
Efficient Content-Based Sparse Attention with Routing Transformers

Aurko Roy^{*1} Mohammad Saffar^{*1} Ashish Vaswani¹ David Grangier¹

Abstract

Self-attention has recently been adopted for a wide range of sequence modeling problems. Despite its effectiveness, self-attention suffers from quadratic compute and memory requirements with respect to sequence length. Successful approaches to reduce this complexity focused on attending to local sliding windows or a small set of locations *independent* of content. Our work proposes to learn dynamic sparse attention patterns that avoid allocating computation and memory to attend to content unrelated to the query of interest. This work builds upon two lines of research: it combines the modeling flexibility of prior work on *content-based* sparse attention with the efficiency gains from approaches based on *local, temporal* sparse attention. Our model, the Routing Transformer, endows self-attention with a sparse routing module based on online k -means while reducing the overall complexity of attention to $O(n^{1.5}d)$ from $O(n^2d)$ for sequence length n and hidden dimension d . We show that our model outperforms comparable sparse attention models on language modeling on Wikitext-103 (15.8 vs 18.3 perplexity) as well as on image generation on ImageNet-64 (3.43 vs 3.44 bits/dim) while using fewer self-attention layers.

1. Introduction

Generative models of sequences have witnessed rapid progress driven by the application of attention to neural networks. In particular, (Bahdanau et al., 2014; Cho et al., 2014; Vaswani et al., 2017) relied on attention to drastically improve the state-of-the art in machine translation. Subsequent research (Radford et al., 2018; Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019) demonstrated the power of self-attention in learning

powerful representations of language to address several natural language processing tasks. Self-attention also brought impressive progress for generative modeling outside of language, e.g. image (Parmar et al., 2018; Menick and Kalchbrenner, 2018; Child et al., 2019) and music generation (Huang et al., 2018; Child et al., 2019).

Self-attention operates over sequences in a step-wise manner: at every time-step, attention assigns an *attention weight* to each previous input element (representation of past time-steps) and uses these weights to compute the representation of the current time-step as a weighted sum of the past input elements (Vaswani et al., 2017). Self-attention (Shaw et al., 2018) is a particular case of attention (Bahdanau et al., 2014; Chorowski et al., 2015; Luong et al., 2015).

Self-attention is commonly used in auto-regressive generative models. These models generate observations step-by-step, modeling the probability of the next symbol given the previously generated ones. At every time step, self-attentive generative models can directly focus on any part of the previous context. In contrast, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have direct interactions with only a local neighborhood of context around the current time step.

This advantage however comes at a price: unlike recurrent networks or convolution networks, the time and space complexity of self-attention is quadratic in n , the length of the sequence. Specifically, for every position $i \leq n$, self-attention computes weights for its whole context of length i , which induces a complexity of $\sum_{i \leq n} i = n(n-1)/2$. This makes it difficult to scale attention based models to modeling long sequences. However, long sequences are the norm in many domains, including music, image, speech, video generation and document level machine translation.

Therefore, an important research direction is to investigate sparse and memory efficient forms of attention in order to scale to tasks with long sequence lengths. Previous work has proposed *data independent* or fixed sparsity patterns bounding temporal dependencies, such as local or strided attention. At each time step, the model

^{*}Equal contribution ¹Google Research. Correspondence to: Aurko Roy <aurkor@google.com>.

attends only to a fix number of time steps in the past (Child et al., 2019). Extensions to local attention have suggested learning the length of the temporal sparsity for each attention module in the network (Sukhbaatar et al., 2019). These strategies draw their inspiration from RNNs and CNNs and bound their complexity by attending only to representations summarizing a *local* neighborhood of the current time step. Their attention matrices (matrices containing the attention weights for every pair of previous, current time-step) are natively sparse and requires instantiating only non-zero entries. While these approaches have achieved good results, fixing the sparsity pattern of a content based mechanism such as self-attention can limit its ability to pool in information from large contexts.

As an alternative to local attention, (Correia et al., 2019) considers content-based sparsity, an approach allowing for arbitrary sparsity patterns. This formulation however does require instantiating a full dense attention matrix prior to sparsification through variants of L_0 -sparsity or sparsemax approximations (Blondel et al., 2019).

The present work builds upon these two lines of research and proposes to retain the modeling flexibility of content-based sparse attention while leveraging the efficiency of natively sparse attention matrices. Our formulation avoids sparsemax variants and relies on clustering of attention instead. Each attention module considers a clustering of the space: the current time-step only attends to context belonging to the same cluster. In other word, the current time-step query is *routed* to a limited number of context through its cluster assignment. This strategy draws inspiration from the application of k -means clustering to Non-negative Matrix Factorization (NMF) (Lee and Seung, 2001; Ding et al., 2005; Kim and Park, 2008), which is relevant to the sparsification of non-negative matrices like attention matrices.

