INVESTIGATING HOPFIELD NETWORKS ON GRAPHS: LEARNING INVARIANCE AND STORING ORBITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building on prior work focused on the clique and hyperclique sub-cases, we investigate the capacity of classical Hopfield networks for storing the orbit of a graph under graph isomorphism. Our key observation is that the orbits of many natural classes of graphs can be efficiently stored in a Hopfield network by minimizing a convex objective, called the Energy Flow. Moreover, only a vanishingly small fraction of examples from the orbit are required for the Hopfield network to strictly memorize the entire orbit. We remark that this phenomenon does not appear to hold for modern Hopfield networks.

1 INTRODUCTION

Classical discrete recurrent neural networks McCulloch & Pitts (1943); Amari (1972); Hopfield (1982) are network architectures that are primarily used for auto-associative memory. They consist of symmetrically connected binary threshold units and map an input bit string to an output bit string by locally minimizing an energy function that encodes certain binary patterns as attractors Hopfield (1982). Albeit widely celebrated as a canonical model for biological computation and auto-associative memory, a practical drawback of networks such as Hopfield's are their limited storage capacity for general datasets. For example, given mutually i.i.d. bits, each with probability of being one set as $1/2$ and organized into N bit strings of length n, then to store all N bit strings with high probability we require N linear in n McEliece et al. (1987). However, for more structured data sets the outlook is far more positive. In particular, Hillar & Tran (2018); Hillar et al. (2021) proved that these recurrent neural networks can achieve robust, exponential storage with respect to storing all (hyper)cliques of a certain size and edge count. Moreover, these works also empirically highlight that minimizing the energy flow (MEF) Hillar et al. (2012) defined on a vanishingly small number of examples suffices to memorize all cliques. Other relevant works include Burns & Fukai (2023) on generalizing Hopfield networks to simplicial complexes.

Extending this line of investigation, a natural direction of research is to explore the ability of networks to memorize the graph isomorphism orbit of a graph given only a few examples. Key questions in this regard include for which types of graph is this possible, what is the critical sampling ratio, and how does invariance emerge over training? Given the brevity of this workshop format and the preliminary nature of our investigations thus far, our focus here is on presenting interesting observations and connections relating to these questions, rather than detailed answers. We observe empirically that the orbit of many types of graphs, both random and highly symmetric, can be memorized by training on just a few examples. Second, we observe that the parameters at the end of training are approximately graph isomorphism invariant: we conclude that an implicit bias towards invariance to the structure in the data must emerge during training. Third and finally, we propose a new yet simple algorithm, based on our networks, for checking if two graphs are *not* isomorphic. We speculate that this approach may open up new approaches and ideas for algorithms for solving the graph isomorphism problem.

2 SETUP

We consider simple undirected binary graphs with v vertices. Each graph is encoded as binary vector $\boldsymbol{x} \in \{0,1\}^n$ where $x_j = 1$ iff the jth vertex pair is present as an edge and is 0 otherwise. A graph isomorphism is a permutation of the vertices which preserves edge adjacency, and the set of graphs

isomorphic to a graph $\boldsymbol{x}$ we denote as $\text{Orb}(\boldsymbol{x})$. Let $Sym_0(n) \subset \mathbb{R}^{n\times n}$ denote the set of symmetric, real $n \times n$ matrices with zeros on the diagonal and $\Theta = Sym_0(n) \times \mathbb{R}^n$. We consider classical Hopfield networks equipped with the energy function $E : \{0,1\}^n \times \Theta \to \mathbb{R}$ defined as

$$E(\boldsymbol{x};\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{h}^T\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{h}) \in \Theta$. Strictly speaking, our networks are different from Hopfield networks since they have learnable weights rather than fixed ("one-shot") ones Hopfield (1982), but for expositional simplicity we shall call them Hopfield networks in what follows.

The input-output map of a Hopfield network with parameters $\boldsymbol{\theta}$ we denote as $H(\boldsymbol{x};\boldsymbol{\theta}) : \{0,1\}^n \times \Theta \to \{0,1\}^n$, where $n = \binom{v}{2}$ and v is the number of vertices. If $H(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{x}$ then $\boldsymbol{x}$ is a fixed point of the recurrent dynamics that define the input-output map; furthermore, we say under this condition that H has *memorized* $\boldsymbol{x}$. A sufficient but not necessary condition for H to memorize $\boldsymbol{x}$ is that $E(\boldsymbol{x};\boldsymbol{\theta}) < E(\boldsymbol{x}';\boldsymbol{\theta})$ for all $\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x})$, where $\mathcal{N}(\boldsymbol{x})$ denotes the set of all binary vectors a hamming distance exactly one away from $\boldsymbol{x}$. If $\boldsymbol{\theta}$ satisfies this property we say that it *strictly memorizes* $\boldsymbol{x}$. Given a vector $\boldsymbol{x} \in \{0,1\}^n$ we consider a dataset $T \subset \text{Orb}(\boldsymbol{x})$. To learn a Hopfield network which memorizes T, we minimize an energy flow objective (MEF) Hillar et al. (2012); Hillar & Tran (2018); Hillar et al. (2021),

$$L(\boldsymbol{\theta};T) := \frac{1}{n|T|}\sum_{\boldsymbol{x}\in T}\sum_{\boldsymbol{x}'\in\mathcal{N}(\boldsymbol{x})}\exp(E(\boldsymbol{x};\boldsymbol{\theta}) - E(\boldsymbol{x}';\boldsymbol{\theta})). \tag{2}$$

