



# UMA ANÁLISE DAS CONTRIBUIÇÕES DE ALAN TURING À ÁREA DE REDES NEURAIS ARTIFICIAIS



**Bolsista: Everton Schumacker Soares**  
e-mail: [ev.schumacker@gmail.com](mailto:ev.schumacker@gmail.com)  
**Co-autores: Levy Boccato, Marcos Fernandes, Diogo Coutinho Soriano**  
**Orientador: Prof. Dr. Romis Attux**

**Universidade Estadual de Campinas (UNICAMP)**  
**Faculdade de Engenharia de Sorocaba (FACENS)**  
**Universidade Federal do ABC (UFABC)**



**Palavras-chave: Redes Neurais - História da Computação - Alan M. Turing**

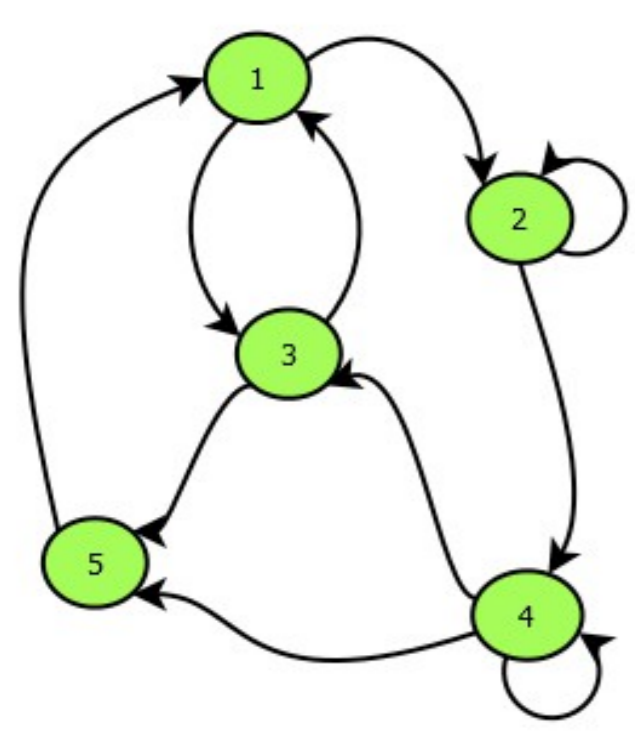
## BREVE CONTEXTUALIZAÇÃO HISTÓRICA

**“I propose to investigate the question as to whether it is possible for machinery to show intelligent behaviour.” – Alan M. Turing**

Alan Mathison Turing (1912 – 1954) foi um pesquisador responsável por contribuições a vários ramos da ciência da computação. Em 1945, ele foi contratado pelo NPL (National Physical Laboratory), então chefiado por Sir Charles Darwin, para o desenvolvimento do *Automatic Computing Engine* (ACE), um computador de propósito geral. Porém, esse projeto acabou sofrendo com vários conflitos e divergências no âmbito do time de desenvolvimento, o que levou Turing a se afastar de sua conclusão.

Durante esse período, Turing dedicou muito de seu tempo a reflexões sobre a natureza da inteligência. Desse modo, não causa surpresa que, em 1948, ele tenha apresentado ao NPL um relatório intitulado *Intelligent Machinery*, no qual foram apresentadas e discutidas máquinas desorganizadas que, modernamente, podemos denominar *redes de Turing*. Esse trabalho foi, ao lado dos esforços de McCulloch e Pitts, um marco na gênese da área de redes neurais artificiais. O trabalho contém, ademais, referências visionárias a assuntos como aprendizado de máquina e otimização evolutiva. Infelizmente, esse trabalho só veio a ser efetivamente conhecido mais recentemente, graças a esforços como os de Copeland [Copeland, 2004] e Teuscher [Teuscher, 2001].

## REDES NEURAIS ARTIFICIAIS DE TURING



**Redes tipo A:** Cada neurônio que compõe essa rede, possui um estado interno, que pode ser “0” ou “1”, e dois terminais de entrada de sinal. Portanto, sinais de entrada de, no máximo, dois neurônios diferentes produzem uma única saída dada pela fórmula:

$$Y(t+1) = 1 - X1(t).X2(t)$$

onde  $X1(t)$  e  $X2(t)$  são os sinais de entrada do neurônio e  $Y(t+1)$  é o sinal de saída, que equivale ao estado do neurônio no instante  $t+1$ . Essa operação corresponde à função lógica NAND.