Our proposed model, Routing Transformer, combines our efficient clustered-based sparse attention with classical local attention to reach excellent performance both for language and image generation. These results are obtained without the need to maintain attention matrices larger than batch length which is the case with the segment level recurrence mechanism used in (Dai et al., 2019; Sukhbaatar et al., 2019). We present experimental results on language modeling (Wikitext-103 and enwik-8) and unconditional image generation (ImageNet-64). Routing Transformer sets new state-of-the-art while having comparable or fewer number of self-attention layers and heads, both on Wikitext-103 (15.8 vs 18.3 perplexity) and on

ImageNet-64 (3.43 vs 3.44 bits/dim). We also report competitive results on enwik-8 (0.99 vs 0.98 perplexity).

2. Related Work

Attention with Temporal Sparsity: Research on efficient attention neural models parallels the advent of attention-based architectures. In the context of speech recognition, (Jaitly et al., 2015) proposed the Neural Transducer which segments sequences in non-overlapping chunks and attention is performed in each chunk independently. Limiting attention to a fixed temporal context around the current prediction has also been explored in (Chorowski et al., 2015), while (Chiu and Raffel, 2017) dynamically segment the sequence into variable sized-chunks.

Hierarchical attention strategies have also been explored: the model first considers which part of the inputs should be attended to before computing full attention in a contiguous neighborhood of the selected area (Gregor et al., 2015; Xu et al., 2015; Luong et al., 2015). Later, hierarchical attention has been simplified by (Liu et al., 2018) that alternates coarse layers (attending to the whole sequence at a lower temporal resolution) with local layers (attending to a neighborhood of the current prediction).

This alternating strategy is also employed by (Child et al., 2019), which introduces bounded and strided attention, i.e. attending to a fixed context in the past at a subsampled temporal resolution. This work formalizes such a strategy using a sparse attention formalism, showing how it relates to full attention with a specific sparsity pattern in the attention matrix. It shows that sparse attention is sufficient to get state-of-the-art results in modeling long sequences over language modeling, image generation and music generation. (Sukhbaatar et al., 2019) builds upon this work and shows that it is possible to obtain further sparsity by letting the model learn the length of the temporal context for each attention module. This work also makes use of the attention cache introduced in (Dai et al., 2019), a memory mechanism to train models over temporal contexts which extend beyond the length of the training batches.

Attention with Content-Based Sparsity: The above work mainly relies on two efficient ideas: attending to less elements by only considering a fixed bounded local context in the past, and attending to less elements by decreasing the temporal resolution of context. These ideas do not allow arbitrary sparsity patterns in attention matrices. Content-based sparse at-

tion has been introduced to allow for richer patterns and more expressive models. (Martins and Kreutzer, 2017; Malaviya et al., 2018) propose to compute attention weights with variants of sparsemax. (Correia et al., 2019) generalizes this approach to every layer in a Transformer using entmax which allows for more efficient inference. This line of work allows for learning arbitrary sparsity attention patterns from data, based on the content of the current query and past context. However, sparsity here cannot be leveraged to improve space and time complexity since sparsemax/entmax formulations require instantiating the full attention matrix prior to sparsification. This is a drawback compared to temporal sparsity approaches. Our work is motivated by bridging this gap and allows for arbitrary sparsity patterns while avoiding to instantiate non-zero entries of attention matrices.

Sparse Computation beyond Attention: Learning models with sparse representations/activations for saving time and computation has addressed in the past in various context. Previous work often refers to this goal as *gating* for conditional computation. Gating techniques relying on sampling and straight-through gradient estimators are common (Bengio et al., 2013; Eigen et al., 2013; Cho and Bengio, 2014). Conditional computation can also be addressed with reinforcement learning (Denoyer and Gallinari, 2014; Indurthi et al., 2019). Memory augmented neural networks with sparse reads and writes have also been proposed in (Rae et al., 2016) as a way to scale Neural Turing Machines (Graves et al., 2014). In the domain of language modeling, a related work is the sparsely gated Mixture-of-experts (MOE) (Shazeer et al., 2017) where sparsity is induced by *experts* and a trainable gating network controls the routing strategy to each sub-network. Another related work is (Lample et al., 2019) who use product quantization based key-value lookups to replace the feed forward network in the Transformer. Our work differs from theirs in that we make use of dynamic key-value pairs to infer sparsity patterns, while their key-value pairs are the same across examples.

