EECS 182 Deep Neural Networks
Spring 2023 Anant Sahai

Homework 8

This homework is due on Thursday, April 6, 2022, at 10:59PM.

1. Redo Midterm

Please redo the midterm exam (linked at https://inst.eecs.berkeley.edu/~cs182/sp23/assets/assignments/mt-sp23.pdf) as homework. Attach your answers to the written part of this homework.

Solution: Solutions for these will be released separately.

2. Hand-Design Transformers

Please follow the instructions in this notebook. You will implement a simple transformer model (with a single attention head) from scratch and then create a hand-designed the attention heads of the transformer model capable of solving a basic problem. Once you finished with the notebook,

- Download `submission_log.json` and submit it to "Homework 8 (Code)" in Gradescope.
- Answer the following questions in your submission of the written assignment:

(a) Design a transformer that selects by contents. Compare the variables of your hand-designed Transformer with those of the learned Transformer. **Identify the similarities and differences between the two sets of variables and provide a brief explanation for each difference.**

Solution: Answers may vary, but weights, keys, and queries are likely more evenly distributed than those the student implemented. However, Vm, attention scores and values should be roughly the same.

Explanation: this occurs since the network is simple enough that there is only one correct attention pattern (only attend to matching positions) and only one correct thing to do with attended positions (use them to copy the original content into the output). In more complicated problems it's rare to be able to predict precisely what features the transformer learns.

(b) Design a transformer that selects by positions. Compare the variables of your hand-designed Transformer with those of the learned Transformer. **Identify the similarities and differences between the two sets of variables and provide a brief explanation for each difference.**

Solution: Answers may vary, but weights, keys, and queries are likely more evenly distributed than those the student implemented. However, attention scores and outputs should be roughly the same, since there is only one correct attention pattern (only attend to the first position) and only one correct thing to do with attended positions (use them to copy the first element into the output).

3. Kernelized Linear Attention (Part II)

This is a continuation of "Kernelized Linear Attention" in HW 7. Please refer to this part for notation and context. In Part I of this problem, we considered ways to efficiently express the attention operation when sequences are long (e.g. a long document). Attention uses the following equation:

$$V_i' = \frac{\sum_{j=1}^{N} \text{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{N} \text{sim}\left(Q_i, K_j\right)}. \tag{1}$$

We saw that when the similarity function is a kernel function (i.e. if we can write $\text{sim}\left(Q_i, K_j\right) = \Phi(Q_i)^T \Phi(K_j)$ for some function Φ), then we can use the associative property of matrix multiplication to simplify the formula to

$$V_i' = \frac{\phi\left(Q_i\right)^T \sum_{j=1}^{N} \phi\left(K_j\right) V_j^T}{\phi\left(Q_i\right)^T \sum_{j=1}^{N} \phi\left(K_j\right)}. \tag{2}$$

If we use a polynomial kernel with degree 2, this gives a computational cost of $\mathcal{O}\left(ND^2M\right)$, which for very large N is favorable to the softmax attention computational cost of $\mathcal{O}\left(N^2 \max\left(D, M\right)\right)$. ($N$ is the sequence length, D is the feature dimension of the queries and keys, and M is the feature dimension of the values. Now, we will see whether we can use kernel attention to directly approximate the softmax attention:

$$V_i' = \frac{\sum_{j=1}^{N} \exp(\frac{Q_i^T K_j}{\sqrt{D}}) V_j}{\sum_{j=1}^{N} \exp(\frac{Q_i^T K_j}{\sqrt{D}})}. \tag{3}$$

(a) Approximating softmax attention with linearized kernel attention

i. As a first step, we can use Gaussian Kernel $\mathcal{K}_{\text{Gauss}}(q,k) = \exp(\frac{-||q-k||_2^2}{2\sigma^2})$ to rewrite the softmax similarity function, where $\text{sim}_{\text{softmax}}(q,k) = \exp(\frac{q^T k}{\sqrt{D}})$. Assuming we can have $\sigma^2 = \sqrt{D}$, **rewrite the softmax similarity function using Gaussian Kernel.**. (*Hint: You can write the softmax* $\exp(\frac{-||q-k||_2^2}{2\sigma^2})$ *as the product of the Gaussian Kernel and two other terms.)*

