Pervasive Attention

2D Convolutional Neural Networks for Sequence-to-Sequence Prediction

Maha Elbayad Laurent Besacier Jakob Verbeek

CoNLL 2018, Brussels, Belgium

NLE seminar, Sept 28, 2018

Overview

- Sequence-to-sequence models:
  - Recurrent vs feed-forward en/decoders.
  - Encoder-decoder interfacing.
- Pervasive attention (our work).
- Experimental results.
- Conclusions.

Sequence-to-sequence models

A conditional language model that assigns probabilities to a sequence of tokens $\mathbf{y} = (y_1, ..., y_{|y|})$ given a conditioning sequence $\mathbf{x} = (x_1, ..., x_{|x|})$.

Using the chain rule:

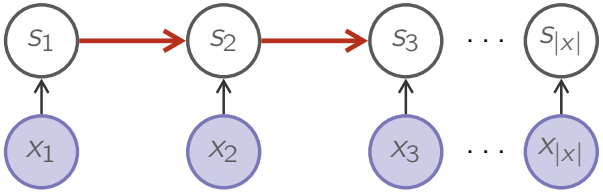
$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|y|} p(y_t|\mathbf{x}, y_{<t})$$

$$p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|y|} f_\theta(\mathbf{x}, y_{<t})$$

How to encode x and $y_{<t}$?

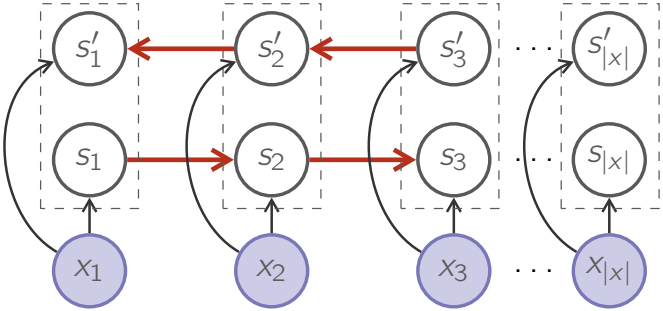
Recurrent encoder-decoders

Encoder-decoder models | Recurrent networks



Encoder

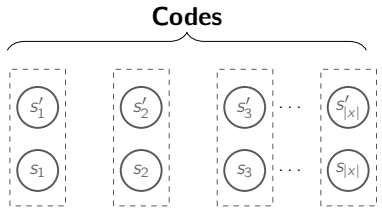
Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)



Bidirectional encoder

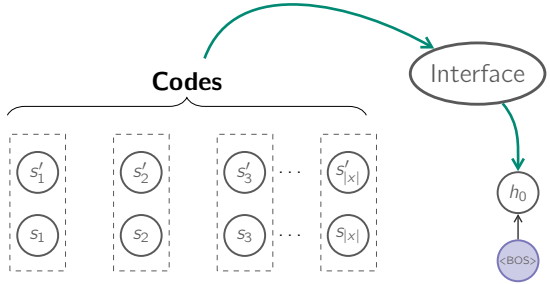
Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

Encoder-decoder models | Recurrent networks



Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

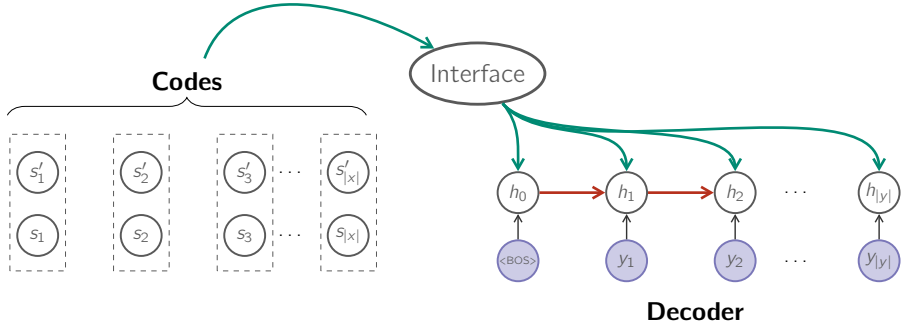
Encoder-decoder models | Recurrent networks