3. Self-Attentive Auto-regressive Sequence Modeling

Auto-regressive sequence models decompose the probability of a sequence $\mathbf{x} = (x_1, \dots, x_n)$ as

$$p(\mathbf{x}) = \prod_{i=1}^n p_{\theta}(x_{i+1}|x_{\leq i}). \quad (1)$$

In neural models, the conditional distribution $p_{\theta}(x_{i+1}|x_{\leq i})$ is modeled by a neural network with learned parameters θ and these parameters are typi-

cally learned to maximize the likelihood of the training data. In particular, Transformer architectures have shown to reach state-of-the-art accuracy in several domains, including language modeling (Vaswani et al., 2017; Radford et al., 2018), image generation (Parmar et al., 2018) and music generation (Huang et al., 2018). Transformer models compose a series of attention modules. Each module refines the input representation by taking a weighted average of the representations from the previous modules.

For every module, the input representation is a sequence of n vectors $\mathbf{x} = (x_1, \dots, x_n)$ from a continuous space of dimension d . Thus one may actually treat the input sequence as a $n \times d$ matrix X . A self-attention layer operates on this representation. It first applies three linear projections,

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \quad (2)$$

where Q, K and V are referred to as *keys*, *queries* and *values*, while W_Q, W_K, W_V are learned projection matrices.

The key and the query matrices determine the $n \times n$ attention matrix $A = \text{softmax}(QK^{\top})$, where the softmax operator over matrices denotes that the softmax function has been applied to each row. A may be interpreted as a matrix of weights in $[0, 1]$ where A_{ij} denotes how much query position i at the next layer must pay attention to key position j at the previous layer. In the case of self-attention for auto-regressive models, queries attend only over keys from previous time-steps, i.e.

$$A = \text{softmax}(\text{ltr}(QK^{\top})) \quad (3)$$

where ltr denotes the lower triangular operator. Given the attention matrix A , the next layer representation X' is computed simply as AV . In summary,

$$X'_i = \sum_{j \leq i}^n A_{ij} V_j, \quad (4)$$

In practice, Transformer (Vaswani et al., 2017) adds several extensions to this basic self-attention mechanism. In particular, the result X' of performing self-attention is scaled by $1/\sqrt{d}$. Moreover, each layer relies on multiple attention *heads*, i.e. each layer performs multiple projections onto triplet (queries, keys, values) and attention is performed for each head. The attention results from all heads are then concatenated. This strategy allows each head to specialize on different aspects of the input sequence. In addition, Transformer further processes the result of attention through a learnable non-linear transformation (multi-layer perceptron, mlp)

followed by a residual connection and a normalization step, i.e.

$$X' = \text{layernorm}(X' + X) \quad (5)$$

$$X'' = \text{layernorm}(\text{mlp}(X') + X), \quad (6)$$

where `layernorm` denotes the parameterized normalization step from (Ba et al., 2016). A full Transformer model is therefore a chain of attention modules (Eq. 6) preceded by an embedding module (learnable representation for symbols and their positions) and followed by a logistic classification module (learnable linear classifier to predict the next symbol).

Our work is interested in the application of the Transformer to long sequences, a challenging problem since space and time complexity of attention is quadratic in sequence length n . We describe various approaches to sparse attention including ours in the next section.

4. Efficient Content-Dependent Sparse Attention

Attention-based models can be problematic for long sequences. For a sequence of length n , the full attention matrix A , as introduced in Section 3, is $n \times n$ -dimensional and can be prohibitive to instantiate. This motivates sparse attention models, i.e. models relying on attention matrices which have a majority of zero entries.

For each query, a sparse attention model defines a set of keys which can be attended to. In the following, we introduce the set S_i as the set of key positions that the query at position i can attend to, i.e.

$$X'_i = \sum_{j \in S_i} A_{ij} V_j. \quad (7)$$

For example, classical causal self attention can attend to every key prior to the current query, which translates to $S_i = \{j \mid j < i\}$. Most previous work on attention sparsity defined such sets purely based on positions, independently of actual query and key vectors. For example, local attention (Luong et al., 2015) considers attending only to a k -long time window prior to the current query, $S_i = \{j \mid i - k \leq j < i\}$. (Child et al., 2019) propose block sparse attention where half the heads perform local attention, and half the heads perform *strided attention* given by $S_i = \{j \mid i - j \pmod{k} = 0, j < i\}$. (Sukhbaatar et al., 2019) is also a variant of local attention where the cardinality of $|S_i|$ is learned from data with an L_1 penalty to trade-off sparsity with modeling accuracy.

