

Hopfield Networks is All You Need

Andrew Pozzuoli

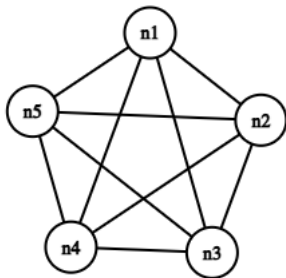
COSC 5P77

April 9, 2021

- 1 Hopfield Network
- 2 Modern Hopfield Network
 - Continuous States
 - Storage and Retrieval
 - Energy Function
 - New Update Rule
 - Convergence and Retrieval Error
 - Well-separated Patterns and Clusters
- 3 Connection to Attention
- 4 Other Features
- 5 Future Work: One Avenue
 - Immune Repertoire Classification
- 6 Conclusion
- 7 Bibliography

- Introduced in the 1982 paper, *Neural networks and physical systems with emergent collective computational abilities* by J. J. Hopfield [1].
- An early concept of a neural network that models itself off of neurobiology.
- A content-addressable associative memory mechanism capable of retrieving stored information from partial input.

Figure: Example Hopfield Network with 5 nodes



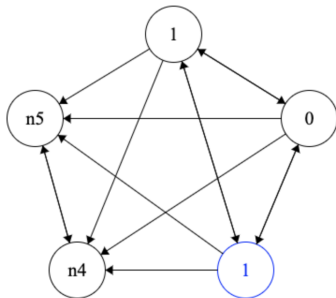
Stored patterns:

- $(1, 0, 1, 0, 0)$
- $(0, 0, 1, 0, 1)$
- $(1, 1, 0, 1, 0)$
- ...

Network trains on the stored patterns to adjust the weights between nodes.

Hopfield Network [1]: Example

Figure: Example Hopfield Network with 5 nodes



Query: (1, 0, 1, 1, 1)

Stored patterns:

- $*(1, 0, 1, 0, 0)^*$
- $(0, 0, 1, 0, 1)$
- $(1, 1, 0, 1, 0)$
- ...

The “1, 0, 1” of the query input causes the network to settle the nodes n4 and n5 into what is compatible with the trained weights via an update rule. In the example, pattern 1 is retrieved after a number of updates.

The 2021 conference paper, *Hopfield Networks is All You Need* introduces a modern Hopfield network that can be integrated into deep learning layers [2].

New features:

- Continuous states.
- New energy function.
- Corresponding new update rule equivalent to attention in transformers.
- Storage of exponentially many patterns.
- Exponentially small retrieval errors.

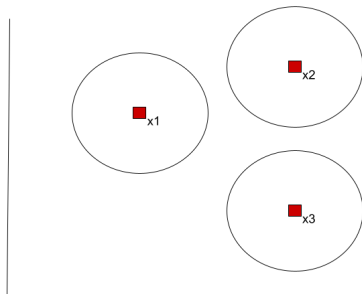
The modern Hopfield network [2] has been updated to store patterns with continuous states.

Binary Hopfield pattern eg.
(1, 0, 1, 0, 0)

Modern Hopfield pattern eg.
(4.56, 9.01, 0.08, 7.32, 5.65)

- The network is differentiable which is necessary for gradient descent in deep learning layers.
- Continuous states are stored as vectors.
- Queries are now vectors rather than partial binary patterns.

Figure: Example of pattern storage



- Around each pattern is a sphere.
- A pattern is stored if every query that falls inside the sphere converges to a single fixed point (for well separated patterns).
- A pattern is retrieved if after iterating the update rule, the function returns this fixed point with some exponentially small retrieval error.

- Vector patterns are represented as points in space.

The number of patterns, N , that the modern Hopfield network can store is given by the following equation:

Figure: Number of patterns [2].

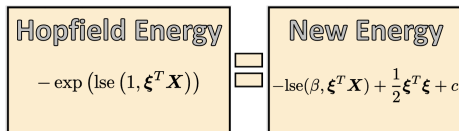
$$N \geq \sqrt{p} c^{\frac{d-1}{4}}$$

Important to note that the parameter d is in the exponent of the lower bound. Therefore, this Hopfield network can store exponentially many patterns with respect to the dimension.

Every Hopfield network is associated with some energy function.

- The goal is to minimize the energy function.
- As the network moves toward retrieving a stored pattern, the energy gets lower.

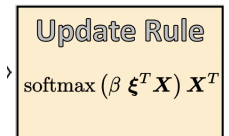
Figure: Hopfield energy function [2]



- X - The pattern stored in the network.
- ξ - The query pattern.
- β and c - Parameters that need to be tuned to the network.
- The term $\frac{1}{2}\xi^T \xi$ prevents the energy from going to infinity due to continuous states.