Decoder

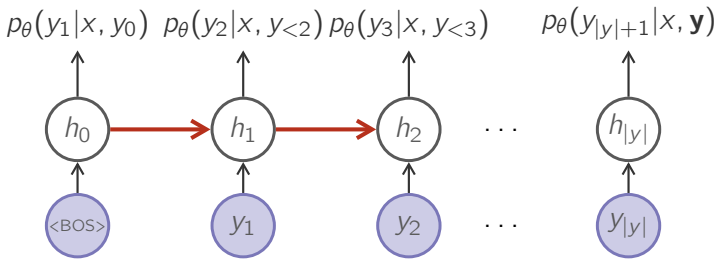
Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

Encoder-decoder models | Recurrent networks



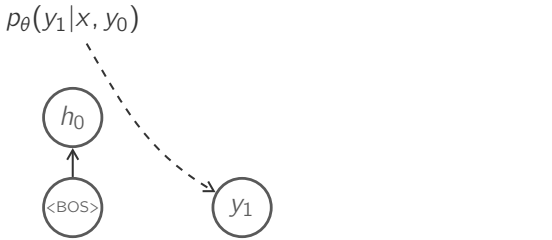
Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

Encoder-decoder models | Recurrent networks



Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

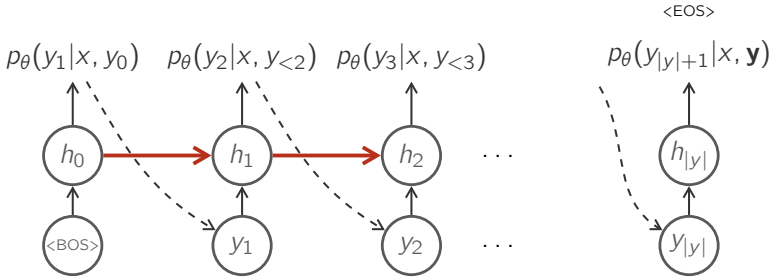
Encoder-decoder models | Recurrent networks



Decoder (inference)

Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

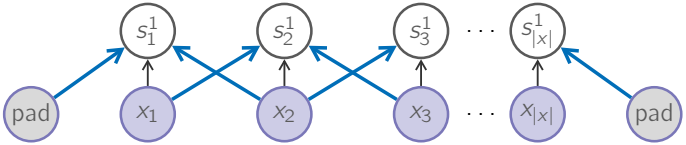
Encoder-decoder models | Recurrent networks



Graves (2013); Sutskever et al. (2014); Cho et al. (2014); Bahdanau et al. (2015)

Feed-forward encoder-decoders

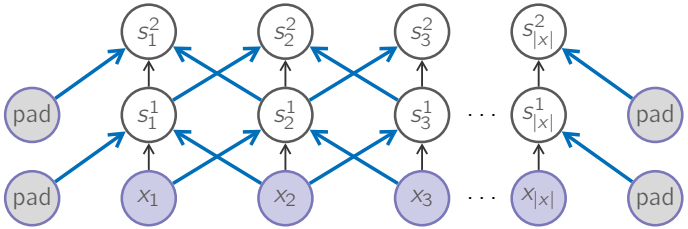
Encoder-decoder models | Convolutional networks



Encoder

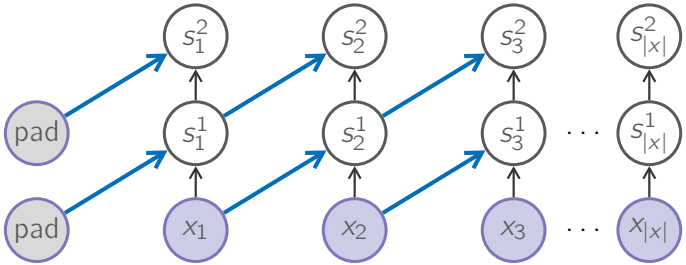
Kalchbrenner et al. (2014); Kim (2014); Gehring et al. (2017b)

Encoder-decoder models | Convolutional networks



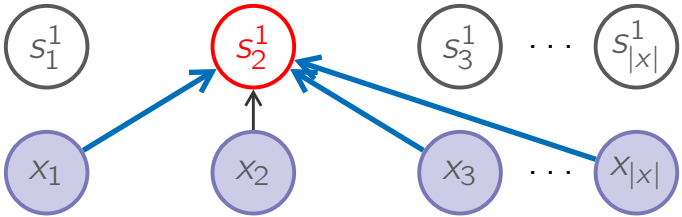
Encoder

Kalchbrenner et al. (2014); Kim (2014); Gehring et al. (2017b)



