Critical Neural Networks in Atari Games

Thomas Pluck[1] and Aaron McAfee[1]
[1]Department of Electronic Engineering
Maynooth University
Maynooth, Ireland
Email: {thomas.pluck.2025, aaron.mcafee.2021}@mumail.ie

***Abstract*—**

I. Introduction

II. Background

A. Reinforcement Learning

Reinforcement Learning (RL) provides a computational framework in which a training agent learns to make sequential decisions through interacting with an enviroment to maximize rewards [**?**]. Traditionally, RL problems are modeled as Markov Decision Processes (MDPs) defined by the tuple (S, A, P, R), where S is the set of states called the state space, A is the set of actions called the action space, P is the transition probablity function and R is the reward function [**?**]. The goal is to find a policy $\pi(a \mid s)$ that maximizes the expected return, either via the state-value function

$$V^{\pi}(s) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s\Big] \tag{1}$$

or the action-value function

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a\Big]. \tag{2}$$

Bellman's optimality equations characterize the unique fixed point Q^* (or V^*) satisfying

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s' \mid s, a) \max_{a'} Q^*(s', a'), \tag{3}$$

enabling dynamic-programming solutions when P and R are known [**?**].

Q-learning is a model-free, off-policy algorithm that uses this optimality form to iteratively update $Q(s, a)$ toward Q^* via

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) \tag{4}$$

$$+ \alpha\Big[r_t + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)\Big] \tag{5}$$

while balancing exploration (e.g. ε-greedy) and exploitation [**?**].

As a variant of Q-learning, Deep Q-Networks [**?**] were introduced, employing convolution Neural Networks (CNNs) to approximate $Q(s, a)$ directly from high-dimensional sensory inputs, emabling end-to-end learning from raw pixels. DQN was able to demonstrate human-level performance on 49 atari games through experience-replay buffers and periodically updated target networks [**?**].

B. Criticality in Neural Systems

The criticality hypothesis posits that biological neural systems self-organize to operate near critical points between ordered and chaotic dynamics [**?**], [**?**]. Empirical evidence includes observations of "neuronal avalanches" in cortical tissue with size distributions following power laws with exponents of approximately -3/2, matching predictions from critical branching processes [**?**].

Neural networks near criticality demonstrate optimal computational properties, including maximized dynamic range [**?**], [**?**], information transmission [**?**], and information storage capacity [**?**]. Conversely, deviations from criticality correlate with neural pathologies [**?**], suggesting that maintaining criticality is essential for healthy brain function.

These findings motivate our approach: rather than training networks that may accidentally drift away from criticality, we leverage RG flow analysis to design networks that intrinsically maintain critical dynamics throughout operation.

C. Edge of Chaos

Dynamical systems can be categorized by the exponential rate of divergence of trajectories initially perturbed by a minimal δ_0, leading to asymptotic divergence $|\delta(t)| \approx e^{\lambda t}|\delta(0)|$ - the value of this λ known as the Lyapunov exponent characterizes stable ($\lambda < 0$), chaotic ($\lambda > 0$) and weakly chaotic or edge of chaos ($\lambda \approx 0$) regimes [].

Each of these regimes supports various diffusion statistics on ensembles, with the power law distributions typical of criticality only emerging at the edge of chaos. This makes the edge of chaos a necessary but not necessarily sufficient condition for dynamical systems to exhibit criticality. []

In ANN systems in particular the weakly chaotic regime can be determined in terms of the L_2 norm of the Jacobian of the neural network itself []. We show in appendix, that in minimizing this specific criterion in terms of ANN parameters, the edge of chaos in ANN systems is actually sufficient for network criticality to arise.

III. Methodology

A. Edge of Chaos Regularizer

The proposed method involves the addition of a regularization term to the loss function, which guides the network

towards the edge of chaos and in a linear approximation induces criticality - the full derivation can found in the appendix of this paper:

$$R_{(layer)} = \frac{2\sigma''(z)\nabla_x^2\sigma(z)}{\sqrt{N}}\left(\frac{1}{N} - \frac{1}{\|\nabla_x\sigma(z)\|}\right) \quad (6)$$

Where N is the number of neurons, x is the input vector, z the pre-activation affine transform of x, σ an elementwise activation non-linearity and ∇ and ∇^2 the gradient and Laplace operators respectively.

B. Enviroment Setup

The experiments carried out in this paper uses the Arcade Learning Enviroment (ALE), which provides a standardized interface to 26 of atari 2600 games as challenge problems for general agents [**?**]. Each game is instantiated via the OpenAi Gym API with deterministic frame-skipping wrappers to ensure consistent dynamics and reproducibility. To handle the varying control schemes across different games, the maximum discrete actionspace is computed among the verified enviroments. Observations of raw frames (210×160×3) are converted to grayscale, resized to 84x84 pixels and max-pooled over two consecutive frames to mitigate flickering artifacts before stacking four frames into an 84x84x4 tensor to encode short-term motion [**?**] Rewards are clipped to [-1, +1] to bound temporal-difference targets across games with widely varying reward scales, and actions are taken from the minimal discrete action set, through Gymnasium's API.

C. Experiments

IV. Results

V. Discussion

VI. Conclusion

Acknowledgment

The authors would like to thank the instructors and teaching assistants of CS637 for their guidance and support throughout this project. We extend our gratitude to our peers for their valuable feedback during presentation rehearsals and draft reviews. Special thanks to Barak Pearlmutter for introducing us to key concepts in reinforcement learning that formed the foundation of this work. We also acknowledge the resources provided by our university's computing facilities that enabled our experimental implementations. Finally, we thank our families and friends for their unwavering support throughout our academic journey.

