Doing More with Less: Computational Role of Information Structure in Neural Networks based on Entropy Maximization

Alexandre Pitti*
ENSEA, Cergy, France
alexandre.pitti@ensea.fr

Claudio Weidmann
CY Cergy Paris University
claudio.weidmann@cyu.fr

Krzysztof Lebioda
Technical University of Munich
krzysztof.lebioda@tum.de

Abstract

We propose a bio-inspired concept based on the maximization of entropy in neural networks for memory storage and higher-order cognitive skills. We emphasize the role of information structure in mapping high-resolution inputs onto extremely low-resolution neurons. Despite the unreliability of neurons due to intrinsic noise and limitations, their interaction allows error-free reconstruction. In particular, we show that the necessary number of neurons for reconstruction grows linearly while the resolution of the input grows exponentially.

Playing with the information structure of neurons, we can make them sensitive to symbolic information in signals, like hierarchical binary trees or the relative order of elements in sequences. These features are a hallmark of symbolic systems and of higher-order cognitive skills.

1 Introduction

Despite the unreliability of biological neurons, the brain efficiently exploits their poor computational resources for fast encoding, robust memory preservation, and also for performing high-level cognitive tasks such as scene understanding and hierarchical planning. In comparison, current machine learning uses a different strategy with artificial neurons designed with virtually infinite precision of their weights, unlimited time and gigantic resource (data and GPU) and energy usage.

Here, we emphasize the role of information structure in neural computation (1) for capturing intrinsic features found in data, and (2) for efficient processing, despite intrinsic noise and errors. First, we show how error-free efficient encoding can be done in imprecise neurons of low resolution R_W (number of distinct synaptic weights) to reconstruct highly precise input X of much higher resolution R_X (cardinality of input alphabet), even though $R_W \ll R_X$. Using Information Theory (IT) and the source coding theorem of Claude Shannon, we demonstrate that unreliable neurons can optimize their codes to approach Shannon limits in terms of information capacity, so that each neuron added to the neural population augments its memory capacity following an exponential rule.

This idea is in line with the principle of Entropy Maximization (PEM) proposed by Barlow, who hypothesized that biologic systems optimize their resources by using efficient codes to maximize information (entropy). The PEM is described in Annex Section 5.1.

*The full affiliation of A. Pitti and C. Weidmann is ETIS UMR8051, CY Cergy Paris University, ENSEA, CNRS, F-95000 Cergy, France.

1st Workshop on NeuroAI @ 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Based on this principle, we use randomness as a key mechanism to create orthogonal neural representations and to disambiguate information during reconstruction. It follows that the neural codes create advantageously compact and efficient representations, despite their low resolution at the level of individual neurons. This computational treatment of information is similar to binary coding in digital processing.

These two mechanisms, randomness and compactness, represent the minimal and sufficient conditions to realize efficient coding in an informative system. In line with other proposals, we hypothesize that these two mechanisms are sufficient to describe many neural computations in the brain.

We present several examples in which these random and compact neurons are used to represent structured information in spatio-temporal data, like trees or directed graphs. These features are a hallmark of symbol processing and the higher-order skills processed in the frontal cortex, such as grammar, action plan, visual geometry and algebra (Dehaene et al., 2015).

Although such compact codes are traditionally associated with Good Old-Fashioned AI systems, they can be associated also with spiking neurons and the bio-inspired learning mechanism of Spike Timing-Dependent Plasticity (STDP), which is a temporal code sensitive to the serial order of spike trains (Thorpe et al., 2001; Van Rullen and Thorpe, 2002).

The following Section 2 presents our motivation and the principle used to describe the architecture. Section 3 presents simulation results. We conclude by pointing out links with brain computation, cognitive development, as well as links between current machine learning and bio-inspired neural models.

2 Methods

Neuron Resolution and Patterns. Appendix Section 5.2 presents the two stages of the algorithm and its pseudo-code Algorithm 1. The first stage consists in encoding part of the signal X into the synaptic weights W, shuffled and quantized. This can be done in one-shot learning for a specified number of synapses k.

Shuffling means randomly permuting the order of the synaptic connections from the input X to the neurons Y, so that each neuron 'sees' the input in a specific order proper to that neuron. Each neuron will have its own specific randomly shuffled repertoire so that one value in $X \in \mathcal{R}_X$ will correspond to different values $W \in \mathcal{R}_W$ for different neurons. Here $\mathcal{R}_X$ is the input set of cardinality R_X, while $\mathcal{R}_W$ is the set of possible synaptic weights, of cardinality R_W. The neural code (codeword) representing input X will take its values within the repertoire $\mathcal{R}_W^k$, which is also the number of possible states taken by the weight matrix $\mathbf{W}$. Here k is the number of synapses; if N neurons code an input sequence X of length L, one has $k = NL$.

For $R_W = 2$, the synaptic weights will have binary values $\{w_0, w_1\}$ and the memory capacity of the neural network will be equal to $k \log 2 = k$ bits ($\log$ is base 2 in this article). Binary vectors have interesting properties to encode hierarchical trees and symbolic patterns.

