Order-chaos transition in deep neural network and its application to the training process

Fan Xu [1] **Ling Feng** [2 1]

[1] *Department of Physics, National University of Singapore, Singapore* [2] *Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR), Singapore*. Correspondence to: Ling Feng 0@criticality.ai.

1. Introduction

Recurrent neural networks can be viewed as high-dimensional dynamical systems. Therefore, they exhibit distinct phases, including order, periodic cycles, and chaos. A key insight in this field is that model performance often peaks near the Edge of Chaos (EoC), a critical boundary between chaotic and order phase[1, 2]. Building on this insight, prior work on Long-short-term memory networks (LSTM) reports that as training processes, the validation loss can exhibit multiple descents, and these sharp descents are closely associated with phase transitions between ordered and chaotic hidden-state dynamics[3].

Despite such insight that the neural network performs most optimally at the edge of chaos, few studies use it as a guiding principle for improving model performance. In this work, we propose a complete chaos-monitoring and control regularizer using LSTM as a proxy. By tracking the amount of stability/instability through estimating the Lyapunov exponent, we adaptively regulate the model to stay near the EoC and achieve accuracy that consistently surpasses baseline training and standard regularization methods such as constant weight decay.

2. Finite Time Lyapunov Exponent during Training

To study LSTM dynamics in practice, we train the standard LSTMs on the IMDB sentiment classification task (50,000 movie reviews). These LSTMs have different learning rates and hidden-state widths. As a standard stability/chaos indicator in nonlinear dynamics, we use the finite-time Lyapunov exponent (FTLE) λ_T, defined as the finite-horizon separation rate between two trajectories initialized with an infinitesimal perturbation. Throughout this work, FTLE refers to the maximum FTLE. As commonly known in dynamical systems, $\lambda_T > 0$ typically indicates chaotic behavior, $\lambda_T < 0$ indicates ordered contraction, and $\lambda_T \approx 0$ corresponds to dynamics near the EoC.

Direct FTLE computation via QR/SVD-based Jacobian propagation is expensive and impractical for LSTMs. We therefore adopt a lightweight, accurate and efficient estimator—Benettin renormalization—which periodically re-normalizes the perturbation over short windows and accumulates the window-wise growth[4]:

$$\widehat{\lambda}_{\Delta t}^{(K)} = \frac{1}{K\,\Delta t}\sum_{k=1}^{K}\log\frac{\|\delta x_{k\Delta t}\|}{\|\delta x_{(k-1)\Delta t}\|}, \tag{1}$$

where Δt is the window length, K is the number of windows, and $T = K\Delta t$. Prior studies use zero input to probe the induced autonomous dynamics and reveal attractor structure in recurrent networks, and have shown that RNNs can exhibit chaotic behavior under such zero-drive conditions[5, 6]. Therefore, for LSTMs, after running LSTM on natural token inputs, we add a tiny Gaussian perturbation to the hidden state as well as cell state to form two nearby trajectories (inference the next token each time step). We then evolve both trajectories on an all-zero artificial input sequence of length 1000 and apply Benettin renormalization with fixed windows; this zero-drive sequence isolates the autonomous dynamics from input entrainment. Fig.1(a) shows how we calculate the FTLE. We repeat this measurement for every saved checkpoint during training. As shown in Fig.1(b), the resulting FTLE traces provide a sharp and repeatable phase marker: the first near-zero approach (or zero crossing) aligns with the first EoC and coincides with the onset of the first loss descent, and similar near-zero events recur before later descents.