These *local* attention sparsity variants are effective in practice since correlation between observations natu-

rally decrease with time for many problems. In our experiments, we actually find that local attention is a surprisingly strong baseline in both image generation and language modeling: for e.g., a scaled up Image-Transformer (Parmar et al., 2018) gets 3.48 bits/dim compared to the 3.44 bits/dim reported in (Child et al., 2019). Similarly, scaled up versions of Transformer with local attention and the relative positional encoding scheme of (Shaw et al., 2018) are able to get 19.8 perplexity on `Wikitext-103` and 1.10 bits per byte on `enwik-8`, while the state-of-the-art results using Transformer-XL (Dai et al., 2019) are 18.3 and 0.99 respectively. From an efficiency perspective, local attention is also interesting since sparsity patterns are regular, contiguous in memory and known in advance.

In this work, however, we are interested in a more generic formulation of attention sparsity and would like the sparsity pattern to be informed by the data, i.e., $S = f(\mathbf{x})$. This approach has several modeling advantages: it can accommodate data without a clear ordering over observations. For temporal data, it can also discover patterns with greater sparsity if some types of queries have a longer lasting effect on future observations than others. Content-based sparse attention should however be carefully implemented if we need to avoid instantiating full attention matrices at any point in time. For instance, (Correia et al., 2019) infer sparsity from data but their formulation instantiates a full attention matrix before finding its sparse counterpart. Next section explains how a natively sparse approach can actually be devised inspired by non-negative matrix factorization (NMF).

4.1. Routing Attention with Clustering

Our strategy follows the motivation we delineated in the previous section: we model sparse attention matrices with a low rank sparsity patterns relying on k -means clustering. Our strategy first assigns queries and keys to clusters. Then only queries and keys from the same cluster are considered for attention.

Precisely, our model projects keys K and queries Q into a routing matrix $R \in \mathbb{R}^{n \times d}$ as follows

$$R = [Q, K] \begin{bmatrix} W_R \\ W_R \end{bmatrix} \quad (8)$$

where W_R is a fixed random orthonormal $d \times d$ routing projection matrix. The vectors of R undergo k -means clustering in order to factorize the full attention matrix. The clustering parameters are the centroid vectors $(\mu_1, \dots, \mu_k) \in \mathbb{R}^{k \times d}$. These parameters are model parameters shared across sequences. There are learned online along with the rest of the parameters, as de-

lineated in (Bottou and Bengio, 1995). Once cluster membership for each position i in the sequence is determined, we denote with C_i the cluster corresponding to the routing vector R_i . This allows us to define our sparse attention strategy as

$$X'_i = \sum_{j \in C_i, j \leq i} A_{ij} V_j \quad (9)$$

where C_i denotes the cluster of the vector R_i . In summary, queries are routed to keys belonging to the same cluster. Therefore, our attention sparsity pattern is of rank k , i.e. FG^\top where F and G are binary matrices denoting cluster memberships of queries and keys respectively. Note that since we route both queries and keys via the routing matrix R , it follows that $F = G$. It is important to note that this low rank property only concerns the sparsity pattern, while the resulting attention matrix $\text{ltr}(FG^\top * A) = \text{ltr}(FF^\top * A)$ can however be of higher rank ($*$ denotes element-wise product).

We work with keys and values which are unitary vectors, projecting them onto the unit ball, immediately before computing them. Note that performing k -means on unitary vectors is equivalent to the *spherical* k -means algorithm. This differentiable normalization (Ba et al., 2016) is useful to link cluster memberships with proximity of queries and keys, as outlined below. We also assume that the projection matrices W_Q and W_K used to infer queries and keys are *close* to each other in max norm. More precisely, we assume the existence of a $\delta \geq 0$ such that $\|W_Q - W_K\|_\infty \leq \delta$. This can be enforced by adding an auxiliary loss or by explicitly setting $W_Q = W_K$. This assumption implies that for any vector $u \in \mathbb{R}^d$ it holds that:

$$uW_Q \leq uW_K + \delta \mathbf{1} \|u\|_\infty, \quad (10)$$

where the inequality is entry-wise and $\mathbf{1}$ is the vector in \mathbb{R}^d with all 1's. In this case we first show that for any pair i, j the queries and keys satisfy the following:

$$\begin{aligned} & (Q_i - K_j)^\top (Q_j - K_i) \\ &= (X_i W_Q - X_j W_K)^\top (X_j W_Q - X_i W_K) \\ &\leq (\delta \mathbf{1} \|X_i\|_\infty + (X_i - X_j) W_K)^\top (\delta \mathbf{1} \|X_j\|_\infty - (X_i - X_j) W_K) \\ &\leq \delta^2 \|\mathbf{1}\|^2 \max\{\|X_i\|_\infty, \|X_j\|_\infty\}^2 - \|(X_i - X_j) W_K\|^2. \end{aligned}$$