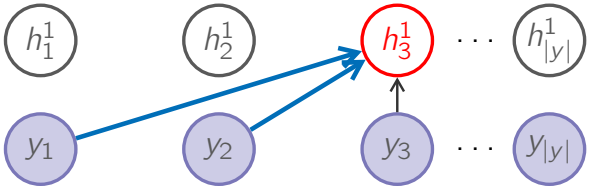
Decoder (Causal convolutions)

Kalchbrenner et al. (2014); Kim (2014); Gehring et al. (2017b)



Encoder (self-attention)

Vaswani et al. (2017)



Decoder (masked self-attention)

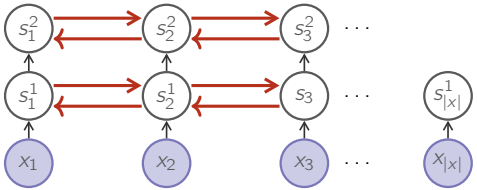
Vaswani et al. (2017)

Encoder-decoder models | Feed-forward

Similar to RNNs:

- Possible interfacing between the encoder & decoder.
- Inference by building step by step the target sequence.

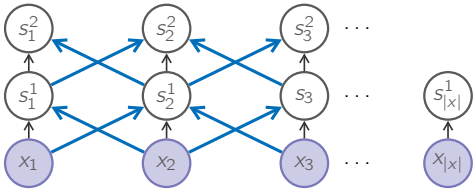
Encoder-decoder models



Recurrent

1. Autoregressive filters (1d).
2. Unbounded dependencies.
3. $\mathcal{O}(T)$ sequential steps.
4. Full context.

Encoder-decoder models



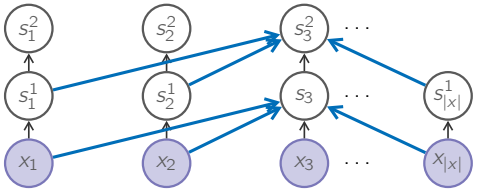
Recurrent

1. Autoregressive filters (1d).
2. Unbounded dependencies.
3. $\mathcal{O}(T)$ sequential steps.
4. Full context.

Convolutional

1. Convolutional filters (1d).
2. Bounded dependencies.
3. $\mathcal{O}(1)$ sequential steps.
4. Incrementally built context.

Encoder-decoder models



Recurrent

1. Autoregressive filters (1d).
2. **Unbounded dependencies.**
3. $\mathcal{O}(T)$ sequential steps.
4. **Full context.**

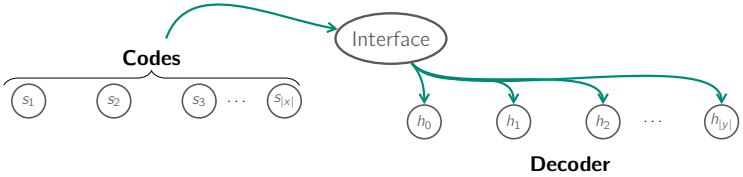
Convolutional

1. Convolutional filters (1d).
2. Bounded dependencies.
3. $\mathcal{O}(1)$ **sequential steps.**
4. Incrementally built context.

Transformer

1. (self-) attention.
2. **Unbounded dependencies.**
3. $\mathcal{O}(1)$ **sequential steps.**
4. **Full context.**

Encoder-decoder interfacing

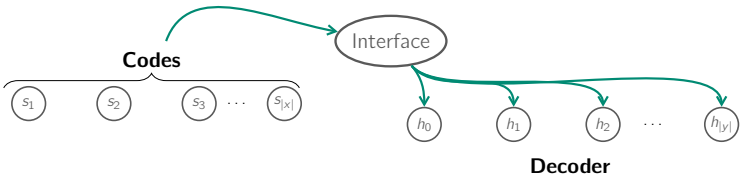


Single vector:

- context $= s_{|s|}$.
- context $= \frac{1}{|s|} \sum s_i$
- context $= \max s_i$

Only at h_0 or at every time step.

Encoder-decoder interfacing



Single vector:

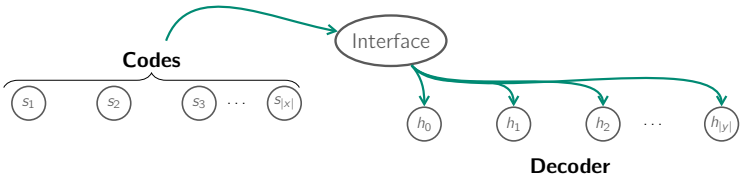
- context $= s_{|s|}$.
- context $= \frac{1}{|s|} \sum s_i$
- context $= \max s_i$

Only at h_0 or at every time step.

