

On the Relationship between Self-Attention and Convolutional Layers

Jean-Baptiste Cordonnier, Andreas Loukas, Martin Jaggi

Presented by Yenson Lau — Layer 6 AI

August 11th, 2021

- Attention can simultaneously attend to every word in an input sequence
- NN architectures using self-attention (SA) only (without convolution) can compete with SA + convolutional architectures on vision tasks
- Do SA layers process images in a similar manner to convolutional layers?

- A constructive theoretical proof that SA can express convolutional layers using relative positional encoding:

Main theorem. *A multi-head self-attention (MHSA) layer with N_h heads of dimension D_h each, final output dimension D_{out} , and a relative positional encoding of dimension $D_p \geq 3$ can express any 2D convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ and $\min(D_h, D_{out})$ output channels.*

- Experiments showing that the first few layers of DNNs using SA do behave similarly to the theoretical construction.

Self-Attention

- Let $\mathbf{X} \in \mathbb{R}^{T \times D_{in}}$ be an input matrix consisting of T tokens in D_{in} dimensions. A self-attention layer $\text{SA} : D_{in} \rightarrow D_{out}$ is expressed as

$$\text{SA}(\mathbf{X})_{t,:} \doteq \text{softmax}(\mathbf{A}_{t,:}) \mathbf{X} \mathbf{W}_{val}, \quad (1)$$

$$T \begin{bmatrix} \text{SA}(\mathbf{x}) \end{bmatrix}^{D_{out}} = T \begin{bmatrix} \text{softmax}(\mathbf{A}_{1,:}) \\ \vdots \\ \text{softmax}(\mathbf{A}_{T,:}) \end{bmatrix}^T \begin{bmatrix} \mathbf{X} \end{bmatrix}^{D_{in}} \begin{bmatrix} \mathbf{W}_{val} \end{bmatrix}^{D_{out}}$$

- We refer to the elements of the $T \times T$ matrix

$$\mathbf{A} \doteq \mathbf{X} \mathbf{W}_{qry} \mathbf{W}_{key}^T \mathbf{X}^T \quad (2)$$

the *attention scores*, and the softmax outputs as *attention probabilities*.

- So far the learnable parameters are the *query*, *key*, and *value* matrices

$$\mathbf{W}_{qry} \in \mathbb{R}^{D_{in} \times D_k}, \quad \mathbf{W}_{key} \in \mathbb{R}^{D_{in} \times D_k}, \quad \text{and} \quad \mathbf{W}_{val} \in \mathbb{R}^{D_{in} \times D_{out}}.$$

For simplicity, ignore residual connections, normalization or constant factors.

- Note that \mathbf{A} is *permutation equivariant* – shuffling the order of the tokens (rows) in \mathbf{X} shuffles the token scores in \mathbf{A} . *This is not desired behavior.*
 - e.g. "the cat ate the fish" is very different from "the fish ate the cat"
- To overcome this, we can add a (fixed or learned) *positional encoding* $\mathbf{P}_{t,:}$, for each token in the sequence, to the input matrix for computing attention scores

$$\mathbf{A} \doteq (\mathbf{X} + \mathbf{P}) \mathbf{W}_{qry} \mathbf{W}_{key}^T (\mathbf{X} + \mathbf{P})^T, \quad (3)$$

where the encoding matrix \mathbf{P} has size $T \times D_{in}$.

Self-Attention

Relative positional encoding

- In *absolute encoding*, a fixed or learned vector $P_{t,:}$ is assigned to each token t . Therefore, for a query token q (the token to compute the representation of) and key token k (the pixel we are attending to),

$$\begin{aligned} A_{q,k}^{abs} &= (X_{q,:} + P_{q,:}) W_{qry} W_{key}^T (X_{q,:} + P_{q,:})^T \\ &= X_{q,:} W_{qry} W_{key}^T X_{k,:}^T + X_{q,:} W_{qry} W_{key}^T P_{k,:}^T \\ &\quad + P_{q,:} W_{qry} W_{key}^T X_{k,:}^T + P_{q,:} W_{qry} W_{key}^T P_{k,:}^T \end{aligned} \quad (4)$$

- *Relative positional encoding* instead considers the positional difference between the query and key tokens:

$$A_{q,k}^{rel} = X_{q,:}^T W_{qry} W_{key}^T X_{k,:} + X_{q,:}^T W_{qry} \hat{W}_{key} r_{\delta} + u^T W_{key} X_{k,:}^T + v^T \hat{W}_{key} r_{\delta}. \quad (5)$$

- The learnable vectors u, v are unique for each head
- The relative positional encoding $r_{\delta} \in \mathbb{R}^{D_p}$ depends only on the shift $\delta \doteq k - q$, and is shared by all layers and heads.
- Key weights are split into W_{key} for the input and \hat{W}_{key} for positional encoding.
- Note the attention scores are now *shift equivariant* rather than permutation equivariant. This is the desired behavior for convolutional / vision tasks.