Therefore, for small enough δ , we get that $(Q_i - Q_j)^\top (Q_j - K_i) \gtrsim 0$ and so we get

$$\|R_i - R_j\|^2 \quad (11)$$

$$= \|W_R(Q_i + K_i) - W_R(Q_j + K_j)\|^2 \quad (12)$$

$$\gtrsim \|W_R\|^2 \left(\|Q_i - K_j\|^2 + \|Q_j - K_i\|^2 \right) \quad (13)$$

$$= 4 - 2(Q_i^\top K_j + Q_j^\top K_i). \quad (14)$$

Note that Equation 14 follows since W_R is a distance preserving transform. Thus, we have the following implication: $\|R_i - R_j\| \leq \varepsilon \Rightarrow Q_i^\top K_j + Q_j^\top K_i \geq 2 - \varepsilon^2/2$. This means that, $\|R_i - R_j\| \leq \varepsilon \Rightarrow Q_i^\top K_j \geq 1 - \varepsilon^2/4$. Therefore, when two time steps $i > j$ are assigned the same cluster due to a small $\|R_i - R_j\|$ distance, it also means that their attention weight $Q_i^\top K_j$ is high. This analysis shows that our clustering routing strategy preserves large attention weights as non-zero entries.

Since, we route attention via the matrix R we dub our model *Routing Transformer*. A visualization of the attention scheme and its comparison to local and strided attention is given in Figure 1. The computational complexity of this variant of sparse attention is $O(nkd + n^2d/k)$. Cluster assignments correspond to the first term, i.e. it compares n routing vectors to all k centroids in a space of size d . Query/key dot products corresponds to the second term, i.e. assuming balanced clusters, each of the n queries is compared to n/k in its cluster through a dot product of dimension d . Therefore the optimal choice of k is \sqrt{n} as in (Child et al., 2019), thereby reducing overall memory and computational cost to $O(n^{1.5}d)$ instead of $O(n^2d)$ (Vaswani et al., 2017).

In practice, we apply regular online k -means to train the cluster centroids. However, in order to infer balanced routing patterns, we define the sets C_i to be of equal size roughly $n/k \sim \sqrt{n}$, i.e. for every centroid μ_i we sort tokens by distance to μ_i and cluster membership is determined by this threshold (top- k). This strategy is simple and efficient. In particular, it guarantees that all clusters have the same size, which is extremely interesting in terms of computational efficiency on parallel hardware like graphic cards. As a downside, this assignment does not guarantee that each point belongs to a single cluster. In the future, we want to investigate using balanced variants of k -means (Banerjee and Ghosh, 2004; Malinen and Fränti, 2014) which is not common in an online setting.

5. Experiments

We evaluate our sparse attention model on various generative modeling tasks including text and image generation. The following sections report our results on Wikitext-103 (Merity et al., 2016), enwik-8 (Mahlone, 2011), as well as ImageNet-64. We find that local attention is a surprisingly strong baseline and that our Routing Transformer outperforms Transformer-XL (Dai et al., 2019) and the Sparse Transformer model of (Child et al., 2019) on all tasks. In all our models, we allocate half the heads to do local attention and

the other half to route attention as in Equation 9. We use the Adam optimizer (Kingma and Ba, 2014) with learning rate 2×10^{-4} with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ following the learning rate schedule described in (Vaswani et al., 2017). We present unconditional samples from our model as a part of the supplementary material.

5.1. Wikitext-103

Wikitext-103 (Merity et al., 2016) is a large public benchmark data-set for testing long term dependencies in word-level language models. It contains over 100 million tokens from 28K articles extracted from Wikipedia with an average of 3.6K tokens per article, which makes it a reference data-set to model long-term textual dependencies. We train a 10 layer Routing Transformer with 16 heads using the relative position encoding of (Shaw et al., 2018) and with attention and ReLU dropout rate of 0.3 each. For routing attention as in Section 4.1 we choose $k = 16$ and attention window to be 256 during both training and evaluation. We describe our results in Table 2 and compare it to other recent work on sparse or recurrent attention such as Adaptive Inputs (Baevski and Auli, 2018) and TransformerXL (Dai et al., 2019) as well as a local attention with relative position encoding baseline (Huang et al., 2018). We find that local attention is a great inductive bias for sparse attention and is better than the adaptive methods proposed in (Baevski and Auli, 2018; Sukhbaatar et al., 2019). Moreover, our Routing Transformer model is able to get a test perplexity of 15.8 improving on the 18.3 obtained by TransformerXL (Dai et al., 2019) while having fewer self-attention layers, and without the need for segment level recurrence.

