Learning Hard Retrieval Decoder Attention for Transformers

Hongfei Xu[1] **Qiuhui Liu**[2] **Josef van Genabith**[1*] **Deyi Xiong**[3,4]
[1]DFKI and Saarland University, Informatics Campus, Saarland, Germany
[2]China Mobile Online Services, Henan, China
[3]Tianjin University, Tianjin, China
[4]Global Tone Communication Technology Co., Ltd.
{hfxunlp, liuqhano}@foxmail.com, josef.van_genabith@dfki.de, dyxiong@tju.edu.cn

Abstract

The Transformer translation model is based on the multi-head attention mechanism, which can be parallelized easily. The multi-head attention network performs the scaled dot-product attention function in parallel, empowering the model by jointly attending to information from different representation subspaces at different positions. In this paper, we present an approach to learning a hard retrieval attention where an attention head only attends to one token in the sentence rather than all tokens. The matrix multiplication between attention probabilities and the value sequence in the standard scaled dot-product attention can thus be replaced by a simple and efficient retrieval operation. We show that our hard retrieval attention mechanism is 1.43 times faster in decoding, while preserving translation quality on a wide range of machine translation tasks when used in the decoder self- and cross-attention networks.

1 Introduction

The Transformer translation model (Vaswani et al., 2017), which has outperformed previous RNN/CNN based sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017), is based on multi-head attention networks. The multi-head attention mechanism, which computes several scaled dot-product attentions in parallel, can be efficiently parallelized at sequence level.

In this paper, we investigate whether we can replace scaled dot-product attention by learning a simpler hard retrieval based attention that attends to a single token only. This simplifies computation and increases speed. We show that this can indeed be achieved by a simple and efficient retrieval operation while preserving translation quality.

Our contributions are as follows:

- We propose a method to learn hard retrieval attention that attends with an efficient indexing operation (resulting in at most h tokens being attended to for h attention heads).
- We empirically show that using the hard retrieval attention mechanism for decoder self- and cross-attention networks increases the decoding speed by 1.43 times while preserving performance on a wide range of MT tasks.

2 Background: the Scaled Dot-Product Attention

The multi-head attention network heavily employed by the Transformer translation model consists of h parallel scaled dot-product attentions, where h is the number of attention heads.

The scaled dot-product attention mechanism takes three inputs: the query sequence Q, the key sequence K and the value sequence V.

It first compares each vector in Q with all vectors in K by dot-product computation to generate the attention score matrix S:

$$S = QK^T \tag{1}$$

where ^{T} indicates matrix transposition.

Next, S is scaled and normalized to attention probabilities P:

$$P = \mathrm{softmax}(\frac{S}{\sqrt{d_k}}) \tag{2}$$

where d_k is the dimension of vectors of K.

Finally, the value sequence V is weighted by the attention probabilities P and accumulated as the attention result:

$$\mathrm{Attention}(Q, K, V) = PV \tag{3}$$

* Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 779–785
November 7–11, 2021.

3 Hard Retrieval Attention

3.1 Training

3.1.1 Forward Propagation

When training the hard retrieval attention mechanism, we have to sharpen the attention probability vectors in P (Eq. 2) into one-hot vectors by multinomial sampling:

$$P_{hard} = \text{sharpen}(P) \tag{4}$$

Since P_{hard} only consists of one-hot vectors, the corresponding attention accumulation operation in Eq. 3 can be achieved efficiently by indexing. Specifically, for the jth one-hot vector $\overrightarrow{p}_{hard\ j}$, we first take the index $i_{max\ j}$ of the one-valued element:

$$i_{max\ j} = \text{argmax}(\overrightarrow{p}_{hard\ j}) \tag{5}$$

where argmax returns the index of the largest element of a vector.

Note that in practice the multinomial sampling directly returns $i_{max\ j}$, but we keep P_{hard} here to explain our approach in Section 3.1.2.

Next, we obtain the hard attention result with a simple retrieval operation on V:

$$\text{Attention}_{Hard}(Q, K, V)[j] = V[i_{max\ j}] \tag{6}$$

3.1.2 Gradient Computation

To compute gradients for the hard attention mechanism, we address the non-differentiability of the sharpening operation in the forward propagation by regarding it as a noise process:

$$P_{hard} = P + Noise \tag{7}$$

where $Noise$ stands for the noise introduced by the sharpening operation.

