Generating Long Sequences with Sparse Transformers

Rewon Child et al., 2019

Rui Yoshinaga

December 5, 2021

Background

- Consider the task of autoregressive sequence generation
- The joint probability of sequence $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ is modeled as

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1}; \theta) \tag{1}$$

  where θ is a network

- θ: Transformer in decoder-only mode
  - Input: sequence of tokens
  - Output: categorical distribution
  - Objective: maximize the log-probability of the data wrt θ

Formularization of the Self-Attention

- Self-attention layer:
  - a matrix of input embeddings $X \to$ output matrix
  - parameterized by a connectivity pattern $S = \{S_1, \ldots, S_n\}$

$$\mathrm{attend}(X, S) = \Big[a(\boldsymbol{x}_i, S_i)\Big]_{i \in \{1,\ldots,n\}} \tag{2}$$

$$a(\boldsymbol{x}_i, S_i) = \mathrm{softmax}\left(\frac{[W_q \boldsymbol{x}_i] K_{S_i}^T}{\sqrt{d}}\right) V_{S_i} \tag{3}$$

$$K_{S_i} = \Big[W_k \boldsymbol{x}_j\Big]_{j \in S_i} \quad V_{S_i} = \Big[W_v \boldsymbol{x}_j\Big]_{j \in S_i} \tag{4}$$

$$\mathrm{attention}(X) = W_p \cdot \mathrm{attend}(X, S) \tag{5}$$

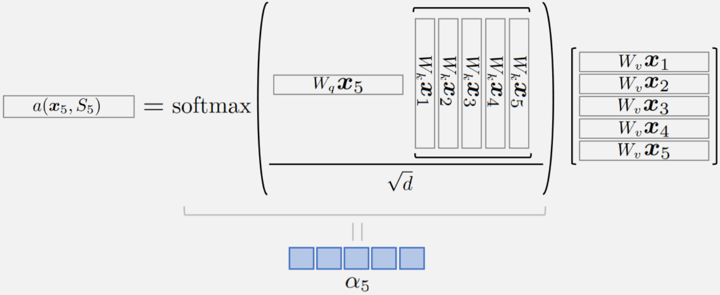
where W_q, W_k, W_v, W_p: the weight matrix

Calculating the Self-Attention

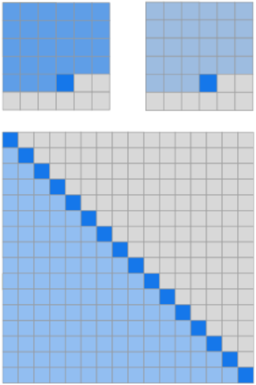
$$\mathrm{attention}(X) = W_p \cdot \mathrm{attend}(X, S) = W_p \cdot \left[a(\boldsymbol{x}_i, S_i)\right]_{i \in \{1,\ldots,n\}}$$

$$= W_p \cdot \mathrm{softmax}\left(\frac{\left[W_q \boldsymbol{x}_i\right]\left[W_k \boldsymbol{x}_j\right]_{j \in S_i}^T}{\sqrt{d}}\right)\left[W_v \boldsymbol{x}_j\right]_{j \in S_i}$$

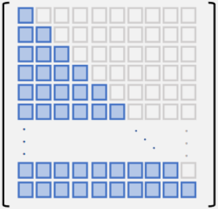
- Example: $i = 5,\ a(\boldsymbol{x}_5, S_5)$
  - $S_5 = \{1, 2, 3, 4, 5\}$



