ANT: Artificial Neural Topology

Woody Hulse, Jonah Schwam, Pavani Nerella, Ilija Nikolov, Taishi Nishizawa
Department of Computer Science
Brown University
Providence, RI, 02912
`woody_hulse@brown.edu`

***Abstract*—Rapid development in artificial intelligence (AI) research has led to increasingly larger artificial neural networks (ANNs). Some of the largest ANNs now have parameter sizes that rival the neuron and synaptic counts of intelligent biological organisms. However, these models have yet to demonstrate the capacity to reason in areas outside of their training domain, leaving a gap in AI research efforts towards artificial general intelligence (AGI). The combined inability of ANNs to replicate the complex graph structure and temporal statefulness of information travel in biological neural circuits we believe results in ANN hypothesis classes that are too narrow for general reasoning. To address both of these issues, we propose ANT, a time-state preserving topological analogy to biological neural networks. In this paper, we show the performance and task-generalizable capabilities of ANT in reinforcement learning (RL) settings compared to conventional artificial neural networks and discuss the viability for larger-scale ANTs for general reasoning tasks.**

I. INTRODUCTION

The past several years have seen monumental research gains in Deep Learning (DL). Recent advancements in artificial neural network architectures such as attention-based models [1], large-scale multi-modal models [2], and generative models have pushed against, and in some cases past, human abilities in certain tasks [3]. Paired with the availability of modern compute resources like Graphics Processing Units (GPUs), the scale of some of the largest ANNs nears the order of magnitude of sophisticated biological agents[1] known to possess complex reasoning behavior. Most models are only designed to be capable within the scope of their training and design, however some notable models, particularly LLMs and RL embodied agents, have invigorated discussions on the bridge between task-specific intelligence and general intelligence in AI agents [4] [5]. These discussions, while in part predicated on more philosophical ideas of self awareness, autonomy, and consciousness, have been unable to conclude the necessary human-like ability to reason–that is, the ability for an agent to complete a task within an area it has not yet experienced [6]. To investigate the reason behind the lack of general reasoning in ANNs given comparable computational complexity requires a consideration of how biological circuits function, as this question is likely explainable by the characteristics important in neurological circuits not present in artificial networks.

TABLE I
PARAMETER COUNT OF A SELECTION OF NEURAL NETWORKS.

Name	Type	Parameters
Human (*Homo sapiens*)	Biological	$1.5 \cdot 10^{14}$ [7]
Cat (*Felis catus*)	Biological	$1.0 \cdot 10^{13}$ [8]
Gemini 1.5 (Google)	Artificial	$^{*}2.4 \cdot 10^{12}$
Claude 3 Opus (Anthropic)	Artificial	$^{*}2.0 \cdot 10^{12}$ [9]
GPT-4 (OpenAI)	Artificial	$1.0 \cdot 10^{12}$ [10]
Rat (*Rattus norvegicus*)	Biological	$4.5 \cdot 10^{11}$ [11]
GPT-3 (OpenAI)	Artificial	$1.7 \cdot 10^{11}$ [10]
Mouse (*Mus musculus*)	Biological	$1.0 \cdot 10^{12}$ [12]
Honey Bee (*Apis mellifera*)	Biological	$1.0 \cdot 10^{9}$ [13]

*denotes unpublished estimates

There are several viable explanations for the gap in generalized cognitive ability between artificial networks and animals, including the constraints on training regimes of most ANNs, a lack of real-world exploration, insufficiently sophisticated artificial neural network architectures, or poor search of the parameter space, among others[2]. While each may contribute in part to the observed disparity, we argue the importance of two fundamental characteristics of biological neural networks not present in ANNs.

Statefulness as a property broadly pertains to the capacity of a system to retain information about its previous states. In neural circuits, this ability allows for the retention of data from earlier sensory inputs or cognitive activities, which in turn influences subsequent neural responses and behavioral outputs. The dynamics of neural statefulness are complex and involve various physical mechanisms, including short-term synaptic plasticity, signal potentiation, and presynaptic excitability. For instance, tetanic activation can temporarily enhance synaptic strength in a phenomenon known as augmentation. The synapse exhibits statefulness by amplifying signals that follow an initial activation signal [14].

Moreover, the graph-like network topology of biological neural circuits facilitates multiple communication pathways among neurons, enhancing redundancy and robustness in neural processing. This structural complexity allows neural circuits to explore a broader hypothesis space[3] compared to layered ANNs, thus enabling the circuits to support and adapt

---

[1] A one-to-one comparison between biological and artificial neural network parameters is impossible due to the complex quantum-chemical nature of neural circuits, so we consider the estimated synaptic count to be the number of "parameters" in biological neural networks. This is by no means rigorous, it is purely illustrative of scale (see Table 1).

[2] Later on, we'll show that the formulation is flawed even in small parameter spaces, showing that at least a significant component of the gap is explainable by the formulation itself.

[3] By hypothesis space, we are referring to the space of all possible functional mappings that a model can produce

to more diverse logical functions. Basset & Bullmore indicate that specific graph structures, such as small-world networks, are particularly efficient in balancing global and local connectivity, which improves both the speed and accuracy of neural communication [15]. The combined attributes of statefulness and graph-like architecture contribute to the ability of neural circuits to undertake complex decision-making processes, adapt to new environments, and learn from prior experiences.

The artificial neural topology is not a purported approximation of the exact function and behavior of biological neural networks. We make no aim to reconstruct the complex electrochemical interactions key in neuron-to-neuron interactions. Rather, our goal is to construct a network with a hypothesis class and hyperparameter space analogous to biological networks, particularly with respect to these state and topological characteristics absent in conventional ANNs.

II. RELATED WORKS

As artificial neural networks evolve, despite the advancements, current ANNs still exhibit substantial deficiencies in generalized reasoning and cognitive flexibility when compared to biological systems. These deficiencies, we hypothesize, stem not from the scale of neural parameters but from fundamental differences in network architecture and dynamic information processing capabilities.

Foundational research by Azevedo et al. [7] provides a detailed comparison of neuronal and nonneuronal cell counts across different species, illustrating that human brains are an isometrically scaled-up version of smaller primate brains. This scaling is not purely in terms of size but also involves complex interconnections and synaptic densities [7]. Further, Herculano-Houzel et al. delve into the isotropic fractionator method and the cellular scaling rules for rodent brains, establishing a quantitative baseline that directly informs the structural complexity needed in ANNs to approach biological realism [20], [21]. These studies suggest that the quest for artificial general intelligence (AGI) may be less about size and more about the intricate connectivity and stateful processing found in natural neural networks.

