

## HYPERNETWORKS LEARN THE BINARY TWO-SPIRAL CLASSIFICATION TASK

### LAS HIPERREDES APRENDEN LA TAREA DE CLASIFICACIÓN BINARIA DE DOBLE ESPIRAL

José L. Segovia-Juárez<sup>1</sup> 

<sup>1</sup>Universidad Tecnológica de los Andes-Abancay-Perú

#### Correspondencia:

José L. Segovia-Juárez  
jsegovij@utea.edu.pe

**Como citar este artículo:** Segovia, J. (2023). Hypernetworks learn the binary two-spiral classification task. *Hatun Yachay Wasi*, 2 (2), 9- 19. <https://doi.org/10.57107/hyw.v2i2.43>

#### ABSTRACT

This research paper explores the application of hypernetwork learning algorithm for machine learning using the binary double spiral classification task. This study experimentally demonstrates a learning rate of 100 % on the dataset with 121 points and up to 98.22 % on the binary double spiral dataset with 225 points. The findings provide insights into the benefits, usefulness, and potential of hypernetworks in solving complex nonlinear classification tasks. The hypernetwork model could be implemented in field programmable gate arrays (FPGAs). The hierarchical and distributed characteristics of hypernetwork models, coupled with the inherent parallelism of FPGAs, render a potent combination for high-speed response tasks. A unique aspect of this approach is the embodiment of biomimetic hierarchical organization principles akin to biological information processing, which proved to be highly effective in addressing complex computational tasks.

**Keywords:** two-spiral task, machine learning, hypernetwork architecture, molecular evolution algorithm

#### RESUMEN

Este artículo explora la aplicación de algoritmos de aprendizaje de *hypernetworks* en el campo del aprendizaje de máquina, utilizando como ejemplo la tarea de clasificar la doble espiral en el dominio binario. Este estudio demuestra experimentalmente una tasa de aprendizaje de hasta el 98,22 % con los datos de doble espiral binaria con 225 puntos, y del 100 % en el conjunto de datos con 121 puntos. Los hallazgos proporcionan una aproximación sobre los beneficios, la utilidad y el potencial de la arquitectura de *hypernetworks* en la resolución de tareas complejas de clasificación no lineal. El algoritmo evolutivo para se puede aplicar en matrices de compuertas programables en campo (FPGAs). Las características jerárquicas y distribuidas de los modelos de *hiperredes*, junto con el paralelismo inherente de las FPGAs, constituyen una combinación potente para tareas de alta velocidad de respuesta. Un



aspecto único de este enfoque es la incorporación de principios de organización jerárquica biomiméticos similares al procesamiento biológico de la información, que mostró ser altamente efectivos para abordar tareas computacionales complejas.

**Palabras clave:** tarea de doble espiral, aprendizaje de máquina, arquitectura *hypernetwork*, algoritmos evolutivos

## INTRODUCTION

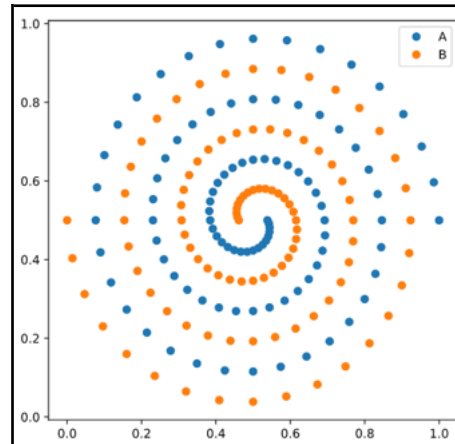
Research and development across various fields is motivated by demanding tasks and practical applications. In the domain of machine learning, these challenges are frequently embodied by curated datasets obtained through empirical means or artificial generation. Certain datasets have been extensively studied and have become established benchmark tasks (Cruz et al., 2022). One such benchmark task is the two-spiral classification task.

The classification of the Two-Spiral Task (TSP Task) has long posed a formidable challenge in neural networks and machine learning, is a well-known benchmark for binary classification. The data consist of points on two intertwined spirals that intersect at the origin, which cannot be linearly separated (Chalup & Wiklendt, 2007).