Figura 1: Rede tipo A

**Redes tipo-B:** equivale a uma rede tipo A na qual cada conexão entre neurônios possui um nó de interferência (Figura 2), gerando a estrutura da Figura 3. Esse nó é constituído por uma rede tipo A específica e possui três estados possíveis:

- I – Inverte o sinal de entrada;
  - II – Anula o sinal de entrada, forçando-o para “1”;
  - III – Alterna o seu comportamento entre I e II.
- Nó de interferência = peso sináptico

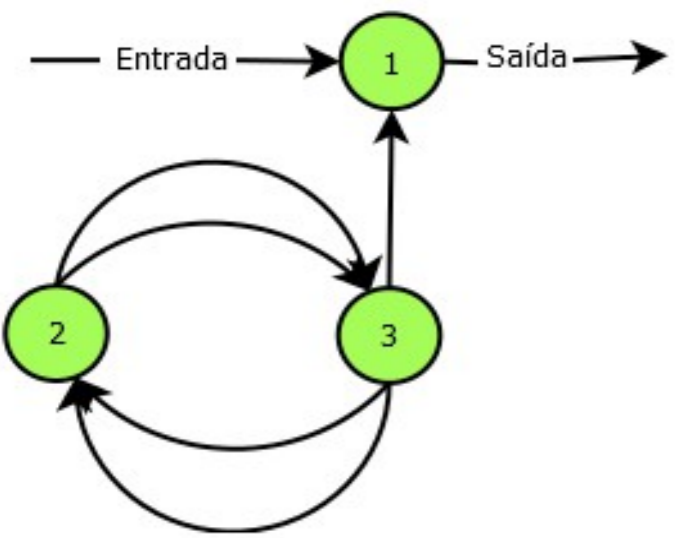


Figura 2: Nó de interferência

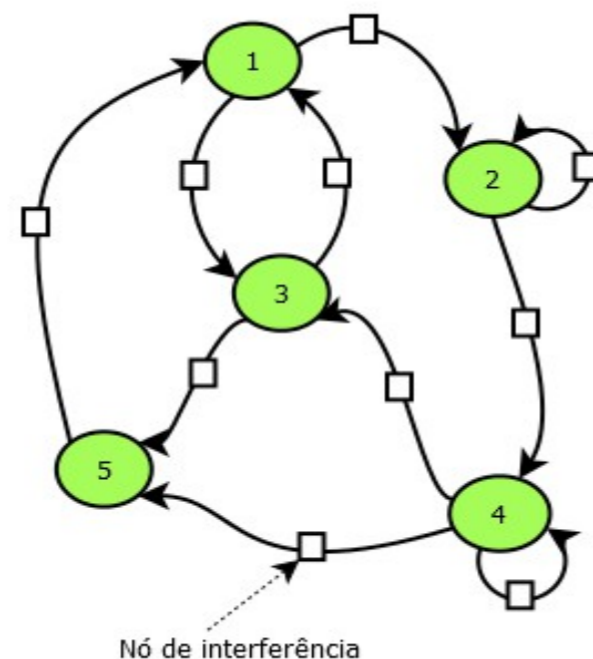


Figura 3: Rede tipo B

**Redes tipo-B com sinal de interferência:** Cada nó de interferência possui dois sinais de interferência externa (Figura 4).

Assim, fica evidente que Turing já havia pensado em um meio de treinar essas redes desorganizadas, alterando seu comportamento por meio de atuação externa.

