RNN の回帰行列を凍結しても統語構造の獲得は損なわれない

上田亮 [1*] 栗林樹生 [2] 神藤駿介 [1] 乾健太郎 [2,3,4]
[1] 東京大学 [2]MBZUAI [3] 東北大学 [4] 理化学研究所
{ryoryoueda,skando}@is.s.u-tokyo.ac.jp
{Tatsuki.Kuribayashi,Kentaro.Inui}@mbzuai.ac.ae

概要

人間と同様の言語能力を達成するために必要十分な構造しか持たないニューラル言語モデルはどのようなものか？ 可能な限り簡素なニューラル言語モデルを出発点としてこの問いに取り組むべく，本稿は Reservoir Computing の基本的なモデルである Echo State Network（ESN）と呼ばれる回帰型のモデルを再訪し，ESN やそれを少し拡張した言語モデルの能力を検証する．実験の結果，適切な初期化のもとでは埋め込み層と出力層のみ訓練し，回帰行列は凍結したとしても統語構造の獲得が損なわれないことが示唆された．

1 はじめに

近年，Transformer [1] をベースとする大規模言語モデル（Large Language Model; LLM）が，その言語モデリング性能や汎用性の高さから自然言語処理（NLP）分野で多大な成功をおさめ注目を集めている．その一方で，人間レベルの言語能力を達成するための必要十分条件は何だったのかという科学的な問いに対して、この成功単体は答えておらず，例えば，Transformer ほど複雑なニューラルネットワークアーキテクチャを使う必要は必ずしもなかったのかもしれない．

本稿では，そのような問題意識から，考え得る限り最もシンプルな回帰型のモデルとして，Reservoir Computing 分野の基本的なモデルである Echo State Network（ESN）[2, 3, 4] を再訪する．ESN に基づく言語モデルや，それを少し拡張したモデルがどの程度の言語モデリング能力や統語構造獲得能力を有するのかを検証し，今後の模索の足がかりとすることを狙う．ESN は，回帰型ニューラルネットワーク（RNN）の一種である．Simple RNN (SRN) に近い構造をもち，特にランダム初期化したあと出力層以外のパラメタ (埋め込み行列と回帰行列) を全て固定 (凍結) したまま訓練するモデルであり，主に時系列データを扱う研究分野で用いられてきた [5]．出力層以外凍結するという著しい制約は，可能な限りシンプルなニューラル言語モデルの出発点とするのに相応しいと考えられる．実は，NLP 分野においても，ESN 文分類器が十分に強力なベースラインモデルとなるという指摘 [6] や，機械翻訳タスクやマスク言語モデリングにおいて Transformer の一部の層を凍結したまま訓練したとしても，通常の Transformer に匹敵する性能が維持されうるという報告 [7] がある．

実験においては，人間の子どもが 13 歳までに暴露されるであろう 1 億単語程度の規模のデータセットを用いて ESN 言語モデルの訓練・評価を行い，訓練後の ESN 言語モデルの統語構造獲得能力を BLiMP データセット [8] 上で評価する．また，元の ESN 言語モデルでは性能に限界があることが分かったため，出力層と埋め込み層を訓練可能とする場合（ESN+i）についても検証を行った．少なくとも我々の実験設定のもとでは，ESN+i 言語モデルは，同データで訓練した Transformer 言語モデルと同等かそれ以上の統語獲得能力を示すことが明らかになり，テキストに基づく言語モデルの統語獲得において，(i) RNN のような再帰性，(ii) 隠れ状態が無秩序に変化しないよう適切に過去の情報を忘却する性質（Echo State Property; ESP）をもつように適切に回帰行列を初期化すること，(iii) 単語埋め込みの学習が少なくとも必要であり，RNN 回帰行列の訓練は必ずしも必要条件ではないことが示唆される．

なお，今回の設定で最も優れた性能を発揮したのは LSTM であり，ゲート機構の必要性なども今後検討に値するだろう．また今回，ESN および ESN+i において様々なモデルサイズを試したが，性能に関して明確なスケーリング則を確認することはできなかった．データセット規模やエポック数の制約の