5.2. enwik-8

The enwik-8 (Mahoney, 2011) is a data-set to benchmark text compression algorithms in the context of the Hutter prize. This data-set consists of the first 100M bytes of unprocessed Wikipedia. It is typically used to evaluate character-level language models. Similar to the prior work of (Dai et al., 2019; Child et al., 2019) we use a sequence length $n = 8192$ and benchmark our results against various baselines including local attention. We train a 24 layer model with 8 attention heads with an attention and ReLU dropout rate of 0.4 each and using the relative position encoding of (Shaw et al., 2018). For routing attention as in Section 4.1 we set $k = 32$ and attention window 256. We report perplexity of 0.99 like TransformerXL and Sparse Transformer, slightly under 0.98 from Adaptive Transformer.

5.3. ImageNet 64×64

In order to evaluate the ability of our model to capture long term dependencies on a modality other than text, we report results on the ImageNet 64×64 data-set as used in (Child et al., 2019). For auto-regressive image generation, this data-set consists of images of $64 \times 64 \times 3$ bytes represented as long sequences of length 12,288 presented in raster scan, red-green-blue order. We train a 24 layer model with 16 attention heads, with half the heads performing local attention, and the other half routing attention as in Section 3. For routing attention we set $k = 8$, attention window 2048, batch size 1 and train our model for roughly 70 epochs as in (Child et al., 2019). We compare our model to a scaled-up ImageTransformer model with local attention (Parmar et al., 2018) and the SparseTransformer model of (Child et al., 2019).

We find that local attention (Parmar et al., 2018) is a strong baseline for image generation, obtaining 3.48 bits/dim when scaled up to 24 layers and 16 heads, compared to later work like Sub-scale Pixel Networks (SPN) (Menick and Kalchbrenner, 2018). Our Routing Transformer model achieves a performance of 3.425 bits/dim (see Table 1) compared to the previous state-of-the-art of 3.437 bits/dim (Child et al., 2019), thereby showing the advantage of the content based sparsity formulation of Section 4.1.

6. Analysis

We evaluate the difference in attention patterns between local and routed attention and compute the Jensen-Shannon divergence between local attention and routed attention for a random subset of heads in our network on the Wikitext-103 data-set. The divergence is computed over the entire sequence length of 4096. We average over 10 runs and report means and standard deviations of the JSD in Table 4. Note that the JSD is always non-negative and is upper-bounded by 0.6931 when computed using the natural logarithm. We observe that the divergence between the different local heads is always very low compared to the divergence between local and routing attention heads, which is almost always very close to the upper-bound of 0.6931. Divergence between different routing attention heads falls somewhere in between, being closer to the upper-bound. This shows that the attention distribution inferred by the routing attention of Section 4.1 is highly non-local in nature and different heads specialize in attending to very different parts of the input.

Model	Layers	Heads	Bits/dim
Glow (Kingma and Dhariwal, 2018)	-	-	3.81
PixelCNN (Van den Oord et al., 2016)	-	-	3.57
PixelSNAIL (Chen et al., 2017)	-	-	3.52
SPN (Menick and Kalchbrenner, 2018)	-	-	3.52
ImageTransformer (Parmar et al., 2018)	24	16	3.48
Sparse Transformer (Child et al., 2019)	48	16	3.44
<i>Routing Transformer</i>	24	16	3.43

Table 1: Results on image generation on ImageNet 64×64 in bits/dim.

Model	Layers	Heads	Perplexity
LSTMs (Grave et al., 2016)	-	-	40.8
QRNNs (Merity et al., 2018)	-	-	33.0
Adaptive Transformer (Sukhbaatar et al., 2019)	36	8	20.6
Local Transformer	16	16	19.8
Adaptive Input (Baeviski and Auli, 2018)	16	16	18.7
TransformerXL (Dai et al., 2019)	18	16	18.3
<i>Routing Transformer</i>	10	16	15.8

Table 2: Results on language modeling on Wikitext-103 data-set. Local Transformer refers to Transformer (Vaswani et al., 2017) with relative position encoding (Shaw et al., 2018) together with local attention. Perplexity is reported on the test set.