In "Temporal Dynamics and Statefulness in Biological Systems," Menzel and Giurfa (2001) explore the cognitive architectures of honeybees, demonstrating how even with a relatively small number of neurons, complex behaviors and decision-making capabilities can arise from highly optimized neural circuitry [13]. This underscores the importance of stateful, dynamic interaction patterns in neural networks, which are often absent in traditional ANNs.

Drawing from Ananthanarayanan et al., who performed cortical simulations with a scale of 10^9 neurons and 10^{13} synapses, it becomes evident that simulating complex neural interactions requires not only computational power but also an innovative approach to neural connectivity and architecture [8]. Their work highlights the potential and the significant challenges in mimicking the structure and function of large-scale neural networks, which directly influences the design philosophy of ANT.

The discussion of task-specific intelligence versus general intelligence in AI has been a focal point of recent philosophical and empirical debates. Mijwil et al. critically examine the implications of large-scale language models on academic integrity, reflecting on their capabilities and limitations in generalized settings [30]. These discussions provide a backdrop against which ANT is positioned, aiming to transcend the typical task-specific optimizations of current ANNs by embracing a more holistic, biologically inspired framework.

There are some network types that are designed with similar approaches to ANTs. Hopfield networks, originally created in 1977, are fully-connected state-preserving networks whose aim is to "memorize" a particular binary network state in such a way that, given random initialization, the state of each neuron of the network will converge toward its memorized state. This type of network is useful in understanding how memory is stored in a neurological sense and some variants have found modern applications as an auxiliary component of deep neural networks [17]. However, these networks follow a strict fully connected structure, only store binary information, and have yet to be formulated in the context of RL.

An alternative approach to improved ANN performance is the spiking neural network (SNN). This network type aims to bridge the gap between biological and artificial neural networks by mimicing neuron-neuron interactions in the brain. Specifically, rather than sending continuous signals throughout the network, SNNs send "spikes" between adjacent neurons when an action potential is reached [25]. This strictly biological approach relies on the foundation that only a strict artificial reconstruction of biological neural networks will bridge the performance gap. From a machine learning theory sense, however, SNNs are still highly restrictive in the hypothesis space. Although they may include some of the properties not present in other ANNs, with an "all-or-nothing" binary approach, SNNs still strictly mediate how information can propagate and have many other issues such as nondifferentiability over the spike discontinuities leading to incorrect gradients and often requires neuromorphic hardware to run efficiently [18].

Our approach with ANT integrates these insights by developing a non-layered, stateful architecture that more closely replicates the interconnected and ongoing processing capabilities found in biological brains. By incorporating a graph-based structure and dynamic information flow, ANT seeks to foster a level of adaptability and cognitive flexibility that conventional ANNs have yet to achieve. Our idea is deeply rooted in a comprehensive understanding of biological neural networks, drawing from seminal works in neuroscience and recent advancements in computational models. By addressing the structural and operational limitations of traditional ANNs, ANT aims to bridge the gap between artificial and biological cognitive systems, paving the way for more sophisticated and versatile AI agents.

III. METHODOLOGY

ANT seeks to improve on existing neural structures primarily through a fundamental expansion in the hypothesis space of ANNs, in particular lifting the restrictions which reqire unidirectional flow of information and gradients through the network and that disallow the storage of a dynamic state as a prior to the model output.

A. *Graph initialization*

ANT is constructed with a randomly locally connected directed graph G of size $|G| = n$. Each vertex v_i has weight w_i, bias b_i, and activation σ_i properties with learnable w_i, b_i, performing a similar mapping from the inputs of a neuron to it's outputs compared to ANNs. For the graph, there are two sets of vertices designated as input and outputs to the network, respectively, representing the sensory and motor components to the network.

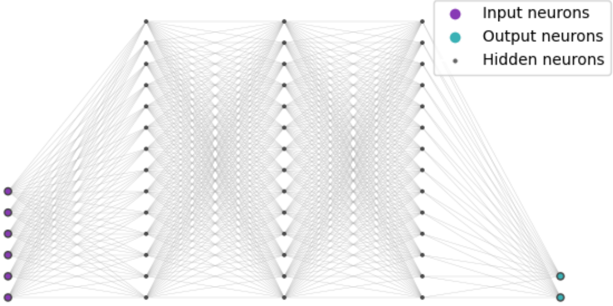

Figure 1. Example ANN architecture (baseline)

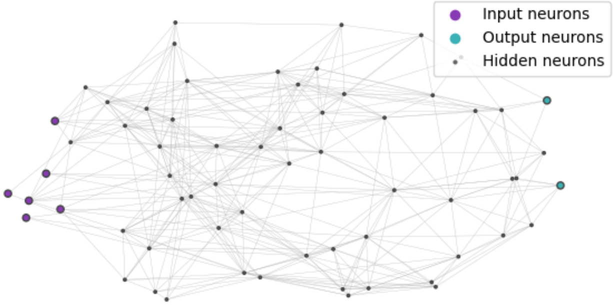

Figure 2. Example ANT architecture (initialization)

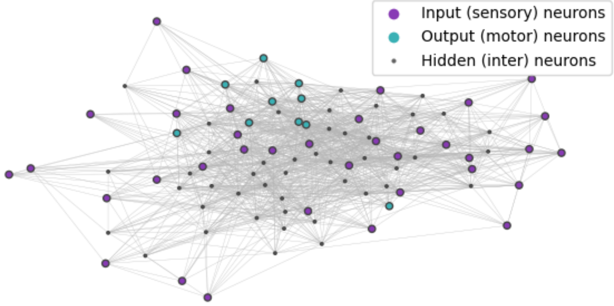

Figure 3. Subgraph of *C. elegans* neural topology [16]

This structure, while not an exact replica of any particular neural topology, when combined with the genetic algorithm (see III.*C*) provides a superset of the hypothesis class which can mimic the neural circuit structures found in biological networks by forming recurrent, potentially cyclic dynamic relationships between neurons.

B. *Update and Learning Mechanism*

Unlike conventional artificial neural networks, ANT, like its biological counterpart, behaves dynamically across time. At each time step t a set of inputs $\mathcal{X}$ of length ℓ are distributed to each of ℓ input neurons (where ℓ is predefined at network initialization). Then, each neuron v_i simultaneously computes $y_i = x_i w_i + b_i$ and applies σ_i on y_i, where x_i at time t is the outputs of inbound neurons at $t-1$, and distributes each vector component of y_i through corresponding outbound edges. The output $\hat{\mathcal{Y}}$ of the network is the concatenated single-dimensional outputs of each designated output neuron.