Standard Transformer

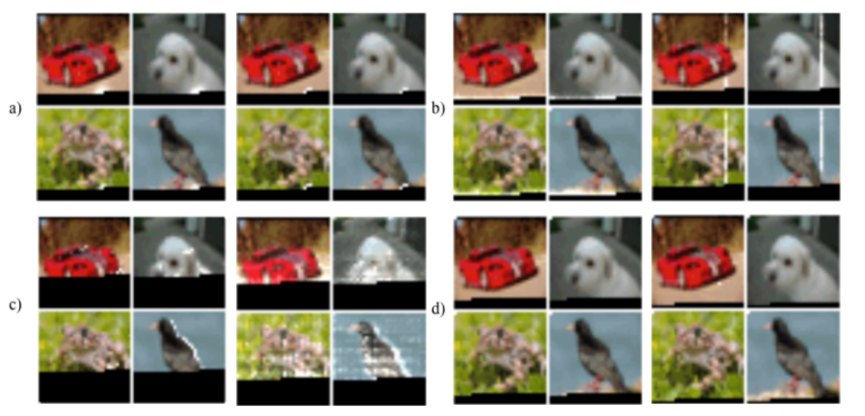


(a) Transformer

$\text{attention matrix} = [\alpha_i]_{i \in \{1,\dots,n\}} =$ 

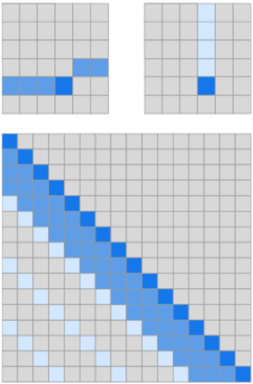
- Full self-attention for autoregressive models defines $S_i = \{j \mid j \leq i\}$
- Become intractable as n grows; $\mathcal{O}(n^2)$
- Replace S_i with efficient $A_i \subset S_i$
  Let $|A_i| \propto \sqrt[p]{n}$; $\mathcal{O}(n \sqrt[p]{n})$
  - In this work $p = 2$

Qualitative assessment of learned attention patterns



- Learned attention patterns from a 128-layer network
  a) Early layers: locally connected pattern like convolution
  b) Layers 19-20: to split into row and column attention
  c) Several layers: global, data-dependent patterns
  d) Layers 64-128: high sparsity, with activating rarely

Sparse Transformer (strided)



(b) Sparse Transformer (strided)

- Let stride $l \approx \sqrt{n}$

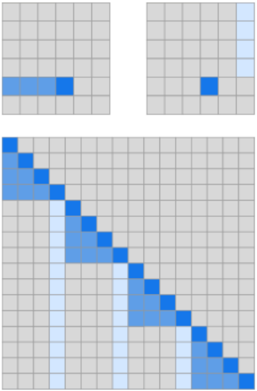
$$A_i^{(1)} = \{t, t+1, \ldots, i\}$$
$$A_i^{(2)} = \{j \mid (i-j) \mod l = 0\}$$

  where $t = \max(0, i-l)$

- Useful for data with periodic structure
  - image, music
- In left figure, $i = 28, l = 5$

$$t = \max(0, 28-5) = 23$$
$$A_{28}^{(1)} = \{23, 24, 25, 26, 27, 28\} \text{ (upper left)}$$
$$A_{28}^{(2)} = \{4, 10, 16, 22, 28\} \text{ (upper right)}$$

Sparse Transformer (fixed)



(c) Sparse Transformer (fixed)

- Let stride $l \approx \sqrt{n}$

$$A_i^{(1)} = \{j \mid \lfloor i/l \rfloor = \lfloor j/l \rfloor\}$$
$$A_i^{(2)} = \{j \mid j \mod l \in \{t, t+1, \ldots, l\}\}$$

  where $t = l - c$, c is hyperparameter

- Useful for data without periodic structure
  - text

- In left figure, $i = 28, l = 6, c = 0$

$$t = 6 - 0 = 6$$
$$A_{28}^{(1)} = \{24, 25, 26, 27, 28\} \text{ (upper left)}$$
$$A_{28}^{(2)} = \{6, 12, 18, 24\} \text{ (upper right)}$$

Factorized attention heads

- The simplest technique:

$$\text{attention}(X) = W_p \cdot \text{attend}(X, A^{(r \mod p)}) \tag{6}$$

where r: index of residual block, p: number of attention heads

  - Use one attention type per residual block
  - Interleave them sequentially or at a ratio

