
> sockeye-train -s train.de -t train.en
  --custom-seq-encoder repeat(6,birnn)
  --custom-seq-decoder pos->res(norm->mh_dot_self_att)->res(mh_dot_src_att)
```

https://github.com/awslabs/sockeye/tree/acl18

Experiments: What to Attend To?

- Best attention on the upper encoder block.
- No gains observed by attention on different encoder layers in source attention mechanism.

Encoder/Decoder Combinations

- Baseline: 6 layer Transformer, 512 hidden units
- Data sets: IWSLT'16, WMT'17
- Metrics: BLEU (and METEOR in the paper)
- 3 runs, reporting mean and standard deviation

$t_{\text{enc}} = res\_nd(\quad) \rightarrow res\_nd(ffl)$

$t_{\text{dec}} = res\_nd(\quad) \rightarrow res\_nd(mh\_dot\_src\_att) \rightarrow res\_nd(ffl)$

Encoder	Decoder	IWSLT EN→DE	WMT'17 EN→DE	WMT'17 LV→EN
self-att	self-att	25.4 ± 0.2	27.6 ± 0.0	18.3 ± 0.0
self-att	RNN	25.1 ± 0.1	27.4 ± 0.1	18.4 ± 0.2
self-att	CNN	25.4 ± 0.4	27.6 ± 0.2	18.0 ± 0.3
RNN	self-att	25.8 ± 0.1	27.2 ± 0.1	17.8 ± 0.1
CNN	self-att	25.7 ± 0.1	26.6 ± 0.3	16.8 ± 0.4
RNN	RNN	25.1 ± 0.1	26.7 ± 0.1	17.8 ± 0.1
CNN	CNN	25.3 ± 0.3	26.9 ± 0.1	16.4 ± 0.2
self-att	*combined*	25.1 ± 0.2	27.6 ± 0.2	18.3 ± 0.1
self-att	*none*	23.7 ± 0.2	25.3 ± 0.2	15.9 ± 0.1

Encoder block	IWSLT	WMT'17
upper	25.4 ± 0.2	27.6 ± 0.0
increasing	25.4 ± 0.1	27.3 ± 0.1
decreasing	25.3 ± 0.2	27.1 ± 0.1

BLEU Scores

Experiments: Network Structure

RNN $\mapsto$ Transformer

- RNN includes multiple source attention layers, multi-headed attention, layer normalization, residual upscaling FF layers, and single headed MLP attention.
- Start with RNN, add multi-headed attention (mh), positional embedding (pos), layer normalization (norm), single headed attention, attention to residual blocks (multi-att), and residual feed-forward layers after attention blocks (ff).
- Gains from multi-headed attention and feed-forward residual layers.

RNN to Transformer

$$\mathbf{U}^{L_s} = dropout \to res\_d(birnn) \to repeat(5, res\_d(rnn))$$

$$\mathbf{Z}^{L} = dropout \to repeat(6, res\_d(rnn)) \to res\_d(dot\_src\_att) \to res\_d(ffl)$$

Model	IWSLT EN→DE	WMT'17 EN→DE	WMT'17 LV→EN
Transformer	25.4 ± 0.1	27.6 ± 0.0	18.5 ± 0.0
RNMT	23.2 ± 0.2	25.5 ± 0.2	-
- input feeding	23.1 ± 0.2	24.6 ± 0.1	-
RNN	22.8 ± 0.2	23.8 ± 0.1	15.2 ± 0.1
+ mh	23.7 ± 0.4	24.4 ± 0.1	16.0 ± 0.1
+ pos	23.9 ± 0.2	24.1 ± 0.1	15.6 ± 0.1
+ norm	23.7 ± 0.1	24.0 ± 0.2	15.2 ± 0.1
+ multi-att-1h	24.5 ± 0.0	25.2 ± 0.1	16.6 ± 0.2
/ multi-att	24.4 ± 0.3	25.5 ± 0.0	17.0 ± 0.2
+ ff	25.1 ± 0.1	26.7 ± 0.1	17.8 ± 0.1

$$\mathbf{U}^{L_s} = pos \to res\_nd(birnn) \to res\_nd(ffl) \to repeat(5, res\_nd(rnn) \to res\_nd(ffl)) \to norm$$

$$\mathbf{Z}^{L} = pos \to repeat(6, res\_nd(rnn) \to res\_nd(mh\_dot\_src\_att) \to res\_nd(ffl)) \to norm$$

Can see that as RNN $\mapsto$ Transformer the BLEU scores increase.

Experiments: Network Structure

CNN $\mapsto$ Transformer

- Neither Transformer nor CNN have dependency between decoder timesteps during traning, use multiple source attention mechanisms, and use different residual structures.