* MBZUAI 滞在中の成果．

影響も有り得るが，初期化手順やアーキテクチャ，モダリティに改善の余地が残っている可能性もある．例えば，回帰行列に脳のような small-world 性や scale-free 性をもたせる [9, 10]，層を重ねて深いネットワークにする [11]，マルチモーダルなデータを用いるなどの改善方法があり得る．

2 ESN 言語モデル

時系列データ（文データ）$\boldsymbol{s} = \{\boldsymbol{u}_t \in \mathbb{R}^{N_{\text{voc}}}\}_{t=1}^{T}$ が与えられたとき，各時刻 t における ESN の隠れ状態 $\boldsymbol{h}_t \in \mathbb{R}^{N_{\text{rec}}}$ と出力ベクトル $\boldsymbol{o}_t \in \mathbb{R}^{N_{\text{voc}}}$ は典型的には以下のように表される [4]：

$$\boldsymbol{h}_t = (1-a)\boldsymbol{h}_{t-1} + af(\boldsymbol{W}_{\text{rec}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{\text{in}}\boldsymbol{u}_t) \tag{1}$$

$$\boldsymbol{o}_t = \boldsymbol{W}_{\text{out}}\boldsymbol{h}_t + \boldsymbol{b}_{\text{out}}. \tag{2}$$

ここで，$f(\cdot)$ は要素ごとの（非線形）写像であり，tanh や ReLU が良く用いられる．$a \in (0, 1]$ は漏れ率と呼ばれる超パラメタであり，残差接続に似た機構をもたらす．行列 $\boldsymbol{W}_{\text{rec}} \in \mathbb{R}^{N_{\text{rec}} \times N_{\text{rec}}}$ と $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{N_{\text{rec}} \times N_{\text{voc}}}$ はランダム初期化後に固定（凍結）し，基本的に訓練中には動かさない（ただし，本稿では，追加で $\boldsymbol{W}_{\text{in}}$ を訓練可能とした条件下での実験も行う）．一方で，行列 $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{N_{\text{voc}} \times N_{\text{rec}}}$ 及びベクトル $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^{N_{\text{voc}}}$ は訓練を通して最適化される．

行列 $\boldsymbol{W}_{\text{in}}, \boldsymbol{W}_{\text{rec}}$ の初期化　行列 $\boldsymbol{W}_{\text{in}}$ はスケール $\sigma_{\text{in}} > 0$ を超パラメタとして以下のように初期化する：

$$(\boldsymbol{W}_{\text{in}})_{ij} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_{\text{in}}^2) \tag{3}$$

$\boldsymbol{W}_{\text{rec}}$ は $\rho_{\text{rec}} \in (0, 1)$，$\lambda_{\text{rec}} \in [0, 1)$ を超パラメタとして以下のように初期化する：

$$\boldsymbol{W}_{\text{rec}} = \frac{\rho_{\text{rec}}}{\rho(\boldsymbol{W}_{\text{rand}} \odot \boldsymbol{W}_{\text{mask}})} \boldsymbol{W}_{\text{rand}} \odot \boldsymbol{W}_{\text{mask}}. \tag{4}$$

ただし，

$$\begin{aligned} (\boldsymbol{W}_{\text{rand}})_{ij} &\overset{\text{i.i.d}}{\sim} \mathcal{N}(0, 1), \\ (\boldsymbol{W}_{\text{mask}})_{ij} &\overset{\text{i.i.d}}{\sim} \text{Bernoulli}(1 - \lambda_{\text{rec}}) \end{aligned} \tag{5}$$

である．$\odot$ は要素ごとの積（Hadamard 積）であり，$\rho(\cdot)$ は行列のスペクトル半径である．ρ_{rec} は $\boldsymbol{W}_{\text{rec}}$ のスペクトル半径を決める超パラメタであり，経験則として 1 に達しないギリギリの値にするのが良いとされている．スペクトル半径は固有値の絶対値の最大値であり，直感的には隠れ状態 $\boldsymbol{h}_{t-1}$ をどの程度 “拡大” しうるかをを定めている．これが大きすぎると過去のノイズを拡大して無秩序な状態に陥るリスクがあるが，1 未満であれば過去の情報を適切に忘却することによって程よい状態が保たれやすくなる．このような性質はしばしば Echo State Property（ESP）と呼ばれる．λ_{rec} は行列の疎性を制御し，ニューロン結合のトポロジーを複雑にし，リッチなダイナミクスを生む効果があるとされる[1)]．

