Depth Growing for Neural Machine Translation

Lijun Wu[1,*], **Yiren Wang**[2,*], **Yingce Xia**[3,†], **Fei Tian**[3], **Fei Gao**[3],
Tao Qin[3], **Jianhuang Lai**[1], **Tie-Yan Liu**[3]
[1]School of Data and Computer Science, Sun Yat-sen University;
[2] University of Illinois at Urbana-Champaign;
[3] Microsoft Research Asia
[1]{wulijun3, stsljh}@mail2.sysu.edu.cn, [2]yiren@illinois.edu,
[3]{Yingce.Xia, fetia, feiga, taoqin, tyliu}@microsoft.com

Abstract

While very deep neural networks have shown effectiveness for computer vision and text classification applications, how to increase the network depth of neural machine translation (NMT) models for better translation quality remains a challenging problem. Directly stacking more blocks to the NMT model results in no improvement and even reduces performance. In this work, we propose an effective two-stage approach with three specially designed components to construct deeper NMT models, which result in significant improvements over the strong Transformer baselines on WMT14 English→German and English→French translation tasks.

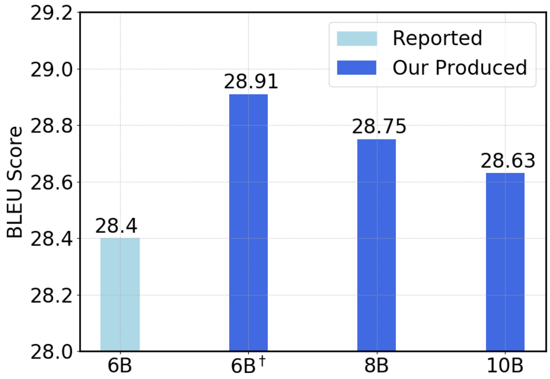

Figure 1: Performances of Transformer models with different number of encoder/decoder blocks (recorded on x-axis) on WMT14 En→De translation task. † denotes the result reported in (Vaswani et al., 2017).

1 Introduction

Neural machine translation (briefly, NMT), which is built upon deep neural networks, has gained rapid progress in recent years (Bahdanau et al., 2014; Sutskever et al., 2014; Sennrich et al., 2015; He et al., 2016a; Sennrich et al., 2016a; Xia et al., 2017; Wang et al., 2019) and achieved significant improvement in translation quality (Hassan et al., 2018). Variants of network structures have been applied in NMT such as LSTM (Wu et al., 2016), CNN (Gehring et al., 2017) and Transformer (Vaswani et al., 2017).

Training deep networks has always been a challenging problem, mainly due to the difficulties in optimization for deep architecture. Breakthroughs have been made in computer vision to enable deeper model construction via advanced initialization schemes (He et al., 2015), multi-stage training strategy (Simonyan and Zisserman, 2014), and novel model architectures (Srivastava et al., 2015; He et al., 2016b). While constructing very deep neural networks with tens and even more than a hundred blocks have shown effectiveness in image recognition (He et al., 2016b), question answering and text classification (Devlin et al., 2018; Radford et al., 2019), scaling up model capacity with very deep network remains challenging for NMT. The NMT models are generally constructed with up to 6 encoder and decoder blocks in both state-of-the-art research work and champion systems of machine translation competition. For example, the LSTM-based models are usually stacked for 4 (Stahlberg et al., 2018) or 6 (Chen et al., 2018) blocks, and the state-of-the-art Transformer models are equipped with a 6-block encoder and decoder (Vaswani et al., 2017; Junczys-Dowmunt, 2018; Edunov et al., 2018). Increasing the NMT model depth by directly stacking more blocks results in no improvement or performance drop (Figure 1), and even leads to optimization failure (Bapna et al., 2018).

There have been a few attempts in previous works on constructing deeper NMT models. Zhou

[*] The first two authors contributed equally to this work. This work is conducted at Microsoft Research Asia.

[†] Corresponding author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5558–5563
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

et al. (2016) and Wang et al. (2017) propose increasing the depth of LSTM-based models by introducing linear units between internal hidden states to eliminate the problem of gradient vanishing. However, their methods are specially designed for the recurrent architecture which has been significantly outperformed by the state-of-the-art transformer model. Bapna et al. (2018) propose an enhancement to the attention mechanism to ease the optimization of models with deeper encoders. While gains have been reported over different model architectures including LSTM and Transformer, their improvements are not made over the best performed baseline model configuration. How to construct and train *deep* NMT models to push forward the state-of-the-art translation performance with larger model capacity remains a challenging and open problem.

