A Case for Novel Networks April 21, 2026
(And how I achieved 100% accuracy)

Abstract - The origins of artificial intelligence (AI) can be traced back to the early stages of World War II; however, its formal inception is marked by the pivotal work of Alan Turing in 1956. Today, AI encompasses a diverse array of solutions, theories, applications, and codebases. One of the foundational codebases that is extensively utilized in Large Language Models (LLMs) is Neural Networks (NN). These networks exist in various forms, typically necessitating an initial structural setup and a training phase. Following training, the models become static and are unable to adapt to new data. In this context, I wish to introduce a novel class of neural networks, referred to as CINS, which represents a significant advancement in functionality, utility, and efficiency.

Introduction

ChatGPT's energy use estimates per query differ greatly. OpenAI says it uses 0.34 watt-hours, while some researchers claim it could be over 20 watt-hours for more complex queries. With 700 million users each week making more than 2.5 billion queries daily, ChatGPT's energy consumption could reach 850 megawatt-hours every day. That's enough to power thousands of electric vehicles.

This usage amounts to nearly 1 trillion queries each year, and the number may rise even higher by 2027, to 8.5 billion daily requests. Overall, ChatGPT's yearly energy consumption is similar to that of 29,000 U.S. homes.

Generative AI use is even larger, as companies rely on OpenAI's models and competitors like Google's Gemini are coming into play. A report predicts that generative AI consumed 15 terawatt-hours (TWh) in 2025 and will jump to 347 TWh by 2030, requiring a lot of new energy resources. This might mean building many large data centers.

Most of the expected increase in energy use will come from AI inference, not training. This rise is driven by the development of AI agents that operate on their own. Some estimates show that the number of daily queries could reach 329 billion by 2030, assuming the global population hits 8.6 billion [1].

Initiatives aimed at enhancing the efficiency of AI models are currently in progress, with organizations concentrating on optimizing their hardware, refining algorithms, and exploring renewable energy options to mitigate their carbon emissions. Progress in developing energy-efficient chips, enhancing cooling methods for data centers, and optimizing software could lead to a decrease in energy consumption over time [2].

The paper by Zhou, Ning, Hong et al. discusses the significant attention that Large Language Models (LLMs) have garnered due to their impressive performance, while also addressing the challenges posed by their high computational and memory requirements for deployment in resource-limited settings. It provides a comprehensive survey of existing literature focused on improving LLM inference efficiency. The authors identify key issues, such as the large model size, quadratic-complexity attention operations, and auto-regressive decoding. They introduce a taxonomy categorizing optimization techniques into data-level, model-level, and system-level strategies. Additionally, the paper includes comparative experiments on notable methods and offers insights into future research directions in the field.

Zhou, Ning, Hong, and others demonstrate that large language models (LLMs) generally require greater computational expenses, memory access costs, and memory utilization during their inference, which negatively impacts efficiency metrics in resource-limited environments. These metrics encompass latency, throughput, power usage, and storage [3].

I would like to present Complex Interconnecting Neural Structures, or CINS. Unlike traditional neural network architectures, CINS is designed to autonomously develop a distinctive structure based on the input data it receives. As inputs are processed and the learning process is initiated, the algorithms generate new structures that build upon existing ones, allowing for adaptation to the desired outcomes or outputs. Furthermore, CINS organizes data inputs into clusters, providing a systematic classification of stimuli, akin to the differentiation of visual and auditory information in the human brain.

The advantages of this innovative neural network architecture are primarily related to enhanced performance, reduced engineering requirements, and extended lifecycle management. Recent testing has demonstrated that CINS performs comparably to classic neural network architectures under baseline conditions. However, when utilizing its advanced structural and learning capabilities, it achieved a perfect score in performance evaluations. Additionally, CINS retains the ability to assimilate new knowledge without compromising existing information, thereby presenting novel paradigms for the development and lifecycle cost management of projects.