文データ $\boldsymbol{s}$ のフォーマット　文データ $\boldsymbol{s} = \{\boldsymbol{u}_t\}_{t=1}^{T}$ において，各 $\boldsymbol{u}_t$ は one-hot 符号化されたベクトルである．$\boldsymbol{u}_1$ は必ず文の始まりを意味する特別な記号（BOS），$\boldsymbol{u}_T$ は必ず文の終わりを意味する特別な記号（EOS）であるとする．

目的関数　ESN の研究ではリッジ回帰で $\boldsymbol{W}_{\text{out}}$ の値を求めることが多いのだが，本稿では近年の言語モデリングの一般的なプラクティスに倣い，文の対数尤度最大化の問題として定式化し，勾配法によって訓練する：

$$\underset{\boldsymbol{\theta}_{\text{tr}}}{\text{maximize}} \ \mathbb{E}_{p_{\text{data}}(\boldsymbol{s})}[\log p(\boldsymbol{s} \mid \boldsymbol{\theta}_{\text{tr}}; \boldsymbol{\theta}_{\text{fr}})] \tag{6}$$

ここで，$\boldsymbol{\theta}_{\text{tr}}, \boldsymbol{\theta}_{\text{fr}}$ はそれぞれ訓練可能パラメタ，凍結パラメタであり，

$$\begin{aligned} \log p(\boldsymbol{s} \mid \boldsymbol{\theta}_{\text{tr}}; \boldsymbol{\theta}_{\text{fr}}) &= \sum_{t=1}^{T-1} \log p(\boldsymbol{u}_{t+1} | \boldsymbol{u}_{1:t}, \boldsymbol{\theta}_{\text{tr}}; \boldsymbol{\theta}_{\text{fr}}) \\ &= \sum_{t=1}^{T-1} \langle \boldsymbol{u}_{t+1}, \log \text{Softmax}(\boldsymbol{o}_t) \rangle. \end{aligned} \tag{7}$$

である．

3 ESN 言語モデルを用いる意義

Transformer が圧倒的な性能を誇るこの時代に，あえて ESN を用いる意義は第 1 節でも触れた通りであるが，本節では補足的な説明を加える．

Transformer のアーキテクチャは自己注意機構，全結合層，残差接続，正規化層，またそれらを多層化したものからなる複雑な構造をしており，言語の認知モデルを考える上でこれほどリッチなアーキテクチャを採用する必要は必ずしもないと考えられる．このような複雑な機構をもつモデルは一般に解釈が難しく，どのようにして言語モデリングや統語構造を学習しているのかを理解するのは困難である場合が多い[2)]．一方で，ESN は非常に簡素な連続時間微分方程式で記述されるニューロンモデルの自然な離散化として導けることが知られている．このことを確かめるため，一時的に t を連続な時間変数として，

1) また，疎行列を伴う行列演算は PyTorch ライブラリの提供する `torch.sparse` によって高速化可能な場合がある．今回は利用できなかったが，今後活用する予定である．

2) 所謂 BERTology も一定の注目を集めてはいるが（[12] など），これは最早 “Transformer を知るための科学” になってしまっていると言わざるを得ないだろう．

ニューロンモデルを以下のように定式化する [13]：

$$\frac{d\boldsymbol{h}_t}{dt} = \frac{1}{\gamma}(-\alpha \boldsymbol{h}_t + f(\boldsymbol{W}_{\text{rec}}\boldsymbol{h}_t + \boldsymbol{W}_{\text{in}}\boldsymbol{u}_t)). \quad (8)$$

