Adversarial Neuro Encoding with Binary Neural Networks

Raul Valencia
School of Computer Science,
University of Auckland
Auckland, New Zealand
rval735@aucklanduni.ac.nz

Chiu Wing Sham
School of Computer Science,
University of Auckland
Auckland, New Zealand
b.sham@auckland.ac.nz

***Abstract*—Adversarial neuro encoding could provide new insights for ciphering information with different perspectives. Nevertheless, it is still underexplored with a handful of publications on the subject. This work proposes the implementation of neuroevolved binary neural networks based on boolean logic functions only (BiSUNA) that apply payload ciphering between two agents to disperse information from an observer. The BiSUNA framework provides three distinctive attributions: it uses an adversarial neural encoding environment to improve the system data transmission; one execution yields a diversity of results given its population heuristics; lastly, it is an unconventional proposal to employ binary neural networks for the solution of symmetric ciphered problems.**

***Index Terms*—Binary Neural Network, BiSUNA, CPA, Adversarial Neurocryptography, Neuroevolution**

I. Introduction

The adoption of Deep Neural Network (DNN) to multiple fields of scientific knowledge has been exponential, areas like computer graphics [1]–[3], astronomy, geology [4] or games [5]. It is no surprise to keep exploring their capabilities in areas where technical proficiency is required to elaborate more complex systems. One such discipline is cryptography: based on a strong mathematics background, a multitude of complex systems use it every day. This subject has improved technology in telecommunications, finance and even media distribution [6].

A well-established crypto-system must have a suitably tested proof to confirm its validity along with its applications, especially if the source code is open and verifiable by multiple independent parties around the world. This paper will explore a recent branch of cryptography, which also establishes associations with other fields such as evolutionary computation and DNN, forming what is going to be known as Neuroevolved Encoding [7].

Neuroevolved Encoding approaches a novel way to elaborate multiple ciphered systems where payloads mask data with sufficient sophistication to hide information from a malicious party, using techniques from the Adversarial Neural Encoding (ANE) framework [8]. This publication moves further this area of research by proposing a binary logic-based architecture better suited to deal with any non-linear discrete sequences.

The main contribution of this work is the application of Binary Neural Network (BNN) to calculate optimal weights/neurons/topology to solve a data encoding problem. The algorithm Binary Spectrum-diverse Unified Neural Architecture (BiSUNA) [9] exploits evolutionary routines to resolve reinforcement learning environments (gradient descent free).

II. Background in Neural Encoding

Work by [10] demonstrated DNN potential with asymmetric key schemes: two parties share a known value (public key) while keeping some information confidential (private key). It revealed how that system was not secure because a genetic algorithm obtained the information both parties were sharing.

Years later, [11] employed DNN in the recreation of the original message from a DES/Triple-DES encrypted payload using a setting known as ”plaintext attack” with an average of 2^{12} playtext-ciphertext pairs.

Another publication was [12], which engaged a DNN to create an energy function to hide binary information on top of gray images, a technique also known as steganography. Similarly, [13] contributed to the utilization of Hopfield neural networks to transmit data within multimedia files. Empiric analysis of the information shared via encoded errors confirmed it was indeed a secure cryptosystem. ANE to organized multiple neural networks to reach different objectives.

On one side, two DNNs encrypted/decrypted payloads; on the other side, a different network tried to improve ciphertext quality within that flow, allowing the system to learn the One-Time Pad algorithm without any human intervention. This work explores further how ANC upgrades the overall encoding performance.

After analyzing some milestones to neural decoding, a pattern that all referenced papers take is the use of traditional neural networks. In other words, their deployments use gradient descent to update weights and approach a single solution of the task at hand, requiring floating-point values for training, even when those obstacles suit better with discrete outputs.

Another issue with traditional DNNs is its fixed topology because training only modifies the value of its connections. Therefore, researchers must test over multiple iterations which model reduces error. This work builds upon the findings of [14] to propose the first full binary neuroevolved encoding system.

III. Binary Spectrum-diverse Unified Neural Architecture

BiSUNA [9] is one of the latest additions to the TWEANN[1] family of algorithms with two distinctive characteristics:

- All neuron operations are logic functions;
- Spectra between agents employs hamming distance.

The second point deserves an explanation. BiSUNA takes a notional ”x-ray” of each agent, named spectrum, to differentiate between individuals. Agents are classified in groups by the function hamming distance, which keeps all actions needed within the realm of binary values, providing easy optimizations for general-purpose processors as well as programmable hardware [**?**].