In this work, we explore the potential of leveraging *deep* neural networks for NMT and propose a new approach to construct and train deeper NMT models. As aforementioned, constructing deeper models is not as straightforward as directly stacking more blocks, but requires new mechanisms to boost the training and utilize the larger capacity with minimal increase in complexity. Our solution is a new two-stage training strategy, which "grows" a well-trained NMT model into a deeper network with three components specially designed to overcome the optimization difficulty and best leverage the capability of both shallow and deep architecture. Our approach can effectively construct a deeper model with significantly better performance, and is generally applicable to any model architecture.

We evaluate our approach on two large-scale benchmark datasets, WMT14 English→German and English→French translations. Empirical studies show that our approach can significantly improve in translation quality with an increased model depth. Specifically, we achieve 1.0 and 0.6 BLEU score improvement over the strong Transformer baseline in English→German and English→French translations.

2 Approach

We introduce the details of our proposed approach in this section. The overall framework is illustrated in Figure 2.

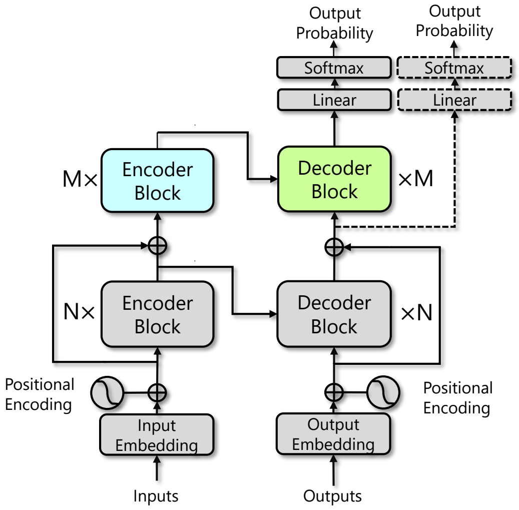

Figure 2: The overall framework of our proposed deep model architecture. N and M are the numbers of blocks in the bottom module (i.e., grey parts) and top module (i.e., blue and green parts). Parameters of the bottom module are fixed during the top module training. The dashed parts denote the original training/decoding of the bottom module. The weights of the two linear operators before softmax are shared.

Our model consists of a bottom module with N blocks of encoder and decoder (the grey components in Figure 2), and a top module with M blocks (the blue and green components). We denote the encoder and decoder of the bottom module as $\texttt{enc}_1$ and $\texttt{dec}_1$, and the corresponding two parts of the top module as $\texttt{enc}_2$ and $\texttt{dec}_2$. An encoder-decoder attention mechanism is used in the decoder blocks of the NMT models, and here we use $\texttt{attn}_1$ and $\texttt{attn}_2$ to represent such attention in the bottom and top modules respectively.

The model is constructed via a two-stage training strategy: in Stage 1, the bottom module (i.e., $\texttt{enc}_1$ and $\texttt{dec}_1$) is trained and subsequently holds constant; in Stage 2, only the top module (i.e., $\texttt{enc}_2$ and $\texttt{dec}_2$) is optimized.

Let x and y denote the embedding of source and target sequence. Let l_y denote the number of words in y, and $y_{<t}$ denote the elements before time step t. Our proposed model works in the following way:

$$h_1 = \texttt{enc}_1(x);\; h_2 = \texttt{enc}_2(x + h_1); \tag{1}$$

$$s_{1,t} = \texttt{dec}_1(y_{<t}, \texttt{attn}_1(h_1)),\; \forall t \in [l_y]; \tag{2}$$

$$s_{2,t} = \texttt{dec}_2(y_{<t} + s_{1,<t}, \texttt{attn}_2(h_2)), \tag{3}$$

which contains three key components specially designed for deeper model construction, including:
(1) ***Cross-module residual connections:*** As shown in Eqn.(1), the encoder $\texttt{enc}_1$ of the bottom module encodes the input x to a hidden repre-

sentation h_1, then a cross-module residual connection is introduced to the top module and the representation h_2 is eventually produced. The decoders work in a similar way as shown in Eqn.(2) and (3). This enables the top module to have direct access to both the low-level input signals from the word embedding and high-level information generated by the bottom module. Similar principles can be found in Wang et al. (2017); Wu et al. (2018).

