リザバー計算に触発された軽量型 Transformer の提案：パラメタ共有を用いた計算の効率化と性能評価

中村 仁 [1]　加藤 万理子 [2]　黒岩 蒼太郎 [3]　崎野 也真人 [4]
田中 剛平 [5]　山下 洋史 [1]　鈴木 秀幸 [1]　白坂 将 [1]
[1] 大阪大学 大学院情報科学研究科　[2] 北陸先端科学技術大学院大学 先端科学技術研究科
[3] 公立はこだて未来大学 大学院システム情報科学研究科
[4] 日産自動車株式会社 R&D　[5] 名古屋工業大学 大学院工学研究科
jin.nakamura@ist.osaka-u.ac.jp　mariko.k@jaist.ac.jp　g2124007@fun.ac.jp
gtanaka@nitech.ac.jp　{h.yamashita, hideyuki, shirasaka}@ist.osaka-u.ac.jp

概要

近年，Transformer は多様な NLP タスクで卓越した性能を示す一方，大規模化による計算資源コストの増大が深刻な課題となっている．本研究では，リザバー計算の仕組みに着想し，Transformer の Encoder をなす Encoder ブロックの大部分を固定層（リザバー層），残余ブロックを学習層とすると同時に，固定層・学習層それぞれにおける層間完全パラメタ共有を組み合わせた軽量アーキテクチャを提案する．提案手法により，学習パラメタと更新コストを大幅に削減した．独英翻訳タスクでは，15%超のパラメタ削減にもかかわらず BLEU スコアを約 28 に維持し，学習収束レートをを考慮した性能指標においても従来手法を上回る性能を示した．さらに，本アーキテクチャは離散力学系として解釈でき，高次元非線形変換の再帰的適用によって翻訳に有効な特徴が獲得される可能性を示唆する．本成果は，NLP と非線形ダイナミクスの結節点として新たな視点を提供するものであり，言語モデルにおける情報表現の理解や省メモリ設計に寄与することが期待される．

1　はじめに

近年，自然言語処理（NLP）分野において，Transformer [1] は機械翻訳や言語モデル，文生成など数多くのタスクで顕著な成果を示している．一方で，近年の大規模化傾向 [2, 3] により，モデルのパラメタ数は数十億を超える例も珍しくなく，膨大な学習・推論コストが問題視されてきた．とりわけ研究開発コストや環境負荷の増大は Green AI [4] の観点からも深刻であり，軽量かつ効率的な Transformer の設計が強く求められている．

これまで，Transformer の軽量化に関して様々な手法が提案されてきたが，多層構造の大部分を積極的に固定して学習コストを削減するアイデアはあまり注目されてこなかった．近年提案された Reservoir Transformer [5] は，一部の層をランダムに初期化し，固定して利用するリザバー計算（Reservoir Computing）[6] の考え方を導入することで，性能を保ったままパラメタ更新量を大幅に削減している．標準的なリザバー計算では単一固定層による高次元非線形変換を再帰的に利用することで高度な時系列処理を行うが，Reservoir Transformer においては層の再帰的利用によるパラメータ数の倹約といったリザバー計算本来の強みが十分に活かされていない．

本研究では，Transformer の Encoder 部に大量の R 層（固定層）を導入し，層間パラメタ共有を組み合わせることでさらなる軽量化を図る．この設計は ALBERT [11] が示した層間共有の有効性と，リザバー計算にみられる再帰性を同時に活用するものであり，固定層と学習層を交互に反復適用する自律離散力学系として捉えられる．

手法の検証には独英翻訳タスクを用い，BLEU スコアなどを指標に Vanilla Transformer や Reservoir Transformer と比較する．結果として，最大 15% 超のパラメタ削減と BLEU ≈ 28 の精度維持が同時に可能であり，パラメタ共有層の増加により学習の立ち上がりに遅れを生じるが，最終性能には大きく影響しないことが示された．提案アーキテクチャの離散力学系としての特徴づけを行うこと（例えば，カオス性の役割の理解 [7]）は，NLP を非線形ダイナミ