In our experiments we use L-BFGS Nocedal (1980); Liu & Nocedal (1989) to minimize equation 2. This differentiable objective has several desirable properties, such as convexity and small size; we refer to Hillar et al. (2021) for more details, including comparisons to other learning approaches.

Finally, we also consider Hopfield networks which are invariant to graph isomorphism. Let $\mathcal{Q}_n$ denote the set of $n \times n$ permutation matrices acting on the vertex pairs which correspond to graph isomorphisms. Defining the action on the parameters as $\boldsymbol{Q}\boldsymbol{\theta} = (\boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{Q}, \boldsymbol{Q}^T\boldsymbol{h})$, then $\boldsymbol{\theta} \in \Theta$ is graph isomorphism invariant if for all $\boldsymbol{Q} \in \mathcal{Q}_n$ we have $\boldsymbol{Q}\boldsymbol{\theta} = \boldsymbol{\theta}$. This in turn implies

$$E(\boldsymbol{Q}\boldsymbol{x};\boldsymbol{\theta}) = E(\boldsymbol{x};\boldsymbol{Q}\boldsymbol{\theta}) = E(\boldsymbol{x};\boldsymbol{\theta}).$$

As per (Hillar & Tran, 2018, Section 5.1), the subset of graph isomorphism invariant parameters forms a three dimensional subspace of Θ. In particular, $\boldsymbol{\theta} \in \Theta$ is invariant if and only if there exist scalars x, y, z such that $h_j = z$ for all $j \in [n]$, $w_{ij} = x$ if edge i is adjacent to edge j (i.e. shares exactly one vertex) and $w_{ij} = y$ if i is not adjacent to j (for all $i, j \in [n]$). To indicate that $\boldsymbol{\theta}$ is graph isomorphism invariant, we write it as a function of these three parameters, $\tilde{\boldsymbol{\theta}}(x, y, z)$.

3 Empirical results

Fix a specific graph, which has an associated collection of isomorphic graphs. A single training set in our experiments consists of a subset of graphs in this collection, drawn uniformly at random. In that case, the score is the total number of bits that are different between the graphs in this train set and their attractors under the network dynamics. A score of zero means that each training sample is a fixed point of the dynamics, i.e. a memory of the network. In the case of the test set, the score is the same as setting the train set to be precisely all graphs that are isomorphic to the fixed graph.

Our key observations are as follows.

- **Observation 1:** as per Figures 1 and 2, Hopfield networks appear to be able to memorize the orbits of a wide variety of graphs, including both irregular, random graphs, e.g., Erdos-Renyi, as well as regular graphs, for example Paley, Johnson and Circulant. The capacity of Hopfield networks for storing group structured data sets (at least in many cases) therefore greatly exceeds the linear constraint for random data.
- **Observation 2:** again considering Figures 1 and 2, across this range of graph types, only a fraction of the orbit needs to be fitted for the the full orbit to be memorized. We speculate that the critical ratio is of the order n and leave a thorough analysis of this to future work.

- **Observation 3:** turning our attention to Figure 3, we observe that the weight matrices of the Hopfield networks trained across all graph types exhibit the same pattern. In fact, these learned parameters lie very close to the three dimensional subspace corresponding to the set of graph isomorphism invariant parameters (see Hillar & Tran (2018) for further details on the characterization of this set).
- **Observation 4:** as per Figure 4 dense associative memory techniques (DAM), for example "Modern Hopfield networks" Krotov & Hopfield (2016); Demircigil et al. (2017), do not appear to demonstrate the same ability as MEF-trained classical Hopfield networks for generalizing or memorizing to the full orbit given only a few examples. Further investigation is naturally warranted.