References

[1] J. M. Beggs and D. Plenz, "Neuronal avalanches in neocortical circuits," *Journal of Neuroscience*, 2003.

[2] J. M. Beggs and N. Timme, "Being critical of criticality in the brain," *Front. Physio.*, 2012.

[3] O. Kinouchi and M. Copelli, "Optimal dynamical range of excitable networks at criticality," *Physical Review E*, 2006.

[4] W. L. Shew, H. Yang, T. Petermann, R. Roy, and D. Plenz, "Neuronal avalanches imply maximum dynamic range in cortical networks at criticality," *Journal of Neuroscience*, 2009.

[5] N. Bertschinger and T. Natschläger, "Real-time computation at the edge of chaos in recurrent neural networks," *Neural Computation*, 2004.

[6] C. Meisel, A. Storch, S. Hallmeyer-Elgner, E. Bullmore, and T. Gross, "Failure of adaptive self-organized criticality during epileptic seizure attacks," *PLoS Comput. Biol.*, 2011.

Appendix

In this appendix, we provide a rigorous derivation of our proposed regularization term that promotes criticality in neural networks. Our approach drives networks to the edge of chaos by explicit Jacobian constraints, incidentally the derived regularizer also pushes the network towards scale-free dynamics - providing a tentative theoretical justification that the edge of chaos may be sufficient for criticality in ANNs.

A. Regularizing to the Edge of Chaos

Let us define a standard feedforward network $a = \sigma(z)$ where preactivation $z = Wx+b$ is defined with weight matrix W, bias b, input x and the put through non-linearity σ to create activation a. A known fact about rank-N operators J is that their Lyapunov exponent collapse when $\|J\|_F^2 = N$ at the so-called "edge of chaos".

We derive our regularizer by letting J be the Jacobian of the feedforward layer $a = \sigma(z)$ and finding explicit derivatives in terms of weights W and biases b to minimize the quantity J - to simplify derivation we will focus entirely on individual entries of b the b_i and assure the reader that much same terms will arise when computing the derivative of W_{ij}

Let us begin by computing the derivative of a_i with respect to x_j for the individual terms of the Jacobian J_{ij}:

$$J_{ij} = \frac{\partial}{\partial x_j}\sigma(z_i) = W_{ij}\sigma'(z_i) \quad (7)$$

We can now compute the Frobenius norm of J and begin computing it's derivative w.r.t. b_i:

$$\frac{\partial}{\partial b_i}\|J\|_F = \frac{\partial}{\partial b_i}\sqrt{\sum_{i,j} W_{ij}^2\sigma'(z_i)^2} \quad (8)$$

$$= \frac{\sum_j W_{ij}^2\sigma'(z_i)\sigma''(z_i)}{\|J\|_F} \quad (9)$$

We note at this juncture that $\frac{\partial^2}{\partial x_j^2}\sigma(z_i) = W_{ij}^2\sigma''(z_i)$ so we may write:

$$\frac{\partial}{\partial b_i}\|J\|_F = \frac{\sigma'(z_i)\nabla^2\sigma(z_i)}{\|J\|_F} \quad (10)$$

Where ∇^2 is the Laplace operator. We would now like to encode the edge of chaos criterion $\|J\|_F^2 = N$ into an explicit quantity that we can minimize using the parameters of our network.

$$\frac{\partial}{\partial b_i}\left(1 - \frac{\|J\|_F}{\sqrt{N}}\right)^2 = \frac{\partial}{\partial b_i}\left(1 - \frac{2\|J\|_F}{\sqrt{N}} + \frac{\|J\|_F^2}{N}\right) \quad (11)$$

$$= 2\frac{\partial}{\partial b_i}\|J\|_F \cdot \frac{\|J\|_F}{N} - \frac{\partial}{\partial b_i}\frac{2\|J\|_F}{\sqrt{N}} \quad (12)$$

$$= \frac{2\sigma'(z_i)\nabla^2\sigma(z_i)}{\sqrt{N}}\left(\frac{1}{N} - \frac{1}{\|J\|_F}\right) \quad (13)$$

B. Properties of the Regularizer

At this juncture we can begin to discuss the properties of the proposed regularizer - while it clearly regularises the network toward the edge of chaos, it is not immediately clear from the form derived that minimizing this quantity leads to criticality and scale-free phenomena.

The simplest observation comes from understanding the nature of the Laplacian and Jacobian when applied to re-scaled linear maps. Given some linear map $L : \mathbb{R}^n \to \mathbb{R}^n$, the effect of rescaling this map as $L'(x) = L(x/\alpha)$ has a quadratic action on the Laplacian such that $\nabla^2 L'(x) = \nabla^2 L(x/\alpha) = (1/\alpha^2)\nabla^2 L(x/\alpha)$, and similarly a linear action on the Jacobian where $J_{L'}(x) = J_L(x/\alpha) \cdot (1/\alpha)I = (1/\alpha)J_L(x/\alpha)$. This demonstrates that the Laplacian scales by a factor of $1/\alpha^2$ while the Jacobian scales by a factor of $1/\alpha$ under coordinate rescaling.

So looking at the derived form above, we can see it as the difference between a quadratic and linear scaling operator (at least in the linear sense) - if we then attempt to minimize this quantity, it follows that we are trying to find a regime in which linear and quadratic scaling are equivalent, ie. the system is scale-free.

The extent that this analysis can be extended to the full quasi-linear case seen in neural networks would require a complete renormalization group analysis to formalize this connection rigorously, as the non-linearities in neural networks introduce complexities beyond the simplified linear map case presented here.