

Redes de Hopfield

Bruno Mamede de Araújo Moura¹, Gustavo Henriques da Cunha¹, Vitor Hugo Palmié Peixoto¹

¹Universidade Federal de São João del-Rei (UFSJ)

brunommoura@aluno.ufsj.edu.br, gustavohenrique13579@aluno.ufsj.edu.br,
torugoppeixoto@aluno.ufsj.edu.br

Abstract

Este trabalho tem como objetivo o estudo da arquitetura de redes neurais conhecida como Redes de Hopfield, com o propósito de explicar e entender seu funcionamento. Além disso, busca testá-la em um problema específico, o problema do Caixeiro Viajante, e comparar seus resultados com diferentes parâmetros.

1 Introdução

Com o avanço da Inteligência Artificial na solução de diversos problemas na Ciência da Computação e em outras áreas, um campo de estudo de grande destaque é o das Redes Neurais Artificiais. Buscando simular a maneira como os neurônios funcionam no cérebro humano, os algoritmos conseguem analisar os dados fornecidos como treinamento e identificar um conjunto de parâmetros capaz de resolver novos problemas.

As Redes de Hopfield [Hopfield, 1982], popularizadas por James Hopfield, que lhes deu o nome, são uma das arquiteturas pioneiras no estudo das Redes Neurais Artificiais. A partir do estudo físico de como os neurônios funcionam, esse modelo é capaz de resolver problemas de satisfação de restrições, como o problema do caixeiro-viajante [Luger, 2005], ou problemas de recuperação de informação.

Neste trabalho, aplica-se uma variação do Modelo de Hopfield para o problema do Caixeiro Viajante, o modelo de Hopfield-Tank [Hopfield and Tank, 1985] [Hopfield and Tank, 1986], que se mostra mais eficiente em problemas de otimização.

Para o ajuste dos parâmetros, são realizados testes com as mesmas entradas de dados e diferentes parâmetros, buscando observar quais configuram os melhores resultados.

2 Redes de Hopfield

Uma Rede Neural Artificial, ao buscar abstrair o funcionamento do cérebro, trabalha com uma estrutura em grafos, composta por nós ou unidades que se conectam por ligações orientadas, as quais possuem um peso que representa a força e o sinal da conexão [Russell and Norvig, 2010].

Dentre as redes neurais, as Redes de Hopfield pertencem a uma classe chamada Redes Neurais Recorrentes. Elas funcionam conectando um nó a outro de tal maneira que os dados de

saída da rede são reutilizados como entradas, realimentando-a. Além disso, esse modelo opera com conexões bidirecionais com pesos simétricos [Russell and Norvig, 2010]. Assim, a Rede de Hopfield também pode ser classificada como uma Rede Neural de Memória Autoassociativa, pois, dado um estado inicial, a rede convergirá para um estado estável [Luger, 2005]. A função de energia é dada pela equação 1, onde a conexão é representada por T_{ij} , V_i é o nível de ativação do nó, e I_i é o *bias*.

$$E = \frac{-1}{2} \sum_{i=1}^N \sum_{j=1}^N T_{ij} V_i V_j - \sum_{i=1}^N V_i I_i. \quad (1)$$

Os nós de processamento, que buscam simular os neurônios, têm dois estados, indicando quando o neurônio está "disparando" ou não. Desta forma, dada uma configuração inicial, a rede, de forma aleatória e assíncrona, atualiza seus valores para cada neurônio. O algoritmo segue a equação 2, na qual cada nó i possui um limiar U_i e atualiza seu valor aleatoriamente em cada iteração [Hopfield, 1982], garantindo que cada nó seja atualizado.

$$V_i \rightarrow \begin{cases} 1 & \text{se } \sum_{j \neq i} T_{ij} V_j > U_i, \\ 0 & \text{se } \sum_{j \neq i} T_{ij} V_j < U_i. \end{cases} \quad (2)$$

O resultado dessas atualizações é uma rede que funciona como uma memória dos dados. Assim, dado um conjunto de treinamento, mesmo sem o armazenamento dos dados de treino, com uma entrada distorcida de um dos dados usados, a rede consegue utilizar essa entrada como estímulo e recuperar o dado original, que é o processo utilizado para resolver problemas de recuperação de dados.