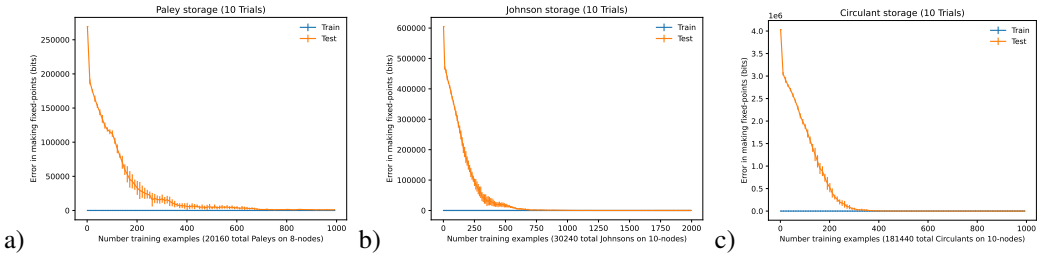

Figure 1: **MEF Accuracy versus number of training samples**. a) Paley graphs on 10 vertices (20160 isomorphic graphs), b) Johnson graphs on 10 vertices (30240 isomorphic graphs), c) Circulant graphs on 10 vertices (181440 isomorphic graphs). For this and Figures 2 and 4, the score on the y-axis is the number of bits different from the input set to its attractors under the network dynamics.

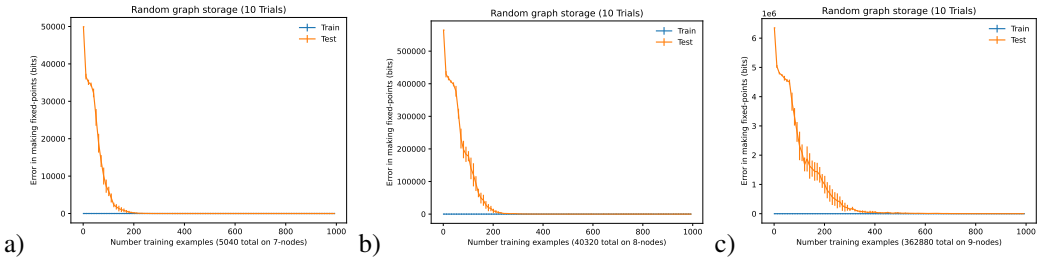

Figure 2: **MEF Accuracy versus number of training samples**. a-c) Three different random Erdos-Renyi graph on $7, 8, 9$ vertices having isomorphism class sizes $5040, 40320, 362880$, respectively.

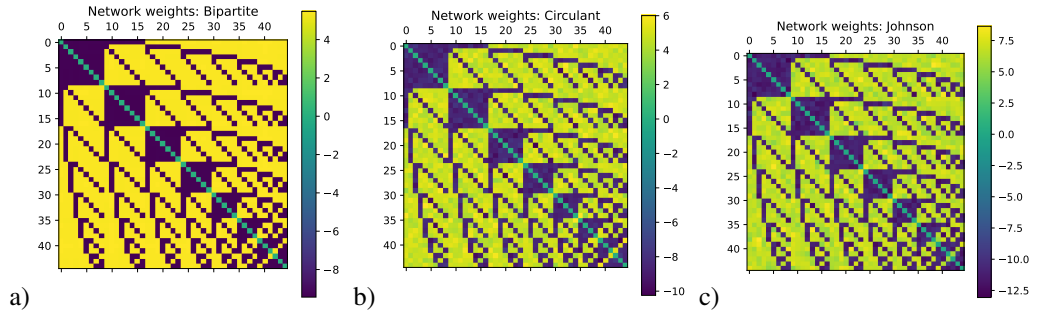

Figure 3: **Weight matrices**. We show Hopfield network parameters for examples of 10-node graphs trained with MEF on 1000 samples of the corresponding isomorphism class. a) Bipartite, b) Circulant, and c) Johnson graphs.

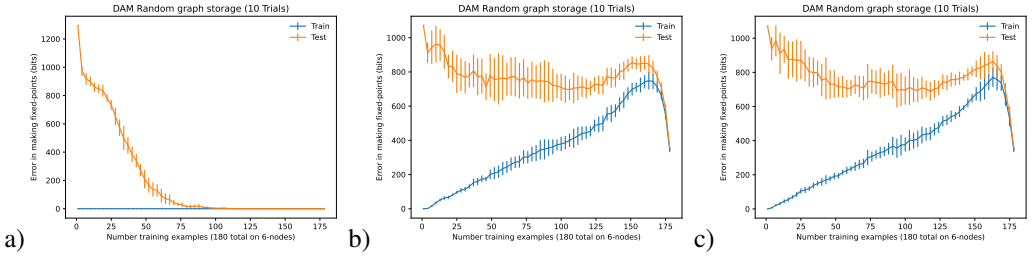

Figure 4: **Comparison with DAM Krotov & Hopfield (2016)**. a) MEF trained Erdos-Renyi random graph on 6 vertices (180 isomorphic graphs), b) DAM trained on same Erdos-Renyi random graph on 6 vertices using degree 3 activation function, c) Same as in b, but with degree 5 activation.