Recent advances in the study of biological neuron interactions reveal complex learning mechanisms within individual dendritic connections and synaptic clefts, as well as the known properties of the soma [26] [27] [28] [29]. With this, we model individual neurons as multiple perceptron relationships between inputs and outputs, rather than a more conventional single perceptron approach. Practically, each neuron then has a two-dimensional weight matrix with a 1-dimensional bias vector.

Because ANT is designed to behave dynamically and learn continually, the gradient computation must be compatible with online updates as well. An ANN performs a gradient update given a loss function L by passing backward the product of an upstream jacobian $\partial L / \partial x_i$ with locally computed input gradients to compute a partial derivative of w_i, b_i, and x_i with respect to loss. For w_i,

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial x_m} \cdot \frac{\partial x_m}{\partial x_l} \cdots \frac{\partial x_k}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_i}, \tag{1}$$

in a network where $v_i \to v_j \to \cdots \to v_l \to v_m$ However, a conventional backpropagation approach from ANNs mandates an acyclic graph to eliminate a circular dependency causing infinite gradient chains, preventing precomputing backpropagation chains. Therefore, gradients must be dynamically passed backward in a similar manner to forward propagation, where only one gradient passing "step" occurs for each t. This links the dependence of each partial gradient component of the chain to the time step at which it was computed, barring cancellation in the chain-rule expansion for L, w_i:

$$\frac{\partial L}{\partial w_{i_{t+k}}} \neq \frac{\partial L}{\partial x_{m_t}} \cdot \frac{\partial x_{m_{t+1}}}{\partial x_{l_{t+1}}} \cdots \frac{\partial x_{k_{t+k-1}}}{\partial x_{j_{t+k-1}}} \cdot \frac{\partial x_{j_{t+k}}}{\partial w_{i_{t+k}}} \tag{2}$$

To mathematically satisfy this approach, we can closely approximate the gradient chain by instead only episodically updating network weights at relatively distant time intervals, removing the weight parameters' dependence on time. Then, for any adjacent neurons $i \to j$ passing gradients backward at $t, t+1$, respectively,

$$\begin{aligned}\frac{\partial x_{k_t}}{\partial x_{j_t}} \cdot \frac{\partial x_{j_{t+1}}}{\partial x_{i_{t+1}}} &= \frac{\partial \sigma_j(y_{j_t})}{\partial y_{j_t}} \cdot \frac{\partial y_{j_t}}{\partial x_{j_t}} \cdot \frac{\partial \sigma_i(y_{i_{t+1}})}{\partial y_{i_{t+1}}} \cdot \frac{\partial y_{i_{t+1}}}{\partial x_{i_{t+1}}} \\ &= \frac{\partial \sigma_j(y_{j_t})}{\partial y_{j_t}} \cdot w_{j_t} \cdot \frac{\partial \sigma_i(y_{i_{t+1}})}{\partial y_{i_{t+1}}} \cdot w_{i_{t+1}} \qquad (3) \\ &= \frac{\partial \sigma_j(y_{j_t})}{\partial y_{j_t}} \cdot w_j \cdot \frac{\partial \sigma_i(y_{i_{t+1}})}{\partial y_{i_{t+1}}} \cdot w_i\end{aligned}$$

which, with an activation function that's near linear[4] across most of $\mathbb{R}$, approximates

$$\begin{aligned}\frac{\partial \sigma_j(y_{j_t})}{\partial y_{j_t}} \cdot w_j \cdot \frac{\partial \sigma_i(y_{i_{t+1}})}{\partial y_{i_{t+1}}} \cdot w_i &\approx 1 \cdot w_j \cdot 1 \cdot w_i \\ &= \frac{\partial x_k}{\partial x_j} \cdot \frac{\partial x_j}{\partial x_i} \qquad (4) \\ &= \frac{\partial x_k}{\partial x_i}.\end{aligned}$$

Applying this principle to the full gradient computation, ANT achieves a parallelizable dynamically learning algorithm, similar to biological networks.[5]

C. *Evolution*

To further emulate its biological counterpart, ANT also performs a genetic search over the hyperparameter space. Specifically, ANT "evolves" by modifying the connectivity and neuron count of the network. For each mutation and with uniform $E \sim \text{Unif}(0,1)$, edges are added with probability $E \leq p \cdot \frac{2e}{n(n-1)}$ and removed with probability $E \leq p_e \cdot \frac{n(n-1)-2e}{n(n-1)}$, where e, n are the current number of edges and neurons, respectively, and p_e is a hyperparameter. The number of new neurons is defined by $N \sim \text{Pois}(p_n)$ with hyperparameter p_n.[6] The evolution of ANT allows for large and complex networks to form, fully taking advantage of the broader search space enabled by the graph structure. Further, the ability of the genetic algorithm to change the topology of ANT generates selectively faster information travel between areas of the network where a locally connected network would otherwise require several time steps to propagate information. This selective minimization of the degrees of separation between distal vertices is important for the learning speed of the at-large network, quick reactionary mechanisms, and other properties of biological circuits [8] [13].

D. *Reinforcement Learning*

Due to the dynamic nature of ANT coupled with the objective of pursuing generalized intelligence, we use reinforcement learning (RL) as a medium of comparison between ANT and ANNs. RL allows us to pursue an ultimate goal of producing continual real-world learning by investigating the

Fig. 1. Evolution of a small ANT.

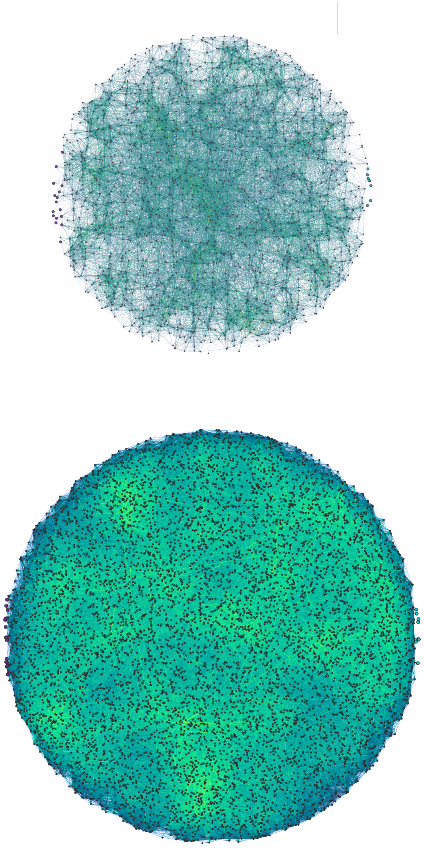

Fig. 2. Evolution of a large ANT. Inter-neuron activity is indicated on the "viridis" color scale.