Preliminaries

2.1 Neural Networks

Artificial neural networks (ANNs) represent a groundbreaking achievement in machine learning, allowing us to tackle a myriad of tasks with some ease. Inspired by the intricate workings of biological neural circuitry, these models are at the frontier of innovation. Although their initial implementation by psychologist Frank Rosenblatt, in the form of the perceptron, faced periods of limited exploration—deemed an "AI winter" by the AAAI during the 1970s and 1980s—their potential has always been profound.

The revival of interest in ANNs was fueled by technological advancements and the development of the backpropagation algorithm, alongside the emergence of recurrent and convolutional neural networks. The 2010s heralded the arrival of AlexNet, a deep neural network that transcended existing image recognition models, igniting what we now know as the AI spring and fostering a renewed passion for deep learning. In 2017, the introduction of the transformer architecture revolutionized our understanding of linguistic complexities, positioning itself as the cornerstone for large language models like GPT-4. Furthermore, diffusion models, first introduced in 2015, laid the groundwork for transformative image generation models such as DALL-E in the 2020s, demonstrating the limitless possibilities of our quest for knowledge and creativity [4].

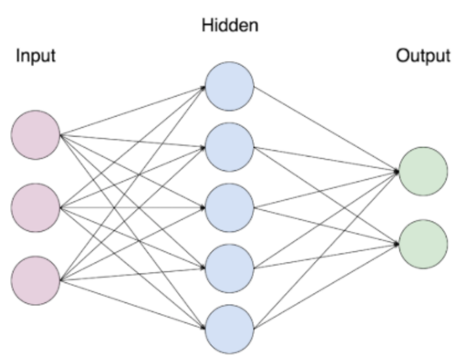

Figure 1 - Simple Neural Network

Latest neural network architectures focus on enhanced efficiency, interpretability, and long-range dependency modeling.

Kolmogorov-Arnold Networks features learnable activation functions instead of weights (fixed at 1). Drawing inspiration from the Kolmogorov-Arnold representation theorem, Kolmogorov-Arnold Networks (KANs) are appealing substitutes for Multi-Layer Perceptrons (MLPs). Unlike MLPs, which utilize fixed activation functions for their nodes ("neurons"), KANs incorporate learnable activation functions for their edges ("weights"). KANs completely eliminate linear weights, with each weight parameter being substituted by a univariate function defined as a spline. These showed that this seemingly minor modification allows KANs to surpass MLPs in both accuracy and interpretability in small-scale AI + Science tasks. When it comes to accuracy, smaller KANs can achieve performance that is on par with or better than that of larger MLPs in function approximation tasks. Both theoretically and through empirical evidence, KANs exhibit more rapid neural scaling laws compared to MLPs. Regarding interpretability, KANs can be easily visualized and engaged with by human users.

Through two instances in mathematics and physics, KANs are illustrated as valuable “collaborators” aiding scientists in (re)discovering mathematical and physical principles. In conclusion, KANs present a noteworthy alternative to MLPs. Despite the slower training process associated with KANs, their enhanced accuracy and interpretability indicate a promising potential to advance current deep learning frameworks that predominantly utilize MLPs. Further investigation is needed to enhance the training efficiency of KANs [5].

Graph Neural Networks Non-Euclidean data structures are mathematical frameworks used in machine learning to model complex, irregular, or hierarchical data that does not fit into flat grid-like structures. Numerous learning tasks involve handling graph data, which offers intricate relational insights among its components. For instance, modeling physical phenomena, identifying molecular fingerprints, forecasting protein interactions, and classifying diseases all necessitate models that can learn from graph-based inputs. In other areas, like processing textual and visual data, analyzing the structures derived from these inputs is an important research focus that also requires graph reasoning models.

Graph neural networks (GNNs) are neural network architectures designed to capture relationships within graphs through the exchange of messages between nodes. In recent years, various GNN variants, including graph convolutional networks (GCN), graph attention networks (GAT), and graph recurrent networks (GRN), have achieved exceptional results across numerous deep learning applications [6].

2.2 Large Language Models

The architecture of typical Large Language Models (LLMs) is predominantly based on the Transformer framework, utilizing a "decoder-only" structure that is particularly effective for text generation, as exemplified by models such as GPT-4, LLaMA, and Claude. Contemporary LLMs can be characterized as extensive deep neural networks.