クスの視点から理解することや，よりよい学習器の設計につながると考えられる．

2 関連研究

2.1 Transformer 軽量化

多数の手法が提案されているが，大きく以下のカテゴリに分かれる：

1. **層やチャネルの一部を削除やスキップ，凍結（freeze）する手法**: LayerDrop [8], AutoFreeze [9] 等
2. **知識蒸留 (Knowledge Distillation)**: 大型モデルから軽量モデルへの知識伝達 [10]
3. **パラメタ共有**: ALBERT [11] では，層間で埋め込みや FFN パラメタを再利用
4. **注意機構の線形化・近似**: Linformer [12], Performer [13] など

本研究はとくに 1, 3 に近く，パラメタ更新において固定される層が存在する点が層削除・層凍結と共通している．また 3 のように層間共有をさらに進め，学習コストを下げる戦略をとっている．

2.2 Reservoir Transformer

Shen ら [5] は，Transformer の一部層をランダムに初期化し，固定しても性能があまり低下しないこと，かつ学習の収束が早まることを示した．これは，ランダムに導入された固定層（ Reservoir 層; R 層）が高次元写像として機能し，後段の学習層（ Learnable 層; L 層）による分離を容易にするためと考察される．しかし同研究では，R 層間でパラメタが共有されていない．本研究では，R, L 層それぞれについて層間パラメタ共有を行い，より積極的にパラメタ数を削減する点が異なる．

2.3 力学系としての Transformer

層間パラメタが共有されている Transformer は，各層を時刻をみなすことで自律離散力学系とみなすことができる．ALBERT を対象としてこのような離散力学系の特性を調べた研究 [7] では，層間伝播によって異なる トークンの内部表現が著しく分離されるカオス性が NLP 性能の向上に貢献することが示唆されている．本研究の提案モデルも，隣接した R，L 層をひとまとめにしたブロックを再帰的に適用する自律離散力学系とみなすことができるため，力学的解析によって内部 トークンの内部表現の伝播特性と学習器性能を関連付けることが可能であると考えられる．

3 提案手法

Encoder-Decoder 型の Transformer をベースとし，Encoder は 8 層，Decoder は 2 層（学習可能）とし，深い Encoder ＋浅い Decoder 構成 [14] を採用した．

固定層（リザバー層, R 層） Self-Attention と FFN レイヤをランダム初期化後に固定し，学習時に更新しないように設定する．Model3, 4 における R 層は，同じパラメタを共有する．

学習層（L 層） 通常の Transformer ブロックを用いる．複数の学習層がある場合は，学習層どうしもパラメタを共有する（Model4）．つまり 1 セットのパラメタで複数層を賄うため，パラメタ数をさらに減少させることができる．

各モデルの特徴 各モデルにおける Encoder の構成を次に示し，固定方法の概要を図 1 に示す．

- Model0: 全層学習 (baseline)
- Model1: R 層（2 層）共有なし + L 層（6 層）共有なし（Reservoir Transformers [5] 再現）
- Model2: R 層（4 層）共有なし + L 層（4 層）共有なし
- Model3: R 層（4 層）共有 + L 層（4 層）共有なし
- Model4: R 層 (4 層) 共有 + L 層 (4 層) 共有

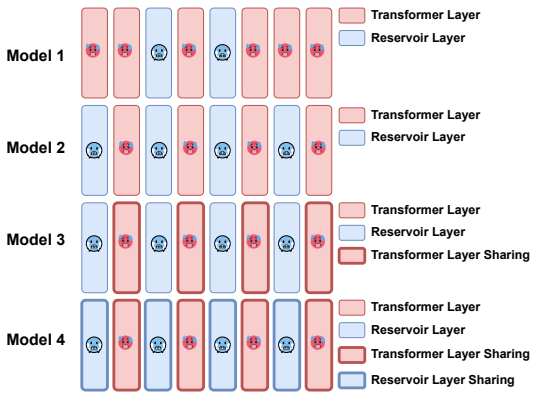

図 1 Model1〜4 における Encoder の概要

4 実験設定

4.1 データセットとタスク

IWSLT'14 De-En 本研究では，IWSLT'14 独英翻訳タスクを用いた．先行研究 [5] と同様，[15] の標準的手順に従った前処理を施し，最終的に約

157,000 対の訓練データ，約 7,100 対の開発データ，約 6,400 対のテストデータを得た．翻訳性能の評価には，大文字・小文字を区別しない（case-insensitive）sacreBLEU [16] を用いており，再現性および他研究との比較可能性を確保している．

4.2 学習条件

- フレームワーク: PyTorch + fairseq
- GPU: NVIDIA A100 (80GB) × 3 枚，分散学習 (DDP)
- Optimizer: Adam($\beta_1 = 0.9, \beta_2 = 0.98$), lr=0.0005, scheduler=inverse_sqrt
- バッチ: max-tokens=4096
- 学習時間: 90 分.