これは入力信号（$\boldsymbol{W}_{\text{in}}\boldsymbol{u}_t$），ニューロン間の結合（$\boldsymbol{W}_{\text{rec}}$），ニューロンの活性化（$f(\cdots)$），ニューロンの興奮状態の減衰（$-\alpha\boldsymbol{h}_t$）から成る，これ以上ないくらいシンプルなニューロンモデルである．これを時間幅 δ で離散化すると，

$$\boldsymbol{h}_t = \left(1 - \frac{\alpha\delta}{\gamma}\right)\boldsymbol{h}_{t-1} + \frac{\delta}{\gamma} f(\boldsymbol{W}_{\text{rec}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{\text{in}}\boldsymbol{u}_t) \quad (9)$$

となる．ここで $\alpha = 1$，$\delta/\gamma = a$ と置き直したものが ESN である．

実は，ESN を用いた計算心理言語学研究は過去に度々行われていた [14, 15, 16] のだが，深層学習モデルが勾配爆発や勾配消失の問題を克服するにつれ，ESN は suboptimal なモデルとみなされ次第に用いられなくなってきたようである [17, 18]．しかしながら，元々暗に効率的な圧縮表現の獲得を期待されて発展してきた深層学習のトレンドは，今や大規模化（スケーリング）に移りつつある．もしもモデルの細かな構造よりも大きさこそが重要であるのならば，この時代にあえて ESN をはじめとするシンプルな回帰型のモデルを再訪し，その限界を再検証することには一定の意義があると考える．

4 実験

4.1 実験設定

訓練用データセットとその前処理　近年の計算心理言語学分野におけるトレンドの 1 つである BabyLM Challenge [19, 20] の提供するデータセット（2023 年版）を訓練用（train），評価用（dev）データセットとして用いる．訓練用データは 100M 単語，評価用データは 10M 単語程度の規模である．元のテキストファイルは文ごとに分割されていないため，NLTK ライブラリの提供する文分割器[3)]を用いて分割した[4)]．更にそれを GPT2 のトークナイザ (gpt2)[5)]を用いてトークナイズした．このとき文ごとのトークン系列の最大長は 128 とし，長さが 6 未満になったものはデータから取り除いた（BOS と EOS を付与したため実質的な最小長は 4）．

統語構造把握能力の評価　言語モデルが統語構造をどの程度正しく把握できるているかを測るため，BLiMP [8] を用いた．BLiMP は統語論に基づく様々な言語現象に関する英語文のミニマルペアを集めたデータセットである．

ESN 言語モデル　超パラメタに関してはそれぞれ $a = 0.8$，$f = \text{ReLU}$，$\sigma_{\text{in}} = 1$，$\rho_{\text{rec}} = 0.993$，$\lambda_{\text{rec}} = 0.5$，$N_{\text{rec}} \in \{64, 128, 256, 512, 1024, 2048\}$ とした[6)]．出力層に加えて埋め込み行列を訓練可能とする ESN+i 言語モデルについても同じ超パラメタを採用した．

比較対象　OpenAI による事前学習済みの GPT2 言語モデル（GPT2 OpenAI），BabyLM Challenge データセットを用いて初めから訓練し直した GPT2 言語モデル（GPT2 Scracth），そして埋め込み次元・隠れ状態サイズ 512 の LSTM 言語モデルを比較対象とする．

言語モデルの訓練方法　今回は各言語モデルを 1 エポックだけ訓練させることとした．バッチサイズを 32 とし，Optimizer には AdamW [21][7)]を用い，既定値をそのまま利用した（例えば学習率は 0.001，荷重減衰は 0.01）．また，ESN，ESN+i，LSTM においては埋め込み層直後と出力層直前にパラメタ 0.1 の Dropout を適用した[8)]．

4.2 実験結果

表 1 に各言語モデルの訓練パラメタ数，総パラメタ数，訓練時の損失，検証時の損失，BLiMP（全体）スコアを示す．GPT2 OpenAI の BLiMP 全体スコアが最も高いが，これは訓練データの規模の違いを考慮すれば当然である（このモデルサイズで達成可能な BLiMP スコアの限界を推し量るのには有用である）．GPT2 OpenAI を除いて，訓練損失，評価損失，BLiMP スコア全てにおいて最も良い結果を出したのは LSTM であった．GPT2 Scratch が LSTM より劣っているのは，エポック数の制限から学習が収束しきっていないことや，単純化のため学習率スケジューリングや勾配クリッピングのヒューリスティクスを省いたことなどが原因と考えられる．ESN の成績はどのモデルサイズにおいても GPT2