Solution:

$$\begin{aligned}\text{sim}_{\text{softmax}}(q,k) = \exp(\frac{q^T k}{\sqrt{D}}) &= \exp(\frac{||q||_2^2}{2\sigma^2}) * \exp(\frac{-||q-k||_2^2}{2\sigma^2}) * \exp(\frac{||k||_2^2}{2\sigma^2}) \\ &= \exp(\frac{||q||_2^2}{2\sigma^2}) * \mathcal{K}_{\text{Gauss}}(q,k) * \exp(\frac{||k||_2^2}{2\sigma^2})\end{aligned}$$

ii. However, writing softmax attention using a Gaussian kernel does not directly enjoy the benefits of the reduced complexity using the feature map. This is because the feature map of Guassian kernel usually comes from the Taylor expression of $\exp(\cdot)$, whose computation is still expensive [1]. However, we can approximate the Guassian kernel using random feature map and then reduce the computation cost. (Rahimi and Recht, 2007)[2] proposed random Fourier features to approximate a desired shift-invariant kernel. The method nonlinearly transforms a pair of vectors q and k using a **random feature map** $\phi_{\text{random}}()$; the inner product between $\phi\left(q\right)$ and $\phi\left(k\right)$ approximates the

[1] https://www.csie.ntu.edu.tw/~cjlin/talks/kuleuven_svm.pdf

[2] https://people.eecs.berkeley.edu/~brecht/papers/07.rah.rec.nips.pdf

kernel evaluation on q and k. More precisely:

$$\phi_{\text{random}}(q) = \sqrt{\frac{1}{D_{\text{random}}}}\left[\sin\left(\mathbf{w}_1 q\right), \ldots, \sin\left(\mathbf{w}_{D_{\text{random}}} q\right), \cos\left(\mathbf{w}_1 q\right), \ldots, \cos\left(\mathbf{w}_{D_{\text{random}}} q\right)\right]^\top. \tag{4}$$

Where we have D_{random} of D-dimensional random vectors $\mathbf{w}_i$ independently sampled from $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_D)$,

$$\mathbb{E}_{\mathbf{w}_i}\left[\phi\left(q\right) \cdot \phi\left(k\right)\right] = \exp\left(-\left\|q-k\right\|^2 / 2\sigma^2\right). \tag{5}$$

Use ϕ_{random} to approximate the above softmax similarity function with Gaussian Kernel and derive the computation cost for computing all the V' here.

Solution:

$$\begin{aligned}\text{sim}_{\text{softmax}}(q,k) &= \exp(\frac{||q||_2^2}{2\sigma^2}) * \mathcal{K}_{\text{Gauss}}(q,k) * \exp(\frac{||k||_2^2}{2\sigma^2}) \\ &= \exp(\frac{||q||_2^2}{2\sigma^2}) * \phi_{\text{random}}(q)^T \phi_{\text{random}}(k) * \exp(\frac{||k||_2^2}{2\sigma^2})\end{aligned}$$

The computation cost of ϕ_{random} is $\mathcal{O}\left(N(M+D)D_{\text{random}}\right)$.)

(b) (Optional) Beyond self-attention, an autoregressive case will be masking the attention computation such that the i-th position can only be influenced by a position j if and only if $j \leq i$, namely a position cannot be influenced by the subsequent positions. This type of attention masking is called *causal masking*.

Derive how this causal masking changes equation 1 and 2. **Write equation 1 and 2 in terms of S_i and Z_i**, which are defined as:

$$S_i = \sum_{j=1}^{i} \phi\left(K_j\right) V_j^T, \;\; Z_i = \sum_{j=1}^{i} \phi\left(K_j\right), \tag{6}$$

to simplify the causal masking kernel attention and derive the computational complexity of this new causal masking formulation scheme.

Solution: The causal masking changes equation 1 as follows,

$$V_i' = \frac{\sum_{j=1}^{i} \text{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{i} \text{sim}\left(Q_i, K_j\right)}. \tag{7}$$

We linearize the masked attention as described below,

$$V_i' = \frac{\phi\left(Q_i\right)^T \sum_{j=1}^{i} \phi\left(K_j\right) V_j^T}{\phi\left(Q_i\right)^T \sum_{j=1}^{i} \phi\left(K_j\right)}. \tag{8}$$

we can simplify equation 8 to

$$V_i' = \frac{\phi\left(Q_i\right)^T S_i}{\phi\left(Q_i\right)^T Z_i}. \tag{9}$$

Note that, S_i and Z_i can be computed from S_{i-1} and Z_{i-1} in constant time hence making the com-

putational complexity of linear transformers with causal masking linear with respect to the sequence length.

4. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **Roughly how many total hours did you work on this homework? Write it down here where you'll need to remember it for the self-grade form.**

Contributors:

- CS182 Staff.
- Olivia Watkins.
- Jake Austin.
- Dhruv Shah.
- Anant Sahai.
- Linyuan Gong.
- Sheng Shen.
- Shaojie Bai.
- Angelos Katharopoulos.
- Hao Peng.