The original dataset for the two-spiral task (Lang & Witbrock, 1988) consisted of 194 data points evenly distributed across two intertwined spirals. These spirals were identical but rotated  $\pi$  radians relative to each other. The data points were represented in Cartesian coordinates  $(x, y)$  and plotted in a configuration where the spiral points formed radial rays, as shown in Figure 1, where one spiral is with orange dots and the other is with blue dots. The task focused on supervised learning of binary classifiers, where the aim was to determine to which spiral each input data point belonged, based on binary labeled targets.

**FIGURE 1**

*Double-spiral dataset plotted on a plane*



Note: The figure illustrates the double spiral dataset plotted on a plane. The objective of the TSP task is to accurately determine the affiliation of an input data point  $(x, y)$  to either the orange (B) or blue spiral (A).

The Double Spiral dataset presents a particularly difficult problem for machine learning algorithms due to the intricate nonlinearity and complexity entailed in accurately separating the two spirals. This is a highly nonlinear problem, solvable with backpropagation (Lang & Witbrock, 1988), fuzzy logic, and adaptive resonance theory neural networks (Carpenter et al., 1992), among others.

The TSP Task has also been used as a benchmark task for evaluating the performance of various machine learning algorithms, including neural networks (Lang & Witbrock, 1988; Singh, 2001). Yang & Kao, (2001) described the TSP Task as an extremely hard classification task and highlighted its use in the neural network community for testing and evaluating learning algorithms.

In addition to neural networks, other techniques have been explored for solving TSP. Ikuta et al. (2010) proposed a chaos glial network connected to a multilayer perceptron (MLP) for solving TSP. The chaos glial network was inspired by astrocytes, glial cells in the brain, and showed better performance compared to conventional MLPs.

In recent years, there has been growing interest in bioinspired approaches to robotics and engineering. Perez-Nieves et al. (2021) demonstrated the successful training of biologically realistic spiking neural networks to perform the TSP with high performance. The researchers investigated the effect of introducing heterogeneity in the time scales of neurons when performing tasks with realistic and complex temporal structures. They found that introducing heterogeneity improved the overall performance, stability, and robustness of the network. The study also suggested that the observed heterogeneity in the brain may play a vital role in its ability to adapt to new environments.

This paper presents our findings on the successful use of the hypernetwork learning algorithm for solving the binary double spiral task. Through our research, we establish the efficacy of hypernetworks in effectively learning the double spiral dataset and gain valuable insights into the performance of alternative machine learning algorithms. Our results have significant implications for the application of machine learning in handling complex datasets and applications beyond the double spiral task.

### **The hypernetwork learning architecture and the binary two-spiral dataset**

The binary two-spiral dataset

As mentioned earlier, the double spiral was initially defined within the Cartesian coordinate plane. However, to apply the hypernetwork architecture for learning, it is imperative that both input and output data reside within the binary domain.

Consequently, the dataset has undergone a redesign process to facilitate binary input/output, resulting in the creation of a binary double spiral, as depicted in Figure 2, with two spirals: one represented by 1 and the other by 0. The organism must learn to discriminate the spiral of “1s” from the spiral of “0s” in the double spiral data set. In this representation, the output values correspond to either 0 or 1, while the axes of the plot are labeled with binary numbers ranging from 1 to the size of the side. These binary numbers are then concatenated to form input vectors. For instance, a binary input vector of “0001 0001” has the desired output of 1, the “0010 1110” output 0, and so forth.

### **The hypernetwork learning architecture**

The hypernetwork architecture is a multilevel model designed for machine learning (Segovia-Juarez et al., 2019). It consists of three hierarchical levels: the molecular level, the cellular level, and the organism level.

- a. The Molecular Level: The atomistic detail of the model where molecules are enzyme-like structures. The interactions between these structures typically involve activation and inhibition processes. These molecules mutate and are then selected based on their performance in the given task
- b. The Cellular Level: At this level, molecular interactions form networks of interactions inside and between cells, driven by protein self-assembly. The synapse, or connections between cells, is dynamic and subjected to intracellular dynamics.
- c. Organismic Level: This is the highest level of organization, where information from cellular and molecular interactions is processed and used for output.

