Self-Attention Dynamics

MAT1510

November 30, 2023

Experiments in multihead self-attention dynamics based on Geshkovski et al. [2] and their code.

Contents

1 The Emergence of Clusters in Self-Attention Dynamics

Transformer based architectures have widespread success throughout all areas of deep learning and recently in large language modelling with the development of ChatGPT. The essential component of the transformer architecture is the self-attention mechanism, allowing a model to learn and distinguish important characteristics of input data. The self attention model lacks a theoretical underpinning to explain its robust performance, with *The Emergence of Clusters in Self-Attention Dynamics* by Geshkovski et al. [2] being a major contribution towards establishing a rigorous description. Tokens are viewed as interacting particles that are advanced by the self attention matrix, allowing the problem to be placed in a dynamical context to derive limiting geometric representations of tokens.

1.1 Mathematical Setting

This work describes the attention mechanism through discrete time dynamics by viewing passage through layers as a time variable. In ResNets, the passage through t-th parametrized layer f_θ is viewed as

$$\dot{x}(t) = f_\theta(x(t))$$

The residual connection modifies the original input with $x(t) + f_\theta(x(t)) = x(t) + \dot{x}(t)$. In self-attention, the parametrized layer acts on n-tokens $x(t) = (x_1(t), \dots, x_n(t))$ with

$$\dot{x}(t) = (P(t)V) \cdot x(t) \qquad \dot{x_i}(t) = \sum_{j=1}^{n} P_{ij}(t)Vx_j(t) \tag{1}$$

where V is the fixed value matrix which is independent of time. The matrix $P(t)$ is the stochastic self-attention matrix depending on keys K and queries Q:

$$P_{ij}(t) = \frac{e^{\langle Qx_i(t), Kx_j(t)\rangle}}{\sum_{\ell=1}^{n} e^{\langle Qx_i(t), Kx_\ell(t)\rangle}} \quad (i,j) \in [n]^2$$

Q, K, V are fixed and independent of time and fixed. This corresponds to weight sharing during repeated applications of the same self-attention matrix throughout the transformer. These dynamics do not incorporate other essential features of the transformer including multiple heads, feedforward layers, and layer-normalization. The paper describes the *limiting* geometric behaviour of tokens $x_i(t)$ in (1) and the main contributions are conditions for various forms of clustering to emerge. Under assumptions on Q, K, V and as $t \to \infty$ the geometric representations of $x(t)$ the clustering to various objects including hyperplanes and polytopes is described below:

V	**Q, K**	**Limiting Representations**
$V = -I$	$Q = K = I$	cluster at origin
$V = I$	$Q^T K > 0$	vertices of convex polytope
$\lambda_1(V) > 0$	$\langle Q\varphi_1, K\varphi_1\rangle > 0$	3 parallel hyperplanes
$\lVert V^2 x\rVert \geqslant \lVert Vx\rVert^2$	$Q^T K > 0$	product of polytope and subspaces

The emergence of this geometry confirms empirical results about leading tokens in the original *Attention Is All You Need* by Vaswani et al. [4]. An early theorem from the paper builds on past results to mitigate quadratic complexity and is later applied to the ALBERT transformer weights.

1.2 Theorems and Examples

One of the theorems in the paper provides a result about the low-rank asymptotic form of the self-attnetion matrix.

Theorem 1 (Low-rank asymptotics). Let each token be one-dimensional $x_i(t) \in \mathbb{R}$ and (Q, K, V) satisfy $V > 0$, $QK^T > 0$. For any $(x_1(0), \dots, x_n(0))$, there is some low rank matrix P^* such that $P(t) \to P^*$ as $t \to \infty$.

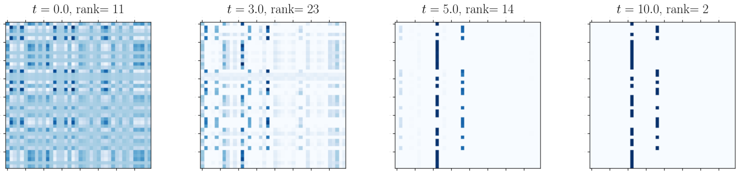


Figure 1: An illustration of asymptotics implied by Theorem 1 for $P(t)$ with $n = 40$ tokens with $Q = K = V = 1$.

