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

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a bio-inspired concept based on the maximization of entropy in neural networks for memory storage and high-order cognitive skills. We emphasize the role of information structure to cut into smaller pieces high resolution inputs into extremely low resolution neurons. Despite the unreliability of neurons due to intrisic 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 exploits efficiently 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 use a different strategy with artificial neurons designed with virtually infinite precision of their weights, unlimited time and gigantic resources (data, and GPU) and energetical cost.

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 unprecise neurons of low resolution R_W to reconstruct back highly precise input X of much higher resolution R_X, with the relationships $R_W \ll R_X$. Using Information Theory (IT) and the source coding theorem of Claude Shannon, we demonstrate that unreliable neurons can maximize their codes to reach Shannon limits in terms of information capacity so that each neuron added to the neural population augments its memory capacity following an exponential scale.

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.

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. This computational treatment of information is similar with the binary codes done in digital processing.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

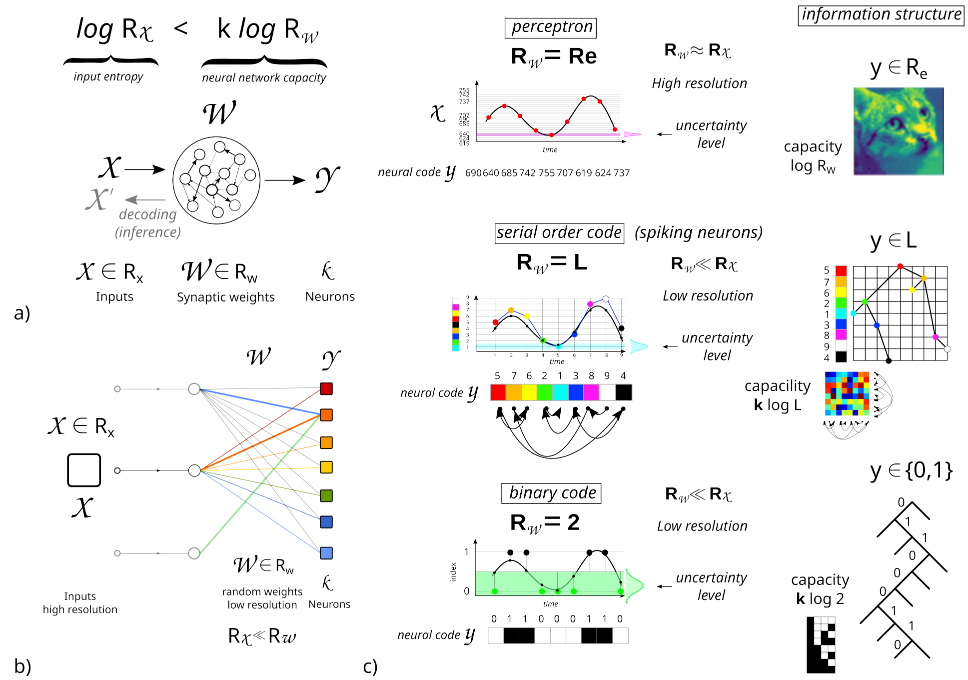

Figure 1: Entropy Maximization principle in neural networks. a) In Information Theory, a neural network can be seen as a network of neurons, a communication channel through which information can pass through. Following this, a high dimensional inputs X of resolution R_X, and entropy $logR_X$ can be conveyed into neurons Y of much lower weights precision W and synaptic resolution R_W. b) The minimal number k of neurons Y necessary to encode input X without loss is given by the Shannon's souce coding theorem. This number depends of the neurons' information structure, which means its resolution R_W. Accordingly, efficient codes can be constructed with a minimal number of neurons, and 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 pattern. For synaptic weights of high resolution, $R_W \approx R_X \approx Re$, 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 X. These codes are often used in symbolic processing. Although discrete codes perform a harsh quantization of the input X, they are faster to compute and to reconstruct the input X.

These two mechanisms, randomness and compacity, 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 computation in the brain.

We present several examples in which these random and compact neurons are used to represent structurally organized 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 these 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. [2001a], Van Rullen and Thorpe [2002].

In this paper, we present in section 2 our motivation and the principle used to describe the architecture. In section 3, we present the results. We conclude then with links with brain computation, cognitive development, and links between current machine learning and bio-inspired neural models.