**FIGURE 2**

*The double spiral data set with 225 points. See text for details*

0001	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0010	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0011	1	0	1	1	1	1	1	1	1	1	1	1	0	1
0100	1	0	1	0	0	0	0	0	0	0	0	0	1	0
0101	1	0	1	0	1	1	1	1	1	1	1	0	1	0
0110	1	0	1	0	1	0	0	0	0	0	1	0	1	0
0111	1	0	1	0	1	0	1	1	1	0	1	0	1	0
1000	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1001	1	0	1	0	1	0	0	0	1	0	1	0	1	0
1010	1	0	1	0	1	1	1	1	1	0	1	0	1	0
1011	1	0	1	0	0	0	0	0	0	0	1	0	1	0
1100	1	0	1	1	1	1	1	1	1	1	1	0	1	0
1101	1	0	0	0	0	0	0	0	0	0	0	0	1	0
1110	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1111	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0

The hypernetwork architecture learns classification tasks through a process known as a variation-selection algorithm, which functions on the structure of the molecular subunits. This process begins with mutations in the molecular subunits to create variations, which are then evaluated against a defined fitness criterion. The best performing molecular subunits, observed from their ability to correctly classify inputs into their appropriate classes, are selected for further rounds of mutation and selection, creating an evolutionary learning cycle. This cycle continues until the hypernetwork can accurately classify all inputs or a termination condition such as a maximum number of iterations or epochs is met (Segovia-Juarez et al., 2019).

Learning in this architecture involves three hierarchical levels: organism level, cellular level, and molecular level. Understanding and adjustment occur at the molecular level and are reflected in the cell performance. This behavior is then scaled up to the overall behavior of the system when classifying the presented input.

For the training of the binary double spiral dataset, an organism with 11 cells was constructed. Four cells for the 8-bit input vector (each input cell receives two bits), one cell for the output (with a 0 or 1 output), and two layers of internal cells, with three cells in each layer. Figure 3 illustrates the topology of cell relationships within the hypernetwork organism. Upon request, the corresponding author can provide access to the software and datasets generated during this study.

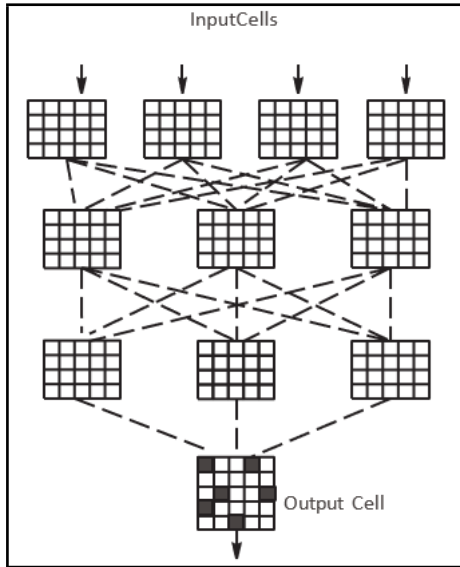
To show hypernetwork learning on the binary double spiral task, a pair of experiments were performed:

Learning the 225 vectors (array of 25 x 11), shown in Figure 2.

Learning the 121 vectors (array of 11 x 11), which is a subset of the previous data set, starts from the center of the plane, as shown in Fig. 7 D.

**FIGURE 3**

A hypernetwork with 4 input cells, two layers of internal cells, and an output layer with 4 cells. This organism was used to learn the double spiral data set.



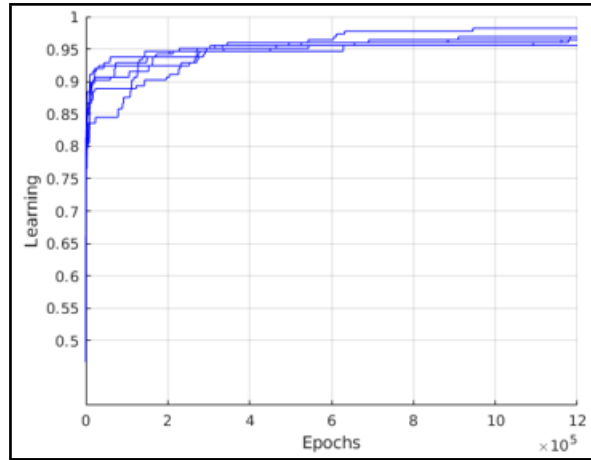
Finally, in order to gain insight into the generalization capabilities of the hypernetwork architecture, we conducted a test by evaluating the trained hypernetwork with 121 points to predict the outcomes with the double spiral with 225 points.

**RESULTS**

Learning the double spiral with 225 points  
 Results of six runs show an average of 96.74% correct classification (an average of 217.5 out of 225 vectors were classified correctly) in 1,200,000 epochs (see Table 1). The learning curves are shown in Figure 4. Each run takes about 60 min on an i7 PC.