Another theorem guarantees clustering towards vertices of convex polytopes. The norm of tokens $x_i(t)$ typically diverges. To account for this, $z_i(t) = e^{-tV}x_i(t)$ are rescaled, motivated by the solution to $\dot{y}(t) = Vy(t) \implies y(t) = e^{tV}y(0)$.

Theorem 2 (Convex polytope asymptotics)**.** Suppose $V = I$ and $Q^TK > 0$. Consider any initial sequence of tokens $z_i(t) \in \mathbb{R}^d$ evolving by

$$\dot{z}_i(t) = \sum_{j=1}^{n} \left(\frac{e^{\langle Qe^{tV}z_i(t), Ke^{tV}z_j(t)\rangle}}{\sum_{\ell=1}^{n} e^{\langle Qe^{tV}z_i(t), Ke^{tV}z_\ell(t)\rangle}} \right) V(z_j(t) - z_i(t)) \tag{2}$$

There exists convex polytope $\mathcal{K} \subset \mathbb{R}^d$ such that for any u, $z_i(t) \to 0$ or $z_i(t) \to \partial\mathcal{K}$ (i.e. a corner) as $t \to \infty$.

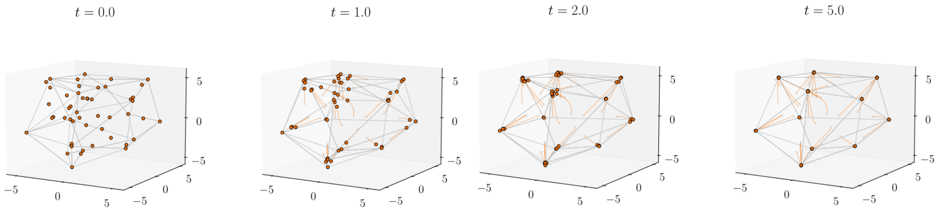


Figure 2: An illustration of Theorem 2 with $n = 40$ tokens with $Q = K = I_3$.

1.3 Themes

These results augment multiple research topics and MAT1510 course themes.

1. **Analysis of attention-based models.** The widespread application of transformers has generated major research interest in the significance of self-attention. The new interacting particle system perspective is promising for further analyses of self-attention and improvements to transformer-based architectures.

2. **Importance of skip connections.** Dong, Cordonnier, and Loukas [1] showed that a lack of skip connections in self-attention yielded trivialized dynamics and a single cluster at the origin.

3. **Quadratic complexity of Transformers.** A major computational challenge is the quadratic complexity of transformers: each self-attention layer has n^2 products $\langle Qx_i, Kx_i\rangle$. Past works have imposed low-rankness in self-attention (*sparse attention*) [5] in order to mitigate the quadratic cost. The transformer model ALBERT uses weight sharing throughout layers with Geshkovski et al. [2] presenting impact of parameter reduction dynamics.

4. **Neural collapse.** The limiting clustered representations and low-rankness of the self-attention matrix share many similarities with the neural collapse phenomenon [3]. The limiting simplex structure provides insight into how self-attention separates different classes.

5. **Clustering in interacting particle systems.** The dynamics presented in this paper are very similar to other non-linear systems modeling clustering. This work is a first application of methods from dynamical systems to rigorously describe trained transformer dynamics.

2 Experiments in Multihead Self-Attention

Geshkovski et al. [2] present a mathematical framework for studying clustering of geometric representations in transformers. The contribution provides existence results for geometric clustering under theoretical conditions on the dynamics and does not address important practical considerations in transformer models.