---

[4] An example of such an activation is $a_n(x) = n\tanh(\frac{x}{n})$ with hyperparameter n, where $a_n(x) = x$ as $n \to \infty$

[5] We use the RMSProp [19] as a parameter optimizer for this work.

[6] We have not implemented neuron pruning, but this could also be included as a random variable or based on activity.

behaviors and convergence properties of ANT against ANNs in complex environments graphically, as well as introduce agents to new environments or alterations in underlying properties of different environments.

Within RL, ANT computes actions unconventionally compared to most other DQN or deep policy networks by taking only the maximum likelihood action. Instead of mandating exploration with a user- or agent-controlled hyperparameter, we found exploration emergent from the dynamic structure of ANT. Intuitively, due to the variability of the input space, ANT cannot overfit in a traditional sense by forming explicit input-output mappings. In other machine learning settings this is a challenge due to ill-defined mappings, but in RL this leads to implicit exploration during the continual development of the network's internal state.

We also use the REINFORCE (REward Increment = Non-negative Factor × Offset Reinforcement × Characteristic Eligibility) [23] policy gradient method used widely across RL. We apply REINFORCE in a semi-online setting. As described before, gradient computation must occur in real time due to the time-dependence of the dynamically changing internal state. To comply with this constraint, we accumulate gradient-reward pairs for T time steps before applying REINFORCE to compute a gradient for the policy:

$$\sum_{t=0}^{T}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\cdot\sum_{k=t}^{T}\gamma^{k-t}\cdot r_k\right] \tag{5}$$

with discount factor γ at each T interval where $T >> 1$.

E. Experimental Setup

Our experiments are predicated on assessing ANT's ability to converge both more quickly and adaptively compared to its ANN counterpart. To assess this, we designed an ANN analogous to ANT with identical weight initializers, mutation parameters, neural activations, and RL integration but within a layered, non-stateful structure. We compare the two with a pair of discrete action space parametrized reinforcement learning environments from Gym: Acrobot and Lunar Lander.[7] Acrobot has a double pendulum with agent able to control the force applied to the top axle. The agent is tasked with applying a sequence of forces such that the end of the double pendulum exceeds a particular height. The agent receives -1 reward at each time step it does not complete the task. Lunar Lander consists of a lunar module initialized with a high position and some velocity, where the agent is tasked to apply leftward, rightward, upward, or no thrust to land gracefully between two target flags positioned on noisy terrain.

We complement these environments with the Context Adaptive Reinforcement Library (CARL) [22], a modification of Gym and several other RL libraries which allows control over environment physics parameters for use in evaluating the generalizability of RL models. In their paper, they posit that if RL models are able to more quickly and reliably converge to high-reward solutions in environments which have object properties that vary, then they are more broadly capable. While imperfect for investigating the capacity for ANT to achieve a general intelligence in cognitive sense, at the scale of only a few dozen neurons we rationalize that quick convergence and adaptability could translate to more complex reasoning behavior for larger-scale, more refined topologies.

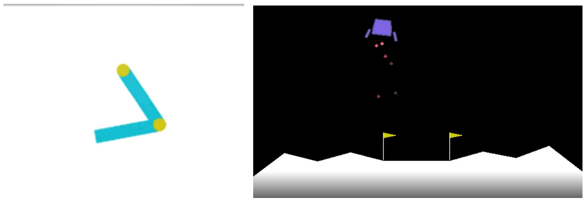

Fig. 3. Acrobot (left) and Lunar Lander (right).

To train parameter efficiency with respect to convergence speed and robustness, we create a "small" version of each model with 36 neurons and roughly ~ 450 trainable parameters and a "large" version of each with 66 neurons and $\sim$1170 trainable parameters. In both cases, the ANN is structured as an MLP with 2 hidden layers while the ANT has a connectivity parameter p_e of 1.4, each with input and output sizes equal to the observation and action spaces of each respective environment.

We first compare the efficacy of an isolated gradient descent approach for training in each environment. For each experiment, we run 1000 episodes of the environment and report the best result across a search of 20 unique learning rate parameters.

We run a similar experiment to test the joint gradient descent-evolutionary learning algorithm. For these experiments, we fix the two initial network structures as in the previous experiment. For 20 evolutionary episodes, we first collect the top $k = 5$ networks from the previous episode, perform perturbations with a neuron addition rate regulated by $N \sim$ Pois(0.5), an edge mutation rate with $P \sim$Binom(0.05), and a weight perturbation with $W \sim \{\mathcal{N}(0, 1 \times 10^{-4})\}_{i=1}^{e}$ for a network with e edges.[8] Each network is then trained for 100 episodes, after which an average reward is computed to determine the top-k. Below is an illustration of the best performing evolutionary path for the ANN and ANT in each environment, as well as close-performing neighbors.

Using the CARL environments, we design an experiment for each environment to assess relative generalizability of ANNs and ANTs. There are two primary characteristics that will measure this: convergence robustness and adaptability. These principles are predicated on the idea that a more generally

---

[7] Gym is an open-source Python library originally developed by OpenAI and now maintained by the Farama Foundation.

[8] We maintain the same graph properties for the ANN when evolving, only adding vertices and edges to layers as opposed to randomly connecting each to the existing graph as in ANT evolution.

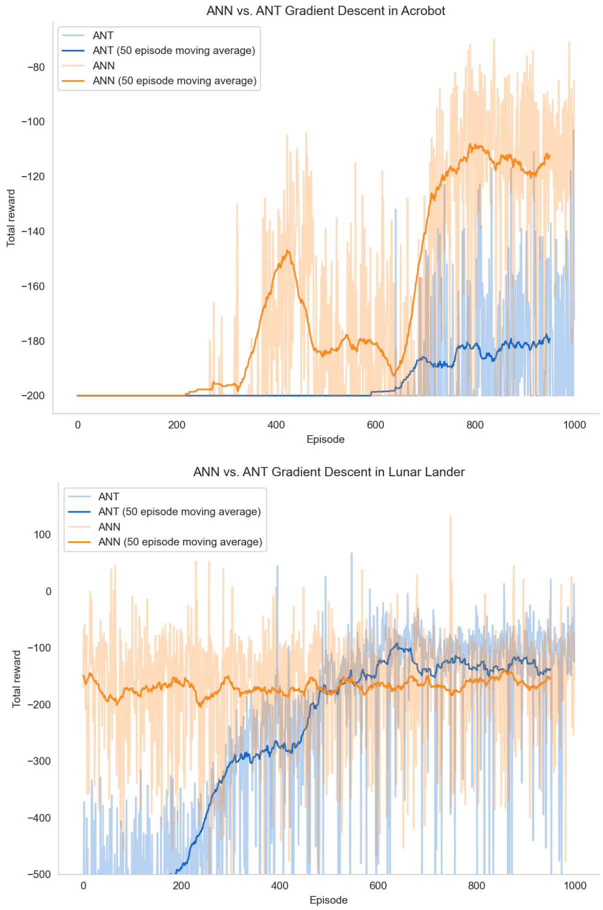

Fig. 4. Single training run of ANN and ANT in Acrobot (top) and Lunar Lander (bottom) with only gradient descent, not necessarily representative of relative performance.