Multi-Head Self-Attention (MHSA)

- In practice it is beneficial to replicate the SA mechanism into multiple heads, by concatenating the output of N_h heads of output dimension D_h and projecting it to dimension D_{out} .

$$\text{MHSA}(\mathbf{X}) \doteq \text{hcat}_{h \in [N_h]} [\text{SA}_h(\mathbf{X})] \mathbf{W}_{out} + \mathbf{b}_{out}. \quad (6)$$

Here $\mathbf{W}_{out} \in \mathbb{R}^{N_h D_h \times D_{out}}$ is the projection matrix and $\mathbf{b}_{out} \in \mathbb{R}^{D_{out}}$ is a bias term.

A handwritten diagram on a blue grid background illustrating the dimensions of the MHSA equation. The equation is written as:

$$\overset{D_{out}}{\text{MHSA}(\mathbf{X})} = \left[\overset{N_h \times D_h}{\text{SA}_1(\mathbf{X})} \mid \dots \mid \text{SA}_{N_h}(\mathbf{X}) \right] \overset{D_{out}}{\mathbf{W}_{out}} + \mathbf{b}_{out}$$

The diagram includes handwritten annotations for dimensions: D_{out} above the first vector, $N_h \times D_h$ above the concatenated matrix, and D_{out} above the \mathbf{W}_{out} vector. A vertical label $N_h \times D_h$ is placed to the left of the \mathbf{W}_{out} vector.

Self-Attention for Images

From 1D sequence to 2D

- Replace query and key tokens with pixels $\mathbf{q}, \mathbf{k} \in [W] \times [H]$, and the input with $\mathbf{X} \in \mathbb{R}^{W \times H \times D}$. Each attention score now associates a query and key pixel.
- For a pixel $\mathbf{p} \in (i, j)$, $\mathbf{X}_{\mathbf{p},:} \doteq \mathbf{X}_{i,j,:}$ and $\mathbf{A}_{\mathbf{p},:} \doteq \mathbf{A}_{i,j,:}$.
- Then, analogously to the 1D case,

$$\text{SA}(\mathbf{A})_{\mathbf{q},:} = \sum_k \text{softmax}(\mathbf{A}_{\mathbf{q},:})_k \mathbf{X}_{\mathbf{k},:} \mathbf{W}_{val}, \quad (7)$$

and relative positional encoding retains the same form as Equation (5).
Similarly MHSA retains the same form as Equation (6).

Self-Attention for Images

Convolution

- Given an image tensor $\mathbf{X} \in \mathbb{R}^{W \times H \times D_{in}}$ and kernel tensor $\mathbf{W} \in \mathbb{R}^{K \times K \times D_{in} \times D_{out}}$,

$$\text{Conv}(\mathbf{X})_{i,j,:} \doteq \sum_{(\delta_1, \delta_2) \in \Delta_K} \mathbf{X}_{i-\delta_1, j-\delta_2, :} \mathbf{W}_{\delta_1, \delta_2, :, :} + \mathbf{b} \in \mathbb{R}^{D_{out}}, \quad (8)$$

where

$$\Delta_K \doteq \left[-\left\lfloor \frac{K}{2} \right\rfloor, \dots, \left\lfloor \frac{K}{2} \right\rfloor \right] \times \left[-\left\lfloor \frac{K}{2} \right\rfloor, \dots, \left\lfloor \frac{K}{2} \right\rfloor \right]$$

is the set of all shifts represented by a $K \times K$ kernel.

- We depart from the original notation slightly by using the *unflipped* convolution. In the paper, the summand is flipped to $\mathbf{X}_{i+\delta_1, i+\delta_2, :} \mathbf{W}_{\delta_1, \delta_2, :, :}$. The theorem in the paper is not changed by this.

Self-Attention as a Convolutional Layer

Main Theorem

- Main theorem.** A multi-head self-attention (MHSA) layer with N_h heads of dimension D_h each, final output dimension D_{out} , and a relative positional encoding of dimension $D_p \geq 3$ can express any 2D convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ and $\min(D_h, D_{out})$ output channels.