For the special case of ordinal codes, the resolution of the synaptic weights R_W is equal to the sequence length L of the input X, with the additional constraint that every weight is used only once. The neural codes will be simply the relative order of the input sequence: e.g., the input vector $[13.3333; 3.14; 5.666]$ will be encoded by the following ordinal code $[3; 1; 2]$, corresponding to the highest value, the lowest value and the value in between (instead of integers 1,2,3, the actual weights might be any distinct numbers w_1, w_2, w_3). Ordinal codes can be seen as encoding a permutation of cardinality L. Thus the memory capacity of an L-input ordinal neuron is $\log L! \approx L \log(L/e)$, using Stirling's approximation of the factorial function and Euler's number $e = 2.718\ldots$. Interestingly, ordinal neurons have been found to process serial order information in the frontal cortex (Pitti et al., 2022a). Ordinal codes have interesting algebraic and combinatorial features for manipulating context-free grammars, and to represent trees and directed graphs (Pitti et al., 2022a).

Retrieving Patterns. The second stage corresponds to the reconstruction or decoding phase. It follows an iterative procedure to refine the signal step by step by a belief vote at the neural population level. This iterative stage is similar to the Expectation-Maximization algorithm or the Boltzmann machine mechanism, found also in predictive coding and active inference (Rao and Ballard, 1999; Friston et al., 2016).

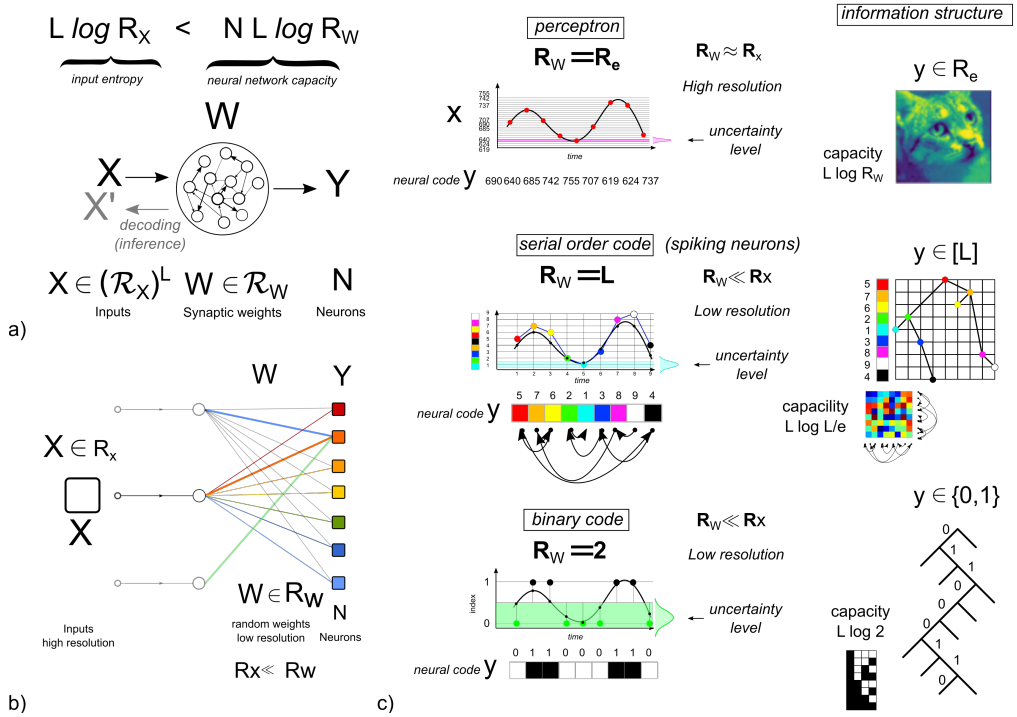

Figure 1: Entropy Maximization principle in neural networks. a) In Information Theory, a neural network can be seen as a communication channel through which information can be transmitted or stored. Following this, an L-dimensional input $\mathbf{X}$ of resolution R_X, and entropy $L \log R_X$, can be conveyed into neurons Y of much lower-precision weights W with synaptic resolution R_W. b) The minimal number N of neurons Y necessary to encode input $\mathbf{X}$ without loss is given by Shannon's source coding theorem. This number depends on the neurons' information structure, which means their number of inputs and their resolution R_W. Accordingly, efficient codes can be constructed with a minimal number of neurons, despite their weak computational capabilities. c) the neurons' resolution R_W, which is the information structure of the neural code, can serve to represent various types of patterns. For synaptic weights of high resolution, $\mathcal{R}_W \approx \mathcal{R}_X \approx \mathbb{R}$, neurons are similar to perceptrons, with neural codes of same resolution as the input, with lots of redundancy. For synaptic weights of low resolution such that $R_W \ll R_X$, neurons with binary codes can be created for $R_W = 2$ and with serial order codes for $R_W = L$, with L the length of the input vector $\mathbf{X}$. These codes are often used in symbolic processing. Although discrete codes perform a harsh quantization of the input $\mathbf{X}$, they are faster to compute and to reconstruct the input.

At each step, the neurons make a decision vote predicting the output value, modeled by a Gaussian distribution centered on the most probable value. Decision votes from multiple neurons are then summed up and refined for the next step.

The harsh quantization in W create large errors in the decision votes at the single neuron level, and large uncertainty intervals. However, the decision vote with multiple neurons allows to discriminate the candidate values and to retrieve the higher resolution of the original signal. Accordingly, the random connections do not affect the belief vote, but create sparse and orthogonal representations such that only the most likely candidate values among the neurons survive, see Section 3.

