Danilo Dordevic: Assignment 6

ddordevic@student.ethz.ch, 21-954-888.

15/01/2023 - 22:19h

Question 1

a.1)

Each head $h \in [N_h]$ in the multi-head attention unit is defined by the matrices $\boldsymbol{W}_q^{(h)} \in \mathbb{R}^{D_i \times D_{hid}}$, $\boldsymbol{W}_k^{(h)} \in \mathbb{R}^{D_i \times D_{hid}}$, $\boldsymbol{W}_v^{(h)} \in \mathbb{R}^{D_i \times D_o}$. These matrices are learnable parameters, so each head contains $D_i \cdot D_{hid} + D_i \cdot D_{hid} + D_i \cdot D_o = \boxed{D_i \cdot (2D_{hid} + D_o)}$ parameters.

Since the multi-head self-attention module contains N_h independent heads, the total number of parameters of all heads is $\boxed{N_h \cdot D_i \cdot (2D_{hid} + D_o)}$.

Additionally, the multi-head self-attention module contains an output matrix $W_{out} \in \mathbb{R}^{N_h D_o \times D_{out}}$ with $\boxed{N_h \cdot D_o \cdot D_{out}}$ parameters, as well as the bias vector $b_{out} \in \mathbb{R}^{D_{out}}$ with $\boxed{D_{out}}$ parameters.

Adding the values in dashed boxed, we get the total number of parameters in the multi-head self-attention module: $\boxed{N_h \cdot D_i(2D_{hid} + D_o) + N_h \cdot D_o \cdot D_{out} + D_{out}}$.

a.2

Let images $\boldsymbol{Z}_{W \times H \times D_i}$ be indexed by a single vector value $\boldsymbol{p} = (i, j)$, which can be interpreted as a linear index, with the following notation: $\boldsymbol{Z}_{\boldsymbol{p},:} = \boldsymbol{Z}_{i,j,:}$.

Self-attention for index $\boldsymbol{q}$ can be written as the following:

$$
\begin{aligned}
\boldsymbol{O}_{\boldsymbol{q},:} &= \text{Self-Attention}(\boldsymbol{Z})_{\boldsymbol{q},:} \\
&= \left[\text{softmax}(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T) \cdot \boldsymbol{Z}\boldsymbol{W}_v\right]_{\boldsymbol{q},:} \\
&= \text{softmax}(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T)_{\boldsymbol{q},:} \cdot \boldsymbol{Z}\boldsymbol{W}_v \\
&= \boxed{\text{softmax}(\boldsymbol{Z}_{\boldsymbol{q},:}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T) \cdot \boldsymbol{Z}\boldsymbol{W}_v} \\
&= \left(\sum_{\boldsymbol{k}} \text{softmax}(\boldsymbol{Z}_{\boldsymbol{q},:}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T)_{\boldsymbol{k}}\boldsymbol{Z}_{\boldsymbol{k},:}\right) W_v
\end{aligned}
$$

Same can be applied to the case of MultiHead-SelfAttention:

$$\begin{aligned}
\text{MultiHeadSelf-Attention}(\boldsymbol{Z})_{\boldsymbol{q},:} &= \underset{h\in[N_h]}{\text{concat}}\left[O_{\boldsymbol{q},:}\right]\cdot\boldsymbol{W}_{out}+\boldsymbol{b}_{out} \\
&= \boxed{\underset{h\in[N_h]}{\text{concat}}\left[\text{softmax}(\boldsymbol{Z}_{\boldsymbol{q},:}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T)\cdot\boldsymbol{Z}\boldsymbol{W}_v\right]\cdot\boldsymbol{W}_{out}+\boldsymbol{b}_{out}} \\
&= \underset{h\in[N_h]}{\text{concat}}\left[\left(\sum_{\boldsymbol{k}}\text{softmax}(\boldsymbol{Z}_{\boldsymbol{q},:}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}^T)_{\boldsymbol{k}}\boldsymbol{Z}_{\boldsymbol{k},:}\right)\boldsymbol{W}_v\right]\cdot\boldsymbol{W}_{out}+\boldsymbol{b}_{out}
\end{aligned}$$

b.1)