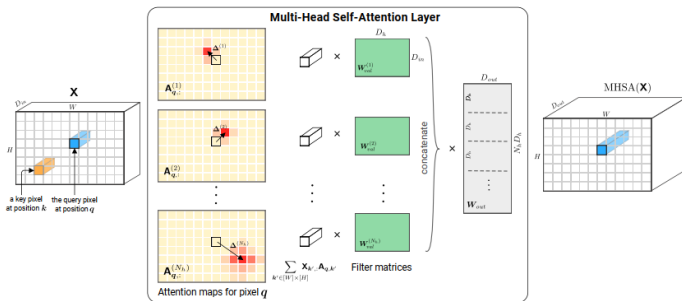


Figure 1: Illustration of a Multi-Head Self-Attention layer applied to a tensor image \mathbf{X} . Each head h attends pixel values around shift $\Delta^{(h)}$ and learn a filter matrix $\mathbf{W}_{val}^{(h)}$. We show attention maps computed for a query pixel at position q .

- The theorem is a consequence of two lemmas.

Self-Attention as a Convolutional Layer

Lemma 1

- **Lemma 1.** Consider a MHSA layer consisting of $N_h = K^2$ heads, $D_h \geq D_{out}$ and let $\mathbf{f} : [N_h] \rightarrow \Delta_K$ be a bijective mapping of heads onto shifts. Further, suppose that for every head

$$\text{softmax}(\mathbf{A}_{q,:}^{(h)})_k = \begin{cases} 1 & \text{if } \mathbf{f}(h) = \mathbf{q} - \mathbf{k}, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Then for any convolutional layer with a $K \times K$ kernel and D_{out} output channels, there exists $\{\mathbf{W}_{val}^{(h)}\}_{h \in [N_h]}$ such that $\text{MHSA}(\mathbf{X}) = \text{Conv}(\mathbf{X})$, for any kernel tensor $\mathbf{W} \in \mathbb{R}^{K \times K \times D_{in} \times D_{out}}$, and for all $\mathbf{X} \in \mathbb{R}^{W \times H \times D_{in}}$.

- *Proof of Lemma 1.* Rewrite Equation (6) as

$$\text{MHSA}(\mathbf{X}) = \mathbf{b}_{out} + \sum_{h \in [N_h]} \text{softmax}(\mathbf{A}^{(h)}) \mathbf{X} \underbrace{\mathbf{W}_{val}^{(h)} \mathbf{W}_{out}^{(h)}}_{\mathbf{W}^{(h)}}. \quad (10)$$

Here $\mathbf{W}_{val}^{(h)} \in \mathbb{R}^{D_{in} \times D_h}$ is the value matrix for head h , or in MATLAB notation,

$$\mathbf{W}_{out}^{(h)} = \mathbf{W}[(h-1) * D_h + 1 : hD_h, :] \in \mathbb{R}^{D_h \times D_{out}}.$$

Self-Attention as a Convolutional Layer

Lemma 1

- *Proof of Lemma 1 (cont).* Consider a single "query" pixel from $\text{MHSA}(\mathbf{X})$:

$$\text{MHSA}(\mathbf{X})_{q,:} = \mathbf{b}_{out} + \sum_{h \in [N_h]} \left(\sum_{k \in [T]} \text{softmax}(\mathbf{A}_{q,:}^{(h)})_k \mathbf{X}_{k,:} \right) \mathbf{W}^{(h)}. \quad (11)$$

The summand $\sum_k \text{softmax}(\mathbf{A}_{q,:}^{(h)})_k \mathbf{X}_{k,:}$ is a weighted average on the rows of \mathbf{X} .

- Applying the assumption from Equation (9), each of the softmax weights pick out a single row $k = q - f(h)$:

$$\text{MHSA}(\mathbf{X})_{q,:} = \mathbf{b}_{out} + \sum_{h \in [N_h]} \mathbf{X}_{q-f(h),:} \mathbf{W}^{(h)}. \quad (12)$$

If we set $K = \sqrt{N_h}$, then it is clearly possible to index all the shifts using $h \in [N_h]$, i.e. $f([N_h]) = \Delta_K$. Therefore we can rewrite the expression as

$$\text{MHSA}(\mathbf{X})_{q,:} = \mathbf{b}_{out} + \sum_{h \in [N_h]} \mathbf{X}_{q-f(h),:} \mathbf{W}_{f(h),:,} = \text{Conv}(\mathbf{X})_{q,:}. \quad \square \quad (13)$$

Self-Attention as a Convolutional Layer

Lemma 1

- Lemma 1 says that, with the right choice of relative positional encoding (and a very restricted set of attention parameters), *a MHSA layer is exactly equivalent to a convolutional layer*.
- The number of linearly independent filters expressible by \mathbf{W} is clearly limited by $\min(D_h, D_{out})$. Therefore, to express D_h convolutional filters, it is best to simply let $D_{out} = N_h D_h$ and have $\mathbf{W}^{(h)} \in \mathbb{R}^{D_h \times D_h}$.
- There is a generalized version of this lemma in the Appendix of the paper that delves into the space of filters spannable for a given set of dimensions D_h, N_h, D_{out} , etc., for interested readers.