3. FTLE-Based Optimizer

We then use the value of FTLE $\lambda_T = 0$ as guiding principle, create a controller to ensure the model approaches it every epoch. During training, we adjust the forget-gate bias (b_f) and output-gate biases (b_o) by proportional to the FTLE value (we use gate biases as controlled objects because they are effective at regulating stability weights when zero-token inputs are used during FTLE estimation):

$$b_f \leftarrow b_f - \eta_b\,\lambda_T, \qquad b_o \leftarrow b_o - \eta_b^{(o)}\,\lambda_T, \tag{2}$$

where $\eta_b^{(o)} \ll \eta_b$ (one order of magnitude smaller in our experiments). We freeze the gradients of these two biases(immune to back-propogation) and treat Eq. (2) as an external control input. When $\lambda_T > 0$ (chaotic regime), we set the optimizer weight decay proportional to FTLE so the regularizer can pull the model back to the edge of chaos; when $\lambda_T \leq 0$ (ordered / near-EoC), we set $wd = 0$ as back-propagation naturally makes the model less stable. This implements an asymmetric penalty that suppresses chaotic excursions only when the model is in the chaotic phase. Fig. 2 reports three hyperparameter settings. Across all settings, our controller (BIAS) consistently outperforms both baseline training (none) and constant weight decay (CWD)[7], while keeping λ_T close to zero. In the high-learning-rate regime, BIAS exceeds the ~90% ceiling of CWD and reaches a best validation accuracy of 91.11% on IMDB.

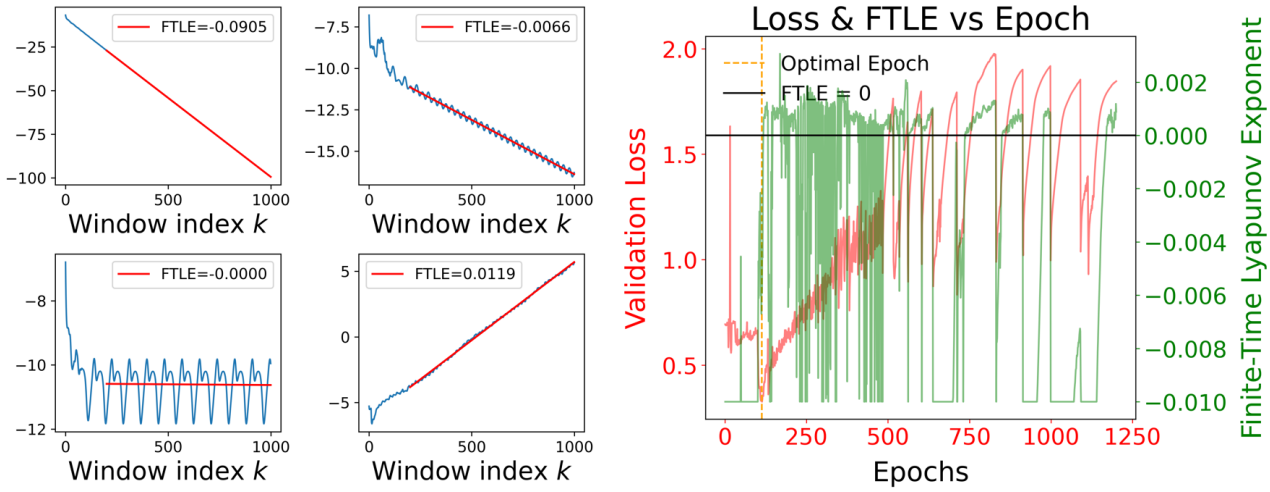

Fig. 1: Finite Time Lyapunov Exponent (FTLE) as chaos indicator on LSTM models: (a) Benettin-style FTLE estimation on all zero input tokens. The blue curve shows the cumulative log-growth $\sum_{k=1}^{K} \log \frac{\|\delta x_{k\Delta t}\|}{\|\delta x_{(k-1)\Delta t}\|}$ (updated every Δt steps with renormalization), and the red line is a linear fit after burn-in; its slope is the estimated (maximum) FTLE. (b) Checkpoint-wise FTLE (green, right axis) and validation loss (red, left axis) across epochs. The horizontal black line marks FTLE=0, and the dashed yellow line indicates the best-performing epoch (optimal epoch). Near-zero FTLE events align with the first optimal valley and recur near later loss descents, consistent with phase transitions around the edge of chaos.