2 Methods

Neurons Resolution and Patterns. We present in the Appendix in section 5.2 the two stages of the algorithm and its pseudo-code 1. The first stage consists on the encoding part of the signal X into the neural weights W, shuffled and quantized. This can be done in one-shot learning for a specified number of neurons k.

Shuffling means permutating randomly 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 the neuron. Each neuron will have its own specific randomly shuffled repertoire so that one value in $X \in R_X$ will correspond to a different value $W \in R_W$. The neural codes (codeword), representing input X, will take its values within the repertoire R_W, the cardinality of the neurons, which is also the number of possible states taken in the weights matrix W.

For $R_W = 2$, the neural codes will be only binary values $[0; 1]$ and the memory capacity of the neural network will be equal to $k \log 2$. Binary vectors have interesting properties to encode hierarchical trees and symbolic patterns.

For the special case where the resolution in the synaptic weights R_W is equal to the sequence length L of the input X, 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.

Ordinal codes can be seen as random permutation of cardinality L. Interestingly, they have been found in the frontal cortex to process serial order information Pitti et al. [2022a]. Ordinal codes have interesting algebraic and combinatorics features for manipulating context-free grammars, and to represent trees and directed graphes Pitti et al. [2022a]. The memory capacity of the ordinal neural are equal to $k \log L$.

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 with the Expectation-Maximization algorithm or the Bolztmann machine mechanism, found also in predictive coding and active inference Rao and Ballard [1999], Friston et al. [2016].

At each step, the neurons make a decision vote to predict the most probable values, following a gaussian distribution centered on the most probable guess. These decision votes are then summed up, and revised for the next step.

The harsh quantization in W create large errors in the decision votes at the unit level, and large uncertainty intervals. However, the decision vote with multiple neurons permits to discriminate the candidate values and to retrieve the higher resolution of the original signal. Accordingly, the random connections don't affect the belief vote, but create sparse and orthogonal representations such that only the least common candidates denominators among the neurons survive, see section 3.

This error-corrrective treatment of information is similar with bayesian inference. The neurons provide a conditional output relative to its likelihood to the class or to the input: $q(x/z) \approx p(z/x)p(x)$. 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 large. The random vectors act upon the belief vote so that only the least common denominators encoded differently by the neurons will win the votes.

3 Experiments

3.1 XP 1: Input Reconstruction for different neural resolutions

We devise 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 resolution of the synaptic weights W: $R_W = \{2, 10\}$ for binary and ten-level quantization.

Fig. 2 shows the results for the resolution $R_W = 2$ only, binary neurons. 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).

The convergence to the original image resolution $R_X = 256$ is achieved rapidly by selecting 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 taken for computing global decision, and stabilizes to a global minimum when a certain number is taken. In line with the source coding theorem, there is a threshold to information capacity and a minimal number of neurons k are necessary to allow encoding. Nonetheless, the relationships among the neural codes are highly nonlinear due to randomization, so a small number is enough to reconstruct back original information. The reconstruction process has some similarities to Diffusion Probabilistic models. In comparison, it is faster to converge as it requires a dizain of steps, see Fig. 2 a-b). However, the iterative process of the decision vote is also highly nonlinear.

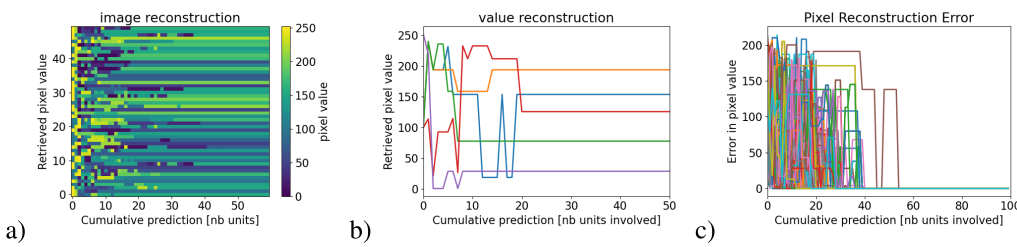

Figure 2: Reconstruction stage. a) cumulative prediction error for pixels' value decision making, by adding one by one neurons. b) retrieved pixels' value, for 5 values only. c) reconstruction error.

