Are Transformers Universal Approximators of Seq2Seq Functions?

Chulhee Yun[†], Srinadh Bhojanapalli[‡], Ankit Singh Rawat[‡], Sashank J. Reddi[‡], Sanjiv Kumar[‡]

[†]MIT, [‡]Google Research NYC

Transformers

- Transformers enable state-of-the-art performance in various NLP tasks. E.g., BERT, GPT2, XLNet, ...
- A **Transformer block** $t^{h,m,r}$ is a map from $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$, defined by two key components:

Self-attention layer

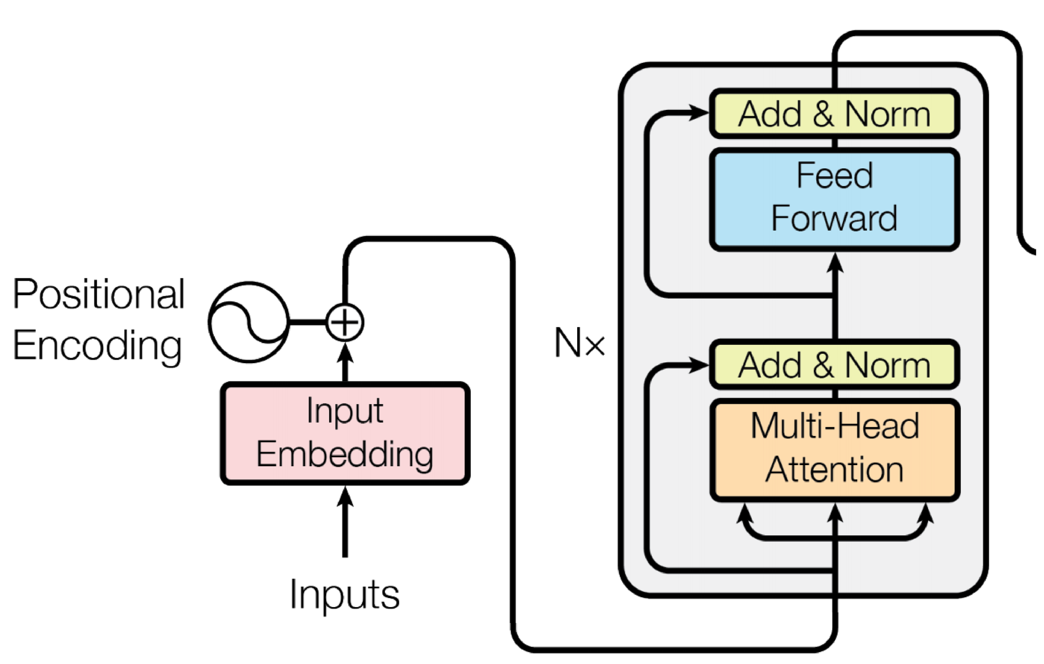
$$\mathrm{Attn}(X) = X + W_o \begin{bmatrix} \mathrm{head}_1(X) \\ \vdots \\ \mathrm{head}_h(X) \end{bmatrix}$$

where

$\mathrm{head}_i(X) = (W_v^i X) \cdot \mathrm{softmax}[(W_k^i X)^T (W_q^i X)]$,
$W_o \in \mathbb{R}^{d\times hm}$, $W_v^i, W_k^i, W_q^i \in \mathbb{R}^{m\times d}$

Token-wise feed-forward layer

$\mathrm{FF}(X) = \mathrm{Attn}(X) + W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{Attn}(X))$,
$W_1 \in \mathbb{R}^{r\times d}$, $W_2 \in \mathbb{R}^{d\times r}$



Our Contributions

What is the class of sequence-to-sequence functions that Transformers can represent?

- Due to parameter sharing, it is not clear if Transformers can represent arbitrary sequence-to-sequence functions.

We show that a Transformer network with large enough depth is able to approximate any ***continuous*** *and* ***permutation equivariant*** *function up to arbitrary accuracy.*

Definitions

- For a function $f : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ is **permutation equivariant** if for any permutation matrix $P \in \mathbb{R}^{n\times n}$, $f(XP) = f(X)P$.
- For a matrix $A \in \mathbb{R}^{d\times n}$, we denote the entry-wise ℓ^p norm of A as $\|A\|_p$.
- Define function distance

$$\mathrm{d}_p(f,g) := \left(\int \|f(X) - g(X)\|_p^p dX\right)^{1/p}.$$

Main Theorems

$$\mathcal{F}_{\mathrm{PE}} := \{f : \mathbb{R}^{d\times n} \circlearrowleft \mid f \text{ is continuous, permutation equivariant, and has compact support.}\},$$
$$\mathcal{T}^{h,m,r} := \{g : \mathbb{R}^{d\times n} \circlearrowleft \mid g \text{ is a composition of Transformer blocks } t^{h,m,r}.\}$$

Theorem 1. A Transformer network with **constant width** and **large enough depth** can approximate any function $f \in \mathcal{F}_{\mathrm{PE}}$ to **arbitrary accuracy**; i.e., let $1 \le p < \infty$ and $\epsilon > 0$, then for any given $f \in \mathcal{F}_{\mathrm{PE}}$, there exists a Transformer network $g \in \mathcal{T}^{2,1,4}$ such that $\mathrm{d}_p(f,g) \le \epsilon$.