3) https://www.nltk.org/api/nltk.tokenize.sent_tokenize.html
4) spaCy ライブラリの文分割器の方がより正確な分割を期待できるが，今回は時間的な制約もあり，ルールベースで高速な処理ができる NLTK の文分割器を用いることにした．
5) https://huggingface.co/docs/transformers/model_doc/gpt2#transformers.GPT2Tokenizer
6) 時間の許す限り Optuna で探索して得た暫定的なものであるため，改善の余地を残している可能性がある．
7) https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html
8) GPT2 Scratch については既定値のパラメタを用いた．

表 1　各言語モデルの訓練パラメタ数，総パラメタ数，訓練時の損失，評価損失，及び BLiMP の全体スコア．訓練損失が評価損失よりも大きいのは，評価損失は 1 エポック後に計算しているのに対し訓練損失は 1 エポック全体の平均を取っていることに加え，訓練時は Dropout によりランダムネスの影響があるためであると考えられる．GPT2 は厳密には回帰行列を持たないが，所謂 causal attention が回帰的な役割をもつため ✓ マークをつけている．読みやすさのため GPT2 Scratch の BLiMP スコアに下線を引き，それより高い ESN+i のスコアは太字にしている．

	埋め込み訓練	回帰行列訓練	訓練/総パラメタ数 [M]	訓練損失↓	評価損失↓	BLiMP↑ [%]
ESN (64)			3/6	6.35	5.94	53.2
ESN (128)			6/12	6.18	5.76	54.5
ESN (256)			12/25	6.03	5.59	53.5
ESN (512)			25/51	5.98	5.51	55.6
ESN (1024)			51/104	6.09	5.56	54.5
ESN (2048)			102/210	6.66	6.03	53.2
ESN+i (64)	✓		6/6	5.26	4.75	58.0
ESN+i (128)	✓		12/12	5.08	4.61	**60.4**
ESN+i (256)	✓		25/25	4.95	4.50	**60.8**
ESN+i (512)	✓		51/51	4.87	4.43	**61.5**
ESN+i (1024)	✓		102/104	4.84	4.40	**62.4**
ESN+i (2048)	✓		205/210	4.88	4.43	**62.2**
GPT2 Scratch	✓	✓	124/124	5.64	4.81	58.6
LSTM (512)	✓	✓	53/53	4.50	4.12	68.4
GPT2 OpenAI	✓	✓	124/124	-	-	82.2

Scratch に劣後しており，十分な学習能力を伴っていないことが示唆される．一方で，埋め込み行列を訓練可能とした ESN+i 言語モデルは $N_{\mathrm{rec}} \geq 128$ のとき GPT2 Scratch を上回る成績を出している．特に，$N_{\mathrm{rec}} \leq 1024$ に関しては GPT2 よりもモデルサイズが小さいのにも関わらずである．以上の結果から，元の ESN 言語モデルは Transformer に劣後するものの，追加で埋め込み行列の訓練を許せば Transformer と同等以上の性能をもつことが示唆される．

5　まとめと今後の展望

本稿では，可能な限りシンプルなニューラル言語モデルから出発して，人間レベルの言語能力を獲得を達成するために必要十分な要素が何であるのかを探求するのを目的として，Reservoir Computing 分野の基本的なモデルである Echo State Network (ESN) を再訪し，ESN に基づく言語モデルやそれを少し拡張したモデルがどの程度統語構造を獲得する能力を持っているかについての検証を行った．実験の結果，ESN をそのまま用いる場合では Transformer に劣後するものの，追加で埋め込み行列 $\boldsymbol{W}_{\mathrm{in}}$ の訓練を許せば Transformer と同等以上の性能をもつことが明らかになった．これは，少なくとも必要十分な言語獲得能力を探求する科学的な研究においては，Transformer のような複雑なアーキテクチャを言語モデルとして採用する必要は必ずしもないことを示唆している．特に (i) RNN のような再帰性，(ii) 過去の情報を適切に忘却できる（ESP）ような回帰行列 $\boldsymbol{W}_{\mathrm{rec}}$ の初期化，(iii) 埋め込み行列 $\boldsymbol{W}_{\mathrm{in}}$ の訓練が少なくとも必要であることが示唆され，回帰行列の訓練は必ずしも必要でないことが示唆された[9)]．なお，我々の今回の設定で最も優れた性能を発揮したのは LSTM であった．ゲート機構の必要性についても今後議論する価値があるだろう．