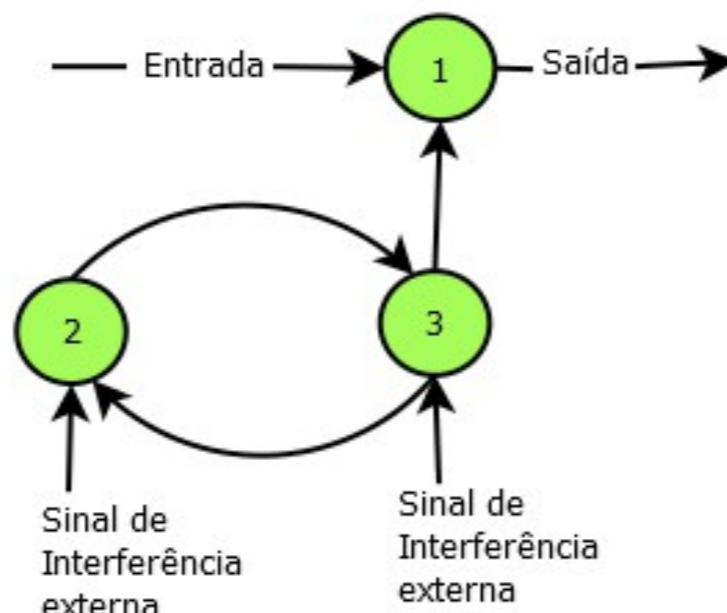


Figura 4: Nó de interferência da rede tipo B com sinal de interferência

## IMPLEMENTAÇÃO

A implementação da rede tipo A foi baseada na formalização feita por Christof Teuscher em [Teuscher, 2001]. Para representar a topologia da rede, isto é, a arquitetura da rede e os pesos sinápticos, Teuscher definiu a matriz de conexão  $C$ :

$$C_{N \times N} = \begin{bmatrix} 0 & \dots & 0 & c_{1(i+1)} & \dots & c_{1N} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & c_{N(i+1)} & \dots & c_{NN} \end{bmatrix}, \text{ na qual } c_{ij} = \begin{cases} 1 & \text{se existe conexão de } i \text{ para } j \\ 2 & \text{se existe conexão dupla de } i \text{ para } j \\ 0 & \text{caso contrário} \end{cases}$$

Definindo a função **NAND**  $\Phi$  por:

$$\phi(x) = \begin{cases} 0 & \text{se } x = 2 \\ 1 & \text{caso contrário.} \end{cases} \quad \vec{\phi}(x) = \begin{bmatrix} \phi(x) \\ \vdots \\ \phi(x) \end{bmatrix}$$

Assim, é possível definir os vetores de estado atual dos neurônios por:

$$\vec{d}_N[t] = \begin{bmatrix} d_1[t] \\ \vdots \\ d_{i+1}[t] \\ \vdots \\ d_{N-O+1}[t] \end{bmatrix}$$

e o vetor de input expandido por:

$$\vec{x}_N^T[t] = \begin{bmatrix} 1 - x_1[t] \\ \vdots \\ 1 - x_i[t] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Desta forma, é possível calcular a expressão de saída de cada neurônio por:

$$\vec{d}_N^T[t+1] = \vec{\phi}(C^T \vec{d}_N^T[t]) - \vec{x}_N^T[t] \quad \vec{y}_O^T[t+1] = \vec{d}_O^T[t+1] = \begin{bmatrix} d_{N-O+1}[t+1] \\ \vdots \\ d_N[t+1] \end{bmatrix}$$

## TREINAMENTO

**Rede tipo-A:** Com o uso de algoritmos genéticos, a arquitetura da rede foi treinada para que ela pudesse simular o comportamento de uma máquina de estados. A mutação, assim como o crossing-over, acrescenta a chance de reorganização da estrutura da rede.

**Redes tipo-B com sinal de interferência:** Diferentemente da Rede tipo-A, o treinamento consiste em evoluir os sinais de interferência da rede.

**Fitness:** Em ambos os casos, a função de fitness é calculada através de um mapeamento entre distância de Hamming entre o sinal de saída e o sinal esperado.

$$h(k) = \begin{cases} 1 & \text{se } y_o[t] \neq s_o[t-1] \\ 0 & \text{caso contrário.} \end{cases} \quad f(k) = \frac{1}{1 + \sum h(k)}$$