This error-correcting treatment of information is similar to Bayesian inference. The neurons provide a conditional output relative to its likelihood to the class or to the input: $q(x/z) \propto p(z/x)p(x)$ (here $p(z)$ may be assumed uniform due to randomization). The likelihood of the neuron z, $p(z/x)$, can be computed, stored, and then used to discriminate the input x. The variance is chosen arbitrarily, it trades off convergence properties with computational complexity. The neuron input randomization acts upon the belief vote so that with high probability the original input value receives the largest cumulative vote, while other values receive a lower, random number of votes.

3 Experiments

3.1 XP 1: Input Reconstruction for different neural resolutions

We test first the reconstruction capabilities on an image of size $L = 512 \times 512$ and pixel's resolution $R_X = 256$. We encode the image in a neural population of $N = 100$ neurons with two different resolutions of the synaptic weights W: $R_W = 2, 10$ for binary and decimal quantization.

Fig. 2 shows the results for the resolution $R_W = 2$ only, i.e. binary weights. The reconstruction process corresponds to a belief vote in which each neural unit is added iteratively during the decision making stage. Fig. 2 a) presents the decision votes for 50 pixels between $[0, 255]$ and the trajectory for five of them is presented in Fig. 2 b). The quadratic error is presented in Fig. 2 c).

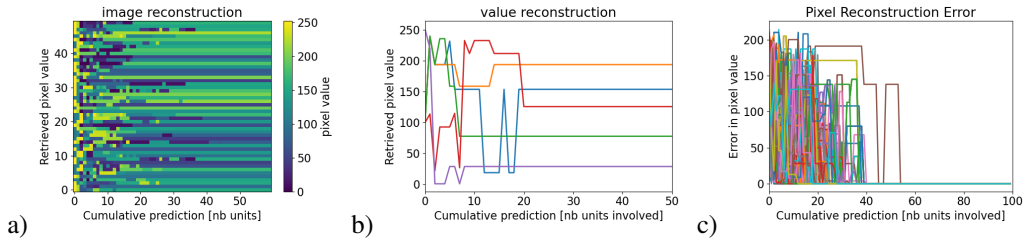

Figure 2: Reconstruction stage. a) cumulative prediction of pixel values, by adding neurons' votes one by one. b) retrieved pixel values, for 5 pixels only. c) reconstruction error.

The convergence to the original image resolution $R_X = 256$ is achieved rapidly by using 20 to 40 neurons during the decision vote. The reconstruction process is also not monotonic as observed in Fig. 2 c). The pixel values change with respect to the number of neurons used for computing the global decision, and stabilize after a certain number is used. In line with the source coding theorem, there is a threshold to information capacity and a minimal number of neurons N is necessary to allow correct decoding. Nonetheless, the relationships among the neural codes are highly nonlinear due to randomization, so a small number is enough to reconstruct the original information. The reconstruction process has some similarities to Diffusion Probabilistic models. In comparison, it is faster to converge as it typically requires some dozen of steps, see Fig. 2 a-b). However, the iterative process of the decision vote is also highly nonlinear.

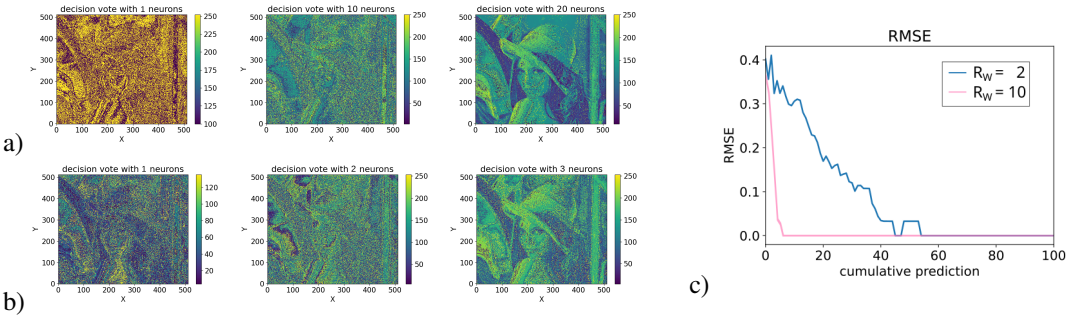

Figure 3: Computational efficiency for different information structure. Image reconstruction with neuron weight resolution $R_W = 2, 10$. a) Reconstruction for $k = 1, 10, 20$ neurons with weights $R_W = 2$. b) Reconstruction for $k = 1, 2, 3$ neurons with weight resolution $R_W = 10$. c) Reconstruction error related to information structure and neuron weight resolution.

Fig. 3 a-b) on the left side corresponds to the decoding of one selected neuron only for weight resolution $R_W = 2$ and $R_W = 10$. Although both neurons have a very low resolution, the behavior of the two neural networks drastically change when combined together during the reconstruction stage. For instance, the binary codes permit to reconstruct perfectly the input with 40 to 50 neurons whereas for $R_W = 10$, ten times less neurons are enough to reconstruct information.