**FIGURE 4**

Six hypernetworks learning the double spiral data set with 225 points.



**TABLE 1**

Results of six hypernetwork organisms for learning the double spiral data set with 225 vectors

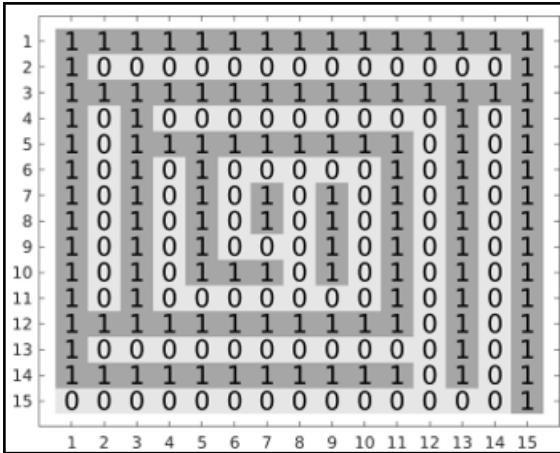
Exp. number	Correct % classification	N°. of correct vectors
1	96.89	218
2	98.22	220
3	96.44	217
4	96.44	217
5	95.56	215
6	96.89	218
Mean	96.74	217.50
95% CL	0.70	1.31

Figure 5 A shows the output obtained by one of the top-performing hypernetwork organisms after 1,200,000 epochs. This hypernetwork achieved a learning rate of 96.89 % for the 225 points. The number of errors encountered during the learning process was seven, as depicted in Fig. 5 B.

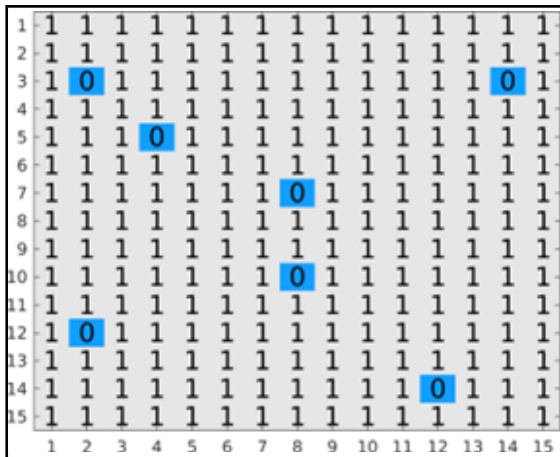
**FIGURE 5**

The final state of the output training the double spiral with 225 input vectors

- A. The output of hypernetwork learning at 1,200,000 epochs.



- B. Errors in the final output. Errors are shown with "0" and correct output with "1".

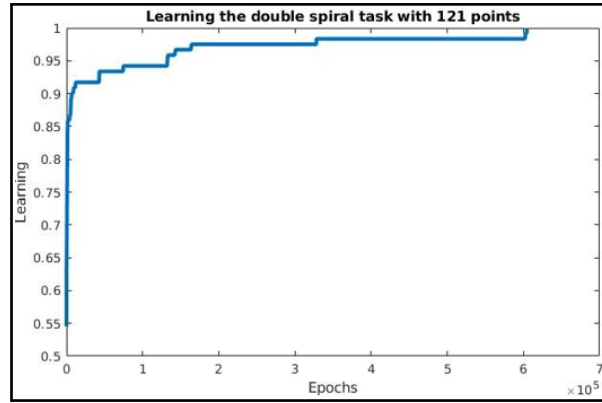


**Learning the double spiral with 121 points**

The hypernetwork shown in Figure 3 was trained with the double spiral data set with 121 points, achieving 100% learning at 604,482 epochs, as shown in Figure 6. At the beginning, it can be observed that the hypernetwork responds to all inputs with a "1", with an error rate of 55 %. As the learning progresses, the desired output is rapidly formed (Fig. 6), successfully learning the complete task at 604,482 epochs.