また，原義通りの ESN では十分な性能を発揮できず，埋め込み行列の訓練を許さなければならなかった原因の候補の 1 つとして，one-hot 符号化が考えられる．ESN を用いる典型的な研究では連続的に変化する時系列データが入出力として用いられるが，one-hot 符号系列は不連続に変化する．また，単語の類似度情報を一切持たないため構造の発見が困難になり得る．さらに，ESN，ESN+i において様々なモデルサイズを試したが，性能に関して明確なスケーリング則は認められなかった．データセット規模やエポック数の制約の影響もあり得るが，初期化手順や，アーキテクチャ，モダリティに改善の余地が残っている可能性もある．今後，回帰行列に脳のような small-world 性や scale-free 性をもたせる，層を重ねて深くする，マルチモーダルな訓練データを用いるなどの改善案を検討する予定である．

9)　ESP の重要性を補強するため，スペクトル半径 ρ_{rec} を変化させた際の実験結果を付録 A に掲載した．

謝辞

本研究は JSPS 科研費 JP23KJ0768，JST ACT-X（JPMJAX24C5）の助成を受けたものです．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

[2] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical report, German National Research Center for Information Technology GMD Technical Report 148, 2001. Erratum note available at https://www.ai.rug.nl/minds/uploads/EchoStatesTechRepErratum.pdf.

[3] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, Vol. 3, No. 3, pp. 127–149, 2009.

[4] Mantas Lukoševičius. **A Practical Guide to Applying Echo State Networks**, pp. 659–686. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[5] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. **Neural Networks**, Vol. 115, pp. 100–123, 2019.

[6] John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. In **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**. OpenReview.net, 2019.

[7] Sheng Shen, Alexei Baevski, Ari S. Morcos, Kurt Keutzer, Michael Auli, and Douwe Kiela. Reservoir transformers. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021**, pp. 4294–4309. Association for Computational Linguistics, 2021.

[8] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. **Trans. Assoc. Comput. Linguistics**, Vol. 8, pp. 377–392, 2020.

[9] Zhidong Deng and Yi Zhang. Collective behavior of a small-world recurrent neural system with scale-free distribution. **IEEE Trans. Neural Networks**, Vol. 18, No. 5, pp. 1364–1375, 2007.

[10] Yuji Kawai, Jihoon Park, and Minoru Asada. A small-world topology enhances the echo state property and signal propagation in reservoir computing. **Neural Networks**, Vol. 112, pp. 15–23, 2019.

[11] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. Deep reservoir computing: A critical experimental analysis. **Neurocomputing**, Vol. 268, pp. 87–99, 2017.

[12] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In **Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers**, pp. 4593–4601. Association for Computational Linguistics, 2019.

[13] Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. **Neural Networks**, Vol. 20, No. 3, pp. 335–352, 2007. Echo State Networks and Liquid State Machines.

[14] Matthew H. Tong, Adam D. Bickett, Eric M. Christiansen, and Garrison W. Cottrell. Learning grammatical structure with echo state networks. **Neural Networks**, Vol. 20, No. 3, pp. 424–432, 2007.

[15] Stefan L. Frank and Michal Čerňanský. Generalization and systematicity in echo state networks. **Proceedings of the Annual Meeting of the Cognitive Science Society**, Vol. 30, No. 30, 2008.

[16] Stefan L. Frank and Rens Bod. Insensitivity of the human sentence-processing system to hierarchical structure. **Psychological Science**, Vol. 22, No. 6, pp. 829–834, 2011. PMID: 21586764.

[17] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)**, pp. 1195–1205. Association for Computational Linguistics, 2018.