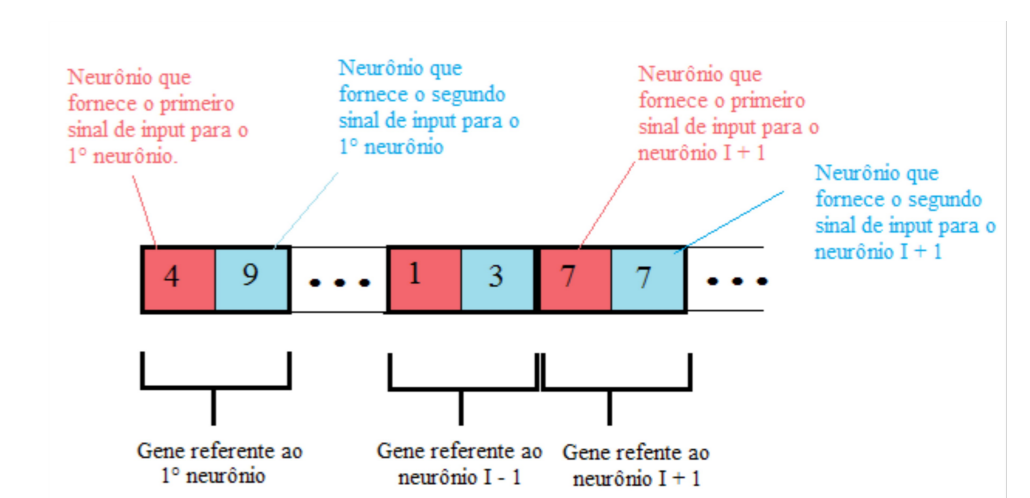


Figura 5: Modelo do vetor que representa um indivíduo de uma população de redes tipo A.

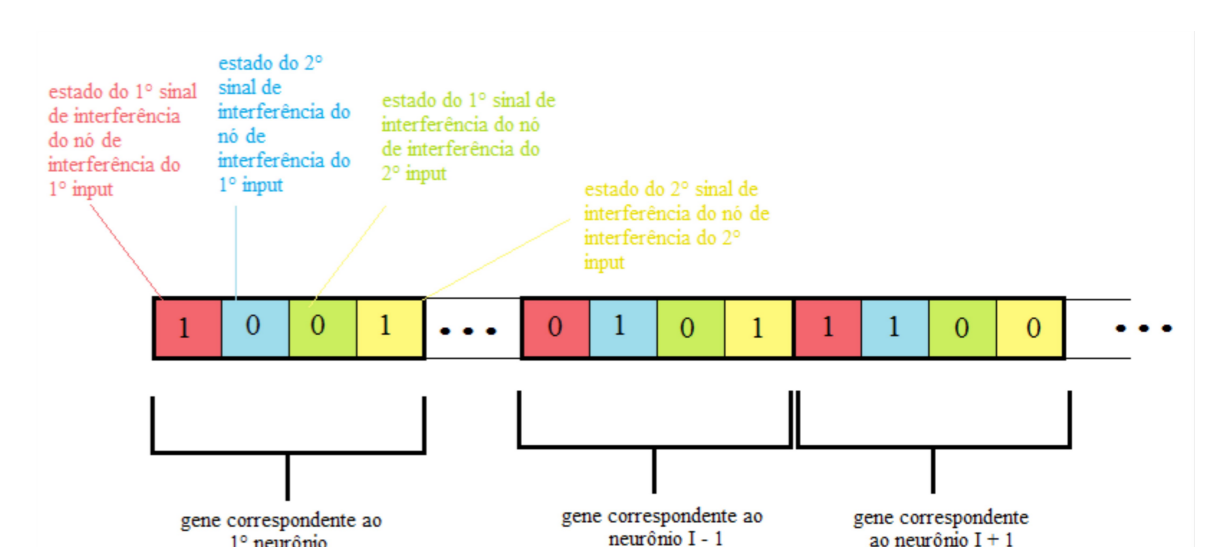


Figura 6: Modelo do vetor que representa um indivíduo de uma população de redes tipo B.