Attention mechanisms:

$$e_i = \text{score}(s_i, h_{t-1}), \forall i$$
$$\alpha = softmax(e_i)_i$$
$$\text{context}_t = \sum_i \alpha_i s_i$$

Encoder-decoder interfacing



Attention mechanisms:

$$e_i = \text{score}(s_i, h_{t-1}), \forall i$$

$$\alpha = softmax(e_i)_i$$

$$\text{context}_t = \sum_i \alpha_i s_i$$

Scoring (k, q):

dot $k^T Wq$

concat $v^T tanh(W[k; q])$

Scaled-dot $\frac{k^T Wq}{dim(k)}$

Breaking the paradigm

State of the art sequence to sequence learning:

- Recurrent (Bahdanau et al., 2015; Luong et al., 2015).
- Convolutional (Gehring et al., 2017b).
- Transformer (Vaswani et al., 2017)

Encoder-decoder + shallow attention interface

Breaking the paradigm

State of the art sequence to sequence learning:

- Recurrent (Bahdanau et al., 2015; Luong et al., 2015).
- Convolutional (Gehring et al., 2017b).
- Transformer (Vaswani et al., 2017)

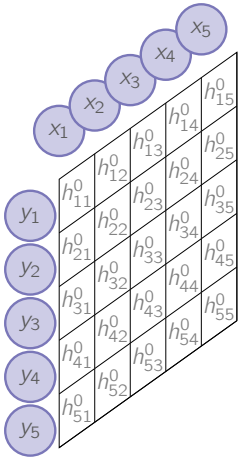
~~**Encoder-decoder + shallow attention interface**~~

How about a built-in deep interface between the two sequences?

No separate encoder and decoder modules!

Pervasive attention

Pervasive attention



The initial 2D grid:

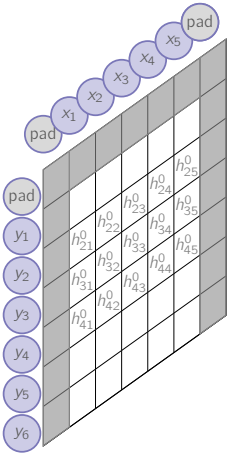
$\forall i, j : \{1, ...|y|\} \times \{1, ...|x|\}$

$u_i = U_{y_i}$ and $v_j = V_{x_j}$

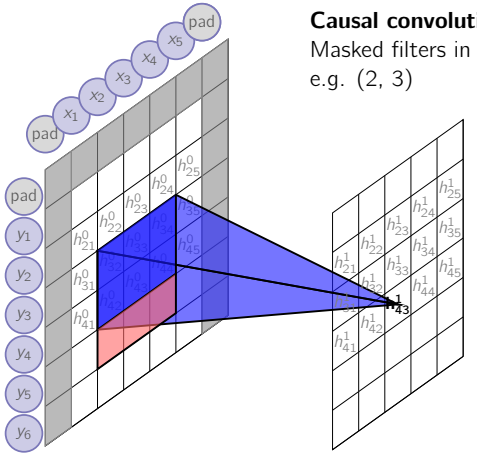
token embeddings from lookup tables.