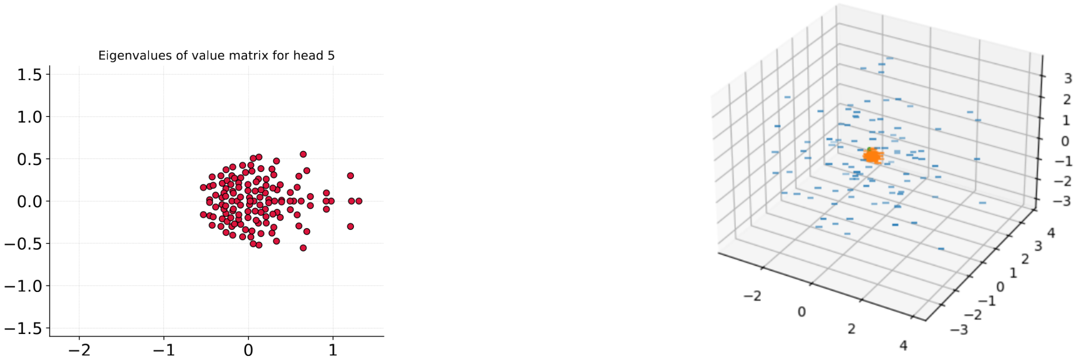
1. **Trained weights**. The use of randomly initialized weights does not elucidate possible data representations induced by clustering: dynamics induced by trained weights may provide further insight into self-attention.
2. **Multi-head self attention**. Multi-head self-attention is essential for large models. The number of heads and their effect on dynamics, clustering patterns and practical considerations require further investigations
3. **Token initialization**. Geshkovski et al. [2] demonstrate existence of clustering patterns for any token initialization. The effect of different token initialization schemes is not emphasized, meanwhile embeddings are trained and are important for self-attention.

2.1 ALBERT Transformer

The ALBERT transformer uses weight sharing across transformer blocks in a BERT architecture. Due to repetitions of same-weight layers, Geshkovski et al. [2] propose that the dynamics of this model are iterated and may demonstrate emergent clustering behaviour.

2.1.1 Single-head Dynamics

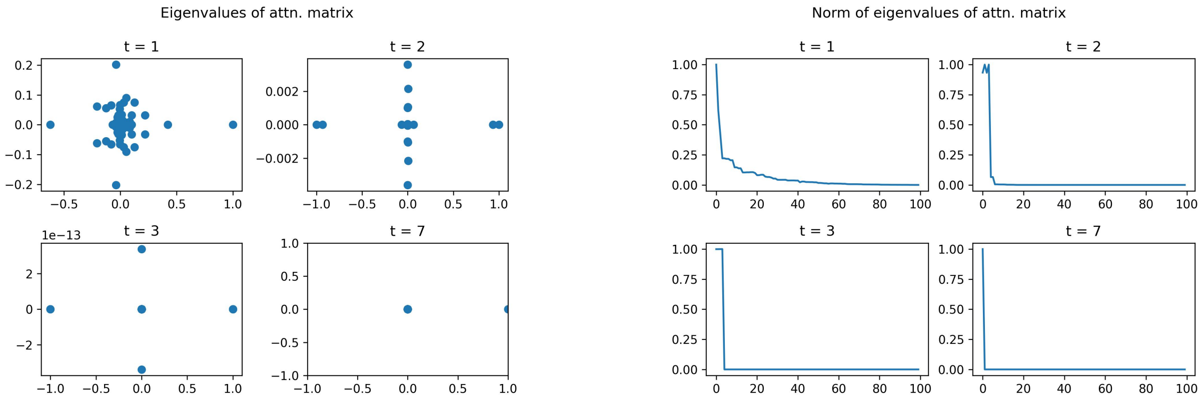
The maximal eigenvalue on head 5 of ALBERT`-xlarge-v2` satisfies the conditions for Theorem 5.2 in [2] (see Figure 3a). The weights W_Q and W_K for this head are passed and the dynamics are iterated in Figure 3b with the same dynamical scheme during which a single cluster emerges. Further analyis of the eigenvalues suggests the emergence of a singular non-zero component during the evolution of self-attention dynamics. The emergence of leading eigenvalues in the self-attention matrix reflects the clustering to a single point. While this is an initial experiment using trained weights, single-head dynamics do not represent the full ALBERT token dynamics.