Model	Layers	Heads	Bits per byte
T64 (Al-Rfou et al., 2019)	64	2	1.13
Local Transformer	24	8	1.10
TransformerXL (Dai et al., 2019)	24	8	0.99
Sparse Transformer (Child et al., 2019)	30	8	0.99
Adaptive Transformer (Sukhbaatar et al., 2019)	24	8	0.98
<i>Routing Transformer</i>	12	8	0.99

Table 3: Results on language modeling on enwik-8 data-set. Local Transformer refers to Transformer (Vaswani et al., 2017) with relative position encoding (Shaw et al., 2018) together with local attention. Bits per byte (bpc) is reported on the test set.

7. Conclusion

Transformer models constitutes the state-of-the-art in auto-regressive generative models for sequential data. Their space-time complexity is however quadratic in sequence length, due to their attention modules. Our work proposes a sparse attention model, the Routing Transformer. It relies on content-based sparse attention motivated by non-negative matrix factorization. Compared with local attention models, it does not require fixed attention patterns but enjoys similar space-time complexity. In contrast with prior work on content-based sparse attention, it does not require computing a

full attention matrix but still selects sparsity patterns based on content similarity.

Our experiments over text and image generation draw two main conclusions. First, we show that a carefully tuned local attention model establishes a strong baseline on modern benchmark, even compared to recent state-of-the-art models. Second, we show that the Routing Transformer redefines the state-of-the-art in large long sequence benchmarks of Wikitext-103 and ImageNet-64, while being very close to do so on enwik-8 as well. Our analysis also shows that routed attention modules offer complementary attention pat-

	$JSD(local local)$	$JSD(local routing)$	$JSD(routing routing)$
layer 0	0.0038 ± 0.0018	0.4706 ± 0.0319	0.1579 ± 0.0576
layer 1	0.3071 ± 0.1217	0.6674 ± 0.0153	0.5820 ± 0.0104
layer 2	0.2164 ± 0.0803	0.5896 ± 0.0249	0.4015 ± 0.0121
layer 3	0.1163 ± 0.0336	0.6047 ± 0.0181	0.4144 ± 0.0264
layer 4	0.1840 ± 0.0562	0.6266 ± 0.0062	0.4191 ± 0.0879
layer 5	0.2284 ± 0.0225	0.6463 ± 0.0155	0.4687 ± 0.0449
layer 6	0.1901 ± 0.0525	0.6471 ± 0.0040	0.5175 ± 0.0469
layer 7	0.1566 ± 0.0685	0.5798 ± 0.0235	0.4350 ± 0.0139
layer 8	0.1638 ± 0.0739	0.5993 ± 0.0148	0.4268 ± 0.0291
layer 9	0.2095 ± 0.0560	0.6127 ± 0.0053	0.3581 ± 0.0019

Table 4: Jensen-Shannon divergence between the attention distributions of a random local attention head and a random head that routes attention as in Section 4.1 per layer on the Wikitext-103 data-set. We report means and standard deviations computed over 10 runs and use the natural logarithm so that divergences are upper-bounded by 0.6931.

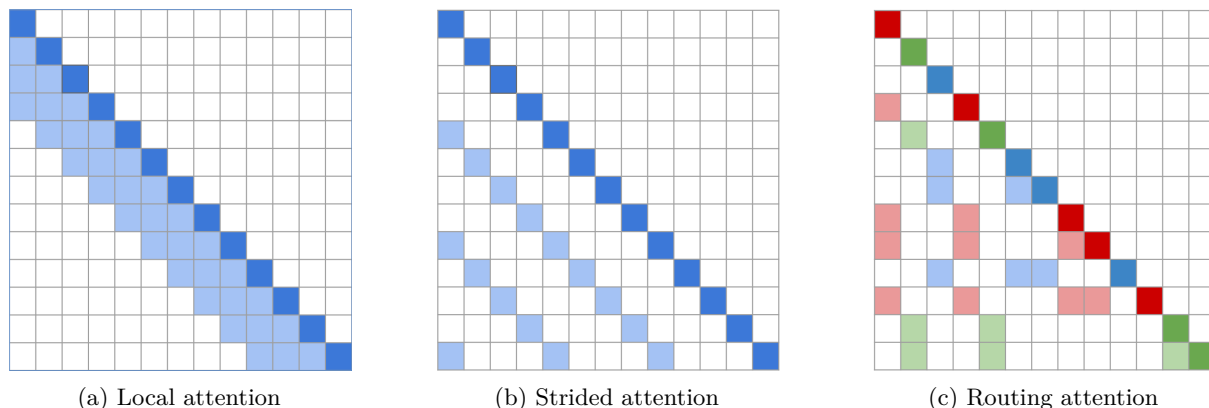


Figure 1: Figures showing 2-D attention schemes for the Routing Transformer compared to local attention and strided attention of (Child et al., 2019). The rows represent the outputs while the columns represent the inputs. For local and strided attention, the colored squares represent the elements every output row attends to. For attention routed as in Section 4.1, the different colors represent cluster memberships for the output token.

terms when compared to local attention.