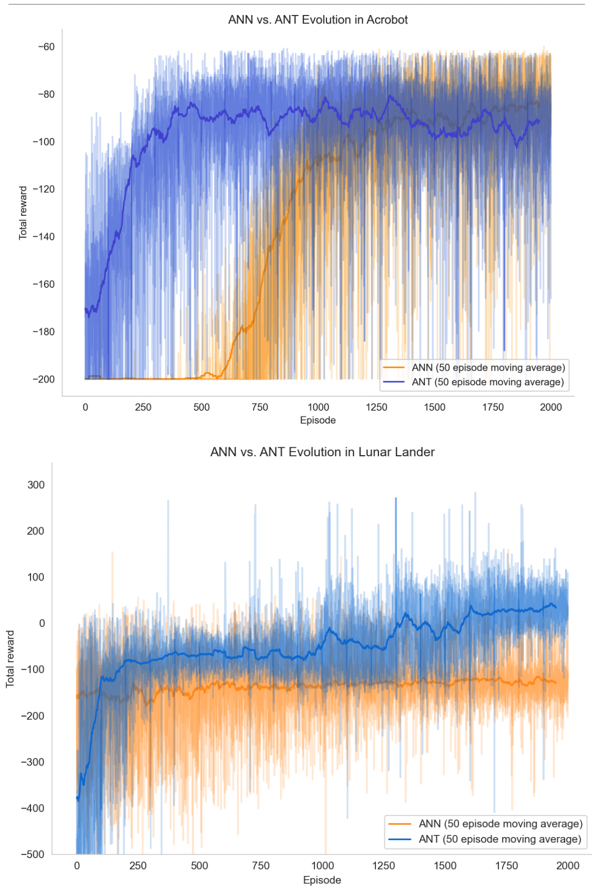

Fig. 5. Single training run of ANN and ANT in Acrobot (top) and Lunar Lander (bottom) with evolution and gradient descent, not necessarily representative of relative performance.

Fig. 6. Sample Acrobot initializations with CARL

intelligent agent is better able to reason within a complex environment and can more quickly adapt to new environments.

To test overall convergence robustness, we float the lengths and vertex masses for each of the pendulums in the Acrobot double pendulum with a uniform distribution ranging from a 50% decrease to a 50% increase for each default value. We then run the same evolutionary experiments on the small and large versions of each model as defined above, where for each episode these distributions are resampled to give new physics parameters.

We test adaptability by evolving an ANN and ANT in Lunar Lander with Earth's gravity (9.81 m/s^2) until each reaches an average test reward of 50. Then, only allowing gradient descent, we evaluate the time to convergence in an environment with the gravity of Mars (3.71 m/s^2). Each network will have to adapt in the way they apply thrust to the module while observing an entirely new action-state pairing mechanism.

F. Robotics

Operating from the Reward is Enough hypothesis, which posits that all aspects of intelligence subserve reward maximization by an agent acting in its environment, ANT can behave analogously to animals through policy optimization [10]. Conversely, this implies that our proposed agent can develop multiple aspects of intelligent behavior given a simple design of such a reward function.

We developed both a virtual and physical agent with analogous environmental setups, where an agent navigates a 2D environment populated with obstacles and food. Upon reaching a food object, the agent receives a reward. The food-finding reward is intended to incentivize the agent to efficiently navigate its environment to move toward the food object. In generalizing the environment to a small sensor-based observation space, the model forgoes assumptions about its environment and embraces continual learning. This further enables the agent to adapt to a non-stationary environment. At each timestep, the agent makes a partial observation of its state, formulated as a 2-dimensional feature vector encoding the distance to the nearest object or obstacle in front of the agent as well as a boolean value indicating if food is present in the agent's field of vision. The agent can take 5 possible actions at each timestep: wait, rotate right, rotate left, move forward, or move backward.

IV. RESULTS

We aim to demonstrate that, while conventional artificial neural networks are unable to converge quickly for small parameter counts, ANT can converge both quickly and robustly with high sample and parameter efficiency. We will first represent a sample learning trajectory for each experiment, then establish the rate of convergence.

A. *Baseline Reinforcement Learning*

With ANN-small and ANT-small as networks defined with 36 neurons and 66 neurons, respectively, we take the mean final reward over 100 different training runs with each model type.

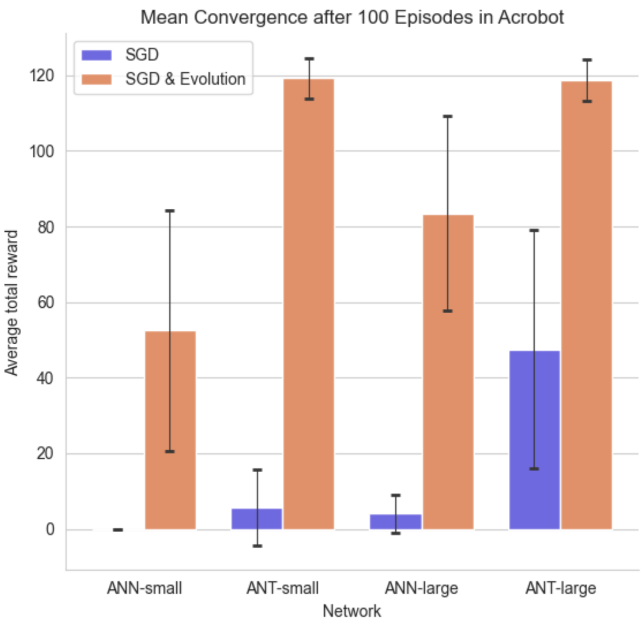

Fig. 7. ANN and ANT 100 episode convergence for Acrobot.