The Fig. 3 a-b) on the left side corresponds to the encoding of one selected neuron only for a specific 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 back 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 and redundancy suppression, the number of neurons required diminishes by a power-law scale.

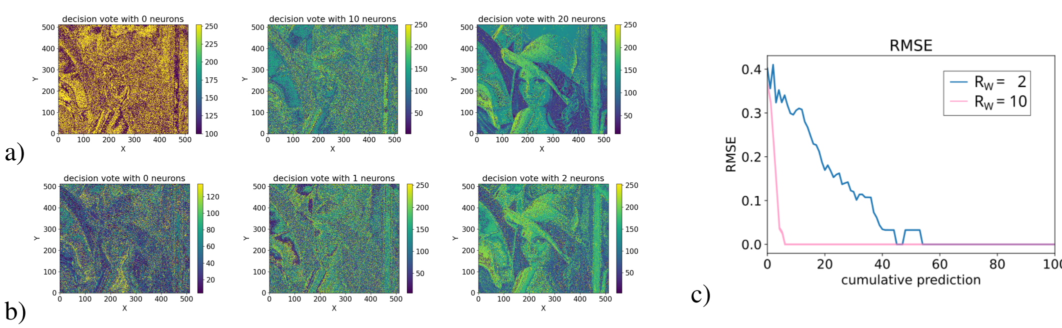

Figure 3: Computational efficiency for different information structure. Image reconstruction with neurons' weight resolution $R_W = \{2, 10\}$. a) Reconstruction for $k = \{1, 10, 20\}$ neurons with weights $R_W = 2$. b) Reconstruction for $k = \{1, 2, 3\}$ neurons with weights resolution $R_W = 10$. c) Error reconstruction related to information structure and neurons' entropy capacity.

3.2 XP 2: serial order planning in Hanoi Tower game

The Towers 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 in 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 of the few examples learned (learning-to-learn).

The states can be represented as nodes in a self-similar graph. Using the graph representation, sequences of discrete states can be constructed and analyzed. The experiment presents the most

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.

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**

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), and versions with the number of states of up to 3125 (5 rods and 5 disks) were also 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 a set of rules, that govern 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 a ratio between the ground truth shortest path with the length of the generated path. 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. 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 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 for 200.000 for the last case is sub-optimal – path generated 50% longer than in the other cases.

Our approach initialized with example sequences generated using a random walk had the best performance, and it was better than the version that uses 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 the 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.

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 [2012, 2001]. Accordingly, although the computational performances of noisy neurons are poor, the interaction of few neurons can increase their performance exponentially, as entropy follows a power-law scale.

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 distribution (randomization). By doing so, neurons have a different information structure from raw input. We emphasize here 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. [2001b]. 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.

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, 2001a.

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

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 predictible (because it could have been infered otherwise) and resemble more to a random signal Atick and Redlich [1992], Olshausen and Lewicki [2013]. It follows that more information can be stored for the same capacity limit within memory.

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 neurons 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 suppresses as well redundancy, and produces discrete neural codes W of 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 each neural code of resolution R_W is not capable to represent completely the original information of higher entropy R_X, we show that only a few number 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 will show that neural networks initialized with random vectors can convey maximal information, and reach out the Shannon's limit in terms of capacity with the equation $\log R_X \approx k \log R_W$, with k the number of neural units.

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 instanciates a new type of neural model, a digital neural network.

5.2 Neural codes implementation

The coding strategy consists of discretizing the items in the sequence in a given repertoire or alphabet of cardinality R_W.

When $R_W = L$, with L the length of the input sequence, theneural code corresponds to an ordinal code, sensitive to the serial order of the elements resent 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; $w_i \in [0, 1]$, $i \in [L]$. 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 the Kohonen networks Kohonen [1982] with a learning rate α fixed to 1.0 for one-shot learning, for the 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, $Y^{(n)} = Y_{\max} = \sum_{r=1}^{L} \frac{1}{r^2}$ for our choice of rank function. Notice that this maximum depends only on the choice of rank function and the sequence length L.

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 the Computer architecture are also , 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 variable 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 distrivution 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 generate data by denoising. In the context of statistics, DPM are modeling energy gradient directly, along entire diffusion process, which can take large number of iterations.

