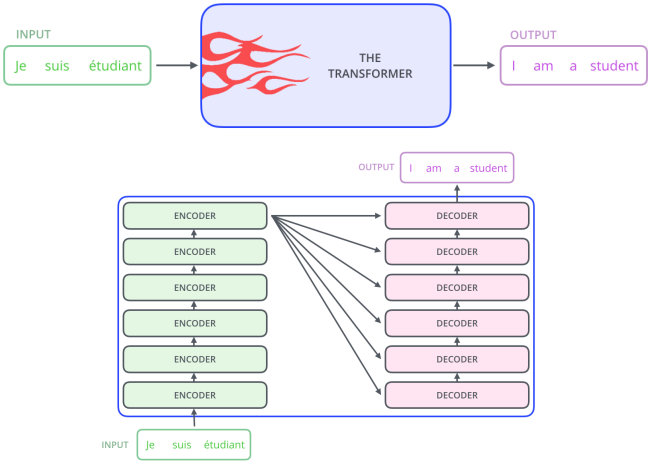
Dissecting Attention: A Numerical Analyst's Perspective

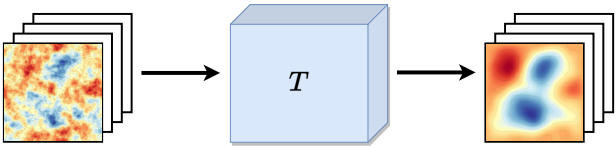
SCML Seminar KAUST, April 2024

What is a Transformer?



The Transformer is a deep neural network architecture to solve the machine translation problem in Natural Language Processing. Source: Jay Alammar. *The Illustrated Transformers*.

Tensor2tensor

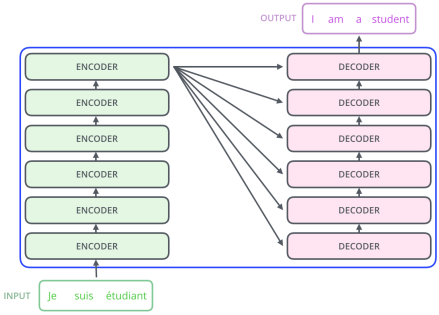


- **seq2seq** or **tensor2tensor** in Neural Machine Translation maps various sized matrices to matrices of the same size.
- A **tensor2tensor** DNN parametrizes the following discrete map for $\mathcal{H} = (L^2(\Omega))^d$

$$T_\theta : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}, \quad T : \mathcal{H} \to \mathcal{H}.$$

- Sentence in one language, embedded into high dimensional spaces, "translated" to another language's embedding after stacking multiple layers of the same module.
- Columns: numbers of latent/embedding dimension/channels (fixed in a given layer). Row: token embedding, patch embedding, or a DoF's embedding.
- The model can be trained on a lower "resolution" (n small) and evaluated at a higher "resolution" ($n_{\text{eval}} \geq n$).

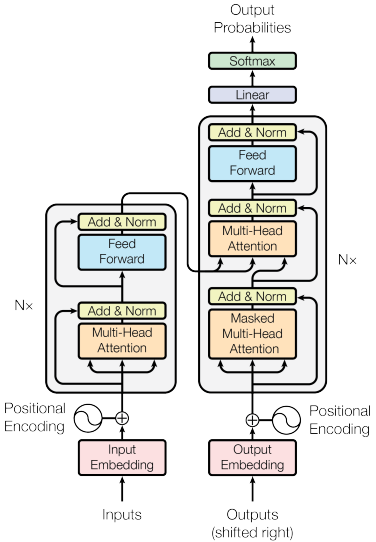
What is a Transformer?



A Transformer consists of a sequence of encoder blocks with *identical* architectures, and decoder blocks with *identical* architectures. Source: Jay Alammar. *The Illustrated Transformers*.

- Like RNN in Neural Machine Translation, the Transformer block in each layer has the ***same*** architecture (and the number of parameters).
- Unlike CNN, after the initial embedding layer (from words to vector), the latent representations propagated in the hidden layers are of the ***same*** discretization size.

What is a Transformer?



Keywords:

- "Multi-head Attention".
- "Feedforward": fully connected MLP with shared weights at each position.
- "Add": skip-connection $x \mapsto x + f(x)$.
- "Norm": layer normalization (a learned diagonal column-scaling of the latent representation).
- "Positional Encoding": a hard-coded mapping to encode different positions in different latent dimensions.

