



Inteligência Artificial
Prof. Luiz Antonio Ferraro Mathias

Neurônios Biológicos

O neurônio é a célula do sistema nervoso responsável pela condução do impulso nervoso. O neurônio é constituído pelas seguintes partes: corpo celular (onde se encontra o núcleo celular), dendritos, axônio e telodendro.

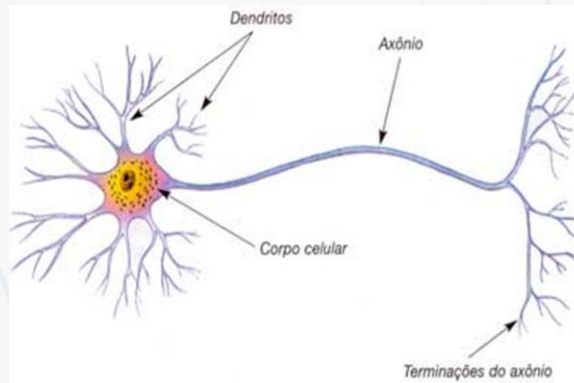
Dendritos: são prolongamentos numerosos dos neurônios especializados na recepção de estímulos nervosos, que podem ser do meio ambiente ou de outros neurônios.

Axônio: é o um prolongamento dos neurônios responsável pela condução dos impulsos elétricos que partem do corpo celular, até outro local mais distante, como um músculo ou outro neurônio. Alguns axônios de neurônios de um humano adulto podem chegar a mais de um metro de comprimento.

Telodendro: é uma ramificação terminal do axônio, onde o impulso passa de um neurônio para o outro, ou para outro órgão.

Neurônios Biológicos

O espaço entre o dendrito de um neurônio e o axônio de outro é o que se chama uma sinapse: os sinais são transportados através das sinapses por uma variedade de substâncias químicas chamadas neurotransmissores.



Neurônios Biológicos

No quadro abaixo, visualizamos um quadro comparativo entre a capacidade de processamento do cérebro humano, formado por células neurais e um computador.

Requisito	Computador	Cérebro humano
Velocidade	Nanosegundo	Milisegundo
Tipo de processamento	Sequencial	Paralelo
Número de unidades de armazenamento	10e9 bits	10e14 sinapses
Número de unidades de processamento	+ - 1024	10e11
Velocidade de processamento	nanossegundos	milissegundos
Controle de processamento	Controle autocrático, centralizado	controle anárquico, distribuído
Armazenamento do conhecimento	estritamente relocável	adaptativo
Material	Metal e plástico	orgânico
Tolerância a falhas	Mínima, quando existe	boa

Neurônios Biológicos

Os neurônios biológicos são de cinco a seis ordens de grandeza mais lentos que as portas lógicas dos microprocessadores.



Conceitos de Redes Naturais Artificiais

Segundo Haykin (2001), uma rede neural é um processador maciçamente e paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural de armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos: o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem; forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

O histórico das redes neurais artificiais é dividido da seguinte forma:

1943 - McCulloch e Pitts apresentam um estudo sobre o comportamento do neurônio biológico com o objetivo de criar um modelo matemático para este;

1949 – Donald O. Hebb definiu o conceito de atualização de pesos sinápticos;



Conceitos de Redes Naturais Artificiais

- c) 1951 – Marvin Minski constrói o 1º neuro-computador chamado “Snark”;
- d) 1957 – Surge o 1º neuro-computador a obter sucesso (Mark I Perceptron), desenvolvido por Frank Rosenblatt, Charles Wighman e outros;
- e) 1958 – Frank Rosenblatt implementa o 1º modelo de rede neural: o Perceptron (1 camada de entrada e 1 camada de saída). Este modelo não podia simular o comportamento de uma simples função XOR (ou-exclusivo);
- f) 1972 – Teuvo Kohonen definiu um novo modelo de rede neural, conhecido como “mapa auto-reorganizável de características”, com isto, introduziu um novo paradigma no estudo das redes neurais, o aprendizado não-supervisionado;



