Do Transformers Need Deep Long-Range Memory?

Jack W. Rae
DeepMind & UCL
London, UK
jwrae@google.com

Ali Razavi
DeepMind
London, UK
alirazavi@google.com

Abstract

Deep attention models have advanced the modelling of sequential data across many domains. For language modelling in particular, the Transformer-XL — a Transformer augmented with a long-range memory of past activations — has been shown to be state-of-the-art across a variety of well-studied benchmarks. The Transformer-XL incorporates a long-range memory at every layer of the network, which renders its state to be thousands of times larger than RNN predecessors. However it is unclear whether this is necessary. We perform a set of interventions to show that comparable performance can be obtained with 6X fewer long range memories and better performance can be obtained by limiting the range of attention in lower layers of the network.

1 Introduction

When we read a book, we maintain representations of the characters and events in the text that help us understand the story. We do this with a selective memorisation process; most of the finer details of the text are quickly forgotten and we retain a relatively compact representation of the book's details.

Early models of natural language used recurrent neural networks (RNNs) such as the Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) which emulated this selective memory approach by modelling the past in a compact state vector. The model learns to store relevant information within its state implicitly in order to optimise the task loss.

The LSTM has reigned as a state-of-the-art language model for over two decades since its inception in the '90s (Melis et al., 2017) and is arguably the most ubiquitous neural sequence model. Unlike human memory systems, however, the LSTM struggles to reason over long-range contexts when reading text. This has been observed in multiple contexts. In the carefully curated LAMBADA benchmark (Paperno et al., 2016) which tests language model predictions on sections of book text that have long term structure as decided by human raters, LSTMs completely fail. Namely LSTMs guess the correct word 0% of the time, where humans are considered to be above 70% accuracy. For regular language modelling, Daniluk et al. (2017) observed that an LSTM augmented with attention would rarely attend beyond seven preceding words of context. Samples from LSTMs language models quickly devolve into generic text devoid of an overall theme. This has lead many to wonder whether there is any non-negligible long-range signal in the task of language modelling.

Recently we have seen that deep attention models can draw long-range signal from text, even when the objective is as simple as next-word prediction. With the advent of the Transformer (Vaswani et al., 2017), significant gains in language modelling performance can be obtained by extending the models' attention to thousands of words. The Transformer-XL (Dai et al., 2019), a Transformer variant specialised for long-range sequence modelling via the introduction of a cache of past activations, obtained state-of-the-art results in the four major LM benchmarks — PTB (Mikolov et al., 2010), LM1B (Chelba et al., 2013), Enwik8 (Hutter, 2012), and WikiText (Merity et al., 2016). In the case of the latter two, Dai et al. (2019) showed the model effectively used over one thousand words of context, and the resulting samples reflect a thematic consistency spanning paragraphs. When Transformers are paired with long contexts and a large amount of data, e.g. GPT-2 (Radford et al., 2019) and Megatron (Shoeybi et al., 2019), the resulting samples are remarkable in their long-range consistency and stylistic realism.

However Transformers abandon the compact and selective representation of the past. They store a hidden activation at every time-step (up to a given

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7524–7529
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

attention range) and every layer within the network. This can consume orders of magnitude more space than prior RNN hidden states, or the original text. E.g. a typical state-of-the-art LSTM language model state size may range from 4KB (Rae et al., 2018) to model Wikipedia articles to 64KB (Jozefowicz et al., 2016) to model news — and is never greater than 1MB. Whereas a current state-of-the-art 18-layer Transformer-XL state size for Wikipedia articles is 112MB. The state is so large because a separate memory (e.g. 1600 vectors of size d=1024) is maintained per layer. If this were found to be unnecessary then we can reduce the state's memory considerably.