Thus, we pass the gradients P^g_{hard} of P_{hard} directly to P to fix the chain rule for back propagation:

$$P^g = P^g_{hard} \tag{8}$$

where P^g stands for the gradient of P.

The retrieval operation of the value sequence V with the matrix P_{hard} consisting of one-hot vectors is equivalent to the matrix-multiplication between P_{hard} and V. Given the gradient of the hard attention result $Attention^g_{Hard}$, the gradients of P_{hard} and V can be computed as:

$$P^g_{hard} = Attention^g_{Hard} V^T \tag{9}$$

$$V^g[i] = \begin{cases} \sum Attention^g_{Hard}[j], & i = i_{max\ j} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $V^g[i]$ is the ith row of the gradient matrix V^g of V.

For efficiency, we use the retrieval operation again instead of the matrix multiplication for the computation of V^g like in the forward pass.

3.2 Inference

Since the largest attention score in Eq. 1 corresponds to the largest probability after scaling and normalization in Eq. 2, we skip the computation of Eq. 2 and directly take the result of Eq. 1 for the computation of retrieval indexes during inference:

$$i_{max\ j} = \text{argmax}(\overrightarrow{s}_j) \tag{11}$$

where $\overrightarrow{s}_j$ stands for the jth row of S.

Next, we can obtain the hard attention results with $i_{max\ j}$ by the simple retrieval operation presented in Eq. 6.

4 Experiment

We implemented our approach based on the Neutron implementation of the Transformer (Xu and Liu, 2019).

To investigate the impact on translation quality of our approach, we conducted our experiments on the WMT 14 English to German and English to French news translation tasks to compare with Vaswani et al. (2017). We also examined the impact of our approach on the pre-processed data of the WMT 17 news translation tasks for 12 translation directions.

The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as test sets for the WMT 14 English to German and English to French news translation tasks. We used the pre-processed data for WMT 17 news translation tasks.[1]

4.1 Settings

We applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with $32k$ merging operations on both data sets to address the unknown word issue.

[1] http://data.statmt.org/wmt17/translation-task/preprocessed/.

Models	En-De	En-Fr
Transformer Base	27.55	39.54
with Hard Dec Attn	27.73	39.39
Transformer Big	28.63	41.52
with Hard Dec Attn	28.42	41.81

Table 1: Results on WMT 14 En-De and En-Fr.

We only kept sentences with a maximum of 256 subword tokens for training. Training sets were randomly shuffled in every training epoch. We followed Vaswani et al. (2017) for experiment settings. We used a beam size of 4 for decoding with the averaged model of the last 5 checkpoints for the Transformer Base setting and 20 checkpoints for the Transformer Big setting saved with an interval of $1,500$ training steps, and evaluated tokenized case-sensitive BLEU.

Though Zhang et al. (2019); Xu et al. (2020b) suggest using a large batch size which may lead to improved performance, we used a batch size of $25k$ target tokens which was achieved through gradient accumulation of small batches to fairly compare with Vaswani et al. (2017). The training steps for Transformer Base and Transformer Big were $100k$ and $300k$ respectively following Vaswani et al. (2017). Parameters were initialized under the Lipschitz constraint (Xu et al., 2020a).

4.2 Main Results

We first examine the effects of using hard retrieval attention for decoder self- and cross-attention networks (reported in our ablation study results in Table 3) on the WMT 14 English-German and English-French task to compare with Vaswani et al. (2017). Results are shown in Table 1.

Table 1 shows that using the hard retrieval attention mechanism for decoder self- and cross-attention networks achieves comparable performance on both tasks under both Transformer Base and Big settings.

4.3 Efficiency Analysis

Comparing the inference of the standard scaled dot-product attention with the hard retrieval attention, we expect the latter to be faster and more efficient than the first as:

- The operation to find the index of the largest element in the vector (Eq. 11) in the hard retrieval attention is more efficient than the scaling and normalization (Eq. 2) in the scaled dot-product attention.
- The operation to retrieve the corresponding vector in V with indexes in the hard retrieval attention is faster than the matrix multiplication in the standard attention.