4.3 評価指標

BLEU 本研究では，最終的な test セット上の BLEU スコアを主要評価指標とする．IWSLT における BLEU スコアは，[17–19] などで示されているとおり，概ね 30〜35 程度が高性能の目安とされる．

AUCC (Area Under the Convergence Curve) 学習時間 (分) vs test BLEU グラフの曲線下面積．学習曲線がグラフ上方に位置するほど AUCC が大きく，「同じ時間でより高スコアに到達」したモデルを評価.

time to best, min loss 最高 BLEU or 最小損失に要する時間 (分).

パラメタ数 固定層共有によりどれだけ削減できたか確認.

5 結果

各々のモデルについて 4 インスタンスの学習を行った結果を比較する．

図 2 に学習時間 (分) を横軸，test BLEU を縦軸に描画した結果を示す．AUCC は R 層導入により一般的に増加するが，共有層の増加により低下する傾向がみられる．これは，共有層の増加により学習の立ち上がりに遅れが生じることを意味する．すべてのモデルについて，90 分制限内での到達 BLEU は 28 程度で baseline とほぼ同等であり，パラメタ共有による性能低下はみられない．

表 1 に各モデルのパラメータ数と time to best を示す．Model4 は約 49.13M までパラメタを削減しながら，time to best が 38.95 ± 5.13 分となった．Baseline である Model0 (57.54M, 30.2 ± 2.1 分) に比べ若干遅延は見られるものの，同程度の最終 BLEU を得られるため省メモリ環境での候補となる．

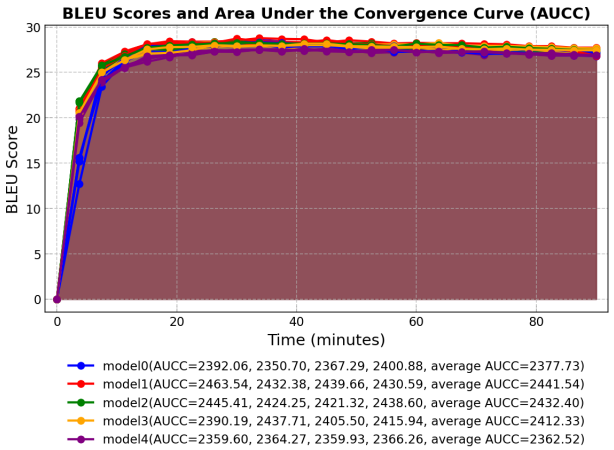

図 2 各モデルにおける学習曲線と AUCC

表 1 収束までの時間と性能の比較

Model	パラメタ数 (M)	time to best (min)	test BLEU
Model0	57.54	30.18 ± 2.45	28.29 ± 0.19
Model1	57.54	28.69 ± 2.38	**28.55 ± 0.21**
Model2	57.54	**25.87 ± 3.59**	28.25 ± 0.10
Model3	53.34	28.72 ± 8.17	28.13 ± 0.14
Model4	**49.13**	38.95 ± 5.13	27.58 ± 0.04

表 2 は，各モデルにおける最小損失 (min loss) およびその到達時刻 (time at min loss) の平均 ± 標準偏差を示したものである．Model4 は最小損失がやや大きめではあるが，BLEU への影響は軽微であり，パラメタ削減メリットと合わせて考えると依然有望である．