(2) ***Hierarchical encoder-decoder attention:*** We introduce a hierarchical encoder-decoder attention calculated with different contextual representations as shown in Eqn.(2) and (3), where h_1 is used as key and value for $\texttt{attn}_1$ in the bottom module, and h_2 for $\texttt{attn}_2$ in the top module. Hidden states from the corresponding previous decoder block are used as queries for both $\texttt{attn}_1$ and $\texttt{attn}_2$ (omitted for readability). In this way, the strong capability of the well trained bottom module can be best preserved regardless of the influence from top module, while the newly stacked top module can leverage the higher-level contextual representations. More details can be found from source code in the supplementary materials.

(3) ***Deep-shallow decoding:*** At the decoding phase, $\texttt{enc}_1$ and $\texttt{dec}_1$ work together according to Eqn.(1) and Eqn.(2) as a shallow network $\texttt{net}_S$, integrate both bottom and top module works as a deep network $\texttt{net}_D$ according to Eqn.(1)~Eqn.(3). $\texttt{net}_S$ and $\texttt{net}_D$ generate the final translation results through reranking.

Discussion

• *Training complexity:* As aforementioned, the bottom module is trained in Stage 1 and only parameters of the top module are optimized in Stage 2. This significantly eases optimization difficulty and reduces training complexity. Jointly training the two modules with minimal training complexity is left for future work.

• *Ensemble learning:* What we propose in this paper is a *single* deeper model with hierarchical contextual information, although the deep-shallow decoding is similar to the ensemble methods in terms of inference complexity (Zhou, 2012). While training multiple diverse models for good ensemble performance introduces high additional complexity, our approach, as discussed above, "grows" a well-trained model into a deeper one with minimal increase in training complexity. Detailed empirical analysis is presented in Section 3.3.

3 Experiments

We evaluate our proposed approach on two large-scale benchmark datasets. We compare our approach with multiple baseline models, and analyze the effectiveness of our deep training strategy.

3.1 Experiment Design

Datasets We conduct experiments to evaluate the effectiveness of our proposed method on two widely adopted benchmark datasets: the WMT14[1] English→German translation (En→De) and the WMT14 English→French translation (En→Fr). We use $4.5M$ parallel sentence pairs for En→De and $36M$ pairs for En→Fr as our training data[2]. We use the concatenation of *Newstest2012* and *Newstest2013* as the validation set, and *Newstest2014* as the test set. All words are segmented into sub-word units using byte pair encoding (BPE)[3] (Sennrich et al., 2016b), forming a vocabulary shared by the source and target languages with $32k$ and $45k$ tokens for En→De and En→Fr respectively.

Architecture The basic encoder-decoder framework we use is the strong Transformer model. We adopt the `big` transformer configuration following Vaswani et al. (2017), with the dimension of word embeddings, hidden states and non-linear layer set as 1024, 1024 and 4096 respectively. The dropout rate is 0.3 for En→De and 0.1 for En→Fr. We set the number of encoder/decoder blocks for the bottom module as $N = 6$ following the common practice, and set the number of additionally stacked blocks of the top module as $M = 2$. Our models are implemented based on the PyTorch implementation of Transformer[4] and the code can be found in the supplementary materials.

Training We use Adam (Kingma and Ba, 2014) optimizer following the optimization settings and default learning rate schedule in Vaswani et al. (2017) for model training. All models are trained on 8 M40 GPUs.

[1] `http://www.statmt.org/wmt14/translation-task.html`

[2] Training data are constructed with filtration rules following `https://github.com/pytorch/fairseq/tree/master/examples/translation`

[3] `https://github.com/rsennrich/subword-nmt`

[4] `https://github.com/pytorch/fairseq`

Table 1: The test set performances of WMT14 En→De and En→Fr translation tasks. '†' denotes the performance figures reported in the previous works.

Model	En→De	En→Fr
Transformer (6B)[†]	28.40	41.80
Transformer (6B)	28.91	42.69
Transformer (8B)	28.75	42.63
Transparent Attn (16B)[†]	28.04	–
Ours (8B)	**29.92**	**43.27**

Evaluation We evaluate the model performances with tokenized case-sensitive BLEU[5] score (Papineni et al., 2002) for the two translation tasks. We use beam search with a beam size of 5 and length penalty 0.6 for both tasks.