Plugging $\boldsymbol{W}_q=\boldsymbol{W}_k=\boldsymbol{0}$ into the expression for $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}^{relative}$ removes the first three terms, as they all contain at least one of the matrices set to $\boldsymbol{0}$. This leaves the new expression:

$$\begin{aligned}
\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}^{relative} &= \boldsymbol{v}^T\widetilde{\boldsymbol{W}_k}\boldsymbol{r}_{\boldsymbol{\delta}} \\
&= \boldsymbol{v}^T\boldsymbol{I}\boldsymbol{r}_{\boldsymbol{\delta}} \\
&= \boldsymbol{v}^T\boldsymbol{r}_{\boldsymbol{\delta}} \\
&= -\alpha^{(h)}\begin{pmatrix}1\\-2\Delta_1^{(h)}\\-2\Delta_1^{(h)}\end{pmatrix}^T\begin{pmatrix}\|\boldsymbol{\delta}\|^2\\\boldsymbol{\delta}_1\\\boldsymbol{\delta}_2\end{pmatrix} \\
&= \boxed{-\alpha^{(h)}(\|\boldsymbol{\delta}\|^2-2\boldsymbol{\delta}_1\boldsymbol{\Delta}_1^{(h)}-2\boldsymbol{\delta}_2\boldsymbol{\Delta}_2^{(h)})}
\end{aligned}$$

b.2

Absolute encoding

Looking at equation 5), we can see that the expression for $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}^{absolute}$ consists of 4 terms, each of which is produced by the same order of operations over vectors and matrices of the same dimensions, so analyzing the computational complexity of only one of them is enough to determine the computational complexity of $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}^{absolute}$. Without the loss of generality, we analyze the first term: $\boldsymbol{Z}_{\boldsymbol{q},:}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{Z}_{\boldsymbol{k},:}^T$. First, multiplying $\boldsymbol{Z}_{\boldsymbol{q},:}\in\mathbb{R}^{1\times D_x}$ and $\boldsymbol{W}_q\in\mathbb{R}^{Dx\times Dx}$ requires $D_x\cdot(2D_x-1)$ operations (there are D_x entries in the result of this vector-matrix multiplication, and each one requires D_x multiplications and D_x-1 additions). This is repeated for $\boldsymbol{W}_k^T\boldsymbol{Z}_{\boldsymbol{k},:}^T$, which also requires $D_x\cdot(2D_x-1)$ operResults. Result of these two multiplications are two D_x-dimensional vectors, which are then multiplied. This requires $operationseratoins.Thetotalcostofcomputingonetermisthus2$*D$_x(2D_x-1)+2D_x-1$. As there are 4 terms, the cost of computing $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}^{absolute}$ is $4\cdot(2*D_x(2D_x-1)+2D_x-1)=O(DAttentionthentionmatrix$A^{absolute} contains N^2 entries, so the cost of computing it is $\boxed{O(N^2D_x^2)}$. To get the self-attention matrix $O^{absolute}$, we need to compute the softmax of the matrix $A^{absolute}$, which takes $\boxed{O(N^2)}$ time. After that, the resulting NxN matrix is multiplied by $\boldsymbol{Z}$ which is NxD_x, requiring $\boxed{O(N^2D_x)}$ operations. Finally the result is

multiplied by a $D_x \times D_x$ matrix $\boldsymbol{W}_v$ to yield $O^{aboslute}$, which takes $\boxed{O(ND_x^2)}$. Summing up all of the expressions in dashed boxes, the final computational complexity of computing self-attention in the case of absolute positional encoding is $\boxed{O(N^2D_x^2 + N^2D_x + ND_x^2)}$

Relative encoding

It is evident from Equation 6, which defines the expression for the entries to the $\boldsymbol{A}^{relative}$ matrix, that the computational costs of computing $\boldsymbol{A}^{relative}$ is the same as computing $\boldsymbol{A}^{absolute}$, as there is the same number of terms with the same operations over matrices and vectors of same dimensions. However, if the relative encoding is gaussian, then the first three terms from equation 6 can be ignored, and computing $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}$ boils down to vector inner product, $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}} = \boldsymbol{v}^T\boldsymbol{r}_{\boldsymbol{\delta}}$, which has complexity $O(D_p)$. Computing the whole $\boldsymbol{A}^{relative}$ matrix requires doing the same computation N^2 times, so its computational complexity is $O(N^2D_p)$. Now that the $\boldsymbol{A}^{relative}$ is found, to get $\boldsymbol{O}^{relative} = \text{softmax}(\boldsymbol{A}^{relative})\boldsymbol{Z}\boldsymbol{W}_v$, we need to perform the softmax operation, and two more matrix multiplications, that are exactly the same as in the absolute encoding part, which take $O(N^2)$, $O(N^2D_x)$ and $O(ND_x^2)$, respectively. the total computation complexity is then $O(N^2D_p + N^2D_x + ND_x^2)$, which, if we assume that $Dp \ll D_x$, boils down to $\boxed{O(N^2D_x + ND_x^2)}$.