The paper presents a novel update rule to minimize the energy function.

Figure: Novel update rule [2]



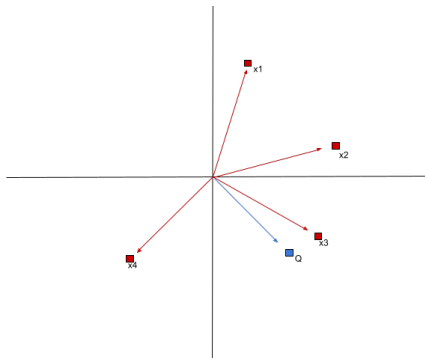
Update Rule

$$\text{softmax}(\beta \xi^T \mathbf{X}) \mathbf{X}^T$$

- \mathbf{X} - The pattern stored in the network.
- ξ - The query pattern.
- β - Parameter that needs to be tuned to the network.

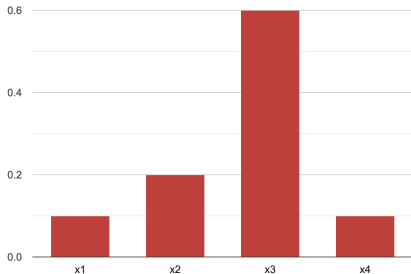
New Update Rule: Step-by-step

Figure: Example stored patterns and query vector



- $\beta \xi^T X$
- The update rule performs a dot product between the query and every stored pattern.
- Vectors that are closer to the query will yield a larger dot product.
- In this example, x_3 yields the largest dot product with Q .

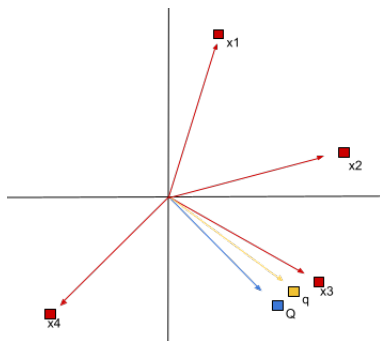
Figure: Example softmax distribution



- $\text{softmax}(\beta \xi^T \mathbf{X})$
- Perform a softmax on the dot products.
- This gives a distribution showing which stored patterns are closest to the query.

New Update Rule: Step-by-step

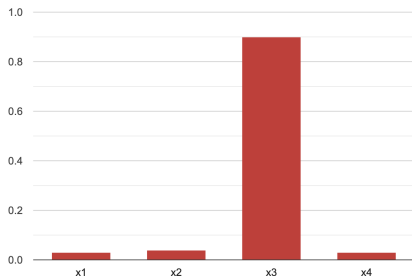
Figure: Result vector



- $\text{softmax}(\beta \xi^T \mathbf{X}) \mathbf{X}^T$
- Multiply each stored vector with the values of the softmax distribution.
- Resulting mapped point is a new vector closer to the stored pattern (q in this figure).
- This result vector becomes a new query vector and has the update rule applied to it.

New Update Rule: Step-by-step

Figure: Softmax on result query from first update



- The resulting softmax distribution heavily favours the stored pattern closest to the original query.
- With each update, the energy function lowers causing the network to move closer to convergence.

Figure: Proof of convergence with one update [2]

Proof. From Eq. (179) we have

$$\|J^m\|_2 \leq 2\beta N M^2 (N-1) \exp(-\beta (\Delta_i - 2 \max\{\|\xi - x_i\|, \|x_i^* - x_i\|\} M)) . \quad (394)$$

After every iteration the mapped point $f(\xi)$ is closer to the fixed point x_i^* than the original point x_i :

$$\|f(\xi) - x_i^*\| \leq \|J^m\|_2 \|\xi - x_i^*\| . \quad (395)$$

For given ϵ and sufficient large Δ_i , we have $\|f(\xi) - x_i^*\| < \epsilon$, since $\|J^m\|_2$ goes exponentially fast to zero with increasing Δ_i . \square

- With each iteration of the update rule, the error between the mapped point and the fixed point we want to retrieve goes to zero exponentially fast for well-separated patterns.
- The paper asserts that an exponentially small error is enough to say that the network converges in one update.

Figure: Well-separated patterns with fixed-point convergence

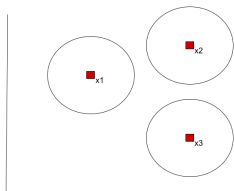
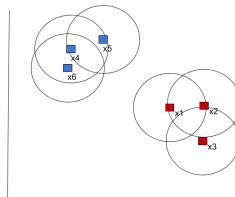
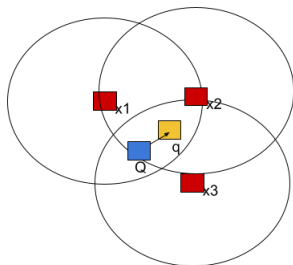


Figure: Clustered patterns with metastable states



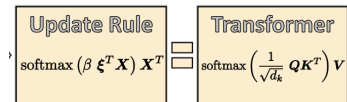
- Depending on the size of the spheres, patterns may not be well-separated and overlap.