(a)

(b)

Figure 3: ALBERT single-head dynamics. (a) satisfies conditions of 5.2, while (b) is the PCA of iterated dynamics with the weight matrices of head 5 of ALBERT-xlarge-v2. Dynamics are implemented with RK-4 with $\Delta t = 0.1$, $T = 5$, blue points at $t = 0$, orange points at $t = 25$, green points at $t = 50$. The final green cluster is not centered at the origin.



(a)

(b)

Figure 4: ALBERT single-head dynamics. Eigenvalues of the attention matrix and their norms during self-attention dynamics with weight matrices of head 5 in ALBERT-xlarge-v2.

2.1.2 Multi-head Dynamics

Token dynamics are implemented with ALBERT-xlarge-v2 weights with the evolution governed by

$$\dot{x}(t) = \sum_{h=1}^{16} P_h(t)x(t)$$

where $P_h(t)$ is the self-attention matrix corresponding to the h-head weights. Viewed from the usual notation of multi-head self attention, we assume $W^O = \begin{bmatrix} I & I & \cdots & I \end{bmatrix}^T$, that value weights W_{V_h} are also I. The trained weight multi-head dynamics are applied to various token initializations:

$$(x_1(t), \dots, x_n(t))$$

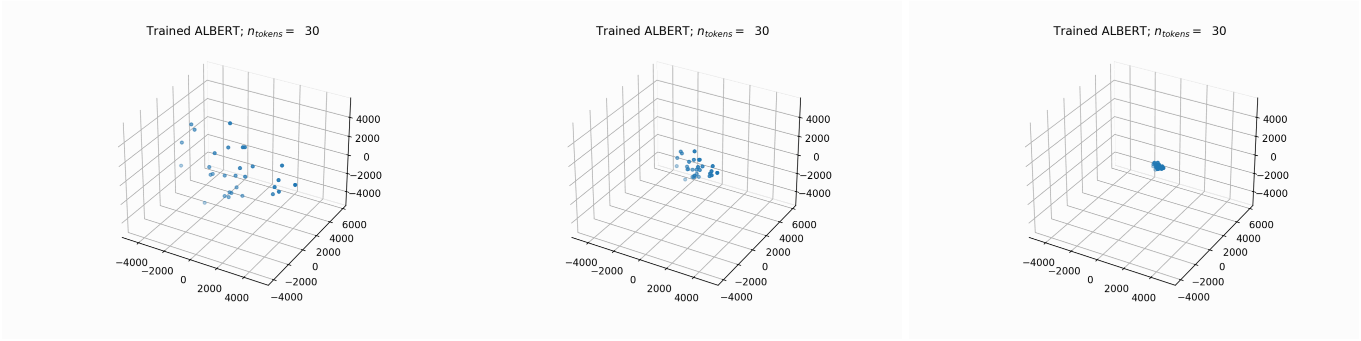
where $x_k(t) \in \mathbb{R}^d$ for d the embedding dimension.

1. **Randomly initialized embedding.** $x_k(t) \in [-5, 5]^d$ are uniformly sampled similarly to [2] for various n. As n increases, dynamics display more complicated clustering patterns with greater stability at cluster points. For small n a single final cluster. As n increases, multiple distinct clusters emerge, and may slowly collapse to a single point. As n increases further, the slow collapse becomes rarer and the distinct clusters appear stable.

2. **Paragraph embedding.** Various paragraphs are embedded and used for the token initialization $x(0)$. Dynamics vary widely depending on the paragraph used, when clustering emerges it centres at one of the vectors present in the initial embedding $x(0)$: i.e. around one of the $x_k(0)$.

2.2 Further Multihead Experiments

Similar experiments to [2] with random weight initializations are repeated for the multihead context. The number of initialized tokens and number of heads in multihead blocks are varied and the effect on the geometry and speed of clustering is studied.