- Marged head:

$$\text{attention}(X) = W_p \cdot \text{attend}(X, \bigcup_{m=1}^{p} A^{(m)}) \tag{7}$$

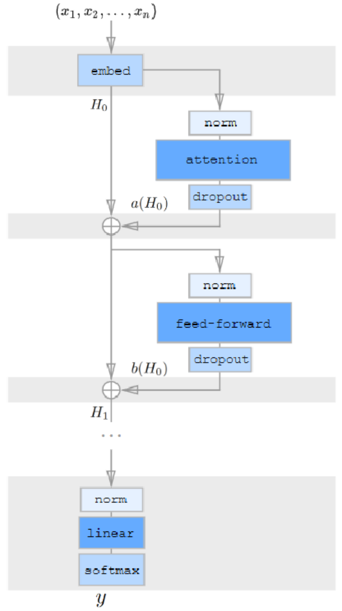
  - Merge factorized attentions, use it as a single attention

- Multi-head attention:

$$\text{attention}(X) = W_p \left[\text{attend}(X, A)^{(i)}\right]_{i \in \{1, \ldots, n_h\}} \tag{8}$$

  - n_h attention products are computed in parallel then concatenated

Sparse Transformer



- Define a network of N layers as follows:

$$H_0 = \mathrm{embed}(X, W_e) \tag{9}$$
$$H_k = H_{k-1} + \mathrm{resblock}(H_{k-1}) \tag{10}$$
$$y = \mathrm{softmax}(\mathrm{norm}(H_N)\, W_{out}) \tag{11}$$

$$a(H) = \mathrm{dropout}(\mathrm{attention}(\mathrm{norm}(H))) \tag{12}$$
$$b(H) = \mathrm{dropout}(\mathrm{ff}(\mathrm{norm}(H + a(H)))) \tag{13}$$
$$\mathrm{resblock}(H) = a(H) + b(H) \tag{14}$$

$$\mathrm{ff}(x) = W_2 f(W_1 x + b_1) + b_2$$
$$f(x) = \mathrm{GELU}(x) = x \cdot \mathrm{sigmoid}(1.702x)$$

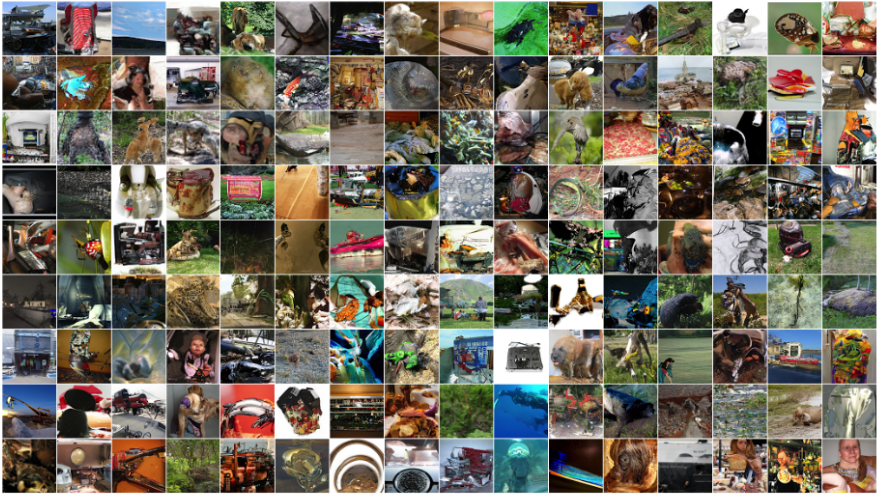
$$\mathrm{embed}(X, W_e) = \left[\boldsymbol{x}_i W_e + \sum_{j=1}^{n_{emb}} \boldsymbol{o}_i^{(j)} W_j \right]_{\boldsymbol{x}_i \in X} \tag{15}$$