This result shows the impact of the neurons' resolution on encoding. For instance, as the architecture of the neural network maximizes entropy through randomness, the number of neurons required diminishes by an exponential rule.

3.2 XP 2: Serial order planning in Tower of Hanoi game

The Tower of Hanoi puzzle is a game where all disks have to be positioned in a specific goal state, moving one disk at a time on the different rods. As the number of disks N augments, the computational complexity increases exponentially as 2^{N-1}. In comparison to machine learning techniques, humans are good at it by grasping the structure of the game (pattern extraction), and by composing new plans based on the few examples learned (learning-to-learn).

The states can be represented as nodes in a self-similar graph. Using this graph representation, sequences of discrete states can be constructed and analyzed. The experiment presents the most well-known variant with 3 rods and 3 disks (27 states). We have tested slightly more complex variants with an extra disk (81 states), 6 disks (729 states); also versions with up to 3125 states (5 rods and 5 disks) were examined.

A set of ordinal patterns, which are sensitive to the serial order of the items present in the sequences, can be extracted from the example chunks; see Fig. 1. These patterns can be thought of as the grammar, or the set of rules, that governs the sequences, and that can be used to both reconstruct missing items within a sequence and to generate novel sequences.

The models were evaluated by calculating the ratio between the ground truth shortest path length and the the generated path length. The averaged results for reinforcement learning algorithms like DQN and PPO, as well as our algorithm with both types of example sequences are presented in Table 1. Our approach used less than 100 neurons with one-shot learning. DQN and PPO were able to generate models that solve the tasks with both start and target nodes fixed, and with random start node and fixed target node. DQN required 200.000 iterations, PPO 50.000 iterations. Besides, DQN was not able to generate a useful model for the case with both start and target nodes set to random, even after 5 million iterations. In the case of PPO, the model trained over 200.000 iterations for the last case is sub-optimal, the generated path was 50% longer than in the other cases.

Our approach initialized with example sequences generated using a random walk had the best performance, better than the version using the shortest path examples. This could be explained by greater variability in the random sequences. Some of the paths are indeed never seen when using only shortest paths. Despite the coarseness of serial order codes, they are capable to capture the relative information structure in sequences, which is pertinent for compositionality (Lebioda et al., 2024).

ALGORITHM	FIXED START FIXED TARGET	RANDOM START FIXED TARGET	RANDOM START RANDOM TARGET
DQN	0.693	0.743	-
PPO	0.736	0.796	0.363
OURS (SHORT)	0.712	0.754	0.798
OURS (RAND)	**0.882**	**0.808**	**0.849**

Table 1: Ratio of shortest path's length (ground truth) to the generated path's length, averaged over 100 trials. The environment was Towers of Hanoi graph with 3 pegs and 3 plates. OURS (SHORT) refers to our algorithm initialized with example sequences that are shortest paths, while OURS (RAND) was initialized with chunks generated by random walk.

4 Discussion

Horace Barlow made the hypothesis that the brain maximizes its computational resources to overcome the unreliability of its neurons due to intrinsic noise and material limitation (Barlow, 2001, 2012). Accordingly, although the computational performances of noisy neurons are poor, the interaction of few neurons can increase their performance exponentially, as their entropies are summed.

Thus, the Principle of Entropy Maximization can explain how information compression, and memory retention can be done in neural networks (Jirsa and Sheheitli, 2022). For this, we presented a

biologically plausible method that removes redundancy in a compact way by limiting neurons' variability (quantization) and by shuffling their alphabet order (randomization). By doing so, neurons have a different information structure from raw input. Here we emphasize its advantage in higher-level cognitive skills; e.g., to extract hierarchical patterns like binary trees and serial order information.

Among all types of information structure, the ability to process hierarchical trees and serial order in sequences has been acknowledged as a sign of higher-order cognition (Dehaene et al., 2015; Rosenbaum et al., 2007). It is interesting that relatively poor capabilities of spiking neurons may be advantageously exploited as they are also sensitive to the spatio-temporal order of spike trains (Thorpe et al., 2001). During cognitive development in infancy, infants appear to rapidly learn from physical interactions the causal and temporal rules present in data as well as their violation. Indeed, whenever a sequence of actions does not correspond to any of the known ordinal rules, it can be evaluated to be either a rule violation or a new 'symbolic' rule, which can then be incorporated into the known repertoire.

References

J.J. Atick and A. N. Redlich. What does the retina know about natural scenes? *Neural Computation*, 4:196–210, 1992.

H. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12(3): 241–253, January 2001. ISSN 0954-898X, 1361-6536. doi: 10.1080/net.12.3.241.253. URL `https://www.tandfonline.com/doi/full/10.1080/net.12.3.241.253`.

H. B. Barlow. Possible Principles Underlying the Transformations of Sensory Messages. In Walter A. Rosenblith, editor, *Sensory Communication*, pages 216–234. The MIT Press, September 2012. ISBN 978-0-262-51842-0. doi: 10.7551/mitpress/9780262518420.003.0013. URL `https://academic.oup.com/mit-press-scholarship-online/book/20714/chapter/180090664`.

C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, pages 1064–1070 vol.2, 1993. doi: 10.1109/ICC.1993.397441.

Trenton Bricken and Cengiz Pehlevan. *Attention approximates sparse distributed memory*. 2021. Publication Title: NeurIPS.