A Deep Neural Network (DNN) is a type of artificial neural network distinguished by multiple hidden layers situated between the input and output nodes. This configuration empowers the model to capture intricate, non-linear relationships and to learn hierarchical features from the data.

Modern LLMs incorporate multiple stacked Transformer Blocks, which comprise the following components:

- Multi-Head Self-Attention: This mechanism enables each token to attend to all other tokens within the sequence, thereby facilitating comprehension of context and dependencies.

- Feed-Forward Network (FFN): This component processes the outputs of attention independently for each token, which introduces non-linearity and enhances the model's ability to learn complex patterns.

- Layer Normalization and Residual Connections: These elements contribute to the stability of training by ensuring a consistent distribution of data throughout the deep model, allowing for improved learning efficiency in models designed to predict subsequent tokens in a sequence while processing extensive datasets through parallelized computation.

2.3 Energy Efficiency

In recent years, studies have shown that increasing the size of language models not only improves their ability to process language but also generates new capabilities to handle more complex tasks that extend beyond traditional NLP (Natural Language Processing) activities. These larger language models are known as large language models (LLMs), which consist of numerous components, including a Transformer block.

A standard Transformer architecture features multiple stacked Transformer blocks. Generally, a Transformer block is made up of a Multi-Head Self-Attention (MHSA) block, a Feed Forward Network (FFN), and a LayerNorm (LN) operation.

The fundamental idea behind the Transformer architecture is its self-attention mechanism, utilized in the MHSA block. Formally, if we represent the input features as X = [x1, x2, ..., xn], the MHSA block performs linear projections on these features to produce queries Q, keys K, and values V.

This self-attention mechanism enables the model to assess the significance of various parts of the input irrespective of their distance, thereby capturing long-range dependencies and intricate relationships within the input sentence. These mechanisms utilize a specific category of deep learning neural networks, but require 'quadratic computational complexities' and more energy as these develop in sophistication [7].

Deep learning neural network structures are well-organized, layered, and interconnected computational models aimed at examining complex data by learning hierarchical representations. They comprise input, hidden, and output layers, with several hidden layers defining their "depth." Prominent architectures include Convolutional Neural Networks (CNNs) for spatial data, Recurrent Neural Networks (RNNs) for sequential data, and Transformer models.

A New Foundational Approach

3.1 Organization

The concept of Complex Interconnecting Neural Structures (CINS) inherently encompasses a systematic organizational framework. The brain consists of various clusters of neurons, such as the hippocampal and temporal lobes. CINS establishes an organizational architecture within the input (stimulus) layer, which it employs to create clusters of middle-layer nodes utilizing a proprietary set of algorithms. The advantages of this organizational structure will be further explained in subsequent discussions regarding processing mechanisms, but it lays the foundation for novel learning methodologies that yield distinctly advantageous characteristics.