In this paper we investigate a simple question: can we use short-range attention for the majority of layers in the Transformer and recover the same performance? The hypothesis is that this should be possible, because many steps of reasoning will only involve short-range correlations, i.e. to piece characters together to form words or phrases. We find indeed it is possible. We recover comparable performance for long-range language modelling by using a small fraction (1/6th) of long-range memories to the baseline TransformerXL. Crucially, we find it matters *where* long-range memories are placed in the network. Placing them in the lower layers of the network is ineffective; placing them in the latter layers or interleaved across the network works much better. We show that such a model trains with $2X$ less time and memory, due to the reduction in expensive attention operations.

2 Background

The *Transformer* is a deep neural network for processing sequences (Vaswani et al., 2017), it processes a window of n consecutive inputs $x_{t-n}, \ldots, x_t$ in parallel. At each layer it reasons over time using *multi-head attention* which we will briefly describe. For a given layer l, let $h_t \in \mathbb{R}^{1 \times d}$ be the hidden activation at time t, and $h_{\leq t} \in \mathbb{R}^{t \times d}$ be the preceding activations in the same window. Let k be the number of attention heads, then $Q_i, K_i, V_i \in \mathbb{R}^{d \times \frac{d}{k}}$ are a set of learnable weight matrices which generate *queries*, *keys*, and *values* per attention head. These are defined to be $q_i = h_t Q_i$ as the query, $k_i = h_{\leq t} K_i$ to be the keys, and $v_i = h_{\leq t} V_i$ to be the values for attention head i. The attention head output is defined to be,

$$attn_i(h_t, h_{\leq t}) = \sigma(q_i k_i^T) v_i$$

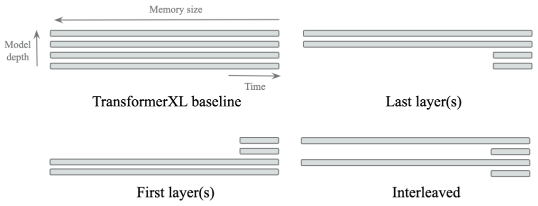

Figure 1: Comparison of arrangement patterns for long-range and short-range memories across the layers of a Transformer. Baseline contains equally long-range memories at every layer.

where $\sigma(\cdot)$ is defined to be the softmax operator. Attention is the linear combination of each attention head, $attn = \sum_{i=1}^{k} W_i \, attn_i$ with a learnable weight.

The attention operation consumes $\mathcal{O}(n)$ compute per step and thus $\mathcal{O}(n^2)$ for the window of inputs at each layer. The *Transformer-XL* (TXL) proposes concatenating the past activations from the same window $h_{\leq t}$ with a memory of size $m \geq n$ of past activations from the preceding windows of inputs (Dai et al., 2019). This results in an attention cost of $\mathcal{O}(n(n+m))$ which can be significantly cheaper than processing all $n+m$ inputs in parallel, which would require $\mathcal{O}((n+m)^2)$. The TXL's memory can be considered to be a state, alike to an RNN. However it requires a considerable space: $l \times m \times d$. For character-level language modelling Dai et al. (2019) use a 24-layer model on Enwik8, with memory size $m = 3800$, and hidden size $d = 1024$; this consumes 356MB at single precision. In contrast, the average article size is 8KB.

3 Experiments

We investigate whether the Transformer-XL can perform comparably with fewer long-range memory (LRM) layers on the two prominent long-range language modelling benchmarks, Enwik8 and WikiText-103.

3.1 Interventions

We perform intervention experiments where we replace the long-range memory, for a given layer, with a short-range memory (SRM) of size $m_s = 128$ for a subset of layers. We choose $m_s = 128$ because the TPUv3 contains a 128x128 matrix multiply unit, and any smaller size (other than zero) is padded up to 128. Thus it is a reasonable small size. We chose $m_s > 0$ such that the oldest activations have some context. Because we only modify the