If the network has positional encoding, one can **remove** permutation equivariance:

$$\mathcal{F}_{\mathrm{CD}} := \{f : \mathbb{D} \to \mathbb{R}^{d\times n} \mid f \text{ is continuous on a compact domain } \mathbb{D} \subset \mathbb{R}^{d\times n}.\}$$
$$\mathcal{T}_{\mathrm{P}}^{h,m,r} := \{g_{\mathrm{P}}(X) = g(X+E) \mid g \in \mathcal{T}^{h,m,r} \text{ and } E \in \mathbb{R}^{d\times n} \text{ is learnable.}\}$$

Theorem 2. Let $1 \le p < \infty$ and $\epsilon > 0$, then for any given $f \in \mathcal{F}_{\mathrm{CD}}$, there exists a Transformer network $g \in \mathcal{T}_{\mathrm{P}}^{2,1,4}$ such that we have $\mathrm{d}_p(f,g) \le \epsilon$.

Contextual Mappings

Def. Consider a finite set $\mathbb{L} \subset \mathbb{R}^{d\times n}$. A map $q : \mathbb{L} \to \mathbb{R}^{1\times n}$ defines a **contextual mapping** if the map satisfies the following:

1. For any $L \in \mathbb{L}$, the n entries in $q(L)$ are all distinct.
2. For any $L, L' \in \mathbb{L}$, with $L \neq L'$, all entries of $q(L)$ and $q(L')$ are distinct.

Self-attention layers can implement a permutation equivariant contextual mapping!

Proof Sketch

- Approximate $f \in \mathcal{F}_{\mathrm{PE}}$ with support $[0,1]^{d\times n}$ by a **piecewise constant** function $\overline{f}$ on δ-cubes at each $L \in \mathbb{G} = \{0, \delta, 2\delta, \dots, 1-\delta\}^{d\times n}$.
- Using FF, **quantize** $[0,1]^{d\times n}$ to the grid $\mathbb{G}$.
- Using Attn, implement a ***perm. equivariant*** **contextual mapping** $q(L)$ on almost all elements of $\mathbb{G}$.
  - If l_1, l_2, l_3 are distinct, then q_1, q_2, q_3 are also distinct.
  - $(l_1, l_2, l_3) \mapsto (q_1, q_2, q_3)$, $(l_3, l_1, l_2) \mapsto (q_3, q_1, q_2)$
  - $(l_1, l_2, l_3) \mapsto (q_1, q_2, q_3)$, $(l_1', l_2', l_3') \mapsto (q_1', q_2', q_3')$, $(l_1, l_2, \tilde{l}_3) \mapsto (\tilde{q}_1, \tilde{q}_2, \tilde{q}_3)$.
- The function q maps even *slightly different* contexts to *completely different* numbers.
- Using FF, map each unique number $q(L)_{:,k}$ to the desired output value $\overline{f}(L)_{:,k}$.

Alternative Architectures

- Realizing contextual mappings is sufficient to enable universal approximation property.
- Cheaper architectures to implement some forms of contextual mappings?

Bi-linear projection

$$\mathrm{BProj}(X) = X + W_O \cdot X \cdot W_P$$

- For random W_P, $(X_1 - X_2)W_P$ is dense for sparse $X_1 - X_2 \implies$ a form of "pairwise contextual mapping."

Depth-wise separable convolutions

$$\mathrm{SepConv}(X) = X + W_O (X * W_C),$$

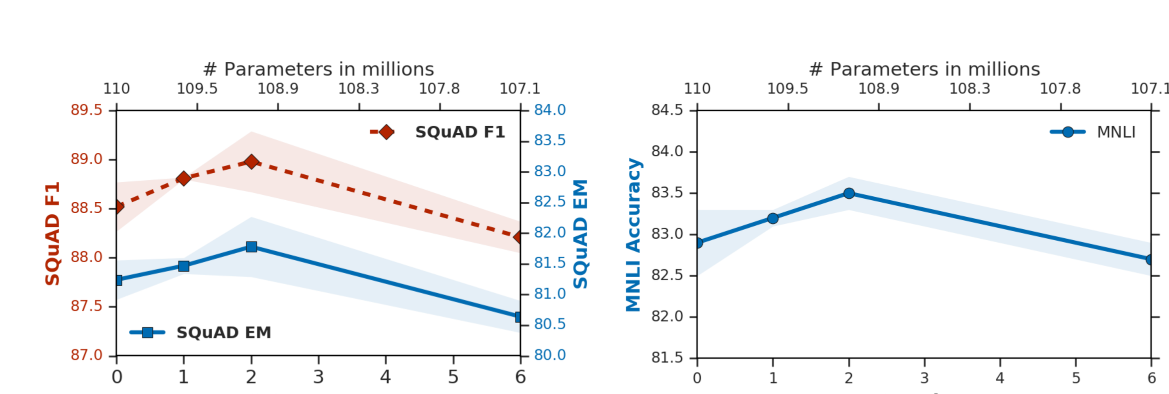
where $W_C \in \mathbb{R}^{d\times k}$, $(X * W_C)_{i,:} := X_{i,:} * (W_C)_{i,:}$.

Performance comparison on $\mathrm{BERT}_{\mathrm{BASE}}$.

Architecture	Avg. Attention	BProj	SepConv	Transformer
# params	88.3M	90M	102.5M	110M
Masked LM acc. (%)	28	59	60	63
MNLI acc. (%)	66	72.3	73	78.2

Hybrid Transformers

- Modify the first few Transformer blocks.
- Replace self-attention layers with depth-wise separable convolutional layers.



(a) SQuAD (b) MNLI