In Figure IV-A, this experiment is run on Acrobot with an expected optimal total reward of approximately 140 for a 200 time step game,[9] with a sufficiently positive result exceeding 100 and a negative result (unable to complete the task) as 0. For models trained only on gradient descent, ANT outperforms both the small and large ANNs significantly, demonstrating an ability to complete the game at remarkably low parameter counts for a quick training cycle and even better convergence for the large model. Given a genetic algorithm with SGD, ANT converges to a near-optimal solution uniformly for both the small and large model, significantly outpacing their ANN counterparts.

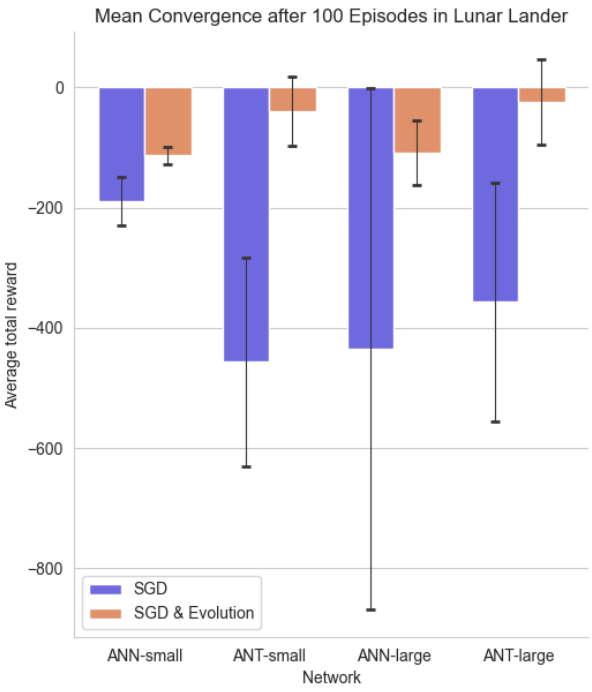

Fig. 8. ANN and ANT 100 episode convergence for Lunar Lander.

For a more challenging game like Lunar Lander with a 50% larger action and observation space on a more complex 2D game, we see that while ANT-small and each of the larger models fail to converge to an optimal solution (above 0) in 100 time steps, ANT again routinely approaches this reward using the combined genetic-gradient descent algorithm, with the 1σ error bar in for both the small and large model indicating that around 30-40% of the trained models achieved an optimal solution.

B. *CARL*

In the first CARL experiment, we find that ANT is resoundingly more robust in converging quickly, showing minimal performance drop-off even for the smaller 36 neuron model.

[9] Acrobot penalizes each non-successful step with a reward of -1, so for illustrative clarity, we report the reward as $200 +$ total reward, where each episode is 200 time steps, so as to avoid strictly negative rewards. We do not do this for LunarLander.

Conversely, Figure 9 shows that both ANNs converge significantly less reliably in the CARL environment, with the small ANN almost always unable to converge within 100 epochs even with a genetic algorithm. The 66-neuron ANT-large network has no significant difference in convergence between the base and CARL acrobot environments.

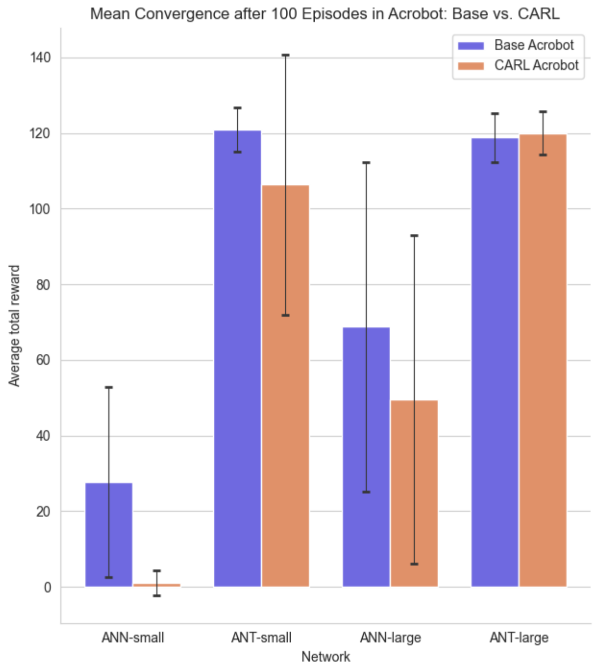

Fig. 9. ANN and ANT trained on static physics variables vs. randomized.

Figure 9 illustrates a broader characteristic of the ANT formulation. Because of a heightened complexity of the hypothesis class and dynamic internals, ANT is able to more reliably fit more complicated environments while being equally computationally performant compared to ANNs. This is further reinforced by the following Lunar Lander experiment, where we assess the quality of generalization outside of the training regime for each ANN and ANT model.

For the second Lunar Lander-based simulations, Figure 10 shows that while the ANN requires nearly 100 episodes to return to an optimal solution, the ANTs are able to maintain a near-optimal solution even within the new environment (reward only drops by 50, on average).

C. Robot Performance

Despite some successes in simulation, the translation to a physical agent failed due to the narrow observation space and discrepancies between the simulated and real-world environments.

For the physical agent, the ultrasonic sensor was prone to inconsistent distance measurements, and the computer vision algorithm frequently failed to detect the food or misidentified other objects as food. These inaccuracies led to suboptimal decision-making and performance failures. Our attempts to rectify these issues through a hard-coded algorithm to locate food were unsuccessful, with a success rate of 20%. The failure of a hard-coded algorithm underscores that the performance failures are, in large part, due to limitations of the physical agent itself rather than the underlying network. This highlights the need for an expanded observation space, which could include integrating multiple sensors to take distance measurements at various angles.

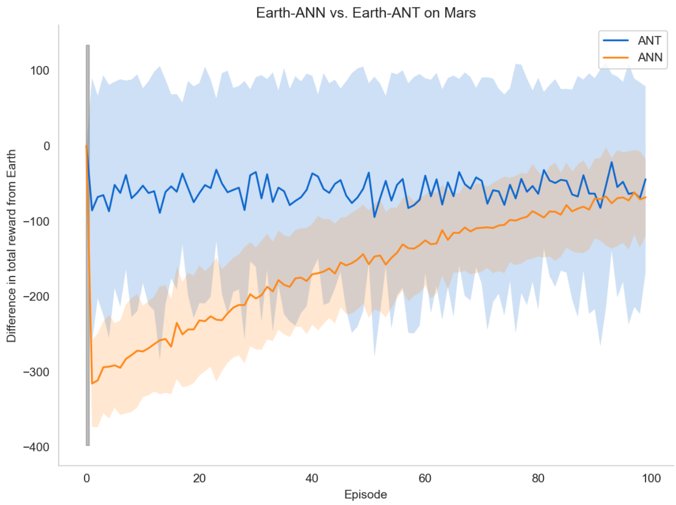

Fig. 10. ANN and ANT ability to adapt to gravity on Mars: 20 Earth-trained models.