学習速度を定量的に評価するため，90 分が経過した時点での累計エポック数および累計ステップ数を測定した．その結果を表 3 に示す．特に，Model4 はパラメタ数を削減したことで，90 分以内に平均 49.5 エポック近く（標準偏差 0.58）学習を進めることができ，ステップ数も約 59,000（標準偏差 168）に達した．これは，R 層が増えるほど逆伝播対象が少なくなり，1 ステップあたりの勾配計算が軽量化されるためであると考えられる．一方で，Model0（完全学習型）はエポック数が平均 42.75（標準偏差 0.50），ステップ数も約 51,000（標準偏差 137.7）にとどまり，同じ 90 分でも学習を進められる割合が相対的に低いことが分かった．

以上の結果から，提案モデル（Model2〜4）では，固定層と層間共有を導入することで逆伝播計算量を抑え，より短時間で多くの更新が可能であることが示唆される．すなわち，限られた GPU 時間やメモ

表 2　最小損失とその到達時刻の比較

Model	min loss	time to min loss (min)
Model0	2.81 ± 0.01	87.15 ± 1.90
Model1	2.89 ± 0.01	86.98 ± 1.65
Model2	2.97 ± 0.01	85.88 ± 3.51
Model3	3.02 ± 0.01	87.99 ± 2.13
Model4	3.10 ± 0.03	87.88 ± 0.82

表 3　学習速度の比較

Model	Epochs @90min	Steps @90min
Model0	42.75 ± 0.50	51002.00 ± 137.70
Model1	45.00 ± 0.00	53762.00 ± 191.04
Model2	48.00 ± 0.00	57654.50 ± 123.24
Model3	48.00 ± 0.00	57623.50 ± 229.35
Model4	**49.50 ± 0.58**	**59262.50 ± 168.09**

リ環境でも高い学習効率を実現する上で，大きな利点となりうる．

6　離散力学系としての考察

6.1　R–L ペアを反復する離散力学系

本研究で提案した Model4 は，層間パラメタが共有されているために，Encoder 全体を「R–L のペア」を繰り返し適用する以下のような自律離散力学系とみなせる：

$$x_{t+1} \;=\; g(x_t) \quad \text{with} \quad g(\cdot) \;\equiv\; f^{\mathbf{L}}\!\left(f^{\mathbf{R}}(x_t)\right),$$

ここで x_t はトークン表現であり，$f^{\mathbf{R}}(\cdot)$，$f^{\mathbf{L}}(\cdot)$ はそれぞれランダム固定層（**R** 層），学習層（**L** 層）に対応する写像である．つまり，R–L ブロックをひとまとまりの関数 g として捉え，その繰り返しによってトークンの内部表現が更新される．

ALBERT との比較　層間パラメタ共有を採用した ALBERT [11] を自律離散力学系とみなした Inoue らの研究 [7] では，トークン表現が過渡的な指数的分離（過渡カオス）を呈することが NLP 性能向上に寄与することが示唆されている．ALBERT は一つのエンコーダブロックが写像 g に対応するが，本研究の Model4 では R–L ブロックペアが写像 g に対応している．自律離散力学系としての Model4 においても同様に過渡カオスが誘発されると期待されるが，それがどのように性能に寄与するかは興味深い話題である．

リザバー計算との関連　本手法は，リザバー計算の枠組み [6] を Transformer に応用した先行研究 [5] をさらに拡張し，R 層どうしや L 層どうしを層間共有によってまとめ上げている点が特徴的である，ALBERT にみられるような層間パラメタ共有により積極的に更新パラメタを圧縮しつつ，性能への影響を軽微にすることに成功している．

6.2　意義と今後の展開

リザバー層と層間共有を組み合わせることでパラメタ数と勾配更新量を大幅に削減でき，学習時の電力や計算資源を抑えられる点は，Green AI の要請に応えるうえで大きな意義がある．また，本モデルでは固定層-学習層ペアの定める力学系が過渡カオスを誘発する可能性があり，Lyapunov 指数といったカオスの定量化指標を用いた解析により，固定層が翻訳精度に与える影響の力学系理論による解釈が与えられる可能性がある．トークンの内部表現が固定層の作用に従って拡散する速さを評価することで，学習の効率化への固定層の貢献を定量的に理解できると期待される．さらに，機械翻訳以外のタスクでもランダムに固定された高次元写像の効果が期待されるため，Masked LM や要約，文分類などへの適用を通じて本手法の汎用性を高めるとともに，R 層 ＋ L 層の新たな活用法が広がると考えられる．