What is Multi-head Attention?

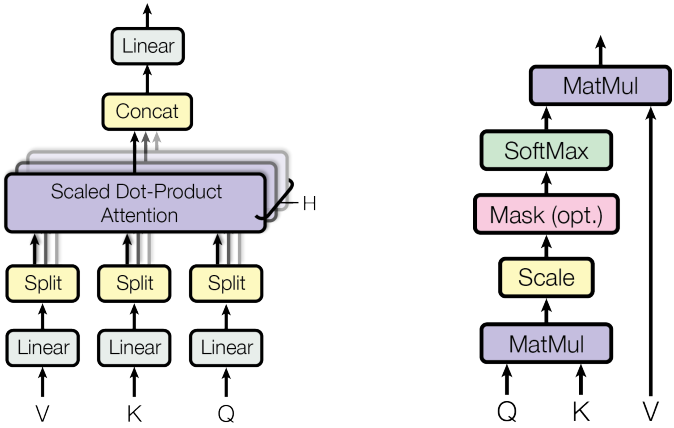
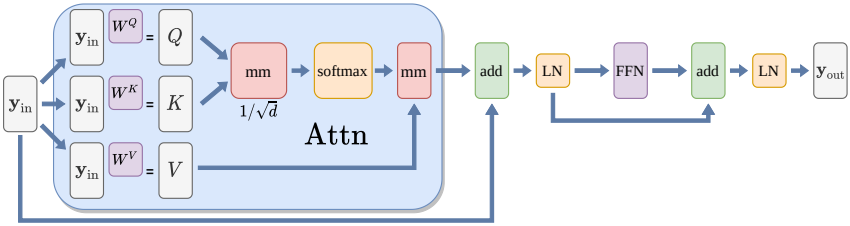


Image source: (Left) Multi-head Attention mapping. (Right) Scaled dot-product attention $\mathrm{Softmax}(QK^\top)V$. Figure 2 in *Attention Is All You Need.*

Single-head Self-Attention

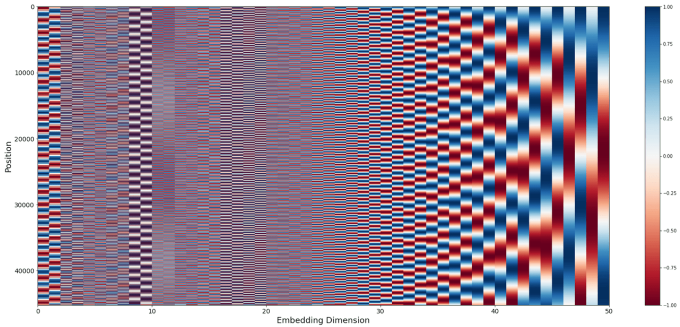


Self-attention mechanism in the classical Transformer in a single attention head.

- $\boldsymbol{y}_{\text{in}}, \boldsymbol{y}_{\text{out}} \in \mathbb{R}^{n \times d}$, input/output embeddings; positional encodings added.
- Latent representations: query Q, key K, value V generated by 3 learnable matrices $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$: $Q = \boldsymbol{y}W^Q$, $K = \boldsymbol{y}W^K$, $V = \boldsymbol{y}W^V$.
- The scaled dot-product attention: $\text{Attn}_s(\boldsymbol{y}) := \text{Softmax}\left(d^{-1/2} QK^\top\right) V$.
- The full attention operator (add&norm, feedforward) is then

$$\text{Attn} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}, \quad \boldsymbol{z} = \boldsymbol{y} + \text{Attn}_s(\boldsymbol{y}), \quad \boldsymbol{y} \mapsto \text{Ln}\left(\boldsymbol{z} + g\Big(\text{Ln}(\boldsymbol{z})\Big)\right).$$

Positional embedding



PE from *Attention is All You Need*.

- Positional embedding (PE): $\boldsymbol{p} \in \mathbb{R}^{n \times d}$ has the same dimension with the latent representation, and $\boldsymbol{y} \mapsto \boldsymbol{y} + \boldsymbol{p}$ for the $\boldsymbol{y}$ right after the input embedding.
- M: maximum discretization size; c: channel index.

$$\boldsymbol{x}_{(i,c)} = \sin\left(\frac{i}{M^{c/d}}\right) \text{ if } c \text{ is even; } \boldsymbol{x}_{(i,2c+1)} = \cos\left(\frac{i}{M^{(c-1)/d}}\right) \text{ if } c \text{ is odd}$$

Positional embeddings

Different interpretations of the latent representation (Query/Key/Value) which is an $\mathbb{R}^{n\times d}$ matrix.