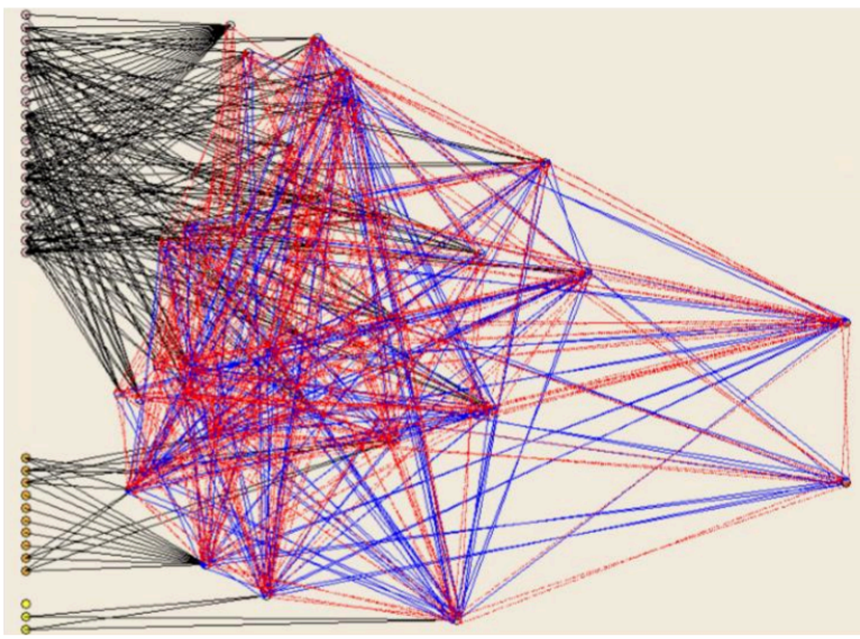

Figure 2 - A Complex Interconnected Neural Structure

In Figure 2 presented above, one can observe the distinct clusters or groupings of middle-layer nodes. Each of these nodes connects solely to a specific set of input stimuli. The inputs, located on the left, are categorized into three distinct groups, with a substantial input section at the top and two smaller clusters positioned at the bottom.

Moreover, CINS facilitates the establishment of connections between the middle-layer groupings and the output layer on the right. These connections signify associations that differ from a conventional neural network connection, which typically utilizes a single weight. As the network undergoes the learning process, these associations are cultivated and modified based on the incoming information. For instance, if a set of input stimuli from the upper grouping indicates a novel pattern or input set, the system generates a new node corresponding to the input stimuli and subsequently associates this new node with elements from both the middle and output layers.

3.2 Asynchronous operation

The aforementioned organizational structure introduces a novel processing requirement. Contrary to synchronous operation, wherein all nodes and connections are processed linearly from the input layer to the output layer, CINS functions asynchronously. As the values of stimuli fluctuate, this triggers a propagation mechanism whereby the node disseminates messages to its connected nodes. This propagation mechanism employs a sophisticated algorithm to manage the messaging while processing middle-layer nodes, thereby sending signals to the output layer as required.

The asynchronicity inherent in this architecture facilitates the processing of extensive networks in segments while preserving the integrity of the learned information. During the testing of the iris model, detailed below, the network processed only 11-12% of its entire structure during operations. This scenario demonstrates the remarkable potential of processing a fraction of an expansive network comprising 1 million inputs to achieve meaningful results.

3.3 Real-Time learning

While CINS can undergo pre-training and function in a static runtime configuration, its primary advantage lies in its capacity for real-time learning and processing. This capability provides significant benefits in data science engineering and reduces life-cycle costs for new projects, as information can be incorporated progressively as the concept and product evolve. Every new concept or product experiences varying stages of maturity; therefore, the integration of real-time learning and adaptation aligns more effectively with practical applications, permitting the assimilation of information as it becomes available or is uncovered.

3.4 Built-in integrity checks

It has been discussed [8] that a limitation of deep neural networks in terms of continual learning has significant practical consequences. Given its robust abilities in representation learning, deep learning has emerged as a key force behind many of the recent breakthroughs in artificial intelligence. Nevertheless, to achieve their high levels of performance, deep neural networks require extensive training on large datasets.

Furthermore, van de Ven, Kudithipudi discuss that once new relevant data becomes available after an extensive training period is completed, quickly updating the network by training solely on the new data is ineffective. To prevent overfitting to the new data, it is necessary to train the network using both the old and new data simultaneously.

Consequently, developing effective continuous learning techniques for deep learning could lead to considerable efficiency gains and a marked reduction in required resources. Another significant potential use of continual learning is correcting mistakes or biases. After training, deep neural networks often exhibit errors or retain biases, yet modifying a network to address these issues is challenging.

The chief issue here is that standard NN architectures connect every node to the preceding layer(s). The connection weights, established during initial training, cannot change without affecting the output of the model. CINS builds and adapts networks so that new information can be assimilated on top of existing ones without disturbing older information.

There are limitations to this, of course; new information may contradict older information, but the algorithms can handle this as well through various features and learning procedures. In the end, data integrity can be preserved if the information is important, thus offering a path toward built-in integrity checks for extended product lifecycles.