Given BiSUNA's versatile nature, it forswears the need for gradient descent to guide the network, which forms an excellent alternative training algorithm, especially areas with discrete values or discontinuities functions. BiSUNA demonstrated its abilities for a similar job in [16].

Given its evolving nature, BiSUNA can encode network arrangements of different topologies while at the same time protects innovation and variety in its population thanks to the utilization of the novelty map. A novelty map is a simple data structure capable of organizing all agents into different groups, placing them according to similarities within each category, to then select the best among each class. The evolutionary loop can be visualized in figure 1.

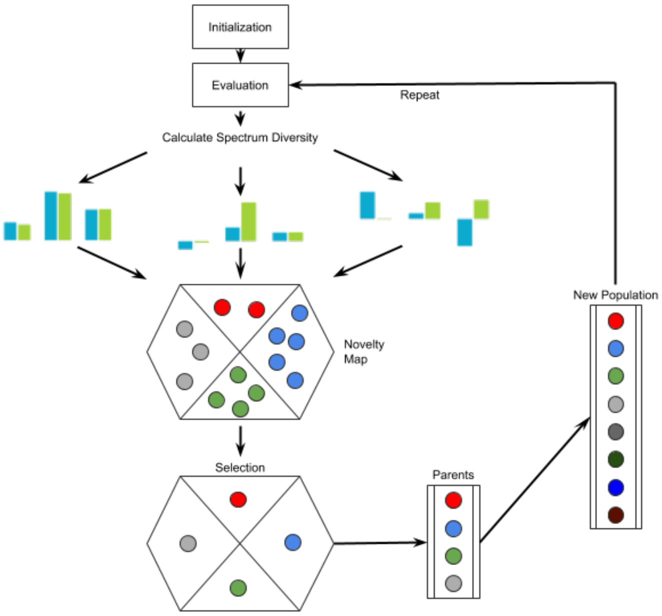


Fig. 1: BiSUNA loop with spectrum and novelty map

A neuroevolved cryptographic system [14] operated float neurons to achieve message transmission. Authors manipulated the original algorithm, adding new neuron types to create barriers between input - ciphertext - output, training agents as one single entity but having two distinctive sections within. It took advantage of ANC to enhance the quality of the encrypted payload using an adversary that tried to decrypt the shared information.

IV. Adversarial Neuro Encoding

Article [17] introduced the term Generative adversarial networks. It states the basic principle of training two or more DNN with seemingly different objectives still in a direction that improves the system results, where a single entity would not be possible.

Well, a similar scenario is employed in the ANC to secure communications, with a minimum of three participants: an encryptor (Alice), a decryptor (Bob) and an eavesdropper (Eve), each with a well-defined task. Alice takes as input plaintext values to generate a ciphertext. Bob acquires Alice's output, to which he applies a decryption function to restore the original plaintext.

Depending on the type of scheme Alice and Bob use, they can either share the same input key (symmetric encryption) or use public/private key pairs (asymmetric encryption). Eve picks the ciphertext as input and wants to decipher those values with the presumption she has more computational power to reverse Alice's function.

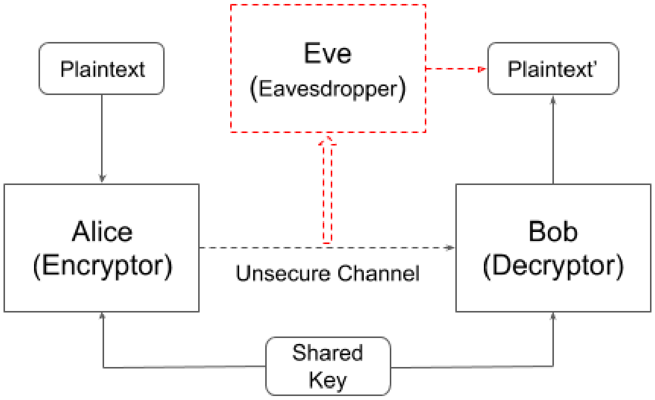


Fig. 2: ANC setting with three participants using symmetric encryption

V. Experiments

After a review of the high-level details on how BiSUNA works and the ANE environment, this section outlines the battery of experiments employed to confirm a successful resolution. The basics of the hardware used to deploy this work were 12 core / 24 threads, fully parallelized agent calculation between Alice, Bob and Eve populations; available RAM was 32GB with peak utilization below 2GB total, with Ubuntu 18.04 running.

The results below show a pre-trained Eve population that classifies payloads and its corresponding encoded text. Also, a ratio of 10 to 1 provided the adversary an unfair advantage over A&B.