2.1 Redes de Hopfield-Tank

Uma evolução das Redes de Hopfield tradicionais, que são discretas, é a Rede de Hopfield-Tank, que opera com uma rede computacional analógica, na qual cada neurônio muda seu estado analógico de forma contínua e simultânea em paralelo, em um espaço de decisão contínuo [Hopfield and Tank, 1985].

Para esse modelo, a evolução dos nós ao longo do tempo é dada pela equação 3.

$$\frac{dU_i}{dt} = \sum_j T_{ij} V_j + I_i - U_i, \quad i = 1, \dots, L \quad (3)$$

Temos que U_i é a entrada, I_i é o *bias*, e V_i é o nível de ativação do nó i , que é uma função da entrada dada pela equação 4, onde g representa a função sigmoide [Hopfield and Tank, 1985].

$$V_i = g(U_i) = \frac{1}{2} \left(1 + \tanh \frac{U_i}{U_0} \right) = \frac{1}{1 + e^{-2U_i/U_0}}. \quad (4)$$

Dada a construção da rede, um conjunto de entradas de voltagem u_i é utilizado, e uma função de energia, definida em relação ao problema, irá convergir para um estado estável [Hopfield and Tank, 1985]. O último estado encontrado é então usado como solução final para o problema.

3 Problema do Caixeiro-Viajante

O problema do caixeiro-viajante é um dos exemplos mais famosos de problemas de otimização combinatória encontrados na computação. Por sua natureza, trata-se de um problema difícil de resolver, pertencente à classe NP-completo. No entanto, é fácil de explicar e ilustrar, o que o torna um problema comum para testes de diferentes algoritmos de aproximação, nos quais se espera encontrar uma solução suficientemente boa em tempo polinomial.

Dado um grafo $G = (V, A)$, onde se têm n vértices V que representam as cidades e arestas A que ligam essas cidades com um certo peso que representa a distância entre elas, é preciso encontrar um circuito de menor distância possível que comece em uma cidade, passe por todas as outras uma vez, e retorne à cidade de origem.

Como observado anteriormente, o PCV é NP-completo; sendo assim, não se conhece um algoritmo em tempo polinomial que encontre uma resposta ótima para o problema, e é por isso que abordagens aproximadas são alternativas populares para resolvê-lo.

4 Modelagem do Problema

O primeiro passo para modelar o PCV em uma Rede de Hopfield-Tank é encontrar um modelo que mapeie os níveis de ativação dos nós para uma solução do problema. Em seu trabalho [Hopfield and Tank, 1985], os autores sugerem a representação como uma matriz quadrada com o número de cidades, onde cada linha representa uma cidade e cada coluna uma posição no ciclo.

A Figura 1 [Hopfield and Tank, 1986] ilustra as conexões dos neurônios, e em cinza, a cidade selecionada para cada posição, no caso da imagem, na ordem D-B-F-C-A-E.

Em seguida, deve-se definir a função de energia, que os autores [Hopfield and Tank, 1985] sugerem como na Equação 5, onde A , B , C e D são constantes que são definidas antes da execução do código.

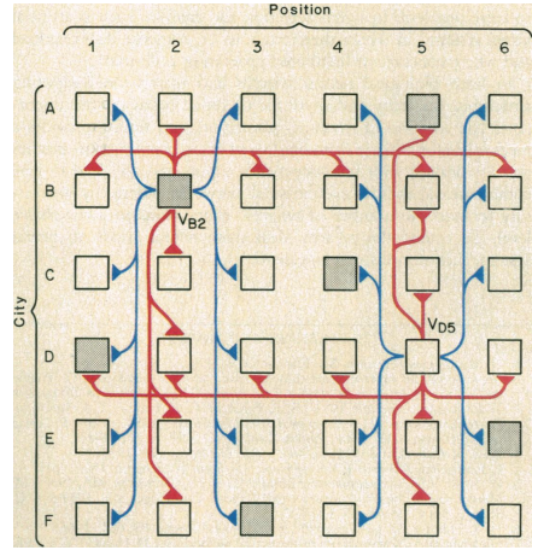


Figura 1: Sintaxe e conexões do circuito de neurônios para a representação do PCV.