3.5 Reduced set processing

An additional feature of this novel architecture is that it does not require the entire network to be processed in order to derive accurate results. In contrast to traditional neural network structures, where a single weight can influence most, if not all, outputs, each weight in this framework has a specific meaning and is associated with only a subset of outputs. Consequently, a cluster of nodes may only necessitate a limited range of stimuli to produce the appropriate response.

In the Iris example presented below, a secondary tertiary set of responses was subsequently integrated to enhance certain information, thereby improving the overall results. This new cluster was trained on top of the existing network and could be processed independently of the prior cluster of information. Notably, the newly introduced cluster had a consequential impact on the outputs generated by the older cluster.

In essence, this new cluster represented only a fraction of the total network, approximately 25%. When processed separately, by merely updating the new input stimuli, only 25% of the network would be activated to generate responses. This phenomenon is significant in relation to both the integrity of learned knowledge and energy efficiency.

3.6 Increased Interactivity
Current agentic and large language models (LLMs) possess the capability to facilitate a degree of interaction by adjusting the context of conversations. Nevertheless, LLMs remain static, with context alterations limited to the token data streams utilized to generate responses. While these models can simulate real-life behaviors, their effectiveness is constrained to peripheral solutions that aim to retain feedback and adaptations.

In contrast to LLMs, CINS offers a unique advantage by enabling users to modify its parameter sets, thereby influencing system behaviors. These behaviors can evolve from initial training derived from an original dataset or through the guidance of specific trainers, and they can subsequently be adjusted by users of the system. It is imperative that appropriate protocols are established by stakeholders to ensure these modifications are conducted with integrity.

Furthermore, new behaviors do not necessarily need to emerge as adaptations of pre-existing ones; they can also manifest as entirely novel behaviors as the product matures, thus providing a pathway for ongoing development without the necessity for additional programming or debugging. While the implementation of robust protocols remains essential, stakeholders have the opportunity to leverage this flexibility to incorporate enhanced business processes, new procedures, and effective checks and balances.

3.7 Reduced energy requirements
The aforementioned advantages of this innovative technology, with the exception of the integrity checks, contribute to enhanced efficiencies by decreasing the energy requirements for the learning process. This reduction is compounded by each benefit, resulting in a significant decrease in energy consumption and overall life-cycle costs.

- Organizational Benefits
  The organization enhances data set engineering, improves training methodologies, and extends life-cycle costs as the project team and product mature. Furthermore, it emphasizes training focus areas, thereby minimizing the energy required to train the complete data set each time new information is presented.

- Asynchronous Operation
  Since the entire parameter set does not need to execute sequentially to yield an accurate response, the processing demands on the system are reduced. Although some overhead is associated with this approach, the advantages become more pronounced as the scale increases.

- Real-Time Learning
  When resources are directed towards specific subsets of training data to optimize system responses, this results in a clear advantage for pre-training and data science

initiatives within projects. Moreover, once deployed, the system is capable of assimilating new information effectively, resulting in substantial savings in time and kilowatt-hours during training.

- Reduced Set Processing
  The introduction of a processing framework that relies on organizational structure, signal transmission based on exceptions, and the ability to concentrate on subsets of data for learning markedly diminishes energy requirements. While this will inevitably impact the self-attention algorithms of large language models, the CINS algorithms are designed to retain features that allow navigation beyond the runtime engine, ensuring that the complete parameter set remains accessible.

- Increased Interactivity
  All products, whether software, process-oriented, or electro-mechanical systems, undergo phases of growth and development. With each iteration, the product matures. As artificial intelligence systems evolve, agents continuously capture user feedback and adapt their behaviors to enhance the product. CINS facilitates this at the neural network layer, providing more robust mechanisms for improved user interactivity.

Getting to 100%

This dataset contains 150 iris flower samples from three distinct species: Setosa, Versicolor, and Virginica. Each sample is characterized by four attributes: sepal length, sepal width, petal length, and petal width. It was introduced by British biologist and statistician Ronald Fisher in 1936 as a case study in discriminant analysis.

The Iris dataset is frequently utilized as an introductory dataset for grasping classification and clustering algorithms in machine learning. By analyzing the characteristics of the iris flowers, researchers and data scientists can classify every sample into one of the three species.