[1] Topology and Weight Evolving Artificial Neural Networks [15]

TABLE I: Default configuration used for all tests

Configuration	Value	Comments
Generations	100,000	A&B's max number of generations
Population Size	100	Max number of agents for each party
Novelty Map Size	20	NM's max number of agents
Add Neuron	0.01	Chance of adding a neuron
Delete Neuron	0.01	Chance of removing a neuron
Add Connection	0.49	Chance of adding a connection
Delete Conn.	0.49	Chance of removing a connection
Neuromodulation	0.1	Chance of being a modulating neuron
Control Neuron	0.2	Chance of being a control neuron
Weight Mutation	0.5	Chance of changing a connection's value
Max steps	10*i	Steps each agent has to collect rewards
Adv. advantage	10	Generations Eve execute for every A&B
Adv. pre-train	true	Set Eve's pre-training
Trials	5	Repetitions of the same configuration

TABLE II: Summary of multiple results obtained with input sizes

Input	Bits	Solved Score	A&B Score	Eve Score	Training (hrs)
3	48	45,000	46,500	-1570	3.97
4	64	80,000	81,738	-1870	6.22
5	80	125,000	127,370	-2530	9.71
6	96	180,000	183,000	-3080	11.55
7	112	245,000	215,524	-3540	16.33

Table I has the BiSUNA configuration for all test. Each input is a 16 bitset, with 3, 5 and 7 input neurons employed, in other words, 48, 80 and 112 bits payload input/output.

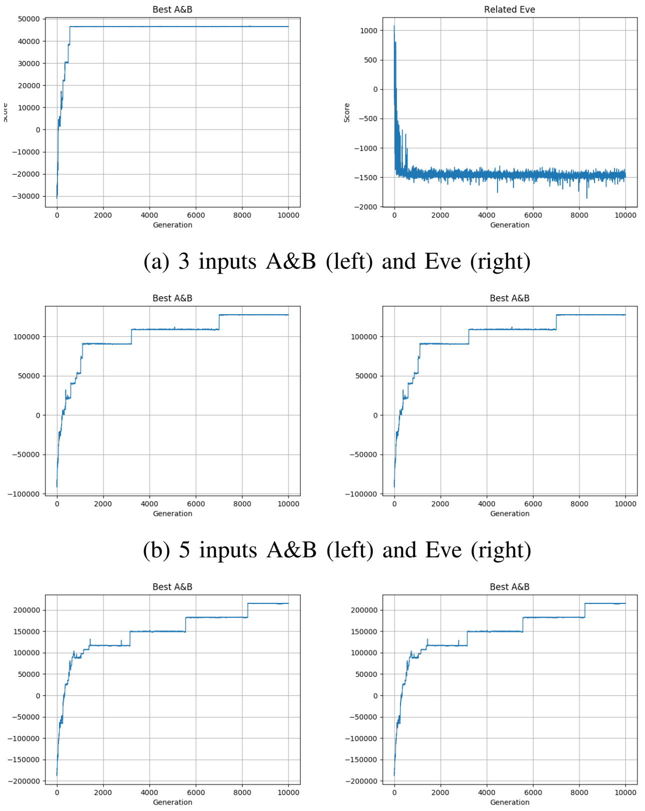
Each test ran for 100,000 generations + Eve's pre-training, taking between 3 to 28 hours to execute depending on the number of inputs and its associated value "Max steps".

With all bases covered, it is time to disclose the results of running the multiple RL environments with the settings previously outlined. Table II summarizes the information obtained for 3 to 8 input size, each with a 16 bitset, with 5 repetitions of 100,000 generations; confirming that in every case A&B reached a solution; on the other hand, despite Eve's initially good performance, it vanished as the contestants were learning to cipher their communication.

Refer to Figure 3 to analyze data gathered for each execution. These graphs show the best agents next to their adversary. These plots show how A&B rewards initially are low. As evolution progresses, they develop a system that boosts their compensation. At the same time, Eve gains are not capable of keeping up with the improvements and its compensation drops, despite having 10 times more computational capacity.

Considering the information provided by Table II, there are some insights to obtain from executing the ENA system with multiple values. First, there is a direct correlation between the number of inputs/bits and scores, growing in proportion to data sizes.

The column "Solved Score" provides the minimum reward A&B population should achieve to be considered a solved problem. Most inputs surpass the required value, except for 7, with one trial performing poorly compared to other executions (83,620 points).



(a) 3 inputs A&B (left) and Eve (right)

(b) 5 inputs A&B (left) and Eve (right)

(c) 7 inputs A&B (left) and Eve (right)

Fig. 3: Test results of running BiSUNA with multiple input sizes in an ENA environment

Training with 8 inputs and over had adverse results, obtaining fewer rewards than minimally established to solve the environment, combined with a substantial increase in training time (about 28.6h). Future work will revisit larger input bitset size.