Overall, our work contributes an efficient attention mechanism that applies to the modeling of long sequences and redefines the state of the art for auto-regressive generative modeling. Our approach could prove useful in domains where the inputs are naturally sparse, such as 3D point clouds, social networks or protein interactions.

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol-

ume 33, pages 3159–3166, 2019. 7

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL <http://arxiv.org/abs/1607.06450>. 4, 5

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018. 6, 7

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL <http://arxiv.org/abs/1409.0473>. 1

Arindam Banerjee and Joydeep Ghosh. Frequency-sensitive competitive learning for scalable balanced

- clustering on high-dimensional hyperspheres. *IEEE Transactions on Neural Networks*, 15(3):702–719, 2004. 5
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3
- Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 606–615, 2019. URL <http://proceedings.mlr.press/v89/blondel19a.html>. 2
- Leon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. In *Advances in neural information processing systems*, pages 585–592, 1995. 5
- Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelnail: An improved autoregressive generative model. *arXiv preprint arXiv:1712.09763*, 2017. 7
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 1, 2, 4, 5, 6, 7, 8
- Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. *arXiv preprint arXiv:1712.05382*, 2017. 2
- Kyunghyun Cho and Yoshua Bengio. Exponentially increasing the capacity-to-computation ratio for conditional computation in deep learning. *arXiv preprint arXiv:1406.7362*, 2014. 3
- Kyunghyun Cho, Bart van Merriënboer, Çağlar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL <http://arxiv.org/abs/1406.1078>. 1
- Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015. 1, 2
- Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers, 2019. 2, 3, 4
- Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 2, 4, 5, 6, 7
- Ludovic Denoyer and Patrick Gallinari. Deep sequential neural network. *arXiv preprint arXiv:1410.0510*, 2014. 3
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1
- Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005. 2
- David Eigen, Marc’Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 3
- Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. *arXiv preprint arXiv:1612.04426*, 2016. 7
- Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 3
- Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2
- Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinulescu, and Douglas Eck. Music transformer: Generating music with long-term structure. 2018. 1, 3, 6
- Sathish Reddy Indurthi, Insoo Chung, and Sangha Kim. Look harder: A neural machine translation model with hard attention. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3037–3043, 2019. 3
- Navdeep Jaitly, David Sussillo, Quoc V Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio. A neural transducer. *arXiv preprint arXiv:1511.04868*, 2015. 2

- Jingu Kim and Haesun Park. Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology, 2008. 2
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>. 6
- Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 7
- Guillaume Lample, Alexandre Sablayrolles, Marc’Aurelio Ranzato, Ludovic Denoyer, and Herv’e J’egou. Large memory layers with product keys. 2019. 3
- Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001. 2
- Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018. 2
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019. 1
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 1, 2, 4
- Matt Mahoney. Large text compression benchmark. URL: <http://www.matmahoney.net/text/text.html>, 2011. 5, 6
- Chaitanya Malaviya, Pedro Ferreira, and André F. T. Martins. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2059. URL <https://www.aclweb.org/anthology/P18-2059>. 3
- Mikko I Malinen and Pasi Fränti. Balanced k-means for clustering. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 32–41. Springer, 2014. 5
- André F. T. Martins and Julia Kreutzer. Learning what’s easy: Fully differentiable neural easy-first taggers. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 349–362, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1036. URL <https://www.aclweb.org/anthology/D17-1036>. 3
- Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018. 1, 6, 7
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 5, 6
- Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240*, 2018. 7
- Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 1, 3, 4, 6, 7
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language_understanding_paper.pdf, 2018. 1, 3
- Jack Rae, Jonathan J Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Timothy Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. In *Advances in Neural Information Processing Systems*, pages 3621–3629, 2016. 3
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 1, 4, 6, 7
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 3
- Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019. 2, 4, 6, 7

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016. [7](#)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, 2017. URL <http://arxiv.org/abs/1706.03762>. [1](#), [3](#), [5](#), [6](#), [7](#)

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. [2](#)

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. [1](#)