Figure: Network settles on a metastable state



- If a query lies between overlapping spheres, the network settles on an average point between them.
- This is called a **metastable state** [2].

Figure: Novel update rule is equivalent to attention in transformers [2]



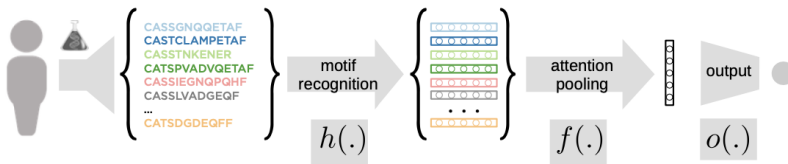
- New update rule happens to be the same as attention in transformers.
- The modern Hopfield network can be used to implement transformer and BERT models.
- When integrated into deep learning layers, this mechanism set state-of-the-art for 10 UCI benchmark collection datasets for small dataset classification.

In addition to attention, the modern Hopfield network provides other features that can be integrated in deep learning layers [2].

- Pooling operations, pattern search, and Long Short-Term Memory which are useful for multiple instance learning.
 - Outperformed competing methods in immune repertoire classification (AUC of 0.832 ± 0.022)
 - State-of-the-art performance on MIL benchmark datasets Tiger, Elephant, and outperformed on USCB Breast Cancer dataset.
- Multiple queries on a training set which is useful for support vector machines, k-nearest neighbour, and vector quantization.
 - Deep learning architectures with this achieved state-of-the-art on the drug design benchmarks for predicting side effects (0.672 ± 0.019) and for predicting β -secretase (0.902 ± 0.023).

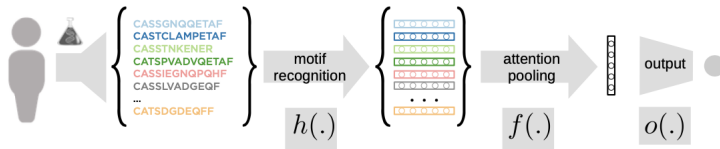
The paper *Modern Hopfield Networks and Attention for Immune Repertoire Classification* (Widrich, 2020) took advantage of the exponential storage capacity of the modern Hopfield network and applied it to Immune Repertoire Classification [3].

Figure: Architecture of DeepRC method for immune repertoire classification integrating modern Hopfield networks (Widrich et al., 2020) [3].



Immune repertoire classification is a multiple instance learning problem

Figure: Architecture of DeepRC method for immune repertoire classification integrating modern Hopfield networks (Widrich et al., 2020) [3]



- Given large bags of amino acid sequences that represent immunoreceptors with the task of classifying the immune status with respect to a particular disease.
- A very small fraction of receptors determines the immune status.
- Highly difficult problem since a single immune repertoire can have millions of sequences with very few indicators for classification.

Results

- DeepRC was a successful classifier of immune repertoire and, on average, outperformed all other methods for immune repertoire classification (average area under ROC curve (AUC) of 0.832 ± 0.022) [3].
- Further exploration into this problem could lead to faster immune testing based on blood samples for example.

The paper *Hopfield Networks is All You Need* introduced a modern Hopfield network with continuous states and a new update rule equivalent to attention in modern transformers [2].

- Exponential pattern storage and exponentially small retrieval error with one update.
- Can be integrated into deep learning layers.
- Provides attention, pooling, memory and association.
- State-of-the-art improvement on immune repertoire classification and three other multiple instance learning problems considered as well as on two drug design datasets.
- Outperformed other machine learning methods on small classification tasks in the UCI benchmark collections.

- [1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982, Publisher: National Academy of Sciences _eprint:
<https://www.pnas.org/content/79/8/2554.full.pdf>, ISSN: 0027-8424.
DOI: 10.1073/pnas.79.8.2554. [Online]. Available:
<https://www.pnas.org/content/79/8/2554>.
- [2] "Hopfield Networks is All You Need," en, p. 94, 2021.
- [3] M. Widrich, B. Schäfl, H. Ramsauer, M. Pavlović, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter, and G. Klambauer, "Modern Hopfield Networks and Attention for Immune Repertoire Classification," en, *arXiv:2007.13505 [cs, q-bio, stat]*, Jul. 2020, arXiv: 2007.13505. [Online]. Available:
<http://arxiv.org/abs/2007.13505> (visited on 03/25/2021).