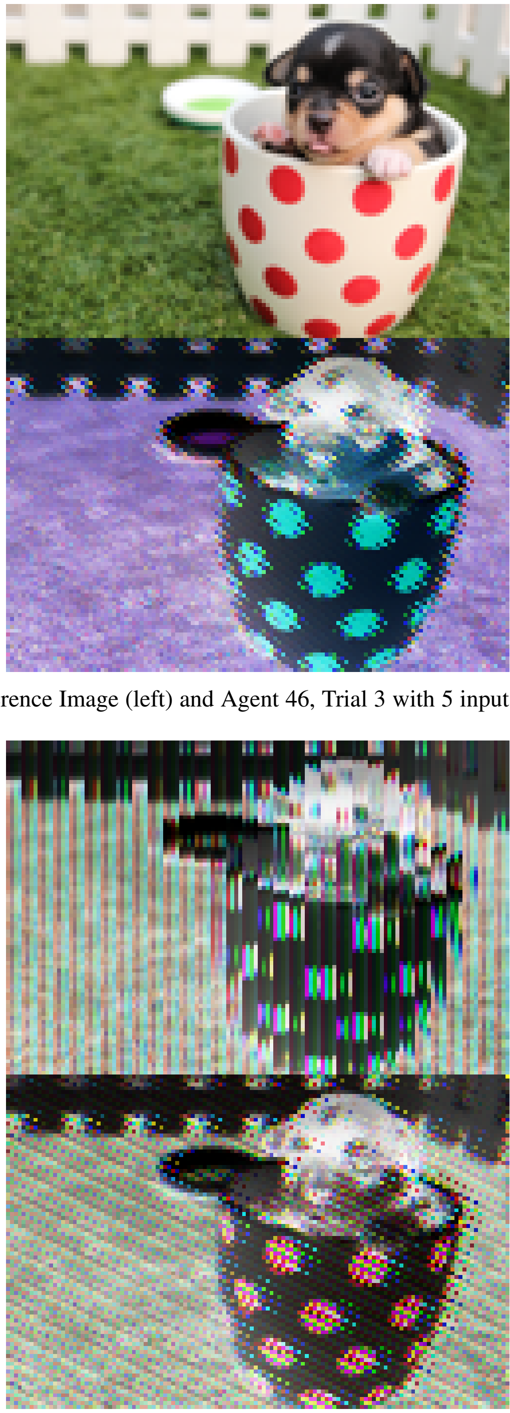
Eve rewards decrease according to the number of bits used, although on a different scale. Any positive final score would have shown that the adversary was obtaining valuable information only from the encoded data, which was not the case for any test.

Lastly, training (measured in hours) takes a substantial amount of time; more parallelization techniques would be necessary to minimize delays. As a summary of how this work compares to the state-of-the-art publications, Table III shows previous work, their main characteristics and differences with the experiments provided here.

To grasp a visual representation of A&B actions, one more test employed images to discern the encoding activity. Figure 4 displays multiple images, first starting with a reference followed by multiple A&B input sizes.

BiSUNA agents reached the objective of this publication given that, without previous knowledge, those populations learned to encode/decode data without human expertise, showing off its capabilities as well as applying these functions within a reasonable time (once trained).

Another point to reflect on, this is still early days for neural encoding, BNN research must continue to implement these systems in real-world applications.



(a) Reference Image (left) and Agent 46, Trial 3 with 5 inputs encode (right)

(b) Agent 96, Trial 4 with 6 inputs (left) and Agent 97, Trial 1 with 7 inputs (right)

Fig. 4: Visual encoding of a reference image

TABLE III: Comparison with previous work

Work	Setting	Topology	Values	Training	Solutions
[18]	ANC	Fixed	Float	GD^{2}	1
[14]	ANC	TWEANN	Float	SUNA	multiple
[8]	CPA	Fixed	Float	GD	1
This	ANE	TWEANN	Binary	BiSUNA	multiple

While similar publications use continuous neurons, this work transforms its operations to binary neural networks. Population metaheuristics demonstrate how multiple solutions adapt to variable circumstances. When a trained ANE agent needs high confidence, BiSUNA would only entail more training to increase the number of solutions along with the confirmation it is a succinct system.

Another important distinction is the application of neuro evolved binary neural networks trained via the BiSUNA technique, in comparison with traditional Gradient Descent (GD) of typically fixed topologies floating-point neural networks.