## RESULTADOS

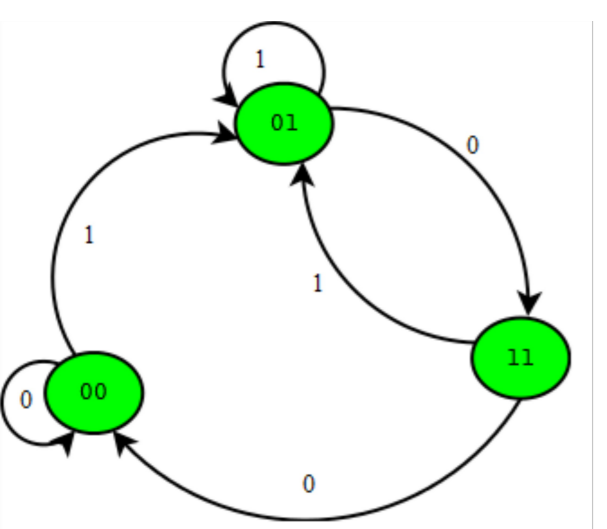


Figura 7: Máquina de Moore de 3 estados, cujo comportamento foi simulado pelas redes.

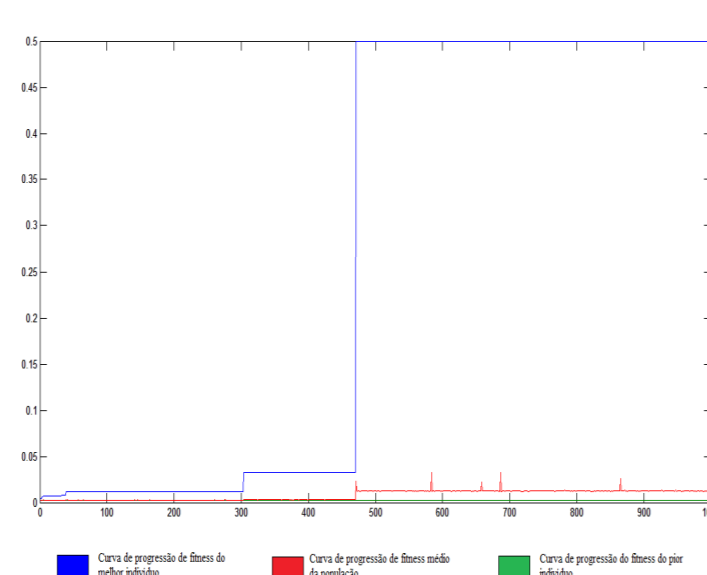


Gráfico 1: Evolução de uma população de redes A.

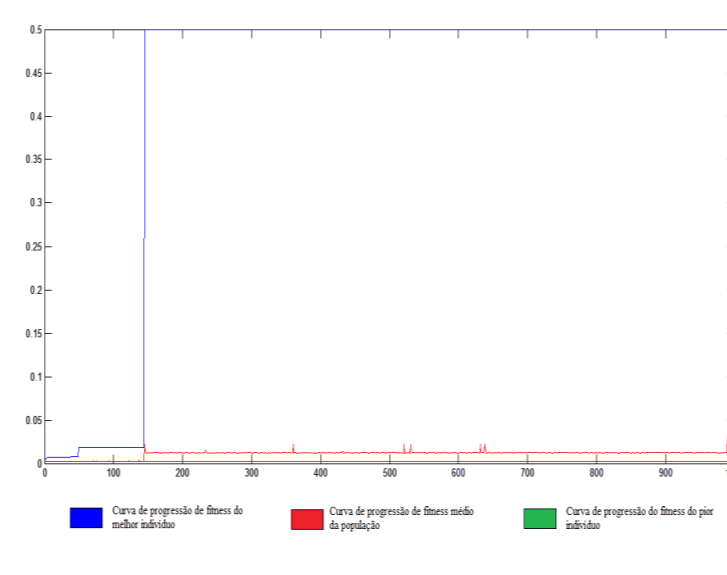


Gráfico 2: Evolução de uma população de redes B.

## REFERÊNCIAS

Copeland, B. J. (2004). *The Essential Turing*. Oxford University Press

Teuscher, C.(2001). *Turing's Connectionism: An Investigation of Neural Network Architectures*. Springer.

Turing, A. M.(1968) *Intelligent Machinery*. In: C. R. Evans, & A. D. Robertson, *Cybernetics: Key Papers*. Baltimore Md. And Manchester: University Park Press.