We tested the efficiency of our approach by recording the time cost of the operations involved in the two attention mechanisms during the forward propagation on the development set of the WMT 14 English-German news translation task with a single GTX 1080 Ti GPU under the Transformer Base setting. Results are shown in Table 2. Table 2 shows that our hard retrieval attention is much faster than scaled dot-product attention.

Even though overall time consumption is not only determined by attention networks, but also by other parts of the Transformer, we suggest the acceleration with hard retrival attention during inference is still significant, as decoding is performed autoregressively in a token-by-token manner and decoder layers have to be computed for many times during inference, while the linear projections for keys and values of the decoder self-attention and cross-attention heads will be computed once only and cached. This makes attention computation consume a larger part of computation during inference than during training, and makes the acceleration of decoder attention layers significant. Using hard attention also saves the computation of the linear projection layer for values, as it only needs to compute the representations of several attended tokens instead of all tokens of the sequence. We report overall decoding speed in Table 3.

4.4 Ablation Study

We conducted ablation studies on the WMT 14 En-De task.

We first test decoding with the hard retrival attention algorithm but with the converged standard Transformer model. Results are shown in Table 4.

Table 4 shows that performing hard decoding with the softly trained model leads to significant loss in BLEU (-1.21). On the one hand this shows that the decoder attention network does not really need to attend too many tokens, on the other hand it shows the importance of our hard attention training approach that closes the gap between training and inference.

We also study applying the hard retrival attention network as different attention sub-layers of the

Operation		Costs	
Std	Hard	Std	Hard
Compare (Eq. 1)		14.40	
Normalize (Eq. 2)	argmax (Eq. 11)	6.22	2.10
Attend (Eq. 3)	Index (Eq. 6)	12.26	2.93
Total		32.88	19.43

Table 2: Time costs (in seconds) of operations for 1000 iterations. Std: standard scaled dot-product attention.

Hard Attention	BLEU	Decoding Speed (sent/s)	Speed-Up
None	27.55	150.15	1.00
Dec Cross-Attn	27.59	187.69	1.25
Dec Cross- and Self-Attn	27.73	214.50	1.43
Enc Self- and Dec Cross- and Self-Attn	26.60	219.20	1.46

Table 3: Results on using hard attention for different attention mechanisms. Speed is measured on the WMT 14 En-De testset with a beam size of 4.

Train	Decode	BLEU
Soft		27.55
Soft	Hard	26.30
Hard		27.73
Hard	Soft	27.80

Table 4: Effects of hard attention training and decoding on WMT 14 En-De.

decoder or both encoder and decoder. Results are shown in Table 3.

Table 3 shows that applying the hard retrieval attention mechanism to encoder self-attention networks significantly hampers performance. We conjecture potential reasons might be: 1) the encoder might be harder to train than the decoder as its gradients come from cross-attention networks while the decoder receives more direct supervision from the classifier, and the hard attention training approach makes the encoder's training even harder. 2) as the hard retrieval attention only attends one token, the multi-head hard retrieval attention can only attend at most the same number of tokens as the number of attention heads. To achieve optimal results, encoder self-attention may need to attend to more tokens. We leave how to use hard retrieval attention in the encoder for future work. Fortunately, the autoregressive decoder is the major factor in time consumption during decoding, and accelerating decoder layers' computation can significantly speed up inference.

4.5 Testing on WMT 17 Tasks

We further examine the performance of using hard retrieval attention for decoder attention networks on all WMT 17 news translation tasks, using the same setting of the Transformer Base as on the WMT 14 En-De task. Results are shown in Table 5. Table 5 shows that hard retrieval attention is able to match the performance in all tested language pairs in both translation directions, with training sets ranging from 0.2M to 52.02M sentence pairs. The largest performance loss (-0.26 BLEU) is on the Cs-En task.

5 Related Work

Zhang et al. (2018) accelerate the decoder self-attention with the average attention network. Xu et al. (2021) propose to replace the self-attention layer by multi-head highly parallelized LSTM. Kim et al. (2019) investigate knowledge distillation and quantization for faster NMT decoding. Tay et al. (2021) investigate the true importance and contribution of the dot product-based self-attention mechanism on the performance of Transformer models.