- Row: a high-dimensional embedding (vector representation) of a token.
- Column: certain discretization of "basis"[1] or "frame"[2] .

Open problems: Positional embeddings

- What role exactly does PE plays in attention?
- How PE shapes the topological structure of the latent representation space?
- How to design "nice" problem-oriented PE to achieve problem-specific attributes of traditional models?
- Is PE "$\approx$" coordinates? ViT: Dosovitskiy et al. (2021); DeiT: Touvron et al. (2021); Swin: Liu et al. (2021).

[1] L. Lu et al. (2021). "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature machine intelligence*.

[2] F. Bartolucci et al. (2023). "Representation Equivalent Neural Operators: a Framework for Alias-free Operator Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Single-layer Single-head Self-Attention

The full self-attention operator: n: discretization size, d: latent dimensions

$$\mathrm{Attn} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}, \quad \boldsymbol{y} \mapsto \boldsymbol{y}_{\mathsf{out}}.$$

$\mathrm{Attn}(\cdot)$ consists the following operations:

- Adding PE (can be learnable): $Pe : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$, $\boldsymbol{y} \mapsto \boldsymbol{y} + \boldsymbol{p}(\theta)$
- Scaled dot-product attention: $Q = \boldsymbol{y}W^Q$, $K = \boldsymbol{y}W^K$, $V = \boldsymbol{y}W^V$

$$\Big(\mathrm{Attn}_s(\boldsymbol{y})\Big)_i = \sum_{j=1}^{n} \frac{\kappa(\boldsymbol{q}_i, \boldsymbol{k}_j)}{\sum_{\ell=1}^{n} \kappa(\boldsymbol{q}_i, \boldsymbol{k}_{j'})} \boldsymbol{v}_j$$

where

$$\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+, \quad \kappa(\boldsymbol{q}_i, \boldsymbol{k}_j) = \exp(\boldsymbol{q}_i \cdot \boldsymbol{k}_j),$$

notice $\kappa(\boldsymbol{q}_i, \boldsymbol{k}_j) = \Big(\boldsymbol{y}W^Q(\boldsymbol{y}W^K)^\top\Big)_{ij}$.

Single-layer Single-head Self-Attention

The full self-attention operator:

$$\mathrm{Attn} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}, \quad \boldsymbol{z} = \boldsymbol{y} + \mathrm{Attn}_s(\boldsymbol{y}), \quad \boldsymbol{y} \mapsto \mathrm{Ln}\left(\boldsymbol{z} + g\Big(\mathrm{Ln}(\boldsymbol{z})\Big)\right).$$

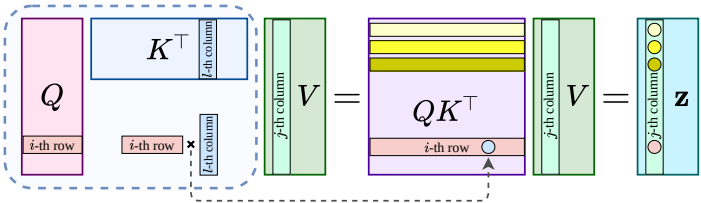
- Ln: layer normalization (LN), which has $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$ learnable as follows

$$\mathrm{Ln}(\boldsymbol{y}) := \frac{\boldsymbol{y} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta},$$

$$\boldsymbol{\mu} := \frac{1}{d}\sum_{j=1}^{d} \boldsymbol{y}^j \in \mathbb{R}^n, \quad \boldsymbol{\sigma}^2 := \frac{1}{d}\sum_{j=1}^{d} (\boldsymbol{y}^j - \boldsymbol{\mu}) \odot (\boldsymbol{y}^j - \boldsymbol{\mu}) \in \mathbb{R}^n.$$

- Note all LN operations are done in an element-wise fashion. After LN is applied, each position in the discretization will roughly follow normal distribution.
- $g(\cdot)$: a *pointwise* feedforward neural net with a "bottleneck" architecture in the base Transformer model. $g(\cdot)$ has width d and is applied at each z_j $(j = 1, \ldots, n)$.