**FIGURE 6**

The learning process of the double spiral with 121 points resulted in achieving 100% accuracy

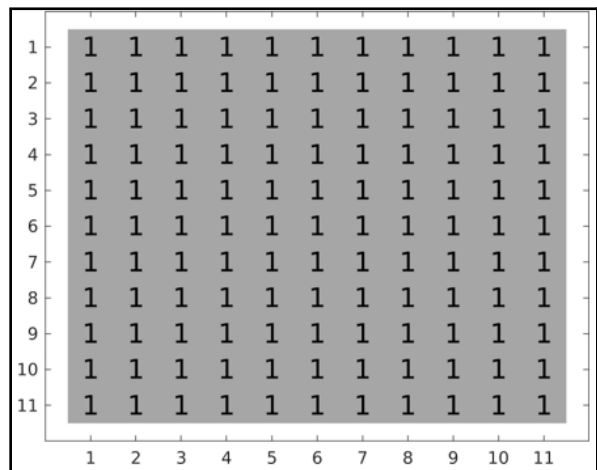


The results obtained while training the double spiral with 121 points are depicted in Figure 7. The figure illustrates the output at different stages of the training process: initially, after one epoch (Fig. 7 A), at epoch 2000 (Fig. 7 B), at epoch 300,000 (Fig. 7 C), and finally, upon reaching 100% learning with the desired output (Fig. 7 D).

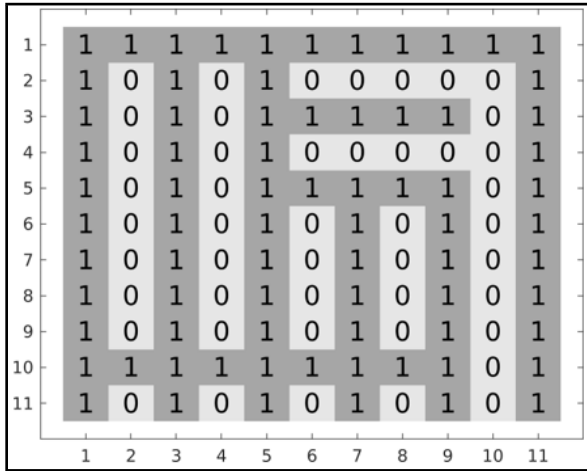
**FIGURE 7**

Learning the binary double spiral with 121 points

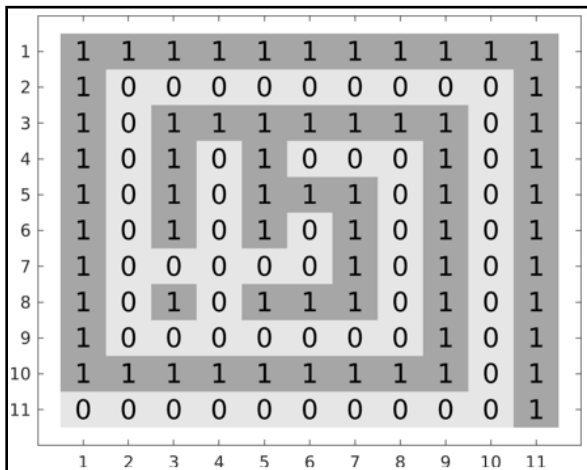
- A. No. Epochs = 1



B. No. Epochs = 2,000



C. No. Epochs = 300,000



D. No. Epochs = 604,482

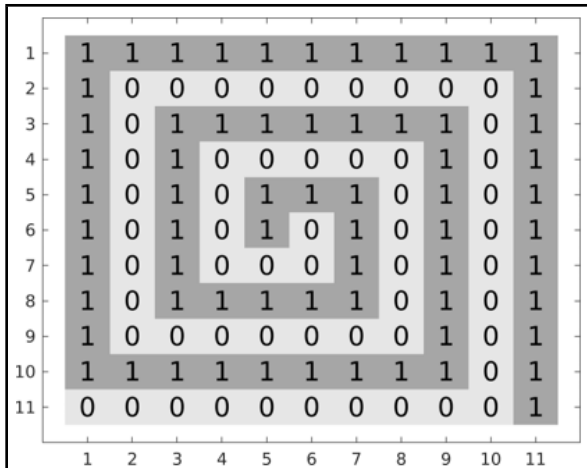
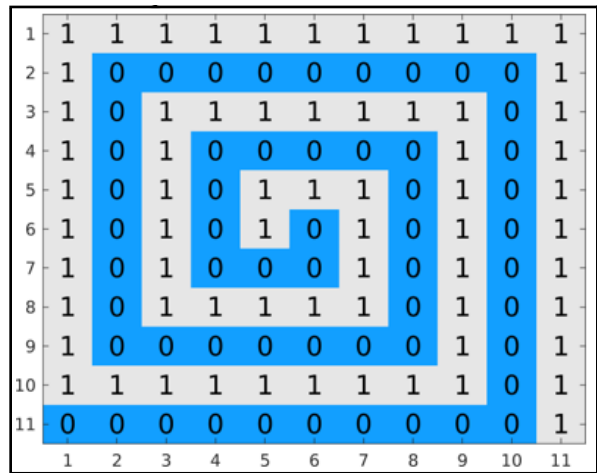