This dataset is particularly favored for its straightforward nature and the clear differentiation between the species based on the provided features. All four features are measured in centimeters.

- Sepal Length: This refers to the length of the sepals of the iris flower.
- Sepal Width: This indicates the width of the iris flower's sepals.
- Petal Length: This describes the length of the petals of the iris flower.
- Petal Width: This indicates the width of the petals of the iris flower.

The target variable signifies the iris flower’s species and includes three categories: Iris setosa, Iris versicolor, and Iris virginica.

- Iris setosa: Recognized for its relatively small size, with unique traits in its sepal and petal dimensions. A casual observer might suggest that the color of this flower has some darker blue tones along with some braided gold.
- Iris versicolor: Intermediate in size, with characteristics that lie between those of Iris setosa and Iris virginica. This flower also shows up with more reds and less blues.
- Iris virginica: Generally larger in size and shows significant variations in sepal and petal dimensions compared to the other two species. Some of these flowers have a more golden tint in their beard.

The Iris dataset can be employed in renowned machine learning frameworks like scikit-learn, TensorFlow, and PyTorch. These frameworks offer tools and libraries for constructing, training, and evaluating machine learning models using the dataset, enabling researchers to exploit the capabilities of these frameworks to test various algorithms and techniques for classification tasks.

Historical Significance of the Iris Dataset

The Iris dataset's historical importance stems from its foundational role in statistical analysis and machine learning. Ronald Fisher's efforts with the dataset laid the groundwork for the creation of numerous classification algorithms that remain in use today. The dataset has withstood the challenges of time and still serves as a benchmark for assessing new machine learning models [9].

Microsoft ML

In conducting a comparative analysis of the latest machine learning technologies and CINS, I utilized the Microsoft Visual Studio 2022 machine learning framework along with the Iris dataset. A notable advantage of the MS ML framework is its exceptional performance in terms of training speeds, which surpasses that of several Python-based systems.

Employing the Data Classification scenario, I loaded the Iris training dataset, which required approximately 8 to 10 seconds to complete the training process, utilizing between 25% and 75% of CPU resources. The model achieved a scoring accuracy ranging from 93% to 97% on the test dataset; however, it consistently encountered difficulties with records 22 and 27. Specifically, record number 27 was identified as an outlier, leading both models to erroneously categorize it as flower type #2 rather than the suggested flower type #3.

CINS using the iome shell.

A CINS model may be developed and configured using a desktop application known as iome. This application was designed for interfacing with networks and equipping models with an engine capable of learning in real-time while simultaneously processing live data. While iome is not optimized for rapid training, it is noteworthy that the CINS model, when applied to the Iris datasets, completed training in approximately three to four scans of the training set, totaling around two minutes, and achieved an accuracy rate of 93-97%. During this process, the CPU utilization remained below 5%. I am confident that employing a dedicated application solely for training purposes would result in significantly faster processing times while maintaining lower CPU usage.

It is important to clarify that the focus on training speed is not a fair comparison. The Microsoft Neural Network (MS NN) is likely to train and execute at a much faster rate than the CINS model, primarily due to the small size of the Iris dataset. The advantages of the CINS approach are expected to become apparent when applied to more substantial datasets. Furthermore, the writer would like to confirm that NNs are better for definite curve-fitting numerical processes and data sets. Their inherent mass connection scheme provides better "mathematical" outputs, yet the CINS show they can compete well.

As previously mentioned, one of the primary benefits of CINS technology is its capability to learn on top of established neural networks. Consequently, for both the challenging records in the training and testing sets, I developed an additional tertiary dataset, referred to as "Color Data," which was utilized for separate training following the initial training on the standard Iris dataset. The primary inquiry was whether this methodology could be implemented effectively[1].

The results were conclusive; the secondary data was successfully learned and processed for the specified records. All color data records were effectively integrated without compromising the integrity of the base trained set.