c.1)

The self-attention matrix can be separated "horizontally" into blocks that correspond to each head of the attention unit. Likewise, the $\boldsymbol{W}_{out}$ matrix can be "vertically" separated into the corresponding blocks of the appropriate height, such that block $\boldsymbol{W}_{out[(h-1)D_o+1:hD_o,:]}$ is multiplied by head $\boldsymbol{O}^{(h)}$, and the results of this block multiplication are summed.

$$
\begin{aligned}
\text{MultiHead-SelfAttention}(\boldsymbol{Z}) &= \underset{h\in[N_h]}{\text{concat}}\left[\text{SelfAttention}_h(\boldsymbol{Z})\right]\boldsymbol{W}_{out} + \boldsymbol{b}_{out} \\
&= \underset{h\in[N_h]}{\text{concat}}\left[\text{softmax}(\boldsymbol{A}^{(h)})\boldsymbol{Z}\boldsymbol{W}_v^{(h)}\right]\boldsymbol{W}_{out} + \boldsymbol{b}_{out} \\
&= \underset{h\in[N_h]}{\text{concat}}\left[\boldsymbol{O}^{(h)}\right]\boldsymbol{W}_{out} + \boldsymbol{b}_{out} \\
&= \sum_{h\in[N_h]} \text{softmax}(\boldsymbol{A}^{(h)})\boldsymbol{Z}\underbrace{\boldsymbol{W}_v^{(h)}\boldsymbol{W}_{out[(h-1)D_o+1:hD_o,:]}}_{\boldsymbol{W}^{(h)}} + \boldsymbol{b}_{out} \\
&= \sum_{h\in[N_h]} \text{softmax}(\boldsymbol{A}^{(h)})\boldsymbol{Z}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \\
\boldsymbol{W}^{(h)} &= \boxed{\boldsymbol{W}_v^{(h)}\boldsymbol{W}_{out[(h-1)D_o+1:hD_o,:]}}
\end{aligned}
$$

Selecting $\boldsymbol{q}$-th entries requires just selecting the corresponding entries in the softmax at-

tention:

$$\text{MultiHead-SelfAttention}(\boldsymbol{Z})_{\boldsymbol{q},:} = \left(\underset{h\in[N_h]}{\text{concat}}\left[\boldsymbol{O}^{(h)}\right]\right)_{\boldsymbol{q},:} \boldsymbol{W}_{out} + \boldsymbol{b}_{out} \tag{1}$$

$$= (\underset{h\in[N_h]}{\text{concat}}\left[\boldsymbol{O}^{(h)}_{\boldsymbol{q},:}\right] \boldsymbol{W}_{out} + \boldsymbol{b}_{out} \tag{2}$$

$$= \sum_{h\in[N_h]} \text{softmax}(\boldsymbol{A}^{(h)})_{\boldsymbol{q},:}\boldsymbol{Z}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{3}$$

$$= \sum_{h\in[N_h]} \text{softmax}(\boldsymbol{A}^{(h)}_{\boldsymbol{q},:})\boldsymbol{Z}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{4}$$

$$= \boxed{\sum_{h\in[N_h]} \left(\sum_{\boldsymbol{k}} \text{softmax}(\boldsymbol{A}^{(h)}_{\boldsymbol{q},:})_{\boldsymbol{k}}\boldsymbol{Z}_{\boldsymbol{k},:}\right)\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out}} \tag{5}$$

c.2)