Figure 8 displays the error analysis of the hypernetwork organism during the learning phase of the double spiral with 121 vectors. Initially, at the beginning of the learning process, the error is observed in half of the bits within the array, as illustrated in Figure 8 A. After 2,000 epochs, there are 17 points within the array that still exhibit errors, as depicted in Figure 8 B. As the training progresses to 300,000 epochs, the number of points with errors reduces to 3, as shown in Figure 8 C. Finally, at the end of the learning process, all the points have been successfully learned, resulting in no errors, as shown in Figure 8 D.

**FIGURE 8**

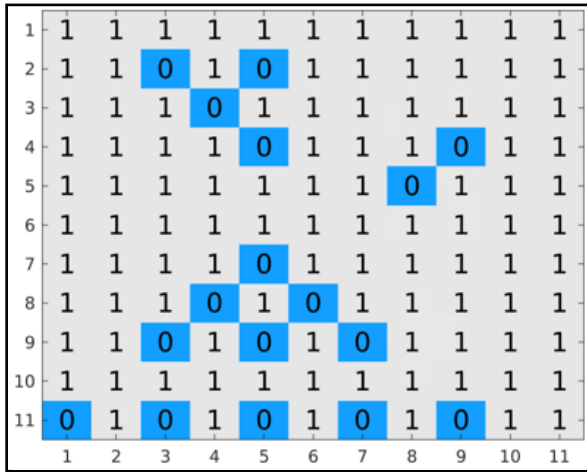
*Errors while learning the double spiral with 121 points. Digit "1" is correct, "0" is an error*

A. No. Epochs = 1

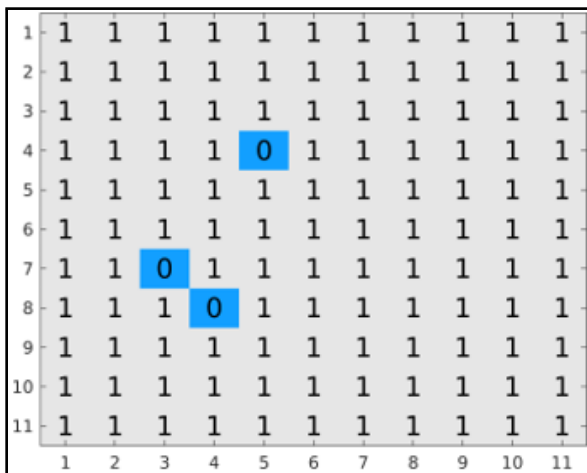




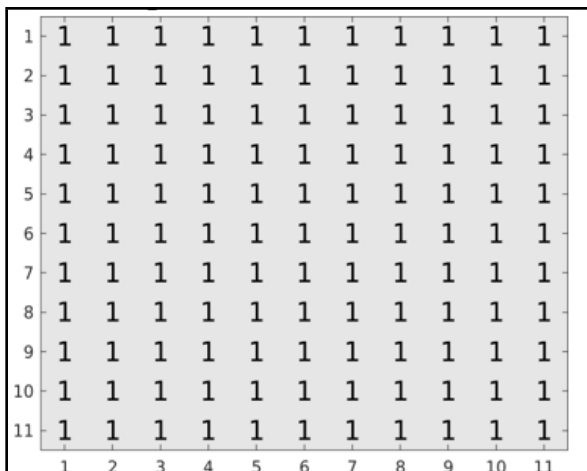
B. No. Epochs = 2,000



C. No. Epochs = 300,000



D. No. Epochs = 604,482



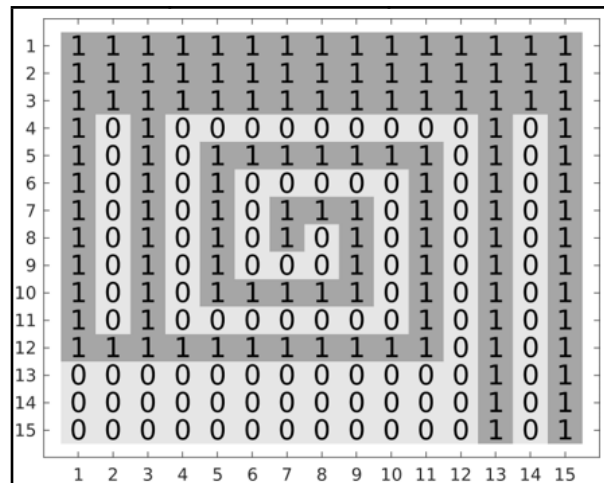
**Generalization with the double spiral task**