Single-layer Single-head Self-Attention

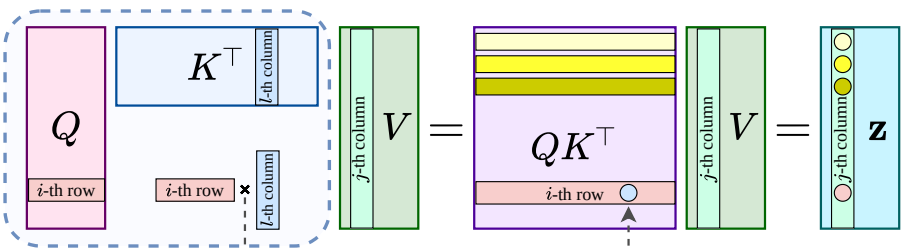


(Still) Open problem

What exactly is the mechanics of the attention mechanism?

- Kernel interpretation, RKHS: Tsai et al. (2019), Wright and Gonzalez (2021), Zhang et al. (2022).
- Fourier (change of basis): Li et al. (2021), Nguyen, Pham, et al. (2022).
- Low-rank or sparse: Y. Xiong et al. (2021), Nguyen, Suliafu, et al. (2021a), Tay et al. (2020), Han et al. (2022).
- Random feature interpretation: Choromanski et al. (2021), Peng et al. (2021).
- Iterative "solver": Yu et al. (2023)

Scaled Dot-product Attention



$$(\boldsymbol{z}_i)_j = h\ \mathrm{Softmax}(QK^\top)_{i\bullet}\,\boldsymbol{v}^j = hm_i^{-1}\exp\left(\boldsymbol{q}_i\cdot\boldsymbol{k}_1,\ldots,\boldsymbol{q}_i\cdot\boldsymbol{k}_\ell,\ldots,\boldsymbol{q}_i\cdot\boldsymbol{k}_n\right)^\top\cdot\boldsymbol{v}^j$$

$$= hm_i^{-1}\sum_{\ell=1}^{n}\exp(\boldsymbol{q}_i\cdot\boldsymbol{k}_\ell)(\boldsymbol{v}^j)_\ell \approx m^{-1}(x_i)\int_\Omega \kappa(x_i,\xi)v_j(\xi)\mathrm{d}\xi,$$

The i-th row in the output computes approx. an integral transform with a non-symmetric normalized learnable low-rank "kernel" function $\kappa(x,\xi)$

$$z(x) \approx \lambda v(x) + m^{-1}(x)\int_\Omega \kappa(x,\xi;\theta)v(\xi;\theta)\mathrm{d}\xi, \qquad \text{where } \boldsymbol{q}_i = q(x_i), \boldsymbol{k}_i = k(x_i), \boldsymbol{v}_i = v(x_i)$$

Nonlocal Methods

Buades, Coll, and Morel[3] proposed that the denoising filter should depend on the signal! This is the earliest prototype of attention.

$$\mathrm{NLM}[u](x) \;=\; \frac{1}{C(x)} \int_\Omega \exp\!\left(-\frac{1}{h^2} \int_\Omega G_\alpha(t) \left| u(x+t) - u(y+t) \right|^2 dt \right) u(y)\, dy,$$

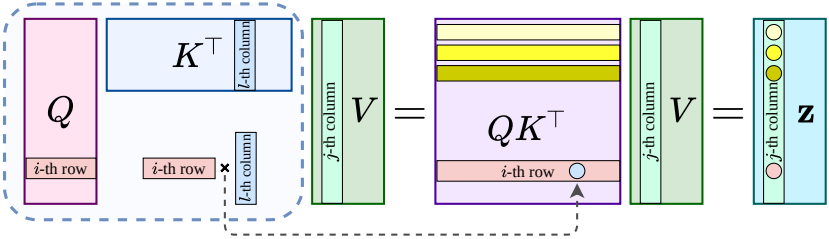
$$C(x) \;=\; \int_\Omega \exp\!\left(-\frac{1}{h^2} \int_\Omega G_\alpha(t) \left| u(x+t) - u(y+t) \right|^2 dt \right) dy.$$

Later this is generalized in papers by Osher and his postdoc using a variational perspective.[4]

[3] A. Buades, B. Coll, and J.-M. Morel (2005). "A review of image denoising algorithms, with a new one". In: *Multiscale modeling & simulation*.

[4] G. Gilboa and S. Osher (2007). "Nonlocal linear image regularization and supervised segmentation". In: *Multiscale Modeling & Simulation*; G. Gilboa and S. Osher (2009). "Nonlocal operators with applications to image processing". In: *Multiscale Modeling & Simulation*.

