Graph Continuous Thought Machines

1st Tofara Moyo
Bulawayo , Zimbabwe
tofaramoyo@gmail.com, Mazusa AI

***Abstract*—We propose a novel Graph Continuous Thought Machine (Graph CTM) architecture that integrates a simulated prefrontal cortex to enable adaptive problem-solving and decision-making. The Graph CTM leverages graph neural networks to process complex data streams, while the simulated prefrontal cortex modulates node activity to selectively focus on relevant information. Through reinforcement learning, the model navigates graph space to converge on optimal solutions, guided by the information contained in learnt node property vectors. The simulated prefrontal cortex regulates the flow of information by adjusting the disposition of nodes to lead to the next instantiation of the graph network. The Graph CTM incorporates an attention mechanism that integrates the internal state of the graph as input, which is modulated by outputs from the model's neural synchronization matrix. This modulation enables the algorithm to selectively focus on specific subgraphs or node subsets,correlating them with the input, effectively emulating short-term and long-term memory mechanisms when attending to both the input and internal representation. By dynamically weighting the importance of different graph components, the model can adaptively process and retain relevant information, facilitating more accurate and context-dependent decision-making.**

I. Introduction

Continuous Thought Machines (CTMs) represent a cutting-edge advancement in artificial intelligence, designed to emulate human-like thought processes through uninterrupted, dynamic neural network operations. Unlike traditional AI systems that rely on discrete inputs and outputs, CTMs operate on a continuous spectrum, processing information in a fluid, real-time manner akin to human cognition.

The key characteristics of the CTM are, continuous processing: CTMs integrate inputs and generate outputs without interruption, enabling seamless interaction with dynamic environments. Adaptive Learning: These machines utilize an advanced neural network architecture that adapts and evolves based on continuous feedback loops, enhancing its predictive and decision-making capabilities.

The architecture of a traditional CTM comprises several key components that work in tandem to process and transform input data. The components are organized in a hierarchical structure, enabling the CTM to capture complex patterns and relationships in the data.

Encoder: The input vector is first transformed by an encoder, facilitating subsequent processing.

Attention Mechanism: The encoded input is then passed through an attention mechanism, which selectively weights the input features, highlighting the most relevant information. The output of the attention mechanism is subsequently fed into a synapse model .

Synapse Model: The synapse model, typically implemented using a U-Net neural algorithm, processes the output from the attention mechanism, generating a feature representation that captures both spatial and temporal dependencies in the data

Neuron-Level Models: The output from the synapse model is then fed into neuron-level models, which are complete neural networks with their own weights and biases. These models process the incoming data, leveraging both the current input and a stored history of previous signals.

Memory Mechanism: A critical aspect of the CTM design is the incorporation of a memory mechanism, which stores a history of incoming signals. This stored history is fed into the neuron-level models, along with the current input, enabling the CTM to capture temporal dependencies and contextual relationships in the data. The combined input is then processed to produce an output, denoted as z. This vector of z's is also used as input to the synapse model together with the output of the encoder block at the next time step.

By integrating these components, traditional CTMs can effectively process complex data streams, capturing both spatial and temporal patterns, and generating meaningful outputs.

In addition to processing current inputs, the Continuous Thought Machine (CTM) also leverages historical output data to inform its decision-making processes. A history of outputs is maintained, which is utilized to calculate the Neural Synchronization Matrix (S). The S matrix is a critical component of the CTM, enabling the model to capture complex relationships between different outputs and identify patterns in the data.

The S matrix is calculated using the following equation:

$S = Z * Z^T$

where: Z is the matrix composed of the collection of z outputs, each representing a specific state or feature of the system.

Z^T is the transpose of the Z matrix.

The dot product of Z and Z^T yields the S matrix, which represents the similarity between different outputs and captures the synchronization patterns between them. This matrix can be used to identify clusters, correlations, and other relationships between outputs, enabling the CTM to make more informed predictions and decisions.

The Continuous Thought Machine (CTM) operates in a phased manner, producing outputs in discrete steps referred to as "ticks." Each tick represents a specific point in the processing pipeline, where the CTM generates an output based on the current input and its internal state. The CTM's loss function plays a crucial role in determining when to terminate the processing. Specifically, the loss function takes into account the confidence of the output at each tick, evaluating the model's certainty in its predictions. This confidence-based approach enables the CTM to adaptively determine when to stop processing, rather than relying on a fixed number of ticks or iterations.