Emmanuel Candes, Justin Romberg, and Terence Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements, 2005. URL `https://arxiv.org/abs/math/0503066`.

Lancelot Da Costa, Thomas Parr, Noor Sajid, Sebastijan Veselic, Victorita Neacsu, and Karl Friston. Active inference on discrete state-spaces: A synthesis. *Journal of Mathematical Psychology*, 99:102447, December 2020. ISSN 00222496. doi: 10.1016/j.jmp.2020.102447. URL `https://linkinghub.elsevier.com/retrieve/pii/S0022249620300857`.

S. Dehaene, F. Meyniel, C. Wacongne, L. Wang, and C. Pallier. The neural representation of sequences from transition probabilities to algebraic patterns and linguistic trees. *Neuron*, 88:2–19, 2015.

Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2):288–299, July 2017. ISSN 0022-4715, 1572-9613. doi: 10.1007/s10955-017-1806-y. URL `http://arxiv.org/abs/1702.01929`. arXiv:1702.01929 [math].

D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

K.J. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, J. O'Doherty, and G. Pezzulo. Active inference and learning. *Neuroscience & Biobehavioral Reviews*, 68:862–879, 2016.

A. Graves, G. Wayne, and I. Danihelka. Neural Turing Machines. *arXiv*, 1410.541v2:1–26, 2014.

E. Guizzo. Closing in on the perfect code [turbo codes]. *IEEE Spectrum*, 41(3):36–42, 2004. doi: 10.1109/MSPEC.2004.1270546.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239. arXiv: 2006.11239.

Viktor Jirsa and Hiba Sheheitli. Entropy, free energy, symmetry and dynamics in the brain. *Journal of Physics: Complexity*, 3(1):015007, March 2022. ISSN 2632-072X. doi: 10.1088/2632-072X/ac4bec. URL https://iopscience.iop.org/article/10.1088/2632-072X/ac4bec.

Pentti Kanerva. Sparse distributed memory. *MIT*, 1988.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv*, 1312.6114, 2014.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

Dmitry Krotov and John Hopfield. Large Associative Memory Problem in Neurobiology and Machine Learning, April 2021. URL http://arxiv.org/abs/2008.06996. arXiv:2008.06996 [cond-mat, q-bio, stat].

Dmitry Krotov and John J. Hopfield. Dense Associative Memory for Pattern Recognition, September 2016. URL http://arxiv.org/abs/1606.01164. arXiv:1606.01164 [cond-mat, q-bio, stat].

Simon Laughlin. A Simple Coding Procedure Enhances a Neuron's Information Capacity. *Zeitschrift für Naturforschung C*, 36(9-10):910–912, October 1981. ISSN 1865-7125, 0939-5075. doi: 10.1515/znc-1981-9-1040. URL https://www.degruyter.com/document/doi/10.1515/znc-1981-9-1040/html.

Simon Laughlin and Terrence Sejnowski. Communication in Neuronal Networks. *Science (New York, N.Y.)*, 301:1870–4, October 2003. doi: 10.1126/science.1089662.

Krzysztof Lebioda, Alexandre Pitti, Fabrice Morin, and Alois Knoll. Serial order codes for dimensionality reduction in the learning of higher-order rules and compositionality in planning. In Michael Wand, Kristína Malinovská, Jürgen Schmidhuber, and Igor V. Tetko, editors, *Artificial Neural Networks and Machine Learning – ICANN 2024*, pages 32–46, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72341-4.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal Hopfield Networks: A General Framework for Single-Shot Associative Memory Models, June 2022. URL http://arxiv.org/abs/2202.04557. arXiv:2202.04557 [cs].

B. A. Olshausen and D. J. Field. Sparse coding of sensory input. *Curr Opin Neurobiol*, 14(6): 481–487, 2004.

B.A. Olshausen and M.S. Lewicki. What natural scene statistics can tell us about cortical representation. *The New Visual Neurosciences. J. Werner, L.M. Chalupa, Eds. MIT Press.*, 2013.

Alexandre Pitti, Mathias Quoy, Catherine Lavandier, Sofiane Boucenna, Wassim Swaileh, and Claudio Weidmann. In search of a neural model for serial order: a brain theory for memory development and higher-level cognition. *IEEE Transactions on Cognitive and Developmental Systems*, 10.1109/TCDS.2022.3168046, 2022a.

Alexandre Pitti, Claudio Weidmann, and Mathias Quoy. Digital Processing based on Randomness and Order in Neural Networks. *PNAS*, 119(33):e2115335119, 2022b.

Julien Pourcel, Ngoc-Son Vu, and Robert M. French. Online Task-free Continual Learning with Dynamic Sparse Distributed Memory. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13685, pages 739–756. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-19805-2 978-3-031-19806-9. doi: 10.1007/978-3-031-19806-9_42. URL https://link.springer.com/10.1007/978-3-031-19806-9_42. Series Title: Lecture Notes in Computer Science.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need, April 2021. URL http://arxiv.org/abs/2008.02217. arXiv:2008.02217 [cs, stat].

R.P. Rao and D.H. Ballard. Predictive coding in the visual cortex a functional interpretation of some extra-classical receptive-field effects. *Nat Neurosci*, 2:79–87, 1999.