Record	Record Issues After Initial Base Training	Record Issues After Tertiary Training
Color Samples: 5	Marginal	Correct
Color Samples: 49, 53, 104, 109, 120	Incorrect	Correct
Test Sample: 24	Marginal	Correct
Test Samples: 22, 27	Incorrect	Correct

Figure 3 - Results of CINS Base and Tertiary datasets

The neural network representation is depicted below. The foundational cluster of nodes constitutes the base learned set, which utilizes the four input sizes. These nodes are interconnected through both positive and negative associations with the three outputs

[1]For this exercise, the 'Color Data' was based on casual observation using several sites like Google Images and [Kaggle IRIS EDA](https://www.kaggle.com/).

corresponding to the flower species; the output at the bottom represents flower species one, followed by flower species two above it, and so forth.

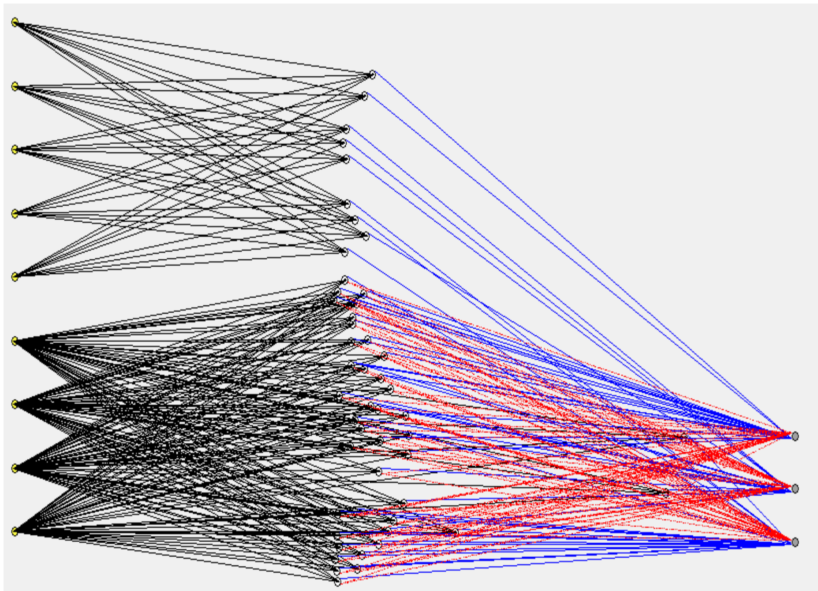

Figure 4 - Resulting CINS network of the Iris size and color data

The distinct grouping or cluster positioned above the foundational dataset represents the newly learned network derived from the color data. As this dataset is presented, it serves as a stimulus to the three existing outputs. Consequently, this color data could be emphasized to enhance the precision of the output.

It is important to note that no attempts were made to correlate erroneous records with an output. One potential approach could involve designating an output for "Wrong" or "Incorrect" and then associating a record number within the model. However, this scenario raises several logical questions regarding the underlying mechanisms that would guide the data collection of erroneous or incorrect records, warranting consideration for a future project.

This assessment does not fully encapsulate a comprehensive evaluation necessary to achieve a perfect score. However, it effectively highlights the advantage of the CINS system, particularly its capacity to acquire new information, which enhances existing knowledge and improves outcomes. Developing a system that evolves, adapts, and assimilates new information without disrupting entire parameter sets represents a significant advancement in this field.

It can be posited that the incorporation of a secondary dataset, such as the Color Data, may introduce potential biases or complications that could impact overall accuracy. Nevertheless, CINS enables the preservation of a robust set of integrity records, which allows for continuous testing that serves to prevent or alert us to such circumstances.

Moreover, emphasizing the model's capacity to update knowledge may inadvertently neglect the significance of sustaining a stable and reliable model characterized by established parameters. While models can certainly be locked down if desired, it is essential to recognize that if the product requires additional information for maturation, the updating feature is indeed justifiable.

Conclusion

Unlocking the Future with Complex Interconnecting Neural Structure (CINS)

The Complex Interconnecting Neural Structure (CINS) technology is set to revolutionize the landscape of continuous learning, user interactivity, and product longevity, all while dramatically slashing energy usage and pre-engineering costs. As we integrate cutting-edge automation and AI tools into our products, the ability of these innovations to adapt and learn new behaviors becomes not just desirable but essential.