$$\begin{aligned} E = & \frac{A}{2} \sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} \\ & + \frac{B}{2} \sum_i \sum_X \sum_{X \neq Y} V_{Xi} V_{Yi} \\ & + \frac{C}{2} \left(\sum_X \sum_i V_{Xi} - n \right)^2 \\ & + \frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{Xi} (V_{Yi+1} + V_{Yi-1}). \end{aligned} \quad (5)$$

A partir das Equações 1 e 3, é possível extrair a fórmula para os pesos da matriz de conexões, dada pela Equação 6 [Hopfield and Tank, 1985].

$$\begin{aligned} E = & -A \delta_{XY} (i - \delta_{ij}) - B \delta_{ij} (1 - \delta_{XY}) \\ & - C - D d_{XY} (\delta_{j,i+1} + \delta_{j,i-1}) \\ & [\delta_{i,j} = 1 \text{ se } i = j \text{ e } 0 \text{ caso contrário}]. \end{aligned} \quad (6)$$

O *bias* de entrada também é derivado das fórmulas anteriores e é dado pela Equação 7, onde n é um parâmetro baseado no número de cidades, mas nem sempre igual a esse valor. Em seus experimentos, Hopfield e Tank [Hopfield and Tank, 1985] usaram esse parâmetro para ajustar o nível de valores positivos do *bias* em relação aos valores negativos vindos de outras conexões. Nos testes, geralmente esse valor foi ajustado para ser um pouco maior que o número de cidades.

$$I_{Xi} = +Cn \quad (7)$$

Dessa forma, as constantes A , B e C fornecem as restrições básicas para o problema do PCV. Com a função de energia E , dado um estado inicial, o circuito deve convergir para uma matriz de permutação, como ilustrado na Figura 1, que representa uma solução para o problema.

5 Implementação

Para a realização dos testes neste trabalho, foi utilizada a biblioteca 'Netomaton' [Antunes, 2021], desenvolvida por Luis Antunes para a linguagem Python. Esta implementação segue como base o artigo de Hopfield e Tank e possibilita o ajuste dos parâmetros antes da execução.

6 Testes e Resultados

Para a realização dos testes, foi utilizada uma instância pequena, com 10 cidades, o mesmo tamanho utilizado por Hopfield e Tank em seu artigo.

O primeiro teste foi realizado utilizando os mesmos parâmetros que os autores encontraram em seu artigo [Hopfield and Tank, 1985], com $A = B = 500$, $C = 200$, $D = 500$ e $n = 15$. Com essas configurações, foi possível encontrar, após 5 execuções, uma distância de 2,88. Para comparar os parâmetros usados pelos autores com outras combinações, foi montada a Tabela 1, que indica cada parâmetro utilizado junto com a média das distâncias obtidas após 5 execuções.

A	B	C	D	n	Média
500	500	200	500	15	3,06
300	300	100	300	12	3,09
400	400	150	400	12	3,23
500	500	200	500	15	3,97

Tabela 1: Tabela com os resultados dos testes.

É possível notar que o conjunto de parâmetros propostos por Hopfield e Tank obteve os melhores resultados. Entretanto, é importante observar que, embora a rede sempre convirja para um ponto de energia mínimo, por trabalhar com soluções inválidas, não é garantido que o resultado será válido. Assim, nos testes conduzidos, os parâmetros propostos pelos autores foram os que apresentaram o maior número de soluções inválidas. Para a construção da tabela, foram consideradas apenas as cinco execuções com soluções válidas. A Figura 2 mostra como fica uma matriz de neurônios em uma solução inválida, onde é possível observar mais de uma cidade em uma mesma posição do passeio.

