RFF for MRFs

November 19, 2020

The exponential kernel, $\exp(\mathbf{x}^T\mathbf{y})$ with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, is widely used to parameterize discrete distributions. It can be well-approximated by Random Fourier Features (RFF) $\exp(\mathbf{x}^T\mathbf{y}) \approx \phi(\mathbf{x})^T\phi(\mathbf{y}), \phi : \mathbb{R}^n \to \mathbb{R}^d$ [2]. Importantly, this approximation is linear, and we can therefore apply reordering tricks based on the distributive and associative property to improve the efficiency of certain operations by polynomial factors.

1 Attention

One such operation is attention, a very popular operation used in neural networks. Naively, attention requires quadratic time complexity. However, the sampled softmax with RFF [2] can be applied to compute attention with linear time complexity [1] (and many others).

1.1 Linear attention with RFF

Given queries $\mathbf{q}_j \in \mathbb{R}^n$, keys $\mathbf{k}_i \in \mathbb{R}^n$, and values $\mathbf{v}_i \in \mathbb{R}^n$ with $t \in [T], i \in [I]$, attention must compute outputs

$$o_t = \sum_i \frac{\exp(\mathbf{q}_t^T\mathbf{k}_i)\mathbf{v}_i}{\sum_j \exp(\mathbf{q}_t^T\mathbf{k}_j)}. \tag{1}$$

This requires $O(TIn)$ time to compute for all o_t. Applying the RFF approximation, we have

$$o_t \approx \sum_i \frac{(\phi(\mathbf{q}_t)^T\phi(\mathbf{k}_i))\mathbf{v}_i}{\sum_j \phi(\mathbf{q}_t)^T\phi(\mathbf{k}_j)} = \phi(\mathbf{q}_t)^T \frac{\sum_i \phi(\mathbf{k}_i)\mathbf{v}_i}{\sum_j \phi(\mathbf{q}_t)^T\phi(\mathbf{k}_j)}. \tag{2}$$

After using the distributive and associate properties, the numerator $\sum_i \phi(\mathbf{k}_i)\mathbf{v}_i$ can be computed once and shared for all queries, reducing the time complexity of computing all o_t to $O(Td + Id)$.

Matrix version Matrix form [1] is more informative, write up later. Just breaks up A matrix into linear decomposition, then applies associative property of matmul.

1.2 Approximation error

2 Application to MRFs

The RFF approximations work well in unstructured distributions. Can we get even tighter approximations when distributions have structure? Does the approximation even work?

2.1 Drop-in substitution of kernel approximation

We start with a linear-chain MRF:

$$p(x) \propto \prod_t \psi(x_{t-1}, x_t) = \prod_t \exp(\mathbf{x}_{t-1}^T\mathbf{x}_t),$$

with the variables $x_t \in \mathcal{X}$ and embeddings $\mathbf{x}_t \in \mathbb{R}^n$. As before, we approximate $\psi_t(x_{t-1}, x_t) = \exp(\mathbf{x}_{t-1}^T\mathbf{x}_t) \approx \phi(\mathbf{x}_{t-1})^T\phi(\mathbf{x}_t)$, with the random projection $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^d$ chosen appropriately. To start, consider computing the partition function of a simple example with $T = 3$:

$$\begin{aligned} Z &= \sum_{x_1} \sum_{x_2} \psi_1(x_1, x_2) \sum_{x_3} \psi_2(x_2, x_3) \\ &\approx \sum_{x_1} \sum_{x_2} \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2) \sum_{x_3} \phi(\mathbf{x}_2)^T \phi(\mathbf{x}_3) \\ &= \sum_{x_1} \phi(\mathbf{x}_1)^T \sum_{\mathbf{x}_2} \phi(\mathbf{x}_2)\phi(\mathbf{x}_2)^T \sum_{x_3} \phi(\mathbf{x}_3). \end{aligned} \tag{3}$$

We can precompute the sum of outer products $\sum_{\mathbf{x}_2} \phi(\mathbf{x}_2)\phi(\mathbf{x}^T)$ as well as other sums independently due to associativity, resulting in time complexity $O(Td + T|\mathcal{X}|d^2)$ in serial, and $O(Td + |\mathcal{X}|d^2)$ if the sums of outer products can be computed in parallel.

Matrix version Probably much clearer here too.

2.2 Approximation error

References

[1] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2020.

[2] Ankit Singh Rawat, Jiecao Chen, Felix X. Yu, Ananda Theertha Suresh, and Sanjiv Kumar. Sampled softmax with random fourier features. *CoRR*, abs/1907.10747, 2019. URL `http://arxiv.org/abs/1907.10747`.