The hypernetwork that learned the double spiral of 121 points was fed with data from the double spiral of 225 points, which includes an additional 104 points (46 %) following the same spatial pattern. The output pattern is shown in Figure 9 A, while the errors are displayed in Figure 9 B. The errors correspond to 30 points deviating from the desired pattern. Out of the additional 104 points, the hypernetwork successfully responded correctly to 74 of them (71.15 %). This indicates that the hypernetwork demonstrates a degree of generalization in this challenging machine learning classification task.

**FIGURE 9**

Testing the hypernetwork trained with 121 points, with the double spiral trained with 225 points

A. Output with the 225 points





- B. Errors when testing an additional 104 points.  
Digit "1" is correct, "0" is an error

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				

**DISCUSSION AND CONCLUSIONS**

We have demonstrated that the hypernetwork learns a challenging task of differentiating the double spiral dataset in binary space. The hypernetwork architecture successfully learns the 121-point dataset with 100 % accuracy and achieves up to 98.22 % learning accuracy in the 225-point task. This architecture exhibits generalization capabilities, albeit limited by the high dimensionality of the problem, as observed in this challenging task. While neural networks can also solve this problem, hypernetworks offer the advantage of being implementable in the analog domain (Segovia et al., 2019). The hypernetwork architecture can be beneficial as a platform for analogous computing, given its biomimetic concept of hierarchical organization and principles of biological information processing involving molecular, cellular, and organismic levels. With the variation-selection learning algorithm acting on the molecular structure, the hypernetwork is capable of evolving new functionality and adapting to changes, analogous to biological systems.

The hypernetwork architecture can be potentially implemented with Field-Programmable Gate

Arrays (FPGAs). FPGAs are programmable logic devices that can be configured by the user after manufacturing to perform various operations, from simple logic gate operations to complex systems on chips or even artificial intelligence systems (Ruiz et al., 2019). FPGAs have gained popularity in various domains, including binary computing, due to their high flexibility, reconfigurability, and parallel computing capacity (Lysecky et al., 2004; Ruiz et al., 2019; Wang et al., 2015).

In the case of a hypernetwork, the FPGA could be programmed to match the distinct requirements of the hypernetwork’s molecular-based learning algorithm. This includes matching and thresholding functions that are vital in the learning process. FPGAs have the added advantage of evolvability just like the hypernetwork model itself. This means that their logic blocks can adapt according to the algorithm acting on the structure of the molecular subunits, hence resulting in a tightly integrated, evolving system that offers potentially fast response times due to the inherent parallelism of FPGA devices.

This compatibility between FPGAs and hypernetwork algorithms makes them a potentially feasible implementation of the hypernetwork architecture, possibly realizing the physical realization of evolvable hardware. FPGAs have been widely used in the field of evolvable computing, which involves the use of evolutionary algorithms to design and optimize hardware systems (Dobai & Sekanina, 2015; Shang et al., 2020; Yao & Higuchi, 1999). The reconfigurable nature of FPGAs makes them well suited for evolvable computing, as they can be dynamically reprogrammed to implement different hardware configurations and adapt to changing requirements.

FPGAs have also been used in evolvable hardware for applications such as neural networks and machine learning (Johnson et al., 2017; Whitley et

al., 2021). FPGAs can be used to implement artificial neural networks, allowing for the evolution of network topologies and parameters. Additionally, FPGAs have been used to evolve analog circuits and explore the dynamics of unlocked FPGAs (Whitley et al., 2021).

This paper presents an innovative technique for training and employing hypernetwork models in analogous computing, highlighting their use in field programmable gate arrays (FPGAs). The hierarchical and distributed characteristics of hypernetwork models, coupled with the inherent parallelism of FPGAs, render a potent combination for high-speed response tasks. A unique aspect of our approach is the embodiment of biomimetic hierarchical organization principles akin to biological information processing, which we demonstrate to be highly effective in addressing complex computational tasks. Through experimental outcomes, this research emphasizes the potency of hypernetworks and FPGAs in the advancement of machine learning and computing paradigms.