- Largest gains from adding residual feed-forward layers.

CNN to Transformer

$$\mathbf{U}^{L_s} = \mathit{pos} \rightarrow \mathit{repeat}(6, \mathit{res\_d}(\mathit{cnn}))$$

$$\mathbf{Z}^{L} = \mathit{pos} \rightarrow \mathit{repeat}(6, \mathit{res\_d}(\mathit{cnn}) \rightarrow \mathit{res\_d}(\mathit{dot\_src\_att}))$$

Model	IWSLT EN→DE	WMT'17 EN→DE	WMT'17 LV→EN
Transformer	25.4 ± 0.1	27.6 ± 0.0	18.5 ± 0.0
CNN GLU	24.3 ± 0.4	25.0 ± 0.3	16.0 ± 0.5
+ norm	24.1 ± 0.1	-	16.1 ± 0.2
+ mh	24.2 ± 0.2	25.4 ± 0.1	16.1 ± 0.1
+ ff	25.3 ± 0.1	26.8 ± 0.1	16.4 ± 0.2
CNN ReLU	23.6 ± 0.3	23.9 ± 0.1	15.4 ± 0.1
+ norm	24.3 ± 0.1	24.3 ± 0.2	16.0 ± 0.2
+ mh	24.2 ± 0.2	24.9 ± 0.1	16.1 ± 0.1
+ ff	25.3 ± 0.3	26.9 ± 0.1	16.4 ± 0.2

$$\mathbf{U}^{L_s} = \mathit{pos} \rightarrow \mathit{repeat}(6, \mathit{res\_nd}(\mathit{cnn}) \rightarrow \mathit{res\_nd}(\mathit{ffl})) \rightarrow \mathit{norm}$$

$$\mathbf{Z}^{L} = \mathit{pos} \rightarrow \mathit{repeat}(6, \mathit{res\_nd}(\mathit{cnn}) \rightarrow \mathit{res\_nd}(\mathit{mh\_dot\_src\_att}) \rightarrow \mathit{res\_nd}(\mathit{ffl})) \rightarrow \mathit{norm}$$

Can see that as CNN $\mapsto$ Transformer the BLEU scores increase.

Experiments: Self-Attention Variations

- Self-attention has advantage because two positions directly connected and no dependencies between consecutive timesteps.
- Self-attention has disadvantage because positional information isn't directly represented and need multiple heads.
- Experiments show attention is more important to decoder side.
- Attention on encoder shows little to no improvement.

Experiments: Self-Attention Variations

Encoder	Decoder	IWSLT EN→DE BLEU	WMT'17 EN→DE BLEU	WMT'17 EN→DE METEOR	WMT'17 LV→EN BLEU	WMT'17 LV→EN METEOR
self-att	self-att	25.4 ± 0.2	27.6 ± 0.0	47.2 ± 0.1	18.3 ± 0.0	51.1 ± 0.1
self-att	RNN	25.1 ± 0.1	27.4 ± 0.1	47.0 ± 0.1	18.4 ± 0.2	51.1 ± 0.1
self-att	CNN	25.4 ± 0.4	27.6 ± 0.2	46.7 ± 0.1	18.0 ± 0.3	50.3 ± 0.3
RNN	self-att	25.8 ± 0.1	27.2 ± 0.1	46.7 ± 0.1	17.8 ± 0.1	50.6 ± 0.1
CNN	self-att	25.7 ± 0.1	26.6 ± 0.3	46.3 ± 0.1	16.8 ± 0.4	49.4 ± 0.4
RNN	RNN	25.1 ± 0.1	26.7 ± 0.1	46.4 ± 0.2	17.8 ± 0.1	50.5 ± 0.1
CNN	CNN	25.3 ± 0.3	26.9 ± 0.1	46.1 ± 0.0	16.4 ± 0.2	47.9 ± 0.2
self-att	*combined*	25.1 ± 0.2	27.6 ± 0.2	47.2 ± 0.2	18.3 ± 0.1	51.1 ± 0.1
self-att	*none*	23.7 ± 0.2	25.3 ± 0.2	43.1 ± 0.1	15.9 ± 0.1	45.1 ± 0.2

Table 5: Different variations of the encoder and decoder self-attention layer.

- Flexible Neural Machine Translation Architecture Combination
- Related Work
- Experiments
- *Conclusion*

Conclusion

- Defined ADL for specifying NMT architectures on composable building blocks.
- Found RNN models benefit from multiple source attention mechanisms and residual feed-forward blocks.
- Found CNN benefits from layer normalization and feed-forward blocks.
- These features explain the effectiveness of transformers, as they make the respective models more transformer "like".
- RNN and CNN models with self-attention on the encoder side are competitive with transformers.

QUESTIONS?