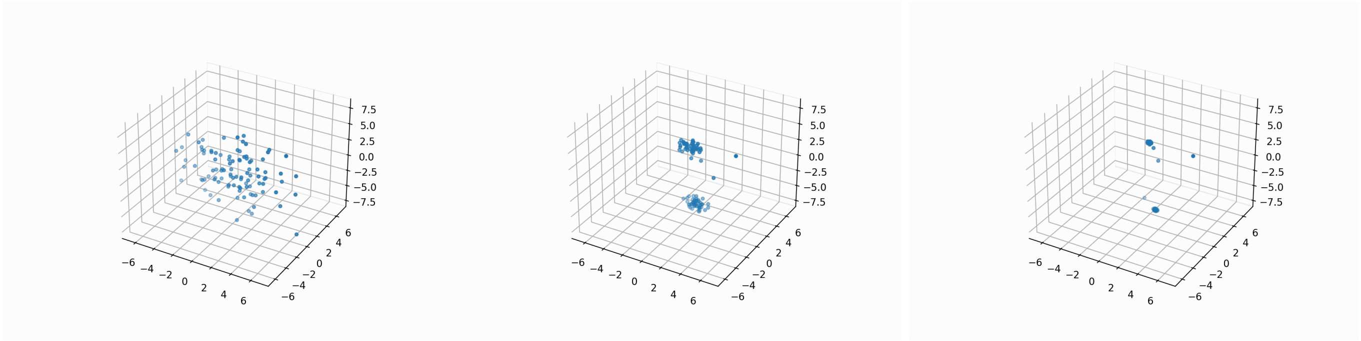


(a) $t = 1$

(b) $t = 7$

(c) $t = 15$

Figure 5: PCA of ALBERT multi-head dynamics. Clustering behaviour of $n = 30$ randomly initialized tokens at $t \in \{1, 7, 15\}$ iterations of the dynamics.



(a) $t = 1$

(b) $t = 10$

(c) $t = 20$

Figure 6: PCA of ALBERT multi-head dynamics. Clustering behaviour of $n = 100$ randomly initialized tokens at $t \in \{1, 10, 20\}$ iterations of the dynamics.

3 Justification

3.1 Maximal eigenvalue of multi-head self-attention matrix

Proposition 1. Let $\mathbf{P}(t) = \left(\sum_{h=1}^{H} P_h(t)\right)$ where $P_h(t)$ is a single-head self-attention matrix with weights for head h where $1 \leqslant h \leqslant H$ and some t. The maximal norm of an eigenvalue corresponding to $\mathbf{P}(t)$ is H and it is attained.

Proof. $P_h(t)$ is stochastic since it is square and every row sums to 1. The row-wise sum of terms of $\sum_{h=1}^{H} P_h(t)$ is therefore H. By rescaling we may express $\mathbf{P}(t) = H \cdot \mathbf{S}(t)$ where $\mathbf{S}(t)$ is stochastic. Each eigenvalue λ of $\mathbf{S}(t)$ satisfies $|\lambda| \leqslant 1$ and there exists at least one $\lambda_0 = 1$. The scaling to $\mathbf{P}(t)$ implies that every λ' of $\mathbf{P}(t)$ satisfies $|\lambda'| \leqslant H$ and there exists $\lambda'_0 = H$. □

4 Github Notes

Documentation for the important piece of code that we may or may not need. Overall, code is certainly digestible.

- `albert_eigenval.ipynb`: Used for analyzing the eigenvalues of each head of pre-trained model.

    1. `albert_get_BV`: Takes as input the pre-trained ALBERT model and an index i specifying the attention head.

        (a) Grabs query, key, value matrices Q, K, V from trained model, as well as dense matrix D.

        (b) Picks out submatrices $Q_i = Q(:, ki : k(i+1))$, similarly for K_i, V_i, where k is the head size. With V_i matrix, this is immbedded in a larger $d \times d$ matrix, maintaining positional encoding, where d is the size of the hidden layer.

        (c) Set $B = \frac{1}{2\sqrt{k}}(Q_i V_i^T) + V_i Q_i^T$ and $V = (V_i D)^T$

    2. `plot_B_spectra`: Would be used to produce plot similar to figure 10 but for matrix B.