Similarly, the ANT network was first trained in a controlled virtual environment, which does not adequately capture the complex dynamics of the physical world. Sensor inconsistencies, variations in motor functions, and dynamic environment elements were not modeled in the simulation and warrant a more intricate simulation environment. Incorporating Partially Observable Markov Decision Processes (POMDPs) could be beneficial in modeling the uncertainties and variabilities inherent in real-world interactions, leading to a more robust and adaptable agent.

D. Performance and Scalability

Due to the dynamic nature of ANT, the complexity of the update procedure runs linearly with respect to the number of neurons, while suboperations within neurons require a linear compute time with respect to the number of connections, leading to a generally quadratic overall compute time. However, ANT is highly parallelizable, with each neuron operation occurring independently of other neurons and with genetic iterations that also run independently of other trials.

ANT is currently written in Python, which has a Global Interpreter Lock (GIL) preventing process threading. We anticipate that a raw C/C++ implementation with threaded neuron operations would be even more computationally advantageous.

As opposed to the few large matrix compositions involved in ANN forward and backward propagation, most of the com-

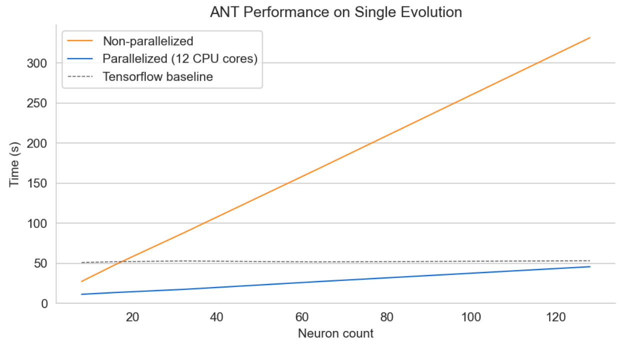

Fig. 11. Speed of ANT with connectivity $\frac{1}{\sqrt{n}}$ (widthwise-expanded ANN).

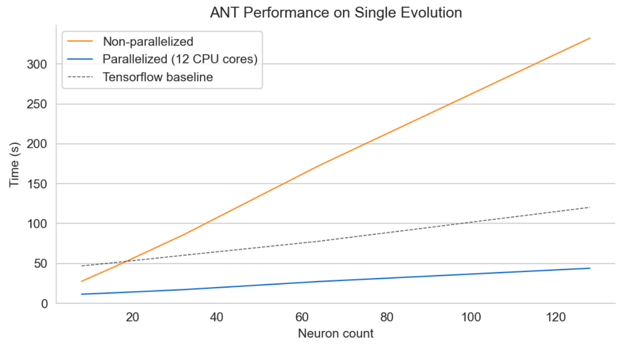

Fig. 12. Speed of ANT with connectivity $\frac{1}{\sqrt{n}}$ (depthwise-expanded ANN).

putations in ANT, while parallelizable, involve many relatively small matrix operations. This imposes a necessary bound on the compute speed of ANT relative to conventional networks–where ANNs can use inbuilt matrix composition optimizations common in today's Graphics Processing Units (GPUs), these improvements aren't reflected to nearly the same degree on smaller compositions.

However, ANT's computation is invariant of a notion of network "depth." Where ANNs require the sequential computation of outputs and gradient depthwise, ANT can compute all gradients in parallel. When shifting the baseline to instead make deeper instead of wider networks, reducing the optimizability of the ANN gradient calculation, ANT becomes a much more advantageous choice:

More experimentation with a more optimized code and better hardware is required to be conclusive on the speed of ANN compared to ANT, though it is likely that the computational tradeoff is problem-specific.

V. SUMMARY AND CONCLUSION

A. *Limitations*

There are several limitations of the ANT formulation which complicate its use in most machine learning contexts. Because it is a dynamic network, ANT is not known to be capable of making distinct input-output functional pairings as in the context of supervised learning. In fact, in all settings which involve sharp discontinuities, heavy discretization of input, or non-temporal input, ANT's formulation seems to preclude it from use.[10]

Further, not studied heavily in this paper is the sensitivity of ANT to initial conditions. As desired network complexity increases, the hyperparameter space grows exponentially as the total number of possible graph structures itself increases at a rate of 2^n where, as we have seen anecdotally, different graph constructions can yield significantly different convergence times. In most cases for particular RL settings, ANT would fail to converge completely given some graph initialization via stochastic gradient descent. This requires the use of a genetic algorithm that, though optimizable, is a much more brute-force approach toward searching for viable network structures and may not be computationally feasible for large parameter spaces.

B. *Future Work*

ANT is a novel neural network formulation which we believe shows significant promise for applications in RL.

The properties of ANT remain largely unstudied. Although we have investigated small-scale applications of ANT in RL studies in this paper, scaled RL applications and alternative machine learning applications are unknown. It is possible that scaling ANT is computationally difficult, however if the results at small scales indicate a similar convergence speeds for higher parameter counts then it would be worthwhile, especially if transfer learning were possible. Alternatively, like many dynamic networks, the convergence properties of ANT may be revealing in studying how biological neural networks may retain information, such as the memory-retention properties of Hopfield networks [24].

We also haven't investigated the graphs created by ANT's evolutionary algorithm. It is likely that they or their internal dynamics have emergent characteristics that could inform future development of ANT or other learning dynamic networks.

C. *Conclusion*

We have introduced the Artificial Neural Topology (ANT), a novel neural network formulation distinct from the conventional artificial neural networks which have dominated ML circles for decades. ANT's formulation allows for a unique combination of effective gradient-based and genetic optimization. We showed that not only can ANTs perform as well given equal conditions compared to ANNs in reinforcement learning tasks, but also that ANTs have more robust and generalizable convergence tendencies than ANNs. While most experimentation was done on a small scale, we illustrate properties that are fundamental to the network and suggest potential for broader reasoning capabilities in applications of larger networks, suggesting a new way forward for creating generally intelligent agents.

---

[10] It is possible that supervised tasks would work if presented in an RL-like setting.

ANT is aimed as a critical unifying subcomponent of the broader mission toward unifying divergent branches of machine learning research. ANT is designed as an artificial analog to the prefrontal cortex, where in each environment we give it automatic perfect control over senses and motor movement. In an embodied setting, this would translate to ANT serving as a central processor for peripheral task-specific models (computer vision, robotics models, etc.) conjoined through a latent space, analogous to how biological organisms are structured.