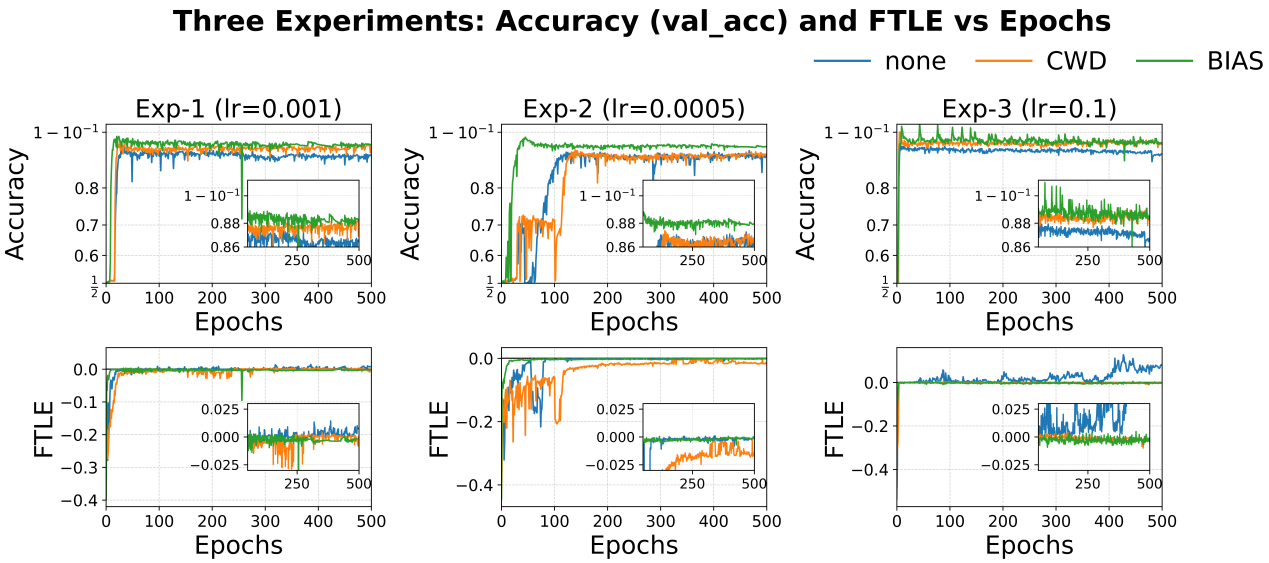

Fig. 2: FTLE-guided controller improves LSTM training under distinct hyperparameters. Columns show three training settings. Top row: validation accuracy vs. epoch (ACC y-axis uses a logit scale). Bottom row: FTLE estimated by Benettin renormalization (horizontal line: FTLE= 0) vs. epoch. We compare baseline training (**none**), constant weight decay (**CWD**), and our FTLE-based controller (**BIAS**: gate-bias controller with FTLE-related weight decay).

References

[1] Ling Feng, Lin Zhang, and Choy Heng Lai. Optimal machine intelligence at the edge of chaos. *arXiv preprint arXiv:1909.05176*, 2019.

[2] Nils Bertschinger and Thomas Natschläger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436, 2004.

[3] Wenbo Wei, Nicholas Chong Jia Le, Choy Heng Lai, and Ling Feng. Multiple descents in deep learning as a sequence of order-chaos transitions. *arXiv preprint arXiv:2505.20030*, 2025.

[4] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems. part 1: Theory. *Meccanica*, 15(1):9–20, 1980.

[5] Thomas Laurent and James von Brecht. A recurrent neural network without chaos. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1612.06212.

[6] Pravish Sainath. Recurrent neural networks as dynamical systems: Exploring chaos and order. OpenReview (Université de Montréal MAT 6215 submission), 2021.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.