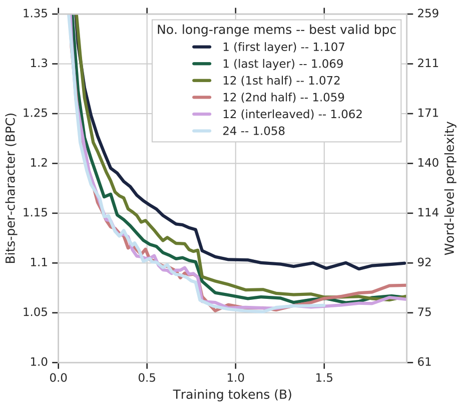

Figure 2: Enwik8 learning curves for varying long-range memory arrangements and no. layers. BPC over the first 500K characters from validation.

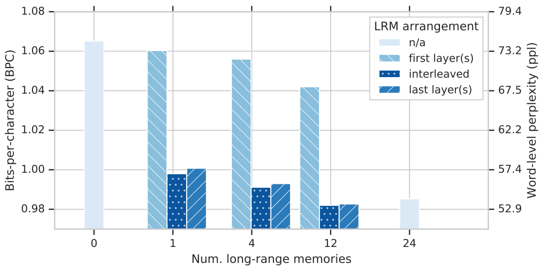

Figure 3: Enwik8 test performance over a varying number of long-range memories and arrangement patterns. Lower is better. Model: 24-layer Transformer-XL, evaluation long-range memory size: 6000 (trained with 2304) and short-range memories size: 128.

memory sizes of the model, which are independent of parameter count, **the number of model parameters is always held constant** (277M for Enwik8 and 257M for WikiText-103).

We consider a model with a varying number of LRMs from l (the number of layers in the network, i.e. the usual case) to a range of fewer values, $\frac{l}{2}$, $\frac{l}{6}$, 1, and 0. We also consider where the LRMs should be arranged within the network; considering (i) interleaved with equal spacing, (ii) the first layer(s) of the network, and (iii) the latter layer(s) of the network. This is displayed visually in Figure 1.

3.2 Model Setup

Aside from memory configurations, we use an identical model setup to Dai et al. (2019). During training we periodically evaluate on the validation set to choose an early stopping criterion. In the case of Enwik8 we periodically evaluate on the first 500K characters of the validation set to speed up model evaluation. We train all models with an overall batch size of 32, using 16 TPUv3 chips running synchronously. We use a window size of $n = 384$, a long-range memory (LRM) size of $m = 2304$. At test-time we extend the LRM size to $m = 6000$, chosen from a sweep over the validation set.

4 Results

We plot the **Enwik8** learning curves for a subset of layer variants in Figure 2. The worst-performing, is the variant with a single long-term memory at the lowest layer (black curve). However perhaps more surprisingly, we see a model with 12 LRMs at the lower layers of the network is actually *worse* than a model with a single LRM on the final layer (dark green). We then see that the full TXL with 24 LRMs is seemingly identical to the 12 LRM models, with either LRMs interleaved across the whole model or LRMs placed in the final 12 layers. Note, we were not able to run these models with multiple seeds per hyper-parameter configuration - but we do generally find language models optimise consistently (e.g. unlike deep reinforcement learning models).

We show the final test performance in bits-per-character (BPC) alongside the corresponding word-level perplexity for models with a varying number of LRMs and LRM arrangements in Figure 3. Position clearly matters, if we place long-range memories in the first layers then performance is significantly worse. We hypothesise that this is because it is better to build up representations with local context before exploiting long-range correlations. For example, we need to piece together characters into an identified named entity (say) before we should query thousands of time-steps back for its prior occurrence.

We followed-up by running an additional arrangement of only placing LRMs in the *middle* layers and found this to be worse than interleaved or final (1.01bpc for 4 long-range memories) which shows there is significant benefit to having some long-range memories in the higher layers.