While the pursuit of generalized reasoning is an important goal in and of itself, which we believe is achieved by ANT in part with a pseudo-cognitive approach by characterizing qualities of biological neural networks, it is equally important to consider how an embodied approach and improvements to the reinforcement learning mechanisms are critical to making agents that can exist and act in the real world.

REFERENCES

[1] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).

[2] Ngiam, Jiquan, et al. ”Multimodal deep learning.” Proceedings of the 28th international conference on machine learning (ICML-11). 2011.

[3] Taloni, Andrea, et al. ”Comparative performance of humans versus GPT-4.0 and GPT-3.5 in the self-assessment program of American Academy of Ophthalmology.” Scientific Reports 13.1 (2023): 18562.

[4] Sonko, Sedat, et al. ”A critical review towards artificial general intelligence: Challenges, ethical considerations, and the path forward.” World Journal of Advanced Research and Reviews 21.3 (2024): 1262-1268.

[5] Li, Yingbo, and Yucong Duan. ”The Wisdom of Artificial General Intelligence: Experiments with GPT-4 for DIKWP.” arXiv preprint (2023).

[6] Goertzel, Ben. ”Artificial General Intelligence: Concept, State of the Art, and Future Prospects” Journal of Artificial General Intelligence, vol.5, no.1, 2014, pp.1-48. https://doi.org/10.2478/jagi-2014-0001

[7] Azevedo FAC, Carvalho LRB, Grinberg LT, Farfel JM, Ferretti REL, Leite REP, et al. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *J Comp Neurol*. 2009 Apr;513(5):532-41. doi:10.1002/cne.21974. PMID:19226510. S2CID:5200449.

[8] Ananthanarayanan R, Esser SK, Simon HD, Modha DS. The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses. In: *Proc Conf High Perform Comput Netw Storage Anal - SC ’09*; 2009. p. 1-12. doi:10.1145/1654059.1654124. ISBN: 978-1-60558-744-8.

[9] Anthropic. Model Card for Claude 3. [Internet]. 2023. Available from: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf

[10] Mijwil Maad, Hiran Kamal, Doshi Ruchi, Dadhich Manish, Al-Mistarehi Abdel-Hameed, Bala Indu. (2023). ChatGPT and the Future of Academic Integrity in the Artificial Intelligence Era: A New Frontier. Al-Salam Journal for Engineering and Technology. 2. 116-127. 10.55145/ajest.2023.02.02.015.

[11] Herculano-Houzel S, Lent R. Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *J Neurosci*. 2005 Mar;25(10):2518-21. doi:10.1523/jneurosci.4526-04.2005. PMC 6725175. PMID: 15758160.

[12] Herculano-Houzel S, Mota B, Lent R. Cellular scaling rules for rodent brains. *Proc Natl Acad Sci U S A*. 2006 Aug;103(32):12138-43. doi:10.1073/pnas.0604911103. PMC 1567708. PMID: 16880386.

[13] Menzel R, Giurfa M. Cognitive architecture of a mini-brain: the honeybee. *Trends Cogn Sci*. 2001 Feb;5(2):62-71. doi:10.1016/S1364-6613(00)01601-6. PMID: 11166636.

[14] Regehr WG. Short-term presynaptic plasticity. Cold Spring Harb Perspect Biol. 2012 Jul 1;4(7):a005702. doi: 10.1101/cshperspect.a005702. PMID: 22751149; PMCID: PMC3385958.

[15] Bassett, D. S., & Bullmore, E. (2006). Small-world brain networks. The Neuroscientist, 12(6), 512-523.

[16] Varshney, Lav & Chen, Beth & Paniagua, Eric & Hall, David & Chklovskii, Dmitri. (2011). Structural Properties of the Caenorhabditis elegans Neuronal Network. PLoS computational biology. 7. e1001066. 10.1371/journal.pcbi.1001066.

[17] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2020, July 16). Hopfield Networks is All You Need. arXiv.org. https://arxiv.org/abs/2008.02217

[18] Neftci, Emre O.; Mostafa, Hesham; Zenke, Friedemann (2019). ”Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks”. IEEE Signal Processing Magazine. 36 (6): 51–63. Bibcode:2019ISPM...36f..51N. doi:10.1109/msp.2019.2931595

[19] Geoffrey Hinton, ”Coursera Neural Networks for Machine Learning lecture 6”, 2018.

[20] Herculano-Houzel S, Lent R. Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *J Neurosci*. 2005 Mar;25(10):2518-21. doi:10.1523/jneurosci.4526-04.2005. PMC 6725175. PMID: 15758160.

[21] Herculano-Houzel S, Mota B, Lent R. Cellular scaling rules for rodent brains. *Proc Natl Acad Sci U S A*. 2006 Aug;103(32):12138-43. doi:10.1073/pnas.0604911103. PMC 1567708. PMID: 16880386.

[22] Benjamins, Carolin, et al. Contextualize Me - The Case for Context in Reinforcement Learning. 2023.

[23] Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 8, 229–256 (1992). https://doi.org/10.1007/BF00992696

[24] John J. Hopfield (2007) Hopfield network. Scholarpedia, 2(5):1977.

[25] Yamazaki K, Vo-Ho VK, Bulsara D, Le N. Spiking Neural Networks and Their Applications: A Review. Brain Sci. 2022 Jun 30;12(7):863. doi: 10.3390/brainsci12070863. PMID: 35884670; PMCID: PMC9313413.

[26] Polsky, A., Mel, B. W. & Schiller, J. Computational subunits in thin dendrites of pyramidal cells. Nature Neuroscience 7, 621–627 (2004).

[27] Tran-Van-Minh, A. et al. Contribution of sublinear and supralinear dendritic integration to neuronal computations. Frontiers in Cellular Neuroscience 9, 67–67 (2015).

[28] Gidon, A. et al. Dendritic action potentials and computation in human layer 2/3 cortical neurons. Science 367, 83–87 (2020).

[29] London, M. & Häusser, M. DENDRITIC COMPUTATION. Annu. Rev. Neurosci. 28, 503–532 (2005)

[30] Mijwil MM, Kamal H, Doshi R, Dadhich M, Al-Mistarehi AH, Bala I. ChatGPT and the Future of Academic Integrity in the Artificial Intelligence Era: A New Frontier. *Al-Salam J Eng Technol*. 2023;2:116-27. doi:10.55145/ajest.2023.02.02.015.

[31] Silver, David, et al. Reward is enough. Artificial Intelligence, vol. 299, Oct. 2021, p. 103535, https://doi.org/10.1016/j.artint.2021.103535.