Self-Attention as a Convolutional Layer

Lemma 2

- **Lemma 2.** *There exists a relative encoding scheme $\{\mathbf{r}_\delta \in \mathbb{R}^{D_p}\}_{\delta \in \mathbb{Z}^2}$ with $D_p \geq 3$ and parameters \mathbf{W}_{qry} , \mathbf{W}_{key} , $\hat{\mathbf{X}}_{key}$, \mathbf{u} with $D_p \leq D_k$ such that, for every $\Delta \in \Delta_K$ there exists some vector $\mathbf{v}(\Delta)$ that yields $\text{softmax}(\mathbf{A}_{q,:})_k = 1$ if $k - q = \Delta$, and zero otherwise.*
- *Proof of Lemma 2.* Start with the relative positional encoding of Equation (5):

$$\mathbf{A}_{q,k}^{rel} = \mathbf{X}_{q,:}^T \mathbf{W}_{qry} \mathbf{W}_{key}^T \mathbf{X}_{k,:} + \mathbf{X}_{q,:}^T \mathbf{W}_{qry} \hat{\mathbf{W}}_{key} \mathbf{r}_\delta + \mathbf{u}^T \mathbf{W}_{key} \mathbf{X}_{k,:} + \mathbf{v}^T \hat{\mathbf{W}}_{key} \mathbf{r}_\delta.$$

Since the desired attention probabilities (9) from Lemma 1 are independent of the input \mathbf{X} , set $\mathbf{W}_{key} = \mathbf{W}_{qry} = 0$.

This leaves only the final term. Setting $\hat{\mathbf{W}}_{key} = [\mathbf{I}_{D_p} ; 0] \in \mathbb{R}^{D_k \times D_p}$ (recall $D_p \leq D_k$) yields

$$\mathbf{A}_{q,k} = \mathbf{v}_{1:D_p}^T \mathbf{r}_\delta.$$

Self-Attention as a Convolutional Layer

Lemma 2

- *Proof of Lemma 2 (cont).* Recall that $\delta \doteq \mathbf{k} - \mathbf{q}$. Now suppose some $\mathbf{v}, \mathbf{r}_\delta$ could be found such that

$$\mathbf{A}_{\mathbf{q}, \mathbf{k}} = -\alpha(\|\delta - \Delta\|^2 + c) \quad (14)$$

so that the maximum score $-\alpha c$ is achieved when $\delta = \Delta$, then

$$\begin{aligned} \lim_{\alpha \rightarrow \infty} \text{softmax}(\mathbf{A}_{\mathbf{q}, \cdot})_k &= \lim_{\alpha \rightarrow \infty} \frac{e^{-\alpha(\|\delta - \Delta\|^2 + c)}}{\sum_{\delta' \in \Delta_K} e^{-\alpha(\|\delta' - \Delta\|^2 + c)}} \\ &= \frac{1_{\delta = \Delta}}{1 + \lim_{\alpha \rightarrow \infty} \sum_{\delta' \neq \delta} e^{-\alpha\|\delta' - \Delta\|^2}} = 1_{\delta = \Delta}. \end{aligned}$$

- Choosing $\mathbf{v}_{1:D_p} = -\alpha(1, -2\Delta_1, -2\Delta_2)$ and $\mathbf{r}_\delta = (\|\delta\|^2, \delta_1, \delta_2)$, yields

$$\begin{aligned} \mathbf{A}_{\mathbf{q}, \mathbf{k}} &= \mathbf{v}_{1:D_p}^T \mathbf{r}_\delta \\ &= -\alpha(\|\delta\|^2 - 2\Delta_1\delta_1 - 2\Delta_2\delta_2) \\ &= -\alpha(\|\delta - \Delta\|^2 - \|\Delta\|^2). \end{aligned}$$