Making Attention more efficient

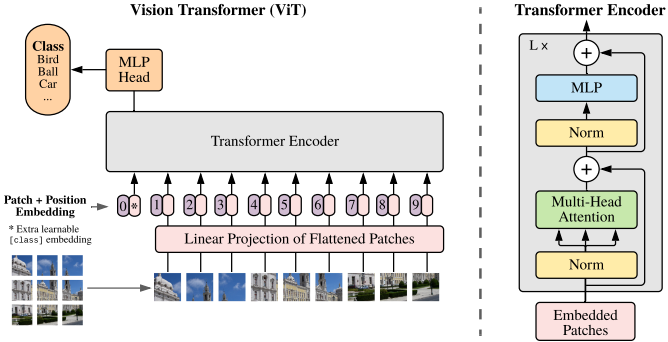


- The positive attention kernel $\mathrm{Softmax}(QK^T)$ characterizes how each position's latent representation vector interact.
- This can be replaced by a simple Fourier kernel[5].
- Computational cost of QK^T scales quadratically with respect to the number of positions n.

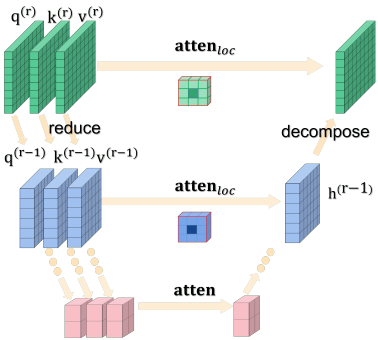
[5] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon (2022). "FNet: Mixing Tokens with Fourier Transforms". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Making Attention more efficient

- Low-rank: Nyströmformer (Y. Xiong et al. (2021)), Fast-Multipole method (Nguyen, Suliafu, et al. (2021b)).
- Sparsification: Child, Gray, Radford, and Sutskever (2019), locality-based feature maps in Reformer Kitaev, Kaiser, and Levskaya (2020),
- Linearization: Wang et al. (2020), RNN interpretation (Katharopoulos, Vyas, Pappas, and Fleuret (2020)).
- Patchify: ViT (Dosovitskiy et al. (2021)).



Hierarchically Nested Attention Neural Operator

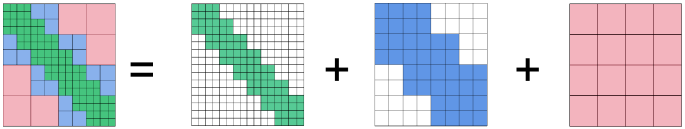


- Use local (windowed) attention to get the representation on each level:

$$\textbf{atten}_{\text{loc}}^{(m)} : \boldsymbol{v}_i^{(m)} = \sum_{j \in \mathcal{N}^{(m)}(i) \cup i} \mathcal{G}(\boldsymbol{q}_i^{(m)}, \boldsymbol{k}_j^{(m)}) \boldsymbol{v}_j^{(m)},$$

$\mathcal{N}^{(m)}(i)$ contains m-th level neighbors of the i-th position in the discrete grid.

Hierarchically Nested Attention Neural Operator



- Aggregate the attention matrix (interaction) spanning multilevels:

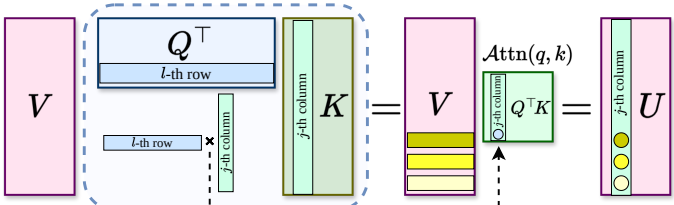
$$\begin{bmatrix} \vdots \\ \boldsymbol{h}_i^{(r)} \\ \vdots \end{bmatrix} = \left(\sum_{m=1}^{r-1} (\mathbf{D}^{(r-1),\top} \cdots \mathbf{D}^{(m),\top} \mathbf{G}_{\text{loc}}^{(m)} \mathbf{R}^{(m)} \cdots \mathbf{R}^{(r-1)}) + \mathbf{G}_{\text{loc}}^{(r)} \right) \begin{bmatrix} \vdots \\ \boldsymbol{v}_i^{(r)} \\ \vdots \end{bmatrix}.$$

where the overall attention matrix on the finest level can be decomposed as

$$\mathbf{G}_h := \sum_{m=1}^{r-1} (\mathbf{D}^{(r-1),\top} \cdots \mathbf{D}^{(m),\top} \mathbf{G}_{\text{loc}}^{(m)} \mathbf{R}^{(m)} \cdots \mathbf{R}^{(r-1)}) + \mathbf{G}_{\text{loc}}^{(r)},$$