VI. Conclusion

This article analyzed the flexibility offered by the BiSUNA algorithm with discrete values; it resolves an application focused on codification, specifically data encoding. With a brief description of how the algorithm works and the environment to test, experiments demonstrated that BiSUNA creates neural encoding systems that can help researchers scrutinize diversity.

As reviewed in section V, BiSUNA resolves the Adversarial Neural Encoding problem using three elements, each with unique populations; when the algorithm converges, it finds a set of multiple solutions that can be analyzed further to confirm which one satisfies the constraints of the target device.

Even though A&B were able to learn to communicate effectively within a reasonable number of iterations (less than 10,000), it was necessary to execute substantially more generations to give double advantage to Eve to confirm that the system encoded data appropriately.

Analyzing visual representations obtained of A&B encoded images, it confirms its validity and the fact that more research is required to use this system in real-world applications. Notwithstanding, thanks to this work, it is possible to move a little step forward in the utilization of binary neural networks to solve reinforcement learning problems.

Future work will focus on a comprehensive mathematical review of the results obtained by the neural networks along the lines of correctness, succinctness and resource utilization. Another area of improvement would be the reduction of training time to make the exploration of various structures a more dynamic process.

References

[1] J. Nalepa, M. Antoniak, M. Myller, P. R. Lorenzo, and M. Marcinkiewicz, “Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation,” *Microprocessors and Microsystems*, vol. 73, p. 102994, 2020.

[2] C.-Y. Lo, F. C. M. Lau, and Chiu-Wing Sham, “Fixed-Point Implementation of Convolutional Neural Networks for Image Classification,” in *2018 International Conference on Advanced Technologies for Communications (ATC)*, Oct 2018, pp. 105–109.

[3] C. Y. Lo and C.-W. Sham, “Energy efficient fixed-point inference system of convolutional neural network,” in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2020, pp. 403–406.

[4] G. Ismayilov and H. R. Topcuoglu, “Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing,” *Future Generation Computer Systems*, vol. 102, pp. 307 – 322, 2020.

[5] Q. Lu, Chiu-Wing Sham, and F. C. M. Lau, “An architecture-algorithm co-design of artificial intelligence for Trax player,” in *2015 International Conference on Field Programmable Technology (FPT)*, Dec 2015, pp. 264–267.

[6] G. M. de Dormale and J.-J. Quisquater, “High-speed hardware implementations of elliptic curve cryptography: A survey,” *Journal of Systems Architecture*, vol. 53, no. 2, pp. 72 – 84, 2007, embedded Hardware for Cryptosystems.

[7] P. Dürr, C. Mattiussi, and D. Floreano, “Neuroevolution with analog genetic encoding,” in *Parallel Problem Solving from Nature - PPSN IX*, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 671–680.

[8] M. Coutinho, R. de Oliveira Albuquerque, F. Borges, L. J. García Villalba, and T.-H. Kim, “Learning perfectly secure cryptography to protect communications with adversarial neural cryptography,” *Sensors*, vol. 18, no. 5, 2018.

[9] R. Valencia, C. Sham, and O. Sinnen, “Using neuroevolved binary neural networks to solve reinforcement learning environments,” in *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Nov 2019, pp. 301–304.

[10] A. Klimov, A. Mityagin, and A. Shamir, “Analysis of neural cryptography,” in *Advances in Cryptology — ASIACRYPT 2002*, Y. Zheng, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 288–298.

[11] M. M. Alani, “Neuro-cryptanalysis of des and triple-des,” in *Neural Information Processing*, T. Huang, Z. Zeng, C. Li, and C. S. Leung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 637–646.

[12] Tai-Wen Yue and Suchen Chiang, “A neural network approach for visual cryptography,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 5, July 2000, pp. 494–499 vol.5.

[13] W. Yu and J. Cao, “Cryptography based on delayed chaotic neural networks,” *Physics Letters A*, vol. 356, no. 4, pp. 333 – 338, 2006. [Online]. Available:

[14] Y. Zhu, D. V. Vargas, and K. Sakurai, “Neural cryptography based on the topology evolving neural networks,” in *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)*, Nov 2018, pp. 472–478.

[15] K. O. Stanley and R. Miikkulainen, “Efficient evolution of neural network topologies,” in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No.02TH8600)*, vol. 2, May 2002, pp. 1757–1762 vol.2.

[16] R. H. V. Tenorio, C. W. Sham, and D. V. Vargas, “Preliminary study of applied binary neural networks for neural cryptography,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 291–292.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[18] M. Abadi and D. G. Andersen, “Learning to Protect Communications with Adversarial Neural Cryptography,” *ArXiv e-prints*, Oct. 2016.