Setting $c = -\|\Delta\|^2$ recovers Equation (14) and completes the proof. □

Self-Attention as a Convolutional Layer

Lemma 2

- The encoding scheme satisfying Lemma 2 is a *quadratic encoding scheme*, and is achieved by setting

$$\begin{aligned} \mathbf{v}^{(h)} &\doteq -\alpha^{(h)}(1, -2\Delta_1^{(h)}, -2\Delta_2^{(h)}), \\ \mathbf{r}_\delta &\doteq (\|\delta\|^2, \delta_1, \delta_2), \\ \mathbf{W}_{qry} = \mathbf{W}_{key} &\doteq \mathbf{0}, \\ \hat{\mathbf{W}}_{key} &\doteq \mathbf{I}. \end{aligned} \tag{15}$$

Here the learned parameters $\Delta^{(h)} = (\Delta_1^{(h)}, \Delta_2^{(h)})$ and $\alpha^{(h)}$ determine the center and width of attention of each head, and $\delta = (\delta_1, \delta_2)$ is fixed and expresses the relative shift between query and key pixels.

- Although the proof requires $\alpha \rightarrow \infty$ to satisfy the assumption (9) of Lemma 1, finite precision arithmetic performs hard attention with sufficiently large enough α . E.g. for Float32, set $\alpha \geq 46$.
- The lemma, and thus the theorem, can be extended in a straightforward manner to cover K -dimensional convolutions, with $D_p = K + 1$.

- **Baseline.** Standard ResNet18 on CIFAR-10
- **Model.**
 - 6 MHSA layers. Each layer consists of a MHSA step, followed by dense, dropout and LayerNorm sublayers.
 - Use 2x2 invertible downsampling to reduce image size (attention coefficients scale quadratically to image size)
 - Fixed size representation of input image is the average pooling of the last layer representations, and fed to a linear classifier
- **MHSA variations.** Different types of relative positional encoding

$$\mathbf{A}_{q,k}^{rel} = \mathbf{X}_{q,:}^T \mathbf{W}_{qry} \mathbf{W}_{key}^T \mathbf{X}_{k,:} + \mathbf{X}_{q,:}^T \mathbf{W}_{qry} \hat{\mathbf{W}}_{key} \mathbf{r}_{\delta} + \mathbf{u}^T \mathbf{W}_{key} \mathbf{X}_{k,:} + \mathbf{v}^T \hat{\mathbf{W}}_{key} \mathbf{r}_{\delta}.$$

- **SA with quadratic embedding:** Retain final term only and fix the variables using Equation (15). The attention widths $\alpha^{(h)}$ and centers $\Delta^{(h)}$ are still learnt.
- **SA with learned embedding:** Retain final term only but learn \mathbf{v} , $\hat{\mathbf{W}}_{key}$, \mathbf{r}_{δ} , with $D_p = D_{out} = 400$. Set $D_h = D_{out}$.
- **SA with content-based attention:** All terms retained (might actually be just first two terms) and all variables learnable. Same dimensions as above.

Experiments

Convergence / testing accuracy

- **ResNet converges faster:**
Probably because SA's inductive bias is not as strong as ResNet, but may also be due to different optimization setup.
- SA models with more learnable parameters converge slower and ends up with lower testing accuracy.

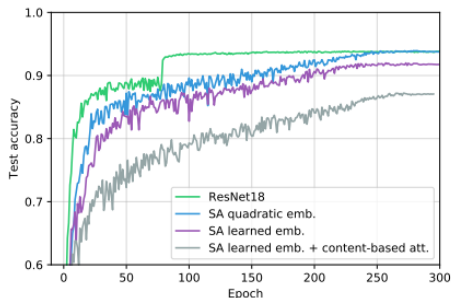


Figure 2: Test accuracy on CIFAR-10.

Experiments

Do self-attention layers actually learn convolutions? Maybe.

- **ResNet converges faster:**
Probably because SA's inductive bias is not as strong as ResNet, but may also be due to different optimization setup.
- SA models with more learnable parameters converge slower and ends up with lower testing accuracy.

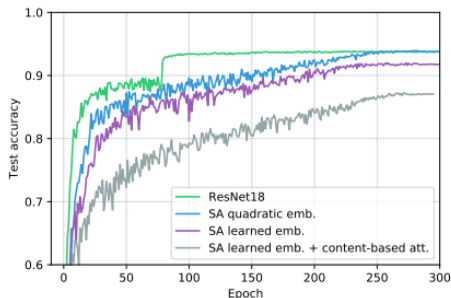


Figure 2: Test accuracy on CIFAR-10.

Thanks!