Galerkin-type Attention

- While it makes sense to ask the kernel to be positive (similarity between rows), it does not to ask the interaction between bases (columns) to be positive.



$$(\boldsymbol{z}^j)_i = z_j(x_i) = h \sum_{\ell=1}^{d} (\boldsymbol{k}^\ell \cdot \boldsymbol{v}^j)(\boldsymbol{q}^\ell)_i \approx \sum_{\ell=1}^{d} \left(\int_\Omega k_\ell(\xi) v_j(\xi) \mathrm{d}\xi \right) q_\ell(x_i).$$

- Resembles a (learnable) Petrov-Galerkin projection.
- How latents interact is similar to the Channel Attention[6].

[6] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon (2018). "CBAM: Convolutional block attention module". In: *Proceedings of the European conference on computer vision (ECCV)*.

Galerkin-type Attention

Consider i-th entry in the j-th column $\boldsymbol{z}^j$ of $\mathbf{z}$, which is the inner product of the i-th row of Q and the j-th column of $K^\top V$:

$$(\boldsymbol{z}^j)_i = h\, \boldsymbol{q}_i^\top \cdot (K^\top V)_{\bullet j}$$

$$\boldsymbol{z}^j = h \begin{pmatrix} | & | & | & | \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n \\ | & | & | & | \end{pmatrix}^\top (K^\top V)_{\bullet j} = h \left((K^\top V)_{\bullet j}^\top \begin{pmatrix} — & \boldsymbol{q}^1 & — \\ — & \vdots & — \\ — & \boldsymbol{q}^d & — \end{pmatrix} \right)^\top$$

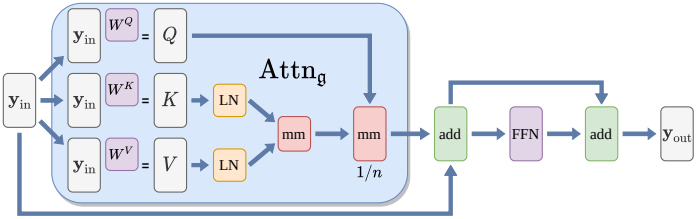
then

$$\boldsymbol{z}^j = h \sum_{\ell=1}^{d} \boldsymbol{q}^l (K^\top V)_{\ell j}, \quad \text{where } (K^\top V)_{\bullet j} = \left(\boldsymbol{k}^1 \cdot \boldsymbol{v}^j, \boldsymbol{k}^2 \cdot \boldsymbol{v}^j, \cdots, \boldsymbol{k}^d \cdot \boldsymbol{v}^j\right)^\top.$$

Rewriting $\langle v_j, k_l \rangle := (K^\top V)_{\ell j}$

$$z_j(x) := \sum_{\ell=1}^{d} \langle v_j, k_l \rangle\, q_l(x), \quad \text{for } j = 1, \cdots, d, \quad \text{and } x \in \{x_i\}_{i=1}^n,$$

Galerkin-projection inspired Attention



- Inspired by the Fourier transform to remove the softmax normalization to improve the computational efficiency: orthogonal $\{q_j(\cdot)\}_{j=1}^d$

$$\min_{a_i} \left\| f - \sum_{i=1}^{d} a_i q_i(\cdot) \right\|^2_{\ell^2(\Omega)}, \quad \text{and} \quad z(x) := \sum_{\ell=1}^{d} \frac{(f, q_\ell)}{(q_\ell, q_\ell)} q_\ell(x),$$

- Inspired by the Gram matrix inverse normalization in the proof of the Ceá type lemma, and layer normalization modifications[7].

[7] R. Xiong et al. (2020). "On layer normalization in the transformer architecture". In: *International Conference on Machine Learning*. PMLR.