[18] Ethan Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Levy. On the predictive power of neural language models for human real-time comprehension behavior. In **Proceedings of the 42th Annual Meeting of the Cognitive Science Society - Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020, virtual, July 29 - August 1, 2020**. cognitivesciencesociety.org, 2020.

[19] Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. Call for papers - the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. **CoRR**, Vol. abs/2301.11796, , 2023.

[20] Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. **CoRR**, Vol. abs/2404.06214, , 2024.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**. OpenReview.net, 2019.

A　スペクトル半径を変化させた際の実験結果

表 2　超パラメタの 1 つであるスペクトル半径 ρ_{rec} を変化させた際の，ESN+i 言語モデル（$N_{\text{rec}} = 512$）の訓練時の損失，評価損失，及び BLiMP の全体スコア．訓練損失が評価損失よりも大きいのは，評価損失は 1 エポック後に計算しているのに対し訓練損失は 1 エポック全体の平均を取っていることに加え，訓練時は Dropout によりランダムネスの影響があるためであると考えられる．

	埋め込み訓練	回帰行列訓練	スペクトル半径 ρ_{rec}	訓練損失↓	評価損失↓	BLiMP↑ [%]
ESN+i (512)	✓		0.5	4.94	4.52	60.6
ESN+i (512)	✓		0.6	4.91	4.49	61.0
ESN+i (512)	✓		0.7	4.89	4.47	61.3
ESN+i (512)	✓		0.8	4.88	4.45	61.1
ESN+i (512)	✓		0.9	4.87	4.44	61.1
ESN+i (512)	✓		1.0	4.87	4.43	61.5
ESN+i (512)	✓		1.1	4.87	4.43	61.3
ESN+i (512)	✓		1.2	4.89	4.43	61.5
ESN+i (512)	✓		1.3	4.90	4.44	61.5
ESN+i (512)	✓		1.4	4.92	4.45	62.0
ESN+i (512)	✓		1.5	1.14×10^{6}	4.45	61.8
ESN+i (512)	✓		1.6	7.80×10^{18}	2.31×10^{17}	58.9
ESN+i (512)	✓		1.7	2.10×10^{30}	7.80×10^{28}	55.2

超パラメタの 1 つであるスペクトル半径 ρ_{rec} を変化させた際の，ESN+i 言語モデル（$N_{\text{rec}} = 512$）の結果を表 2 に示す．表から読み取れるように，スペクトル半径 ρ_{rec} が大きすぎても小さすぎても十分な性能が発揮されず，$\rho_{\text{rec}} \approx 1$ 付近で最も言語モデリングの性能が良くなることが分かる．スペクトル半径 ρ_{rec} が小さすぎる場合はモデルが十分に学習しきれていない（underfit している）．逆に，スペクトル半径 ρ_{rec} が大きすぎる場合（今回の場合は $\rho_{\text{rec}} \geq 1.5$），損失が極端に大きな値になってしまい有意味な学習ができていないことが示唆される．これは，隠れ状態 $\boldsymbol{h}_t$ が無秩序に “拡大” される傾向によって Echo State Property（ESP）が損なわれてしまったためであると考えられる．

また，興味深いことに，ESP を保つための一般的な経験則である $\rho_{\text{rec}} < 1$ を多少逸脱しても（今回の場合は 1.4 まで）性能が保たれている．これは，今回の研究において ReLU$(\cdot)$ を非線形関数として用いたことに起因すると考えられる．例えば，$\boldsymbol{x}$ をゼロを平均とするランダムなベクトルとする場合，$\boldsymbol{y} = \text{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, 0)$ の成分の約半数はゼロになると考えられるため，おおよそ $\sqrt{2}\|\boldsymbol{y}\|_2 \approx \|\boldsymbol{x}\|_2$ が成り立つと考えられる．従って，$\rho_{\text{rec}} < \sqrt{2} \approx 1.41$ の範囲内であれば，隠れ状態が無秩序に “拡大” されてしまうリスクは低いと考えられる．ただし，他の非線形関数を採用する場合に同様の議論が成り立つとは限らないため，やはり $\rho_{\text{rec}} < 1$ を経験則としておくのが安全なのであろう．