Edmund T. Rolls. Pattern separation, completion, and categorisation in the hippocampus and neocortex. *Neurobiology of Learning and Memory*, 129:4–28, March 2016. ISSN 10747427. doi: 10.1016/j.nlm.2015.07.008. URL https://linkinghub.elsevier.com/retrieve/pii/S107474271500129X.

Edmund T. Rolls and Alessandro Treves. The neuronal encoding of information in the brain. *Progress in Neurobiology*, 95(3):448–490, November 2011. ISSN 03010082. doi: 10.1016/j.pneurobio.2011.08.002. URL https://linkinghub.elsevier.com/retrieve/pii/S030100821100147X.

David A. Rosenbaum, Rajal G. Cohen, Steven A. Jax, Daniel J. Weiss, and Robrecht van der Wel. The problem of serial order in behavior: Lashley's legacy. *Human Movement Science*, 26(4):525–554, August 2007. ISSN 01679457. URL https://linkinghub.elsevier.com/retrieve/pii/S0167945707000280.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *CoRR*, abs/1503.03585, 2015. URL http://arxiv.org/abs/1503.03585. arXiv: 1503.03585.

S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14:715–725, 2001.

Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael Mozer, Kenji Kawaguchi, Yoshua Bengio, and Bernhard Schölkopf. Discrete Key-Value Bottleneck, July 2022. URL http://arxiv.org/abs/2207.11240. arXiv:2207.11240 [cs].

J.H. van Hateren. A theory of maximizing sensory information. *Biological Cybernetics*, 68(1):23–29, 1992.

R. Van Rullen and S. Thorpe. Surfing a spike wave down the ventral stream. *Vision Research*, 42:2593–2615, 2002.

5 Annex

5.1 Principle of Entropy Maximization

Our approach is based on the Maximization of Entropy principle (ME), which is a principle rooted in Thermodynamics and used then in Information Theory. In biology, ME has been proposed as a core concept for the efficient encoding of information in the brain by redundancy minimization (Barlow, 2001, 2012; Laughlin, 1981; Rolls and Treves, 2011; Jirsa and Sheheitli, 2022). ME is complementary to the Free-Energy minimization principle for the brain, proposed by Karl Friston (Da Costa et al., 2020), and to the sparse coding of neural information (Olshausen and Field, 2004; Rolls, 2016). The hypothesis of efficient encoding states that neurons must encode information as efficiently as possible in order to maximize neural resources (van Hateren, 1992; Atick and Redlich, 1992; Laughlin and Sejnowski, 2003). To do so, an optimal code must suppress the redundancy present in data and keep the useful information only. Removing redundancy means suppressing information that can be reconstructed by inference. As a consequence, useful information is also more compact, less predictable (because it could have been inferred otherwise) and resembles more a random signal (Atick and Redlich, 1992; Olshausen and Lewicki, 2013). It follows that more information can be stored for the same memory capacity limit.

Following the principle of ME, we devise a similar treatment of information embedded into neural networks to maximize the data storage within, with the most compact neural codes, and to achieve a large capacity memory system (Pitti et al., 2022b). For this, we introduce two important mechanisms, namely quantization and permutation, in order to create neuron synaptic weights W with random connections and low resolution R_W. On the one hand, the quantization of signals X of resolution R_X into a neural code W of resolution R_W, with $R_W \ll R_X$, produces a harsh discretization of data values that is easier to manipulate for neurons. It may suppress redundancy as well, and produces discrete neural codes W with fewer states R_W, and of lower entropy. On the other hand, the random connections from the original signal contribute to differentiate the neural representations for each neuron. Although a single neural code of resolution R_W is not capable to represent completely the original information of higher resolution R_X, we show that only a small number of neurons is enough to reconstruct it perfectly without loss. Accordingly, randomness does not destroy information, but helps to disambiguate it in dense codes with few units.

We show that neural networks initialized with random vectors can convey maximal information, and approach Shannon's limit in terms of capacity with the equation $\log R_X \approx k \log R_W$, with k the number of neural weights.

This use of the ME principle is in line with the definition of entropy proposed by Boltzmann and reformulated by Shannon for digital computing. We suggest therefore that our model instantiates a new type of neural model, a digital neural network.

5.2 Neural code implementation

The coding strategy consists of discretizing the items in the sequence into a given repertoire or alphabet of cardinality R_W. In experiment XP1 this is done by simple uniform quantization, i.e. $W_i = \lfloor X_i/q \rfloor \cdot q + q/2$ for an appropriate step size q (yielding the desired number of quantization levels R_W).

As outlined in Section 2, the ordinal neurons used in experiment XP2 have resolution $R_W = L$, with L the length of the input sequence, and the constraint that every weight is used only once. Then the neural code corresponds to an ordinal code, sensitive to the serial order of the elements present in the sequence; i.e. their relative amplitude or temporal order.

In this case, the ordering function $\mathrm{rank}(A_n, \boldsymbol{s}, i)$, $n \in [N]$, $i \in [L]$, specifies as output the rank under order A_n of the item s_i located at position i within the sequence $\boldsymbol{s} = [s_1, s_2, \ldots, s_L]$. The ordered alphabet $A_n = [\pi_1^{(n)}, \pi_2^{(n)}, \ldots, \pi_R^{(n)}]$ is a permutation of the original repertoire, and N is the number of output neurons, equal to the number of representations of the same sequence in different permuted orders. We implement the rank function $\mathrm{rank}(A_n, \boldsymbol{s}, i) = 1/r$ as the inverse of the rank r for a particular index i, which can be obtained easily with the `argsort()` function in the C, MATLAB, or python languages.