Following the calculation of the Neural Synchronization Matrix (S), the Continuous Thought Machine (CTM) employs a randomized selection process to generate two vectors from the S matrix entries. These vectors are then utilized in the final stages of output generation.

The CTM randomly selects entries from the S matrix to form two vectors:

Vector 1: This vector is fed into the attention block, which consists of the output from the input encoder. The attention mechanism selectively weights the input features, leveraging the information captured in the S matrix to focus on the most relevant aspects of the input data.

Vector 2: This vector is fed into a decoder, which generates the final output of the CTM. The decoder processes the vector, incorporating the contextual information and relationships captured in the S matrix, to produce a meaningful and accurate output.

II. Graph CTM

This paper introduces a significant enhancement to the Continuous Thought Machine (CTM) framework, which we term the Graph CTM. The proposed update involves replacing the traditional Synapse model and neuron-level models with a Graph Convolutional Network (GCN) integrated with an attention mechanism.

The Graph CTM architecture leverages the strengths of GCNs in processing complex graph-structured data, enabling the model to capture nuanced relationships and dependencies between different nodes or entities. By incorporating an attention mechanism, the Graph CTM can selectively focus on the most relevant nodes and edges in the graph, further enhancing its ability to extract meaningful insights.

The Graph CTM consists of the following key components:

Graph Convolutional Network (GCN): The GCN replaces the traditional Synapse model and neuron-level models, enabling the Graph CTM to process graph-structured data and capture complex relationships between nodes.

Attention Mechanism: The attention mechanism is integrated with the GCN, allowing the model to selectively focus on the most relevant nodes and edges in the graph.

The proposed Graph Continuous Thought Machine (CTM) architecture features a hierarchical structure, where the output of each Graph Convolutional Network (GCN) layer is another graph, generated at each processing step or "tick."

The overarching picture can be conceptualized as there being a three-dimensional tensor block of potential neurons, or a dispositional neural model,where each presentation of the graph represents an instantiation of only those neurons that are currently active, denoted by the nodes of the graph.Where the nodes in the current graph over lay on the dispositional neural model, whith the nodes of the graph identifying with the indices of the dispositional neural model tensor.

Each node in the graph is represented by a learnable property vector, which encodes the knowledge and experiences accumulated by that node. This property vector captures two essential aspects:

Learned representations: The node's property vector represents what the neuron has learned from previous experiences, encoding patterns and relationships in the data.

Disposition to cause subsequent node activations: The node's property vector also influences the activation of nodes in the next graph , representing the connections and relationships between nodes across different layers within the dispositional neural model.

The Graph CTM's hierarchical architecture enables dynamic node instantiation, where only the neurons (nodes) in the dispositional neural model that are currently firing are instantiated at each processing step.

The Graph Convolutional Network (GCN) is initialized with a random instantiation of neurons at the commencement of each algorithmic run. This random initialization allows the model to explore different regions of the graph space, enabling the discovery of novel solutions.To optimize the GCN's performance, a reinforcement learning framework is employed. The model navigates through the graph space, searching for a path that best represents the solution to the problem at hand. Through trial and error, the model adapts its traversal strategy, learning to identify the most promising paths and converging on an optimal solution.

In the proposed Graph Continuous Thought Machine (CTM) architecture, each node in every graph is associated with a value z, distinct from its property vector, representing the node's output at a given processing step or ”tick.” A comprehensive history of these outputs is maintained for each node across the entire 3D tensor of dispositional neurons.The history of node outputs is utilized to construct the Z matrix, which is subsequently used to compute the S neural synchronization matrix.

As in the traditional CTM, following the computation of the S matrix, entries are randomly selected to form two vectors. One vector is fed into the input of the Graph Convolutional Network (GCN) with attention, while the other vector is fed into a decoder to produce the final output. This process enables the model to leverage the complex relationships and patterns captured in the S matrix, generating accurate and meaningful outputs.

A key innovation in the updated Continuous Thought Machine (CTM) is the incorporation of a prefrontal cortex-inspired mechanism within the dispositional neural network. This mechanism is designed to determine when a solution has been reached, effectively serving as a stopping criterion for the neurons in this network.

The stopping criterion is based on two complementary approaches:

Confidence-based stopping: The algorithm terminates when the confidence level reaches its maximum value, indicating that the solution has been identified with high certainty.

Reward-based stopping: In reinforcement learning regimens, the agent's expected reward is used as a stopping criterion. The algorithm terminates when the agent reaches a state with the highest expected reward, signifying that the optimal solution has been attained.