Most previous research focuses on efficient modeling of the self-attention mechanism for very long sequences. These are generally not effective on sequences of normal lengths. Dai et al. (2019) introduce the notion of recurrence into deep self-attention network to model very long term dependency efficiently. Ma et al. (2019) combine low rank approximate and parameter sharing to

Lang	Data (M)	En→xx		xx→En	
		Std	Hard	Std	Hard
De	5.85	27.48	27.56	32.89	32.68
Fi	2.63	22.23	22.15	26.15	26.03
Lv	4.46	16.38	16.43	18.12	18.20
Ru	25.00	28.20	28.04	31.52	31.30
Tr	0.20	15.79	15.81	15.58	15.69
Cs	52.02	21.89	21.78	27.62	27.36
Avg.		22.00	21.96	25.31	25.20

Table 5: Results on WMT 17 news translation tasks. xx denotes the language in row headers. None of the differences are statistically significant.

construct a tensorized Transformer. Kitaev et al. (2020) replace dot-product attention by one that uses locality-sensitive hashing and use reversible residual layers instead of the standard residuals. Zhang et al. (2020) propose a dimension-wise attention mechanism to reduce the attention complexity. Katharopoulos et al. (2020) express the self-attention as a linear dot-product of kernel feature maps and make use of the associativity property of matrix products. Wang et al. (2020) approximate the self-attention mechanism by a low-rank matrix. Beltagy et al. (2020) introduce an attention mechanism that scales linearly with sequence length. Child et al. (2019) introduce sparse factorizations of the attention matrix.

On using hard (local) attention for machine translation, Luong et al. (2015) selectively focus on a small window of context smoothed by a Gaussian distribution. For self-attentional sentence encoding, Shen et al. (2018) train hard attention mechanisms which select a subset of tokens via policy gradient. Geng et al. (2020) investigate selective self-attention networks implemented with Gumble-Sigmoid. Sparse attention has been found beneficial for performance (Malaviya et al., 2018; Peters et al., 2019; Correia et al., 2019; Indurthi et al., 2019; Maruf et al., 2019). Our approach learns explicit one-to-one attention for efficiency, pushing such research efforts to the limit.

6 Conclusion

We propose to learn a hard retrieval attention which only attends to one token rather than all tokens. With the one-to-one hard attention matrix, the matrix multiplication between attention probabilities and the value sequence in the standard scaled dot-product attention can be replaced by a simple and efficient retrieval operation.

In our experiments on a wide range of machine translation tasks, we show that using the hard retrieval attention for decoder attention networks can achieve competitive performance while being 1.43 times faster in decoding.

Acknowledgements

We thank our anonymous reviewers for their insightful comments. Hongfei Xu acknowledges the support of China Scholarship Council ([2018]3101, 201807040056). Josef van Genabith and Hongfei Xu are supported by the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP). Deyi Xiong is partially supported by Natural Science Foundation of Tianjin (Grant No. 19JCZDJC31400) and the joint research center between GTCOM and Tianjin University.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Xinwei Geng, Longyue Wang, Xing Wang, Bing Qin, Ting Liu, and Zhaopeng Tu. 2020. How does selective mechanism improve self-attention networks? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2986–2995, Online. Association for Computational Linguistics.

Sathish Reddy Indurthi, Insoo Chung, and Sangha Kim. 2019. Look harder: A neural machine translation model with hard attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3037–3043, Florence, Italy. Association for Computational Linguistics.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. 2019. A tensorized transformer for language modeling. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2232–2242. Curran Associates, Inc.

Chaitanya Malaviya, Pedro Ferreira, and André F. T. Martins. 2018. Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 370–376, Melbourne, Australia. Association for Computational Linguistics.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725. Association for Computational Linguistics.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4345–4352. International Joint Conferences on Artificial Intelligence Organization.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021. Synthesizer: Rethinking self-attention for transformer models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity.

Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. 2020a. Lipschitz constrained parameter initialization for deep transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 397–402, Online. Association for Computational Linguistics.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Meng Zhang. 2021. Multi-head highly parallelized LSTM decoder for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 273–282, Online. Association for Computational Linguistics.

Hongfei Xu, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. 2020b. Dynamically adjusting transformer batch size by monitoring gradient direction change. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3519–3524, Online. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Shuai Zhang, Peng Zhang, Xindian Ma, Junqiu Wei, Ningning Wang, and Qun Liu. 2020. Tensorcoder: Dimension-wise attention via tensor representation for natural language modeling. *CoRR*, abs/2008.01547.