The equations of the neurons Y sensitive to ordinal information in a sequence are as follows. The neurons' output Y is computed by forming the dot product between the ordering function $\mathrm{rank}(A_n, \boldsymbol{s}, i)$ and the synaptic weights $w_i \in \{1/r\}_{r=1}^{L}$, $i \in [L]$. For an input sequence of L items taken in the repertoire of cardinality R and for a population of N ordinal neurons, we have:

$$Y^{(n)} = \sum_{i=1}^{L} \mathrm{rank}(A_n, \boldsymbol{s}, i)\, w_i^{(n)}, \quad n \in [N]. \tag{1}$$

The updating rule of the weights is that of Kohonen networks (Kohonen, 1982) with a learning rate α fixed to 1.0 for one-shot learning, for neuron $Y^{(n)}$, we have:

$$\Delta w^{(n)} \quad = \quad \alpha(\mathrm{rank}(A_n, \boldsymbol{s}) - w^{(n)}). \tag{2}$$

Thus after complete learning, the weights $w^{(n)} = \mathrm{rank}(A_n, \boldsymbol{s})$ and the neuron's output becomes maximal, $Y^{(n)} = Y_{\max} = \sum_{r=1}^{L} r^{-2}$ for our choice of rank function. Notice that this maximum depends only on the choice of rank function and the sequence length L.

The complete procedure for ordinal neurons including decoding is outlined in Algorithm 1. The decoding step 2 is iterated until a tentative solution satisfying the learned ranks of all N neurons is found. Then a global decision vote takes place in step 3. In case of quantizing neurons as in XP1, step 2 for reconciling the ranks is not needed, and iteration over k may be stopped early if the reconstructed value stabilizes.

5.3 Related Works

This approach exploiting information structure is original in Machine Learning and AI. However, some similar features can be found in current neural architectures inspired by Physics and Biology, such as the Diffusion Probabilistic Models, the Variational Auto-Encoder and the Modern Hopfield Networks (Ramsauer et al., 2021; Millidge et al., 2022), or by Computer architecture using discrete codes as neural addresses, such as the Sparse Distributed Memory (Kanerva, 1988; Bricken and Pehlevan, 2021; Pourcel et al., 2022) or others (Graves et al., 2014; Träuble et al., 2022). We report a comparison of computational features and pros and cons in Section 5.3.

Furthermore, it is noteworthy that random matrices have been exploited successfully already in the last decades for fast and accurate sampling and reconstruction in Telecommunication (Berrou et al., 1993; Guizzo, 2004) and in Sensing (Candes et al., 2005; Donoho, 2006). They are now considered as standard methods for optimal codes.

5.3.1 Link with Diffusion Probabilistic Models and Variational Auto-Encoders

Variational Auto-Encoders– Variational Auto-Encoders allow statistical inference such as inferring the value of one random variable from another random variable (Kingma and Welling, 2014). They are meant to map the input variable to a multivariate latent distribution.

In the mathematical expression of VAE neurons, the mean and variance parameters of Gaussian functions are in the place of the synaptic weight values to be optimized. Using the so-called reparametrization trick, the randomness variable ε is injected into the latent space z as external input in VAE. In this way, it is possible to backpropagate the gradient without involving stochastic variables during the update.

In comparison, our approach quantizes information by removing redundancy directly, in one-shot, without regression, by selecting the desired uncertainty level. In effect, it creates large interval bins that correspond with the uncertainty margin of Gaussian functions (mean and variance). The neurons with random distribution can represent the missing value by intersecting their belief votes within their respective interval range.

Diffusion Probabilistic models– In thermodynamics, diffusion refers to the flow of particles from high-density regions towards low-density regions. In Machine Learning, this is done by gradually adding noise to input (Sohl-Dickstein et al., 2015; Ho et al., 2020). The reverse process generates

Algorithm 1 Pseudo-code of the algorithm

```
s = [s_1, s_2, ..., s_L],                          ▷ a sequence of L items,
item s_i ∈ [R] = {1, 2, ... R}                     ▷ items randomly selected
neurons n ∈ [N]                                    ▷ neural population of N neurons
random alphabets A = [A_1, A_2, ..., A_N],         ▷ of cardinality R
original alphabet A_0 = [1, 2, ... R]

s_k = A_k[s], k ∈ [N]                              ▷ sequence s in the new alphabet A_k

#1 encoding, one-shot learning for demonstration purpose
for k = 1, 2, ..., N do                            ▷ for each neuron k
    W_k = rank(A_k, s_k)                           ▷ learn the relative ordinal code
end for

#2 decoding, similar to a Hill-Climbing gradient error
for k = 1, 2, ..., N do                            ▷ for each neuron k
    initialize Err_k, Err_bak,
    s_bak = s_noise                                ▷ with s_noise ∈ [R]^L
    while Err_k ≠ 0 do
        s'_k = s_bak + s_noise                     ▷ with s_noise ∈ [R]^L
        Y^(k) = Σ rank(A_k, s'_k) W_k,
        Err_k = (Y^max − Y^(k))^2
        if Err_k ≤ Err_bak then                    ▷ keep values
            s_bak = s'_k
            Err_bak = Err_k
        end if
    end while s_k = s_bak
end for

#3 global decision, similar to a Gaussian Mixture Model
initialize σ, s'
for i = 1, 2, ..., L do
    initialize cumul_sum[i, j] = 0, ∀j ∈ [R]
    for k = 1, 2, ..., N do
        initialize μ = s'_k[i], τ = (π^(k))^(−1)   ▷ inverse permutation τ
        for j = 1, 2, ..., R do                    ▷ or j in a range around μ
            G(j) = 1/(σ√2π) e^(−(j−μ)^2/2σ^2)      ▷ in alphabet A_k
            cumul_sum[i, τ(j)] += G(τ(j))          ▷ in alphabet A_0
        end for
    end for
    s'[i] = argmax(cumul_sum[i, :])                ▷ return max item
end for
return s'
```