To do this, we leveraged the learnable property vectors of the nodes in this network. Specifically, we mapped the dimensions of these property vectors to the numbers in the cyclic group Z12, a mathematical structure commonly used in music theory.

The learning process for these property vectors was constrained to produce what would be meaningful musical chords, represented as multiclass vectors were they in fact keys indices on a musical keyboard. This can be achieved by incorporating a plugin that computes the perceptual consonance of these property vectors, quantifying the degree to which the resulting chords are musically coherent. The consonance value would then be included in the loss function, guiding the learning process towards producing property vectors that correspond to musically meaningful chords.

We needed to learn two things for this to work. Firstly, each of the property vectors of the nodes in the dispositional neural network should be associated with a meaningful musical chord. Secondly Once the algorithm was very confident of the solution or had reached a state with the highest expected reward, the property vectors of all of the currently firing nodes should harmonize.

III. Analysis

A. Self Design of Neural Pathways

In the traditional Continuous Thought Machine (CTM), the encoder generates an encoding that is subsequently processed by an attention block. This attention block operates on the encoding in conjunction with a vector comprising random entries from the neural synchronization matrix. The integration of this random vector enables the algorithm to modulate the attention mechanism, allowing it to adaptively focus on specific aspects of the encoder's output that are most relevant to solving the task at hand.

The modulation of attention facilitated by the neural synchronization matrix allows the model to selectively concentrate on different parts of the input data. In the context of image classification, for example, this might involve scanning through relevant regions of the image to gather information necessary for increasing confidence in the correct class label.

In the Graph Continuous Thought Machine (Graph CTM), the modulation of attention is extended beyond the traditional encoder-attention framework to also encompass the internal state of the dispositional neural model. This means that as the model learns, it can selectively focus on specific areas of the dispositional network, effectively building a model of memory.

The internal attention mechanism in the Graph CTM allows the model to attend to specific nodes or regions of the dispositional neural network, which can be thought of as a form of ”internal spotlight” that shines on the most relevant information. This internal attention mechanism is modulated by the neural synchronization matrix, which enables the model to selectively concentrate on specific aspects of its internal state.

B. Prefrontal cortex

A key innovation in the proposed algorithm is the incorporation of a simulated prefrontal cortex within the dispositional neural model. This novel component plays a crucial role in regulating the algorithm's traversal through graph space, enabling more efficient and effective exploration of the solution space.Through the use of reinforcement learning, the path traced out by the algorithm in graph space is guided by the information contained in the property vectors of the nodes in the simulated prefrontal cortex. These property vectors capture relevant patterns and relationships in the data, allowing the algorithm to adaptively navigate the graph and converge on optimal solutions.

Solution states would correlate with high levels of harmony within the prefrontal cortex nodes of the dispositional neural network. The simulated prefrontal cortex in the graph neural model is designed to mimic the functionality of the prefrontal cortex in the human brain, where it regulates the activity of other neurons through a gating mechanism. In the graph neural model, this functionality is achieved through the modulation of node activity, where the prefrontal cortex influences which nodes in the dispositional network fire,leading to the next graph, effectively controlling the flow of information.

IV. Conclusions

In conclusion, the proposed Graph Continuous Thought Machine with a simulated prefrontal cortex offers a novel and promising approach to adaptive problem-solving and decision-making. By integrating graph neural networks with a simulated prefrontal cortex and attention modulation, the model can effectively process complex data streams, selectively focus on relevant information, and adaptively navigate graph space to converge on optimal solutions. The incorporation of a simulated prefrontal cortex and neural synchronization matrix enables the model to emulate short-term and long-term memory mechanisms, facilitating more accurate and context-dependent decision-making. Further research and development of this model may lead to significant breakthroughs in creating more sophisticated and human-like artificial intelligence systems.

References

[1] Continuous Thought Machines Luke Darlow, Ciaran Regan, Sebastian Risi, Jeffrey Seely, Llion Jones

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in International conference on machine learning. PMLR, 2015, pp. 1889–1897.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[5] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in International conference on machine learning. PMLR, 2016, pp. 1928–1937.

[6] Symbol Based Self Guided Neural Architecture Search in a Spiking Neural Network Authors: Tofara Moyo https://www.vixra.rxiv.org/abs/2410.0160

[7] Discovery of Novel STEM Documents Authors: Tofara Moyo https://rxiv.org/abs/2411.0124