    3. `plot_V_spectra`: Produced figure 10 using the eigenvalues of the matrix V from `albert_get_BV`.

- `clustering-Tformers.py`: Preforms main experiments

    1. `get_dynamics`. Denote $f(z)$. Returns the dynamics $z'(t) = (z_i(t))_{i=1}^n$ at some time-step t. Takes as input current values of z, attention matrix, value matrix V, and index i specifying token z_i. In particular, computes

    $$z_i'(t) = \sum_j P_{ij}(t) V(z_j(t) - z_i(t)) \tag{3}$$

    2. `transformer`. Solves the n systems

    $$\dot{z}_i(t) = \sum_j P_{ij}(t) V(z_i(t) - z_j(t)) \tag{4}$$

    via 4th order Runge-Kutta on $t \in [0, T]$.

        (a) Calculates attention matrix

        $$P_{ij}(t) = \frac{e^{\langle Q e^{tV} z_i(t), K e^{tV} z_j(t) \rangle}}{\sum_{\ell=1}^n e^{\langle Q x_i(t), K x_\ell(t) \rangle}} \quad (i, j) \in [n]^2 \tag{5}$$

        at each time step $\{t_i\}_{i=1}^m$, and stores in 3-dimensional array

        (b) Computes constants

        $$k_1 = \delta f(z_i(t_j)) \quad k_2 = \delta f(z_i(t_j) + k_1/2) \quad k_3 = \delta t f(z_i(t_j) + k_2/2) \quad k_4 = \delta t f(z_i(t_j) + k_3) \tag{6}$$

        where $\delta = t_{i+1} - t_i$, and set

        $$z_i(t_{j+1}) = z_i(t) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{7}$$

    Iterates over above steps for all time steps and each token z_i.

    3. `visuals`. Just a function for generating the visuals, provided the desired type of geometry. Their automation process here is very nice, we should try to implement something like this if we end up observing a variety of geometries

- `conv-coord.py`: Generates figure 7b.

- `cupy-Tform`: Algorithm which provides data to `diverge-coord.py`, displying the diverging clusters in the case where V is *not* positive-definite (i.e. has negative eigenvalues)

  1. Generate some random sparse square matrix V of size d, compute eigenvalues and eigenvectors, and set $V = TDT^{-1}$ where T is matrix of eigenvectors of V and D diagonal matrix of eigenvalues. Artificially making random Hermitian matrix?
  2. Function `get_dynamics` same as described in `clustering-Tformers.py`.
  3. Set initial $z(0) = (z_i(0))$ values to be $T\vec{x}_0$ where T is the matrix of eigenvectors of (new) V.
  4. Solve for dynamics of z using the 4th order Runge-Kutta method described in `clustering-Tformers.py`

- `diverge-coord.py`: Used to generate figure 15.
- `leaders.py`: Used to generate figure 4.
- `lowrank-attention.py`: Preforms experiment in section 2 of paper, figure 3, 11, 12.

References

[1] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. "Attention is not all you need: pure attention loses rank doubly exponentially with depth". In: Proceedings of Machine Learning Research 139 (2021). Ed. by Marina Meila and Tong Zhang, pp. 2793–2803. URL: https://proceedings.mlr.press/v139/dong21a.html.

[2] Borjan Geshkovski et al. "The emergence of clusters in self-attention dynamics". In: (2023). arXiv: 2305.05465 [cs.LG].

[3] Vardan Papyan, X. Y. Han, and David L. Donoho. "Prevalence of neural collapse during the terminal phase of deep learning training". In: *Proceedings of the National Academy of Sciences* 117.40 (Sept. 2020), pp. 24652–24663. DOI: 10.1073/pnas.2015509117. URL: https://doi.org/10.1073/pnas.2015509117.

[4] Ashish Vaswani et al. "Attention is All you Need". In: 30 (2017). Ed. by I. Guyon et al. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[5] Sinong Wang et al. "Linformer: Self-Attention with Linear Complexity". In: (2020). DOI: 10.48550/ARXIV.2006.04768.