data by denoising. In the context of statistics, DPM are modeling energy gradients directly, along the entire diffusion process, which can take a large number of iterations.

In comparison, our method generates Gaussian random distributions from input by combining the shuffling and quantization operations. Quantization reduces the certainty level of one random variable to model priors (mean value). Each individual neuron learns a randomized version of the original sequence X by discretizing it, yielding the weights learned by that neuron.

Similar to VAE, each item in the sequence is encoded also separately as a latent vector; i.e. the vector of weights of synapse i of every neuron. Thus, the larger the number of neurons used to encode one item, the more precise its reconstruction is.

5.3.2 Link with Sparse Distributed Memory

A similarity exists between our approach and the Sparse Distributed Memory (SDM) architecture proposed by Pentti Kanerva (Kanerva, 1988) and recently investigated by several teams (Bricken and Pehlevan, 2021; Pourcel et al., 2022). SDM has been reintroduced recently for its analogy with a computer-like memory content retrieval based on addresses. Addresses are high-dimensional random binary vectors that separate memory patterns from each other.

The Dynamic SDM (DSDM) proposed by Vu and colleagues (Pourcel et al., 2022) modifies the SDM architecture to make the addresses data-driven and dynamically learnt. This work permits the challenging scenario of continual learning under an online, completely task-free and class-incremental (data incremental) setting, where learning and evaluating can be carried out at any point of time.

The variant SDMLP (Bricken and Pehlevan, 2021) aims to reduce catastrophic forgetting by using a Multi-Layered Perceptron (MLP) with mechanisms derived from the SDM model. The first mechanism is the utilization of the Top-k activation function, which means using only the k most active neurons of a layer in each learning step. This choice permits to have neurons specialized in some tasks, while others are free to learn other tasks. This mechanism reduces the chances for a neuron to be overwritten during the learning phase of another task, and thus to reduce catastrophic forgetting.

In comparison to our model, the quantized vectors extracted from the memory sequence and encoded into the synaptic weights play the same role as the random binary vectors used in the SDMs to allocate memory addresses. The SDM architectures use the Hamming distance for selection of the closest neurons for categorization, we use instead an Euclidean metric based on the Gaussian function, and centered on the mean value of the neuron output, to deliver a belief vote. Although very similar, this approach is more compatible with the Bayesian treatment of information of Gaussian mixture models for inference.

5.3.3 Link with Modern Hopfield Networks

Our approach has many similarities with the Modern Hopfield Networks (MHN) (Krotov and Hopfield, 2016; Demircigil et al., 2017; Krotov and Hopfield, 2021). The MHN version of 2016 exploits a dense binary weight matrix to encode data. A polynomial interaction function between neurons is proposed to update the value, which has a nonlinear effect on the decision making process.

This new version of the Hopfield network has showed many advantages in terms of reconstruction, robustness against noise, memory preservation against catastrophic forgetting and rapid convergence and stability. Moreover, a new exponential interaction function has been introduced, and a theoretical result demonstrated it to achieve the maximum capacity limit (Demircigil et al., 2017). A recent version of it has been developed for encoding continuous values (Krotov and Hopfield, 2021).

In comparison, our approach provides two parameters, the level of random permutation and discreteness of the synaptic weights, to describe the capacity limit of a given neural network. These parameters modulate directly the degree of redundancy or efficacy of the neural codes. In line with Information Theory, we show that the capacity limit of a neural network depends then on its number of neurons N and the resolution of its synaptic weights R_W, but also on the resolution of the input R_X, or its repertoire size. For the case of binary weights, we have $N = \log R_X$, the minimal number of (one-input) neurons required to encode one value $X \in R_X$.

The reconstruction phase in MHN uses polynomial and exponential interaction functions to retrieve the store information. Besides, in our case, the reconstruction phase exploits Gaussian functions for the interaction between neurons to deliver a belief vote. It corresponds also to a decision making process compatible with Bayesian inference.

MHN makes a distinction between discrete and continuous values. Instead, Information Theory treats information uniformly for the two cases, with the quantization of information dependent on their resolution. Similarly, we do not make any separation between the discrete and continuous cases to encode information in our neural network. That is, a combination of discrete neurons of low entropy can encode information at finer resolution and higher entropy.