A preliminary result on the Galerkin-type Attention

Theorem (Approximation capacity of a single layer of Galerkin attention [8])

$\mathbb{Q}_h \subset \mathcal{Q}$ and $\mathbb{V}_h \subset \mathcal{V}$ are the current approximation space, suppose there exists a continuous key-to-value map that is bounded below on the discrete approximation space, i.e., the functional norm of $v \mapsto b(q, v)$ is bounded below for any q, then for g_θ consists a Galerkin attention composed with a channel reduction map

$$\min_\theta \|f - g_\theta(\mathbf{y})\| \leq \underbrace{c^{-1}}_{\|b(q,\cdot)\|_{\mathbb{V}_h'} \geq c} \underbrace{\min_{q \in \mathbb{Q}_h} \max_{v \in \mathbb{V}_h} \frac{|b(\Pi f - q, v)|}{\|v\|}}_{\text{(Error of the Petrov-Galerkin projection)}} + \underbrace{\|f - \Pi f\|}_{\text{(Consistency)}}.$$

- Intepretation: for a "query" (a function in a Hilbert space), to deliver the best approximator in "value" (trial space), the "key" space (test space) has to be big enough so that for every value there is a key to unlock it.
- discrete Ladyzhenskaya-Babuška-Brezzi inf-sup condition: why Transformers have capacity to generalize so well with respect to the length of the sequence.

[8] C. (2021). "Choose a Transformer: Fourier or Galerkin". In: *Advances in Neural Information Processing Systems (NeurIPS)*

Sketch of the proof

- For the continuous bilinear form $b(\cdot,\cdot) : \mathcal{Q} \times \mathcal{V} \to \mathbb{R}$, $b(q,\cdot) : v \mapsto b(q,v)$ is bounded below on $\mathbb{V}_h \subset \mathcal{V}$ for any $q \in \mathbb{Q}_h$:

$$c\|q\|_{\mathcal{H}} \leq \sup_{v \in \mathbb{V}_h} \frac{|b(q,v)|}{\|v\|_{\mathcal{V}}}.$$

  The Riesz map by $b(\cdot,\cdot)$ from the value space to the key space is injective (or key-to-value is surjective). We can verify c is independent of the sequence length for the scaled dot-product attention (without softmax).

- Consider an incoming function f's projection in $\mathbb{Q}_h$: f_h (query). By the inf-sup condition above,

$$\|f_h - g_\theta(\mathbf{y})\|_{\mathcal{H}} \leq c^{-1} \sup_{v \in \mathbb{V}_h} \frac{|b(f_h - g_\theta(\mathbf{y}), v)|}{\|v\|_{\mathcal{V}}}.$$

- The rest is just to show

$$\min_{\theta} \max_{v \in \mathbb{V}_h} \frac{|b(f_h - g_\theta(\mathbf{y}), v)|}{\|v\|_{\mathcal{V}}} \leq \min_{q \in \mathbb{Q}_h} \max_{v \in \mathbb{V}_h} \frac{|b(f_h - q, v)|}{\|v\|_{\mathcal{V}}}.$$

Sketch of the proof

- Solving the min-max problem

$$\min_{q\in\mathbb{Q}_h}\max_{v\in\mathbb{V}_h}\frac{|\langle\Phi f_h, v\rangle_h - b(q,v)|}{\|v\|_{\mathcal{H}}}$$

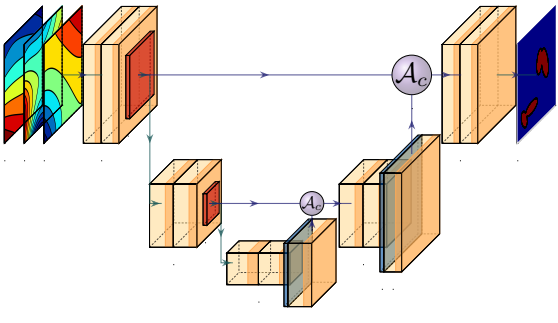
is equivalent to solving the following saddle point problem:

$$\begin{cases}\langle w, v\rangle_h + b(p,v) = \langle\Phi f_h, v\rangle_h, & \forall v\in\mathbb{V}_h,\\ b(q,w) = 0, & \forall q\in\mathbb{Q}_h.\end{cases}$$