As products evolve and product teams advance, continuous growth is natural—it's something we should all embrace! Science continually unveils fresh discoveries that inspire us to adapt and innovate, making CINS's advanced capabilities a game-changer in the industry.

The structural architecture of CINS is designed to harness unparalleled organization and connectivity, offering powerful advantages that far surpass existing technologies. We are just scratching the surface of what this remarkable technology can achieve! While this paper introduces the foundational architecture, functions, and benefits of CINS, it's the advanced features—like its ability to unlearn harmful behaviors, build confidence, and measure expectations against actual outcomes—that equip us with extraordinary tools for even more sophisticated automation. The future is bright with CINS—let's embrace the possibilities!

Feldhake Consulting LLC specializes in assisting organizations in the development of business plans and project case studies for new products, as well as addressing manufacturing challenges. By providing us with your data, we can collaborate to identify optimal configurations and methodologies to achieve outstanding results. Should you wish to learn more or request a demonstration of CINS, please reach out to our founder at mfeldhake@fconsult.llc.

Bibliography

CONSULTING ENGINEER with more than 30 years of experience, combining a strong understanding of new and emerging technologies such as AI/ML and advanced energy systems with product innovation, electrical and industrial engineering, and cross-functional leadership acumen to advance cutting-edge solutions in large-scale efforts for Fortune 500 companies such as MillerCoors, Anheuser Busch, Peabody, Owens Corning, Chrysler, and Boeing. Dynamic changemaker with a proven record of influencing stakeholders at all levels, communicating between various internal and external parties to bolster synergies on new product development and commercialization. Effective strategist able to connect R&D objectives with long-term business plans while consistently identifying avenues for additional revenue, cost savings, and process improvement. Executive leader establishing teams and cultures, delivering top-quality projects spanning all aspects from strategic planning to execution within budget and time constraints.

Special Note: Grammarly was used in the development of this article [10].

References

1. Smith MS. AI Energy Use: The Hidden Cost of ChatGPT Queries. IEEE Spectrum. Published October 2025. Accessed April 1, 2026. https://spectrum.ieee.org/ai-energy-use
2. How Much Energy Do Google Search and ChatGPT Use?. How Much Energy Do Google Search and ChatGPT Use? - RW Digital - Vancouver Digital Marketing Agency. RW Digital - Vancouver Digital Marketing Agency. Published October 31, 2024. Accessed April 1, 2026. https://www.rwdigital.ca/blog/how-much-energy-do-google-search-and-chatgpt-use/
3. Zhou Z, Ning X, Hong K, et al. A Survey on Efficient Inference for Large Language Models. arXiv.org. Published 2024. Accessed April 1, 2026. https://arxiv.org/abs/2404.14294 Pg. 1
4. History of artificial neural networks. Wikipedia.org. Published August 18, 2019. Accessed April 16, 2026. https://en.wikipedia.org/wiki/History_of_artificial_neural_networks
5. Liu Z, Wang Y, Vaidya S, et al. KAN: Kolmogorov–Arnold Networks. Openreview.net. Published 2025. Accessed April 16, 2026. https://openreview.net/forum?id=Ozo7qJ5vZi
6. Zhou J, Cui G, Hu S, et al. Graph Neural Networks: A Review of Methods and Applications. arXiv.org. Published 2018. Accessed April 16, 2026. https://arxiv.org/abs/1812.08434
7. Zhou Z, Ning X, Hong K, et al. A Survey on Efficient Inference for Large Language Models. arXiv.org. Published 2024. Accessed April 1, 2026. https://arxiv.org/abs/2404.14294 Section 2
8. van de Ven, Kudithipudi, Continual Learning and Catastrophic Forgetting. Arxiv.org. Published 2024. Accessed April 10, 2026. https://arxiv.org/html/2403.05175v1
9. GeeksforGeeks. Iris Dataset. GeeksforGeeks. Published February 28, 2024. Accessed April 10, 2026. https://www.geeksforgeeks.org/machine-learning/iris-dataset/
10. Grammarly. (2026). Grammarly (Version 2.0) [Large language model]. https://www.grammarly.com