7　結論

本研究では，Transformer Encoder の一部層をランダム固定（リザバー化）し，さらに層間パラメタ共有を導入することで，大幅なモデル圧縮と学習コスト削減を両立する手法を提案した．IWSLT De-En 翻訳タスクにおける実験では，最大 15% 超のパラメタ削減を達成しつつ BLEU ≈ 28 を維持し，AUCC も baseline に匹敵するなど，トレードオフが比較的緩やかなことが確認された．さらに，離散力学系として見たときにカオス的性質が R 層で活かされ，学習層が効率的に情報を抽出している可能性がある．

今後，Lyapunov 指標や同期オフセット測定といった力学系解析による最適層数の探索を行うことで，R 層と共有層がもたらす効率化の有用性を明らかにできる可能性がある．さらに内部カオスを評価することで，R 層の非線形変換が翻訳精度に及ぼす影響を定量化できれば，さらなる省メモリ化や層構成の最適化につながると期待される．

謝辞

本研究は，JST ALCA-Next（JPMJAN23F2），JST Moonshot（JPMJMS2021）の助成を受けたものである．また，提案モデルにおける開発の一部は，独立行政法人情報処理推進機構 (IPA) における未踏ターゲット事業「リザバーコンピューティング技術を活用したソフトウェア開発分野」の助成を受けたものである．

参考文献

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In **Advances in Neural Information Processing Systems (NIPS)**, 2017.

[2] T. Brown, B. Mann, N. Ryder, et al. Language models are few-shot learners. In **Advances in Neural Information Processing Systems (NeurIPS)**, 2020.

[3] J. Kaplan, S. McCandlish, T. Henighan, and T. Salimans. Scaling laws for neural language models. **arXiv:2001.08361**, 2020.

[4] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)**, 2019.

[5] S. Shen, A. Baevski, A. Morcos, K. Keutzer, M. Auli, and D. Kiela. Reservoir transformers. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)**, 2021.

[6] H. Jaeger. Adaptive nonlinear system identification with echo state networks. In **Advances in Neural Information Processing Systems**, 2003.

[7] K. Inoue, S. Ohara, Y. Kuniyoshi, and K. Nakajima. Transient chaos in bidirectional encoder representations from transformers. **Physical Review Research**, 4(1):013204, 2022.

[8] A. Fan, E. Grave, and A. Joulin. Reducing transformer depth on demand with structured dropout. In **International Conference on Learning Representations (ICLR)**, 2020.

[9] Y. Liu, S. Agarwal, and S. Venkataraman. AutoFreeze: Automatically freezing model blocks to accelerate fine-tuning. **arXiv preprint arXiv:2102.01386**, 2021.

[10] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In **NIPS Workshop**, 2015.

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In **International Conference on Learning Representations (ICLR)**, 2020.

[12] S. Wang, B. Zhang, Y. Guo, L. Huang, and Z. Sun. Linformer: Self-attention with linear complexity. **arXiv preprint arXiv:2006.04768**, 2020.

[13] K. Choromanski, V. Likhosherstov, D. Dohan, et al. Rethinking attention with performers. In **International Conference on Learning Representations (ICLR)**, 2021.

[14] J. Kasai, N. Pappas, H. Peng, J. Cross, and N. A. Smith. Deep encoder, shallow decoder: reevaluating the speed–quality tradeoff in machine translation. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, 2020.

[15] S. Edunov, M. Ott, M. Auli, D. Grangier, and M. Ranzato. Classical structured prediction losses for sequence to sequence learning. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, 2018.

[16] M. Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation (WMT)**, Volume 1: Research Papers, pages 186–191, Brussels, Belgium, October 31 – November 1, 2018. Association for Computational Linguistics.

[17] O. Hrinchuk, E. Bataev, et al. NVIDIA NeMo offline speech translation systems for IWSLT 2023. In **Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT)**, 2023.

[18] A. Xu, B. Ouyang, et al. CMU's IWSLT 2024 simultaneous speech translation system. In **Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT)**, 2024.

[19] C. Wang, A. Finch, and E. Sumita. The NICT translation system for IWSLT 2014. In **Proceedings of the 11th International Conference on Spoken Language Translation (IWSLT)**, 2014.