$$h_{ij} = f(u_i, v_j)$$

$$= \begin{cases} concat([u_i, v_j]) \\ u_i \odot v_j \end{cases}$$

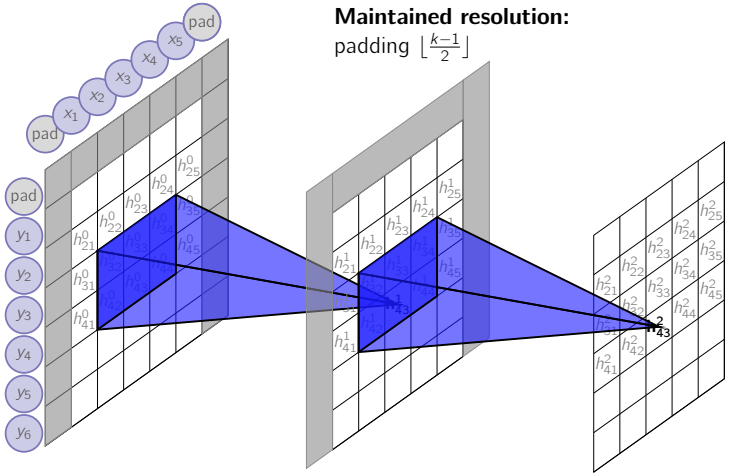


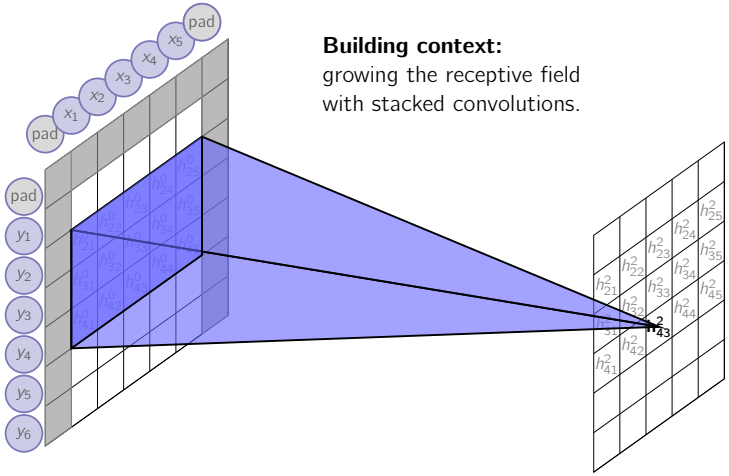
Padding:
Left and right for the source.
Only left for the target.



Causal convolution:
Masked filters in the target direction
e.g. (2, 3)

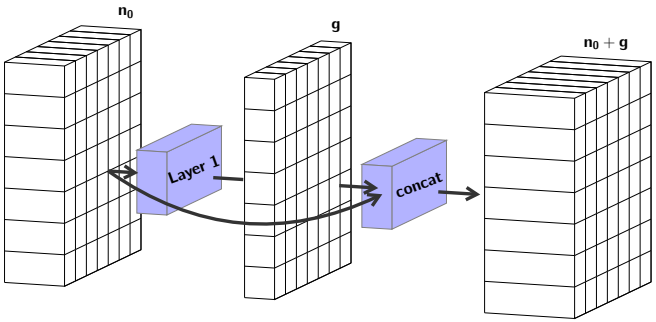
Pervasive attention | 2D convolutional network





Building context:
growing the receptive field with stacked convolutions.

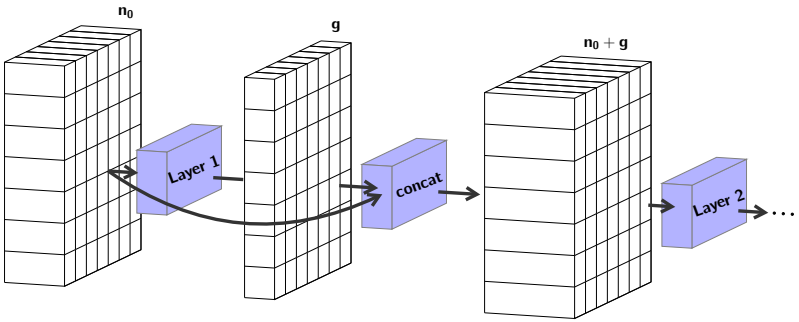
Pervasive attention | DenseNets (Huang et al., 2017)



Layer = BN + ReLU + Conv(1) + BN + ReLU + Conv(k)

Growing the input feature maps by $\mathbf{g} = \mathbf{32}$

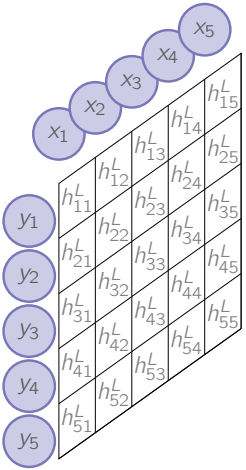
Pervasive attention | DenseNets (Huang et al., 2017)



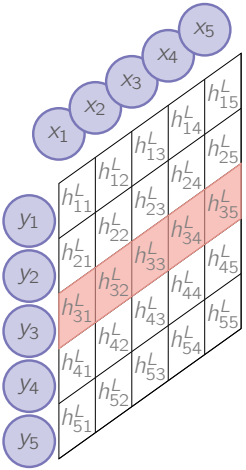
Growing the feature maps up to $\mathbf{n_0 + L \times g}$

n_0 : initial features

L : number of layers **(single block)**.