We can rewrite the defining equation for convolution, so that it uses the previously introduced vector indexing with $\boldsymbol{q}$ instead of scalars (i, j). Additionally, the pixel shift (δ_2, δ_2) is rewritten using the vector notation $\boldsymbol{\delta} = (\delta_1.\delta_2)$:

$$\text{Convolution}(\boldsymbol{Z})_{i,j,:} = \sum_{(\delta_1,\delta_2)\in\Delta_K} \boldsymbol{Z}_{i+\delta_1,j+\delta_2,:}\mathbf{W}^{\text{conv}}_{\delta_1,\delta_2,:,:} + \boldsymbol{b}_{out}$$

$$\text{Convolution}(\boldsymbol{Z})_{\boldsymbol{q},:} = \boxed{\textstyle\sum_{\Delta\in\Delta_K} \boldsymbol{Z}_{\boldsymbol{q}+\Delta,:}\mathbf{W}^{\text{conv}}_{\Delta,:,:} + \boldsymbol{b}_{out}}$$

To prove that the multi-head self-attention for a chosen pixel $\boldsymbol{q}$ is equal to the convolution, we start from equation 5, and utilizing the softmax assumption from Theorem 1, we get:

$$\text{MultiHead-SelfAttention}(\boldsymbol{Z})_{\boldsymbol{q},:} = \sum_{h\in[N_h]} \underbrace{\left(\sum_{\boldsymbol{k}} \text{softmax}(\boldsymbol{A}^{(h)}_{\boldsymbol{q},:})_{\boldsymbol{k}}\boldsymbol{Z}_{\boldsymbol{k},:}\right)}_{=1,\text{ for k such that } f(h)=\boldsymbol{q}-\boldsymbol{k}} \boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{6}$$

$$= \sum_{h\in[N_h]} \underbrace{\text{softmax}(\boldsymbol{A}^{(h)}_{\boldsymbol{q},:})_{\boldsymbol{q}-f(h)}}_{1} \boldsymbol{Z}_{\boldsymbol{q}-f(h),:}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{7}$$

$$= \sum_{h\in[N_h]} \boldsymbol{Z}_{\boldsymbol{q}-f(h),:}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{8}$$

$$= \sum_{h\in[N_h]} \boldsymbol{Z}_{\boldsymbol{q}+f(h),:}\boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{9}$$

$$= \sum_{\Delta\in\Delta_K} \boldsymbol{Z}_{\boldsymbol{q}+\Delta,:}\boldsymbol{W}^{(f^{-1}(\Delta))} + \boldsymbol{b}_{out} \tag{10}$$

$$= \sum_{\Delta\in\Delta_K} \boldsymbol{Z}_{\boldsymbol{q}+\Delta,:}\mathbf{W}^{\text{conv}}_{\underbrace{f(h)}_{\Delta},:,:} + \boldsymbol{b}_{out} \tag{11}$$

$$= \boxed{\sum_{\Delta\in\Delta_K} \boldsymbol{Z}_{\boldsymbol{q}+\Delta,:}\mathbf{W}^{\text{conv}}_{\Delta,:,:} + \boldsymbol{b}_{out}} \tag{12}$$

The transition from equation 9 to equation 10 is possible thanks to the symmetry of the shifts in the set Δ_K. The equality between equation 10 and equation 11 hold because of the bijective map $f : [N_h] \to \Delta_K$, where we utilized the function to switch the sum from going over the head indices h to going over the shifts $\Delta = f(h)$. Moreover, here it can be seen that a mapping can be made between attention head weights and the convolutional filters: $\boldsymbol{W}^{(h)} = \boldsymbol{W}^{f^{-1}(h)} = \mathbf{W}^{\text{conv}}_{f(h),:,:}$.

d.1)

Following are the expressions that I will use later in the proof.

$$\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}} = -\alpha \left(\|\delta - \Delta\|^2 + c \right), \delta = \boldsymbol{q} - \boldsymbol{k}$$
$$\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{j}} = -\alpha \left(\|q - j - \Delta\|^2 + c \right), j \neq k$$

The following is the proof that when α is large, the attention of the model for a given query pixel will focus only on one pixel:

$$\begin{aligned}
\lim_{\alpha\to\infty} \text{softmax}(\boldsymbol{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} &= \lim_{\alpha\to\infty} \frac{\exp \boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}}{\exp(\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{k}}) + \sum_{\boldsymbol{j}\neq\boldsymbol{k}} \exp(\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{j}})} \\
&= \lim_{\alpha\to\infty} \frac{\exp(-\alpha c)}{\exp(-\alpha c) + \sum_{\boldsymbol{j}\neq\boldsymbol{k}} \exp(\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{j}})} \\
&= \lim_{\alpha\to\infty} \frac{\exp(-\alpha c)}{\exp(-\alpha c) + \sum_{\boldsymbol{j}\neq\boldsymbol{k}} \exp(-\alpha\|\boldsymbol{k} - \boldsymbol{j}\|^2 - \alpha c)} \\
&= \lim_{\alpha\to\infty} \frac{1}{1 + \sum_{\boldsymbol{j}\neq\boldsymbol{k}} \exp(-\alpha\|\boldsymbol{k} - \boldsymbol{j}\|^2)} \\
&= \frac{1}{1 + \sum_{\boldsymbol{j}\neq\boldsymbol{k}} \lim_{\alpha\to\infty} \exp(-\alpha\|\boldsymbol{k} - \boldsymbol{j}\|^2)} \\
&= \frac{1}{1 + 0} \\
&= \boxed{1}
\end{aligned}$$

Since it holds that $\sum_{\boldsymbol{k}'} \text{softmax}(\boldsymbol{A}_{\boldsymbol{q},:})_{\boldsymbol{k}'} = 1$, all other values $\lim_{\alpha\to\infty} \text{softmax}(\boldsymbol{A}_{\boldsymbol{q},:})_{\boldsymbol{j}},\ \boldsymbol{j} \neq \boldsymbol{k}$ necessarily must be 0.

d.2)

This is proven by just expanding the expression for $\boldsymbol{A}^{relative}_{\boldsymbol{q},\boldsymbol{k}}$ and applying the property of the 2-norm.

$$
\begin{aligned}
\boldsymbol{A}^{relative}_{\boldsymbol{q},\boldsymbol{k}} &= \boldsymbol{v}^T \boldsymbol{r}_{\boldsymbol{\delta}} \\
&= -\alpha^{(h)} \begin{pmatrix} 1 \\ -2\Delta_1^{(h)} \\ -2\Delta_2^{(h)} \end{pmatrix}^T \begin{pmatrix} \|\boldsymbol{\delta}\|^2 \\ \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \end{pmatrix} \\
&= -\alpha^{(h)} (\|\boldsymbol{\delta}\|^2 - 2\boldsymbol{\delta}_1 \boldsymbol{\Delta}_1^{(h)} - 2\boldsymbol{\delta}_2 \boldsymbol{\Delta}_2^{(h)}) \\
&= -\alpha^{(h)} \left(\|\delta\|^2 - 2\delta^T \Delta\right) \\
&= -\alpha^{(h)} \left(\|\delta\|^2 + \|\Delta\|^2 - 2\delta^T \Delta - \|\Delta\|^2\right) \\
&= -\alpha^{(h)} \left(\|\delta - \Delta\|^2 - \|\Delta\|^2\right) \\
&= -\alpha^{(h)} \left(\|\delta - \Delta\|^2 + c\right), \boxed{c = -\|\Delta\|^2}
\end{aligned}
$$

e.1)

In order to keep the dimensions of the resulting images the same, as well as the pixel values equal, the image needs to be padded with $\lfloor K/2 \rfloor$ zeros on each side. Here it is assumed that K is the kernel size of the convolutional filter.

e.2)

Yes, a multi-head self-attention layer can express an arbitrarily-dilated convolution, because each head is not limited in terms of the pixels (values at pixel shifts) it attends to. So there exist configurations in which attention heads attend to pixels that form a pattern equivalent to dilated convolution.

e.3)

Yes, a multi-head self-attention layer can learn to simulate a strided convolution, provided that a pooling layer is applied to the result of the multi-head self-attention layer's output, in order to reduce dimensions to the dimensions that correspond to the actual strided convolution output dimensions and to ignore values that would be skipped in the strided convolution due to the stride. The reasoning is similar to the one in section , as heads can attend to values at different pixel shifts.

Question 2

At the beginning of training, all of the heads are mostly locally focused, not spread out and positioned randomly. As time goes on, some of the heads from the lower layers tend to spread their attention to a larger area, while keeping the centers near the query pixel.

On the higher layers, it happens that some heads form a circle of attention that is more narrow, and their centers are spread out farther from the query pixel. However, in all layers, there is a present tendency of the attention heads to cover a larger area with attention than at initialization time. As the behavior to support the Main Theorem, I would expect the centers of attention of all heads to form a shape of a kernel that a convolution operation is performed with, by the end of training.

https://colab.research.google.com/drive/1JkRZJ31R5kLTw0VOWsc9SzKlw-BlRBqO?usp=sharing