Crucially, we are able to match (and slightly exceed) the full model's test performance with 12 LRMs, and even a model with 4 LRMs is very close (**0.9846** w/ 24 vs **0.9916** w/ 4 interleaved). It is worth noting that our TXL baseline actually outperforms the published version on Enwik8: 0.985 BPC (ours) vs 0.993 (Dai et al., 2019), which provides credence to the quality of the experimental setup.

Num. LRMs	Memory (GB)	Time / token (us)
24	3.4	405
12	2.8	273
4	1.1	191
1	0.50	155
0	0.20	143

Table 1: Profiling a 24-layer TXL training on Enwik8.

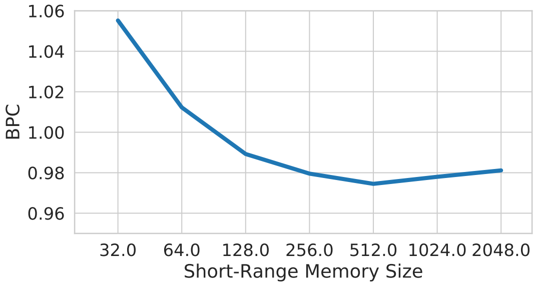

Figure 4: Enwik8 test performance for varying short-range memory length (at both train and test). TransformerXL model uses 4 interleaved long-range memories (trained 2304, tested 6000) and 20 short-range memory layers.

We also inspect word-level language modelling on **WikiText-103**, using the same 18-layer TransformerXL parameters (Dai et al., 2019). We obtain a baseline test perplexity of 18.3 (matching the published value), and obtain **18.4** and **18.6** for interleaved and last-layer spacing respectively when using $l/6$ (i.e. 3) LRMs. We also try placing 3 LRMs on the first three layers and obtain 20.1 perplexity. We remark that (i) long-range memory is important for a significant improvement in performance, (ii) it is better to not place LRMs in the shallow layers, and (iii) it is not necessary to have as many long-range memories as model-layers for comparable modelling performance.

4.1 Performance

We show the performance of training the Transformer-XL with a varying number of LRMs for the Enwik8 architecture in Table 1. This shows the latency (per input token) and peak activation memory consumption during a training iteration on Enwik8 for a range of long-range memory layers. We see the reduction of long-range memories from 24 layers to 4 layers cuts the activation peak memory by 3X. Thus it can be a worthwhile and simple performance improvement.

4.2 Varying Short-Range Memory

In the preceding experiments we fix the short-range memory (SRM) length to 128 and vary the frequency and arrangement of long-range memory layers. We now consider varying the length of SRM for an architecture with $\frac{l}{6}$ long-range memories to determine whether this impacts modelling performance.

We train (and evaluate) the model with twenty SRM lengths from 32-2048, and incorporate four interleaved LRM layers (trained at 2304, evaluated at 6000). The results are plotted in Figure 4. Shortening the memory size to less than 128 provides no speedup for our TPU training setup, as matrices are multiplied in 128x128 blocks, however it incurs a drop in modelling performance. Furthermore increasing the memory size beyond 512 further slows the model down and reduces modelling performance. We see an optimal SRM length is around 512 steps which obtains **0.974**BPC on Enwik8 — a non-trivial performance boost over the 0.99BPC TransformerXL baseline. Thus we conclude that limiting the range of attention can not only speed up the model but improve performance.

5 Related Work

There have been several recent works exploring deep sequence models with a small attention window per layer. Wu et al. (2019) proposed the *dynamic convolution*, where the model directly produces a set of weights over a sequence in memory and then combines them with a convolution. The attention window is thus restricted to the convolution kernel size — a couple of words. Wu et al. (2019) show comparable performance to the Transformer at sentence-level machine translation. However they do not investigate longer-context applications.