In comparison, our method generates gaussian random distribution from input by combining the shuffling and quantizing operations. Quantization reduces the certainty level of one random variable to model priors (mean value).Each individual neuron learns a random permuted order of the original sequence X corresponding to a discrete version of it, a codeword; i.e., the horizontal red row in the weight matrix W.

Similar with VAE, each item in the sequence is encoded then separately as a latent vector; i.e., the vertical green column in the weight matrix. Thus, the larger the number of neurons used to encode one item, the more precise the reconstruction is.

Algorithm 1 Pseudo-code of the algorithm

$\boldsymbol{s} = [item_1, item_2, \ldots, item_L]$, ▷ a sequence of L items,
items $\in [R] = \{1, 2, \ldots R\}$ ▷ items randomly selected
neurons $\in [N]$ ▷ neural population of N neurons
random alphabets $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_N]$, ▷ of cardinality R
original alphabet $\boldsymbol{A_0} = [1, 2, \ldots R]$

$\boldsymbol{s}_k = \boldsymbol{A}_k[\boldsymbol{s}]$, $k \in [N]$ ▷ sequence $\boldsymbol{s}$ in the new alphabet $\boldsymbol{A}_k$

#1 encoding, one-shot learning for demonstration purpose
for $k = 1, 2, \ldots, N$ **do** ▷ for each neuron k
  W_k = rank($\boldsymbol{A}_k$, $\boldsymbol{s}_k$) ▷ learn the relative ordinal code
end for

#2 decoding, similar with a Hill-Climbing gradient error
for $k = 1, 2, \ldots, N$ **do** ▷ for each neuron k
  initialize Err_k, Err_bak,
  $\boldsymbol{s}\_bak = \boldsymbol{s}\_noise$ ▷ with $\boldsymbol{s}\_noise \in [R]^L$
  while $Err_k \neq 0$ **do**
    $\boldsymbol{s}'_k = \boldsymbol{s}\_bak + \boldsymbol{s}\_noise$ ▷ with $\boldsymbol{s}\_noise \in [R]^L$
    $Y^{(k)} = \sum \text{rank}(\boldsymbol{A}_k, \boldsymbol{s}'_k)\, W_k$,
    $Err_k = (Y^{max} - Y^{(k)})^2$
    if $Err_k \leq Err\_bak$ **then** ▷ keep values
      $\boldsymbol{s}\_bak = \boldsymbol{s}'_k$
      $Err\_bak = Err\_k$
    end if
  end while $\boldsymbol{s}_k = \boldsymbol{s}\_bak$
end for

#3 global decision, similar to a Gaussian Mixture Model
initialize σ, $\boldsymbol{S}'$
for $i = 1, 2, \ldots, L$ **do**
  initialize $cumul\_sum[i, j] = 0, \forall j \in [R]$
  for $k = 1, 2, \ldots, N$ **do**
    initialize $\mu = \boldsymbol{s}'_k[i]$
    for $j = 1, 2, \ldots, R$ **do** ▷ or j in a range around μ
      $G(\pi_j^{(k)}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(j-\mu)^2/2\sigma^2}$ ▷ in alphabet $\boldsymbol{A}_k$
      $cumul\_sum[i, j] += G(j)$ ▷ in alphabet $\boldsymbol{A}_0$
    end for
  end for
  $\boldsymbol{S}'[i] = argmax(cumul\_sum[i, :])$ ▷ return max item
end for
return $\boldsymbol{S}'$

5.3.2 link with Sparse Distributed Memory and Modern Hopfield Networks

A similarity exists between our approach and Sparse Distributed Memory 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 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 utilisation 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, where other 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's version of 2016 exploits a dense and binary weight matrix to encode data. A polynomial interactive 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, for robustness against noise, memory preservation against catastrophic forgetting and rapid convergence and stability. Moreover, a new interactive function has been introduced using an exponential one, and theoretical result has been demonstrated to achieve maximum capacity limit Demircigil et al. [2017]. A recent version of it has been developped 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, the resolution of its synaptic weights R_W but also, the resolution of the input R_X, or its repertoire size. For the case of binary neurons, we have $N = logR_X/log2$, the minimal number of neurons required to encode one value $X \in R_X$.

The reconstruction phase in MHN use polynomial and exponential interactive 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 to their resolution. Similary, we don't 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 of bigger resolution and high entropy.