How to encode x and $y_{<t}$ in $f_\theta(\mathbf{x}, y_{<t})$?



Aggregate the hidden states (channel by channel):
With $H_3 = [h^L_{31}, ..., h^L_{3|x|}] \in \mathbb{R}^{d \times |x|}$

- Max/avergae pooling:
  - $h_3 = \max_{1 \leq i \leq |x|} H_3.$
  - $h_3 = \frac{1}{|x|} \sum_{1 \leq i \leq |x|} H_3.$
- Self-attention:
  $\rho = softmax(H_3^T W + b)$
  $h_3 = H_3 \rho.$
- A combination of the above!

Experiments

Experiments

- How do we fare against sota encoder-decoder models?
- What are some key parameters of our model?

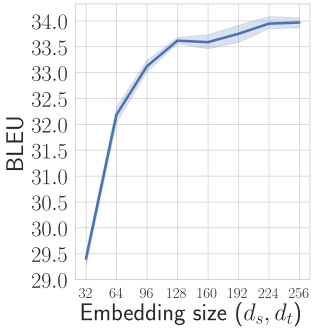
Experiments

- How do we fare against sota encoder-decoder models?
- What are some key parameters of our model?

Task : translation of TED and TEDx talks: German↔English.

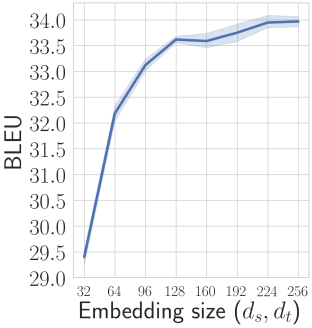
- Train 160k, dev 7k, test 6.5k.
- Lowercasing and tokenization (Moses).
- Subwords (BPE, Sennrich et al. (2016)).

Experiments: IWSLT'14 | Ablation study

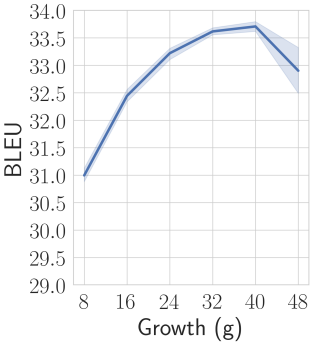


L=20, g=32

Experiments: IWSLT'14 | Ablation study

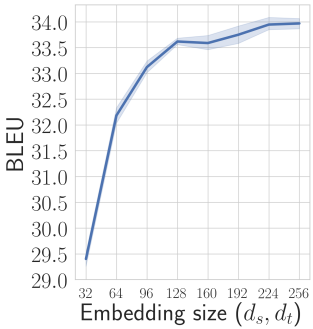


L=20, g=32

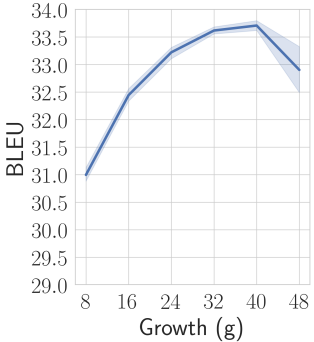


d=128 , L=20

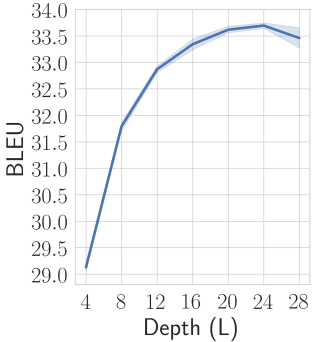
Experiments: IWSLT'14 | Ablation study



L=20, g=32



d=128 , L=20



d=128, g=32

Experiments: IWSLT'14 | Ablation study

Model	BLEU	Flops$\times 10^5$	#params
Average	31.57 $\pm$ 0.11	3.63	7.18M
Max	**33.70** $\pm$ **0.06**	3.44	7.18M
Attn	32.09 $\pm$ 0.12	3.61	7.24M
[Max, Attn]	**33.81** $\pm$ **0.03**	3.51	7.24M

Our model ($L=24$, $g=32$, $d_s=d_t=128$) with different **aggregators**.