## REFERENCES

- Carpenter, G., Grossber, S., Markuzon, N., John, R., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5), 698–712. DOI:10.1109/72.159059
- Chalup, S., & Wiklendt, L. (2007). Variations of the two-spiral task. *Connection Science*, 19(2), 183–199. <https://doi.org/10.1080/09540090701398017>
- Cruz, A., Mayer, K., & Arantes, D. (2022). RosenPy: An Open-Source Python Framework for Complex-Valued Neural Networks. *SSRN*. <https://doi.org/http://dx.doi.org/10.2139/ssrn.4252610>
- Dobai, R., & Sekanina, L. (2015). Low-level flexible architecture with hybrid reconfiguration for evolvable hardware. *ACM Trans. Reconfigurable Technol. Syst.*, 8, 1–24. <https://doi.org/10.1145/2700414>
- Ikuta, C., Uwate, Y., & Nishio, Y. (2010, May 30 - June 2). *Chaos glial network connected to multi-layer perceptron for solving two-spiral problem*. [Conference]: International Symposium on Circuits and Systems (ISCAS). Paris, France <https://doi.org/10.1109/iscas.2010.5537060>
- Johnson, A., Liu, J., Millard, A., Karim, S., Tyrrell, A., Harkin, J., & Halliday, D. (2017-11-13 December). *Homeostatic fault tolerance in spiking neural networks utilizing dynamic partial reconfiguration of fpgas*. [Conference]. 2017 International Conference on Field Programmable Technology (ICFPT). Melbourne, VIC, Australia. <https://doi.org/10.1109/fpt.2017.8280139>
- Lang, K., & Witbrock, M. (1988). Learning to tell two spirals apart. *The 1988 Connectionist Models Summer School At: Pittsburgh, PA*, 52–59. <https://doi.org/10.13140/2.1.3459.2329>
- Lysecky, R., Stitt, G., & Vahid, F. (2004). Warp processors. *ACM Trans. Des. Autom. Electron. Syst.*, 11, 659–681. <https://doi.org/10.1145/1142980.1142986>
- Perez, N., Leung, V., Dragotti, P., & Goodman, D. (2021). Neural heterogeneity promotes robust learning. *Nature Communications*. <https://doi.org/10.1038/s41467-021-26022-3>
- Ruiz, J., Ramirez, G., & Khanna, R. (2019). Field Programmable Gate Array Applications—A Scientometric Review. *Computation*, 7, 63. <https://doi.org/10.3390/computation7040063>
- Segovia, J., Colombano, S., Flores, A., Hidalgo, D., & Mejía, M. (2019). Graph classification with

the hypernetwork, a molecule interaction based evolutionary architecture. *2019 IEEE International Conference on Big Data (Big Data)*, 5384–5393. <https://doi.org/10.1109/BigData47090.2019.9005979>

Shang, Q., Chen, L., Cui, J., & Lu, Y. (2020). Hardware evolution based on improved simulated annealing algorithm in cyclone v fpsocs. *IEEE Access*, *8*, 64770–64782. <https://doi.org/10.1109/access.2020.2984950>

Singh, S. (2001). Quantifying structural time varying changes in helical data. *Neural Computing and Applications*, *10*(2), 148–154. <https://doi.org/10.1007/s005210170006>

Wang, D., Zhou, E., Brake, J., Ruan, H., Jang, M., & Yang, C. (2015). Focusing through dynamic tissue with millisecond digital optical phase conjugation. *Optica*, *2*, 728. <https://doi.org/10.1364/optica.2.000728>

Whitley, D., Yoder, J., & Carpenter, N. (2021). Resurrecting fpga intrinsic analog evolvable hardware. *The 2021 Conference on Artificial Life*. [https://doi.org/10.1162/isal\\_a\\_00448](https://doi.org/10.1162/isal_a_00448)

Yang, J., & Kao, C. Y. (2001). A robust evolutionary algorithm for training neural networks. *Neural Computing and Applications*. <https://doi.org/10.1007/s521-001-8050-2>

Yao, X. & Higuchi, T. (1999). Promises and challenges of evolvable hardware. *IEEE Trans. Syst. Man Cybern. C*, *29*, 87–97. <https://doi.org/10.1109/5326.74067>