4 The Hopfield Network Not Graph Isomorphic Check (HNNGIC)

These observations prompt investigation into the potential for using Hopfield networks to check for graph isomorphisms, a fundamental and important problem in computer science. To this end we propose Algorithm 1, which we refer to as the *Hopfield Network Not Graph Isomorphic Check* (HNNGIC). As the name suggests, this algorithm provides a check if two graphs are not graph isomorphic, returning true in certain cases when they are not graph isomorphic and unknown otherwise.

Algorithm 1: Hopfield Network Not Graph Isomorphic Check (HNNGIC)

Input: two graphs $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \{0,1\}^n$ and computational budget B
Output: True or Unknown

Step 1: minimize $L(\hat{\boldsymbol{\theta}}(x, y, z); \boldsymbol{x}_1)$ within computational budget B, return (x^*, y^*, z^*).;
Step 2: **if** $L(\tilde{\boldsymbol{\theta}}(x^*, y^*, z^*); \boldsymbol{x}_1) < 1/n$ **then**
  if $H(\boldsymbol{x}_2; \tilde{\boldsymbol{\theta}}(x^*, y^*, z^*)) \neq \boldsymbol{x}_2$ **then**
    return True
  end
else
  return Unknown
end

The idea behind this algorithm is simple: first given two graphs we pick one, i.e., $\boldsymbol{x}_1$, arbitrarily at random. We then attempt to train the Hopfield network by minimizing the energy flow defined on this single graph, but restrict the parameters to lie on the graph isomorphism invariant subspace. If the resulting parameters (x^*, y^*, z^*) achieve a loss less than $1/n$ then by definition this implies for every $\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}_1)$ that $E(\boldsymbol{x}_1; \tilde{\boldsymbol{\theta}}(x^*, y^*, z^*)) < E(\boldsymbol{x}'; \tilde{\boldsymbol{\theta}}(x^*, y^*, z^*))$, therefore $\boldsymbol{x}_1$ is strictly memorized. Combining this with the fact that $\tilde{\boldsymbol{\theta}}(x^*, y^*, z^*)$ is graph isomorphism invariant, then this implies every point in the orbit of $\boldsymbol{x}$ under graph isomorphism is also strictly memorized, i.e., every graph isomorphism of $\boldsymbol{x}$ is stored as a memory in the Hopfield network. If the other graph $\boldsymbol{x}_2$ is graph isomorphic to $\boldsymbol{x}_1$, then it must be a fixed point of the associated input-output map. Therefore $\boldsymbol{x}_2$ and $\boldsymbol{x}_1$ cannot be graph isomorphic if $\boldsymbol{x}_2$ is not a fixed point of the invariant Hopfield network which stores $\boldsymbol{x}_1$. Note, just because $\boldsymbol{x}_2$ is a fixed point does not mean we can conclude that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are isomorphic. Indeed, there may be other fixed points not related to the orbit of $\boldsymbol{x}_1$; these are sometimes called "spurious states" in the landscape of attractors.

5 Discussion and Future Work

In this note, we highlight a number of intriguing capabilities of Hopfield networks with respect to their capacity for storing group structured data, an ability not observed in modern Hopfield networks. Our observations prompt a number of interesting and currently open questions: in particular, i) what graphs can be strictly memorized by graph isomorphism invariant networks, ii) can we characterize the fixed points of such networks, and iii) what is the critical ratio for generalization and how does it depend on graph structure in question? We believe answering such questions will prove valuable in

a wider machine learning context by shedding light, albeit in a simple setting, how group structure in data facilitates an escape from the curse of dimensionality, the emergence of invariance more generally, and its interaction with both implicit and explicit forms of regularization. In addition, we hope our findings stimulate thought and discussion towards new energy-based models, which combine the best parts of both classical and modern Hopfield networks.

REFERENCES

S-I Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206, 1972.

Thomas F Burns and Tomoki Fukai. Simplicial hopfield networks. *arXiv preprint arXiv:2305.05179*, 2023.

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*, 168:288–299, 2017.

Christopher Hillar, Jascha Sohl-Dickstein, and Kilian Koepsell. Efficient and optimal binary Hopfield associative memory storage using minimum probability flow. *In: 4th neural information processing systems (NIPS) workshop on discrete optimization in machine learning (DISCML): structure and scalability.*, pp. 1–6, 04 2012.

Christopher Hillar, Tenzin Chan, Rachel Taubman, and David Rolnick. Hidden hypergraphs, error-correcting codes, and critical learning in Hopfield networks. *Entropy*, 23(11):1494, 2021.

Christopher J Hillar and Ngoc M Tran. Robust exponential memory in Hopfield networks. *The Journal of Mathematical Neuroscience*, 8:1–20, 2018.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

Robert McEliece, Edward Posner, Eugene Rodemich, and Santosh Venkatesh. The capacity of the hopfield associative memory. *IEEE transactions on Information Theory*, 33(4):461–482, 1987.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.