Rae et al. (2019) propose shortening the range of attention for Transformers by compressing the distant past. They find the first layers of the model are the most compressible, and obtain state-of-the-art in several long-range language model benchmarks (WikiText-103 and Enwik8). However they do not consider restricting the range of attention for a subset of layers to save compute and space. Sukhbaatar et al. (2019) propose an *adaptive attention* scheme for the TransformerXL where the model can learn to modulate the size of its attention window per attention head. They observe the neural network converges to using smaller attention spans for lower layers in the network, which adds additional evidence to the finding that long-range memories are not useful in these lower layers. Because Sukhbaatar et al. (2019) place the range of attention in the optimisation problem it is very flexible. In this study we promote interpretability by making a set of direct interventions to the memory size across layers. This does result in less generality, as we explicitly create two types of attention ranges, where adaptive attention can select many. However ultimately the two approaches of generality and interpretability complement one another.

(Fan et al., 2020) show that one can train a transformer by having all layers attend to a single memory that is the linear combination of all layers' memories. Thus at training all layers' memories are maintained, but at evaluation or generation time there can be a single memory. This gives evidence that we do not need to store many separate representations for long-range memory to perform well at test time, but the approach does require storing them during training — and incurs significant slowdown to the model.

6 Discussion

We explore a set of interventions to the Transformer-XL's architecture that are very simple to implement, i.e. a few lines of code, but shed light on the fundamental workings of the model when modelling long sequences of text. In our set of interventions, we only modify the flow of information within the network, versus the number of trainable parameters. Thus we do not have confounding factors of varying network capacity.

Our finding is that we do not need long-range memories at every layer of the network. Comparable performance can be obtained with a fraction (1/6th) of long-range memories if they are spaced equally across the network, or in the latter layers. We hypothesise this is because modelling long-range correlations is best done when representations are first formed from short-range correlations. We also find a real performance drop using a single long-range memory, proving long-range dependency is not superfluous to the task.

This study has implications for practitioners interested in speeding up deep Transformer-XL models. There have been a number of long-range transformer variants published in the past year (Lample et al., 2019; Rae et al., 2019; Roy et al., 2020; Kitaev et al., 2020) which aim to extend the range of attention via sparsity or compression. However these models maintain the use of uniform memory capacity for each layer. Here we show that long-range attention does not need to be scaled for every layer, and thus these architectures can be further sped-up with this observation.

This study also has implications for researchers using a single long-range memory, which has typically been the approach in traditional RNN + attention systems. For example, the Differentiable Neural Computer (Graves et al., 2016) and recent memory-augmented agents for reinforcement learning, which utilise a distinct working memory with a single long-range episodic memory (Fortunato et al., 2019). Perhaps performance could be improved by adding additional layers of *episodic* memories.

The practice of storing deep long-range memories is not scalable if we wish for neural networks to have the kinds of large-horizon reasoning that humans possess. We believe the solution of maintaining a small number of long-range memories is a step towards tractable lifelong memory.

References

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Michal Daniluk, Tim Rocktaschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling. *Proceedings of International Conference on Learning Representations (ICLR)*.

Angela Fan, Thibaut Lavril, Edouard Grave, Armand Joulin, and Sainbayar Sukhbaatar. 2020. Accessing higher-level representations in sequential transformers with feedback memory. *arXiv preprint arXiv:2002.09402*.

Meire Fortunato, Melissa Tan, Ryan Faulkner, Steven Hansen, Adrià Puigdomènech Badia, Gavin Buttimore, Charles Deck, Joel Z Leibo, and Charles Blundell. 2019. Generalization of reinforcement learners with working and episodic memory. In *Advances in Neural Information Processing Systems*, pages 12448–12457.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al.

2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Marcus Hutter. 2012. The human knowledge compression contest. *URL http://prize. hutter1. net*, 6.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. *arXiv preprint arXiv:1907.05242*.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2017. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Jack W Rae, Chris Dyer, Peter Dayan, and Timothy P Lillicrap. 2018. Fast parametric learning with activation memorization. *arXiv preprint arXiv:1803.10049*.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2020. Efficient content-based sparse attention with routing transformers. *arXiv preprint arXiv:2003.05997*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*.

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.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*.