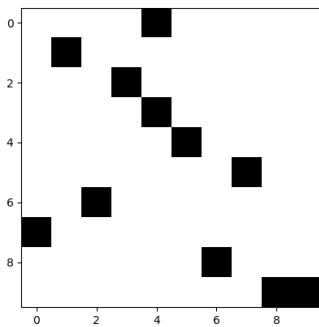


Figura 2: Matriz de neurônios em uma solução inválida.

Para uma solução válida, a Figura 3 ilustra sua matriz de neurônios e a Figura 4 mostra o grafo do circuito.

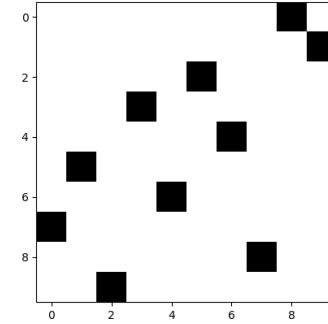


Figura 3: Matriz de neurônios em uma solução válida.

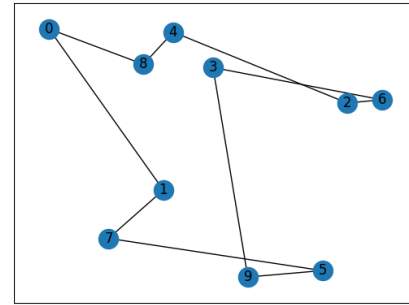


Figura 4: Grafo do circuito em uma solução válida.

Na Figura 3, diferentemente da Figura 2, não existem dois neurônios "disparando" em uma mesma coluna, o que indica que há apenas uma cidade por posição no passeio, resultando em uma solução válida para o problema.

Para analisar o desempenho do algoritmo, também foi feita uma comparação do método com o *Simulated Annealing*, uma metaheurística famosa que apresenta bom desempenho no PCV. Com ela, nas mesmas 5 execuções realizadas na rede neural, foi constantemente obtido o valor de 2,55, possivelmente o ótimo global do problema.

7 Conclusão

Pelos resultados alcançados, é possível concluir que, apesar da interessante aplicação das redes neurais na resolução de problemas de otimização, a rede de Hopfield-Tank não é o método mais adequado para a solução do Problema do Caixeiro Viajante. Além de frequentemente produzir soluções inválidas, o método é inferior tanto em termos de resultados quanto de desempenho quando comparado a outras heurísticas, como o *Simulated Annealing*, que neste trabalho conseguiu, de maneira eficiente, encontrar um resultado melhor que a rede.

No entanto, o desenvolvimento de redes neurais para problemas de otimização não deve ser desencorajado. Neste trabalho, foi utilizada uma das primeiras versões do algoritmo, e desde então, várias otimizações e melhorias foram realizadas, além do surgimento de novas arquiteturas.

Ademais, as redes neurais possuem uma alta capacidade de paralelização. Assim, com a popularização de GPUs cada vez mais potentes, essas redes podem ser a chave para aproveitar o novo poder computacional adquirido nos últimos anos.

Referências

- [Antunes, 2021] Luis Antunes. netomaton. <https://github.com/lantunes/netomaton/tree/master>, 2021.
- [Hopfield and Tank, 1985] John J. Hopfield and David W. Tank. “neural” computation of decisions in optimization problems. *Biol Cybern*, 52:141–152, 1985.
- [Hopfield and Tank, 1986] John J. Hopfield and David W. Tank. Computing with neural circuits: A model. *Science*, 233:625–633, 1986.
- [Hopfield, 1982] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–2558, 1982.
- [Luger, 2005] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison Wesley, Edinburgh Gate, Harlow, Essex CM20 2JE, England, 5th edition, 2005.
- [Russell and Norvig, 2010] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey 07458., 3rd edition, 2010.