

Random Fourier Features for Scaling Markov Random Fields

November 23, 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):

$$\exp(\mathbf{x}^T \mathbf{y}) \approx \phi(\mathbf{x})^T \phi(\mathbf{y}), \quad (1)$$

with random projections $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ [2]. Importantly, this approximation is linear, and therefore admits 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 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, such as Random Feature Attention, LinFormers, etc.).

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 computes outputs

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

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

$$o_t \approx \sum_i \frac{(\phi(\mathbf{q}_t)^T \phi(\mathbf{k}_i)) \mathbf{v}_i^T}{\sum_j \phi(\mathbf{q}_t)^T \phi(\mathbf{k}_j)} = \frac{\phi(\mathbf{q}_t)^T \sum_i \phi(\mathbf{k}_i) \mathbf{v}_i^T}{\phi(\mathbf{q}_t)^T \sum_j \phi(\mathbf{k}_j)}. \quad (3)$$

The terms $\sum_i \phi(\mathbf{k}_i) \mathbf{v}_i^T$ and $\sum_j \phi(\mathbf{k}_j)$ can be computed once for all queries, reducing the time complexity of computing all o_t to $O(Td + Id^2)$.

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

todo

2 Linear Chain 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 $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^d$ chosen appropriately. To start, consider computing the partition function of a simple example with $T = 3$:

$$\begin{aligned} Z &= \sum_{x_1} \sum_{x_2} \psi_1(x_1, x_2) \sum_{x_3} \psi_2(x_2, x_3) \\ &\approx \sum_{x_1} \sum_{x_2} \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2) \sum_{x_3} \phi(\mathbf{x}_2)^T \phi(\mathbf{x}_3) \\ &= \left(\sum_{x_1} \phi(\mathbf{x}_1)^T \right) \left(\sum_{x_2} \phi(\mathbf{x}_2) \phi(\mathbf{x}_2)^T \right) \left(\sum_{x_3} \phi(\mathbf{x}_3) \right). \end{aligned} \tag{4}$$

We can precompute the sum of outer products $\sum_{\mathbf{x}_2} \phi(\mathbf{x}_2) \phi(\mathbf{x}_2)^T$ independently, resulting in time complexity $O(Td^2 + T|\mathcal{X}|d^2)$ in serial, and $O(Td^2 + |\mathcal{X}|d^2)$ if the sums of outer products can be computed in parallel. We can also apply a divide-and-conquer approach to further reduce time to $O(c_{\text{mm}}(\log T + \log |\mathcal{X}|))$ on a parallel machine, where c_{mm} is the cost of a matrix multiplication. Compared to the original time complexity of $O(T|\mathcal{X}|^2)$, this is particularly beneficial if $d \ll |\mathcal{X}|$.

Matrix version Probably much clearer here too. todo

2.2 Approximation error

todo

3 Tree MRFs

We proceed from first order linear-chain MRFs, where nodes only 2 neighbours, to the next simplest model: trees. With tree MRFs, nodes may have more than 2 neighbours depending on the arity of the tree, but the dependencies are still simple and exact inference is tractable.

3.1 The distributive property is not enough

3.2 Approximation error

Tensor version

3.3 Approximation error

4 Application to PCFGs

todo

4.1 Approximation error

todo

References

- [1] Krzysztof Choromanski, Valerii Likhoshesterov, 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>.