3.2 Results

We compare our method (*Ours*) with the Transformer baselines of 6 blocks ($6B$) and 8 blocks ($8B$), and a 16-block Transformer with transparent attention (*Transparent Attn (16B)*)[6] (Bapna et al., 2018). We also reproduce a 6-block Transformer baseline, which has better performance than what is reported in (Vaswani et al., 2017) and we use it to initialize the bottom module in our model.

From the results in Table 1, we see that our proposed approach enables effective training for deeper network and achieves significantly better performances compared to baselines. With our method, the performance of a well-optimized 6-block model can be further boosted by adding two additional blocks, while simply using Transformer (8B) will lead to a performance drop. Specifically, we achieve a 29.92 BLEU score on En→De translation with 1.0 BLEU improvement over the strong baselines, and achieve a 0.6 BLEU improvement for En→Fr. The improvements are statistically significant with $p < 0.01$ in paired bootstrap sampling (Koehn, 2004).

3.3 Analysis

To further study the effectiveness of our proposed framework, we present additional comparisons in En→De translation with two groups of baseline approaches in Figure 3:

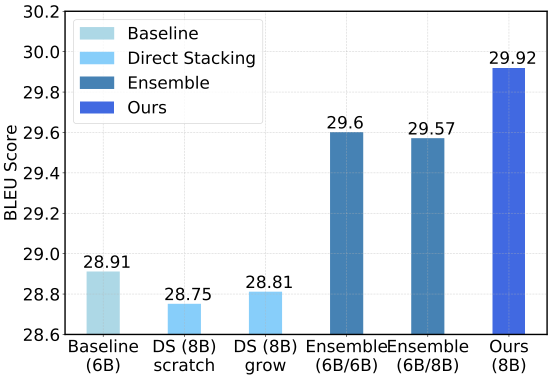

Figure 3: The test performances of WMT14 En→De translation task.

(1) Direct stacking (*DS*): we extend the 6-block baseline to 8-block by directly stacking 2 additional blocks. We can see that both training from scratch (*DS scratch*) and "growing" from a well-trained 6-block model (*DS grow*) fails to improve performance in spite of larger model capacity. The comparison with this group of models shows that directly stacking more blocks is not a good strategy for increasing network depth, and demonstrates the effectiveness and necessity of our proposed mechanisms for training deep networks.

(2) Ensemble learning (*Ensemble*): we present the two-model ensemble results for fair comparison with our approach that involves a two-pass deep-shallow decoding. Specifically, we present the ensemble performances of two independently trained 6-block models (*Ensemble* 6*B*/6*B*), and ensemble of one 6-block and one 8-block model independently trained from scratch (*Ensemble* 6*B*/8*B*). As expected, the ensemble method improves translation quality over the single model baselines by a large margin (over 0.8 BLEU improvement). Regarding training complexity, it takes 40 GPU days (5 days on 8 GPU) to train a single 6-block model from scratch, 48 GPU days for a 8-block model , and 8 GPU days to "grow" a 6-block model into 8-block with our approach. Therefore, our model is better than the two-model ensemble in terms of both translation quality (more than 0.3 BLEU improvement over the ensemble baseline) and training complexity.

[5] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[6] We directly use the performance figure from (Bapna et al., 2018), which uses the `base` Transformer configuration. We run the method of our own implementation with the widely adopted and state-of-the-art `big` setting, but no improvement has been observed.

4 Conclusion

In this paper, we propose a new training strategy with three specially designed components, including cross-module residual connection, hierarchical encoder-decoder attention and deep-shallow decoding, to construct and train deep NMT models. We show that our approach can effectively construct deeper model with significantly better performance over the state-of-the-art transformer baseline. Although only empirical studies on the transformer are presented in this paper, our proposed strategy is a general approach that can be universally applicable to arbitrary model architectures, including LSTM and CNN. In future work, we will further explore an efficient strategy that can jointly train all modules of the deep model with minimal increase in training complexity.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016a. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the wmt2018 news translation task: How i learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. The university of cambridges machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 504–512.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep neural machine translation with linear associative unit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 136–145.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference on Learning Representations*.

Lijun Wu, Fei Tian, Li Zhao, Jianhuang Lai, and Tie-Yan Liu. 2018. Word attention for sequence to sequence text understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.

Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.