- Lastly, the scaled dot-product attention has capacity to represent this best approximator's vector representation $\boldsymbol{p}$: let $\Lambda = \mathrm{blkdiag}\left\{(BM^{-1}B^{\top})^{-1}, 0\right\}$, B and M be the Gram (mass) matrices associated with $b(\cdot,\cdot)$ and $\langle\cdot,\cdot\rangle_h$

$$\begin{aligned}\widetilde{Q} &:= \mathbf{y}\widetilde{W}^{Q} \leftarrow \mathbf{y}W^{Q}U,\\ \widetilde{K} &:= \mathbf{y}\widetilde{W}^{K} \leftarrow \mathbf{y}W^{Q}U\Lambda,\\ \widetilde{V} &:= \mathbf{y}\widetilde{W}^{V} \leftarrow \mathbf{y}W^{V}M^{-1},\\ \text{and }\ \boldsymbol{p} &= \widetilde{Q}(\widetilde{K}^{T}\widetilde{V})\boldsymbol{\zeta}.\end{aligned}$$

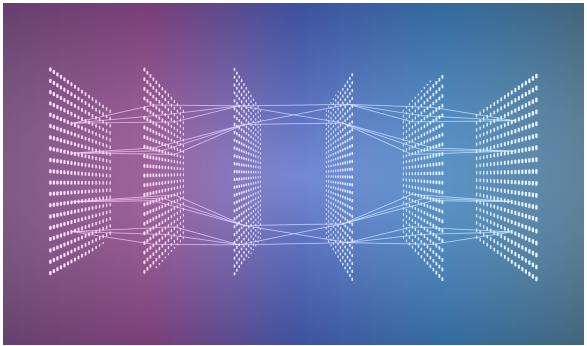
Making linear Attention more efficient



- U-Net meta-architecture [9].
- Input: the concatenation of discretizations of ϕ and $\nabla\phi$.
- Output: the approximation to the index map $\mathcal{I}^D$.
- : 3×3 convolution + ReLU;
- : normalization;
- : interpolation;
- : cross attention from the coarse grid to the fine grid;
- : input and output discretized functions.

[9] O. Ronneberger, P. Fischer, and T. Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer

Representational capcity



- The embedding layer in Transformer constructs an ultra-high-dimensional vector representation of each token in a sentence or each patch in an image.
- In the encoder layer, this (latent) representation interacts with itself nonlinearly to get a "better" representation.
- This interaction can be position-wise (row-wise), or channel-wise (column-wise).

Image source: *Transformers: What They Are and Why They Matter*, Mehreen Saeed.

Representational capcity

Open problem about representational capacity

How to prove the universal representation (approximation) theorem for Transformer when the number of layers increase?

- What is representational power of (stacked) attention layer exactly[10]?
- Random feature model[11]: each channel (column) of the latent representation is similar to an RF-RR model

$$\boldsymbol{f} \mapsto \Phi(\boldsymbol{f};\boldsymbol{\theta}) = \frac{1}{d}\sum_{j=1}^{d} \alpha_j(\boldsymbol{\theta}) g(\boldsymbol{f};\boldsymbol{\theta})$$

where

$$\boldsymbol{\theta} = \operatorname{argmin} \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{u}_i - \Phi(\boldsymbol{f}_i;\theta)\|_V^2 + \text{regularizations}$$

[10] C. Yun et al. (2020). "Are Transformers universal approximators of sequence-to-sequence functions?" In: *International Conference on Learning Representations*.

[11] A. Rahimi and B. Recht (2008). "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning". In: *Advances in neural information processing systems*.

Representational capcity and positional embedding

- PE plays an important role in shaping the representational capacity.
- Learnable PE[12], rotational-invariant PE[13], etc.

Theorem (Universal representater theorem (informal simplified version)[14])

Given fixed n and d, the function class of Transformers $\{u(\boldsymbol{y}) : u(\boldsymbol{y}) = g(\boldsymbol{y} + \boldsymbol{x}),$ where $g := g_\ell \circ \cdots \circ g_1\}$ with the absolute fixed PE $\boldsymbol{x}$ is a universal approximator for continuous functions that map a compact domain in $\mathbb{R}^{n \times d}$ to $\mathbb{R}^{n \times d}$.

Open problem: representational capacity

Can the theoretical results on the approximation capacity of Transformer with different PEs be reflected in specially designed experiments?

[12] J. Gehring et al. (2017). "Convolutional sequence to sequence learning". In: *International conference on machine learning*. PMLR.

[13] J. Su et al. (2021). "Roformer: Enhanced transformer with rotary position embedding". In: *arXiv preprint arXiv:2104.09864*.

[14] S. Luo et al. (2022). "Your Transformer May Not be as Powerful as You Expect". In: *Advances in Neural Information Processing Systems*