Experiments: IWSLT'14 | State-of-the-art

Word-based	De-En	Flops ($\times 10^5$)	# prms	En-De	# prms
Conv-LSTM (MLE) (Bahdanau et al., 2017)	27.56				
Bi-GRU (MLE+SLE) (Bahdanau et al., 2017)	28.53				
Conv-LSTM (deep+pos) (Gehring et al., 2017a)	30.4				
NPMT + language model (Huang et al., 2018)	30.08			25.36	
BPE-based					
RNNsearch* (Bahdanau et al., 2015)	31.02	1.79	6M	25.92	7M
Varational attention (Deng et al., 2018)	**33.10**				
Transformer** (Vaswani et al., 2017)	32.83	3.53	59M	27.68	61M
ConvS2S** (MLE) (Gehring et al., 2017b)	32.31	1.35	21M	26.73	22M
ConvS2S (MLE+SLE) (Edunov et al., 2018)	32.84				
Pervasive attention (ours)	**34.10**± **0.04**	3.51	9M	**27.77**± **0.1**	7M

(*): results obtained using our implementation.
(**): results obtained using FairSeq (Gehring et al., 2017b).

Experiments: IWSLT'14 | State-of-the-art

Word-based	De-En	Flops ($\times 10^5$)	# prms	En-De	# prms
Conv-LSTM (MLE) (Bahdanau et al., 2017)	27.56				
Bi-GRU (MLE+SLE) (Bahdanau et al., 2017)	28.53				
Conv-LSTM (deep+pos) (Gehring et al., 2017a)	30.4				
NPMT + language model (Huang et al., 2018)	30.08			25.36	
BPE-based					
RNNsearch* (Bahdanau et al., 2015)	31.02	1.79	6M	25.92	7M
Varational attention (Deng et al., 2018)	**33.10**				
Transformer** (Vaswani et al., 2017)	32.83	3.53	59M	27.68	61M
ConvS2S** (MLE) (Gehring et al., 2017b)	32.31	1.35	21M	26.73	22M
ConvS2S (MLE+SLE) (Edunov et al., 2018)	32.84				
Pervasive attention (ours)	**34.10± 0.04**	3.51	9M	**27.77± 0.1**	7M
Transformer** (Vaswani et al., 2017)	**35.61**	3.42	46M	**29.5**	

(*): results obtained using our implementation.
(**): results obtained using FairSeq (Gehring et al., 2017b).

Alignment visualization

For i a position in y, the max-pooling operator partitions the n channels by assigning them across the source tokens j.

$$B_{ij} = \{d \in \{1, \ldots, f_L\} | \, j = \arg\max(H^L_{ijd})\}$$

Visualizing $\alpha_{i,j} \propto |B_{ij}|$

Alignment visualization | Example 1

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters.
  - Source-target interactions pervasive in the architecture.
  - Proper re-encoding of source at every target token.
  - Max-pooling generates implicit alignment.
- Compares favorably to encoder-decoder models

Conclusion

- Joint-coding approach, alternative to encoder-decoder
  - 2D CNN with masked filters.
  - Source-target interactions pervasive in the architecture.
  - Proper re-encoding of source at every target token.
  - Max-pooling generates implicit alignment.
- Compares favorably to encoder-decoder models
- Ongoing work:
  - Training on larger corpora (WMT).
  - More efficient hybrid 1D-2D architectures
  - Simultaneous training for both directions.

Thank you for your attention.

References

Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A., and Bengio, Y. (2017). An actor-critic algorithm for sequence prediction. In *ICLR*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. (2018). Latent alignment and variational attention. *arXiv preprint arXiv:1807.03756*.

Edunov, S., Ott, M., Auli, M., Grangier, D., and Ranzato, M. (2018). Classical structured prediction losses for sequence to sequence learning. In *NAACL*.

Gehring, J., Auli, M., Grangier, D., and Dauphin, Y. (2017a). A convolutional encoder model for neural machine translation. In *ACL*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. (2017b). Convolutional sequence to sequence learning. In *ICML*.

Graves, A. (2013). Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. (2017). Densely connected convolutional networks. In *CVPR*.

Huang, P., Wang, C., Huang, S., Zhou, D., and Deng, L. (2018). Towards neural phrase-based machine translation. In *ICLR*.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *ACL*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *ACL*.

Luong, T., Pham, H., and Manning, C. (2015). Effective approaches to attention-based neural machine translation. In *EMNLP*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *ACL*.

Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In *NIPS*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS*.