Experiments

- Task: Density modeling
  - Image: CIFAR-10, ImageNet 64x64
  - Text: EnWik8
  - Audio: Classical music
- Evaluaton: Bits/byte
  - Negative log-likelihood per byte
- Results:
  - Sota in images and text
  - Easily adaptable to raw audio

Model	Bits per byte
CIFAR-10	
PixelCNN (Oord et al., 2016)	3.03
PixelCNN++ (Salimans et al., 2017)	2.92
Image Transformer (Parmar et al., 2018)	2.90
PixelSNAIL (Chen et al., 2017)	2.85
Sparse Transformer 59M (strided)	**2.80**
Enwik8	
Deeper Self-Attention (Al-Rfou et al., 2018)	1.06
Transformer-XL 88M (Dai et al., 2018)	1.03
Transformer-XL 277M (Dai et al., 2018)	**0.99**
Sparse Transformer 95M (fixed)	**0.99**
ImageNet 64x64	
PixelCNN (Oord et al., 2016)	3.57
Parallel Multiscale (Reed et al., 2017)	3.7
Glow (Kingma & Dhariwal, 2018)	3.81
SPN 150M (Menick & Kalchbrenner, 2018)	3.52
Sparse Transformer 152M (strided)	**3.44**
Classical music, 5 seconds at 12 kHz	
Sparse Transformer 152M (strided)	**1.97**

ImageNet 64x64



Unconditional samples from ImageNet 64x64

Classical music from raw audio

- Model: Strided Sparse Transformer 152M parameters
- Task: 12kHz audio generation
  - 65535 sequence length = 5 second audio at 12kHz
  - Trained models on classical music dataset
- Samples: `https://openai.com/blog/sparse-transformer/`
  - Clearly demonstrate global coherence
  - Exhibit a variety of play styles and tones

- Sequence length vs. Model capacity
  - The largest model which entirely fit into 16GB V100 accelerators
  - We could use factorized self-attention on sequences over 1M timesteps, albeit with extremely few parameters (3M)

Sequence length	Parameters	Bits per byte
65,536	152M	1.97
262,144	25M	2.17
1,048,576	3M	2.99

Conclusion

- Sparse Transformer
  - Reduce the computation in the self-attention
- Factorized Self-Attention
  - Separate the self-attention across several attention patterns
- Better performances on density modeling of long sequences
  - Sota in images and text, Easily adaptable to raw audio

Learnable Embedding

- Add n_{emb} embeddings to each input location

$$\text{embed}(X, W_e) = \left[\boldsymbol{x}_i W_e + \sum_{j=1}^{n_{emb}} \boldsymbol{o}_i^{(j)} W_j\right]_{\boldsymbol{x}_i \in X}$$

where $\boldsymbol{x}_i$: one-hot encoded ith element in the sequence
$\boldsymbol{o}_i^{(j)}$: one-hot encoded position of $\boldsymbol{x}_i$ in the jth dimension

- For images: $n_{emb} = 3$ (row, column, channel)
- For text and audio: $n_{emb} = 2$ (row, column)

Model details in Experiments

CIFAR-10

- Strided Sparse Transformer
  - 2 heads, 128 layers, $d = 256$
- CIFAR-10: 3,072 contexts

Text

- Fixed Sparse Transformer
  - 8 heads, 30 layers, $d = 512$, stride $= 128$, $c = 32$, merged head
- EnWik8 dataset: 12,288 contexts

ImageNet 64x64

- Strided Sparse Transformer
  - 16 heads, 48 layers, $d = 512$, stride $= 128$

Model Comparison

Model	Bits per byte	Time/Iter
Enwik8 (12,288 context)		
Dense Attention	1.00	1.31
Sparse Transformer (Fixed)	**0.99**	0.55
Sparse Transformer (Strided)	1.13	0.35
CIFAR-10 (3,072 context)		
Dense Attention	2.82	0.54
Sparse Transformer (Fixed)	2.85	0.47
Sparse Transformer (Strided)	**2.80**	0.38

- Running signitícantly faster than full attention
- Converged to lower error