Conceitos de Redes Naturais Artificiais

- g) 1982 – Surge o “modelo de Hopfield” desenvolvido pelo físico John Hopfield. Este modelo é caracterizado por ser do tipo “feedback”, isto é, uma conexão das entradas com as saídas. A saída será sempre a mesma para um mesmo padrão de entrada;
- h) 1986 – David Rumelhart e James McClelland publicam “Parallel Distributed Processing” estimulando o campo de pesquisa;
- i) 1987 – Ocorre a 1ª conferência de Redes Neurais Artificiais em tempos modernos, a IEEE (The Institute of Electrical and Electronics Engineers) International Conference on Neural Networks. Nasce a International Neural Networks Society (INSS).



Conceitos de Redes Naturais Artificiais

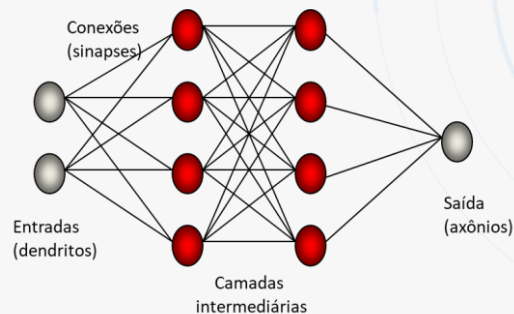
A discussão em torno da Redes Neurais Artificiais é norteadada pela análise dos seguintes aspectos:

- a) Neurônio: unidade computacional básica da rede;
- b) Arquitetura: estrutura topológica de como os neurônios são conectados;
- c) Aprendizagem: processo que adapta a rede de modo a computar uma função desejada ou realizar uma tarefa.



Conceitos de Redes Naturais Artificiais

A arquitetura de uma RNA é distribuída em camadas.



Uma RNA é composta por várias unidades de processamento, que geralmente são conectadas por canais de comunicação que estão associados a determinado peso. O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento de dados.



Conceitos de Redes Naturais Artificiais

Os elementos de processamento são os neurônios da rede, local onde é realizado todo o processamento. Tais elementos possuem duas qualidades importantes:

- Necessitam de informações locais. A saída do elemento de processamento é uma função dos pesos e das entradas;
- Elementos de processamento produzem apenas um valor de saída que é propagado através das conexões de elemento para outro (emissor para receptor), ou para fora da rede quando for um elemento da camada de saída.



Operação de uma RNA

A operação de uma unidade de processamento pode ser resumida da seguinte forma (ilustrado na próxima figura):

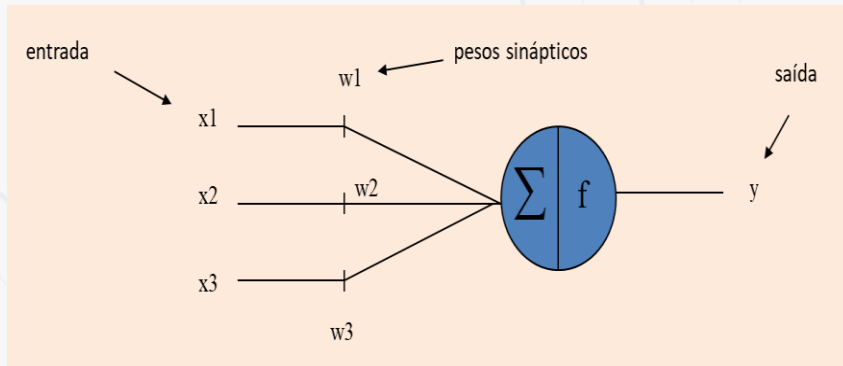
- a) Sinais são apresentados à entrada;
- b) Cada sinal é multiplicado por um número (peso) que indica a sua influência na saída da unidade;
- c) É feita a soma ponderada dos sinais que produz um nível de atividade:

$$y = f\left(\sum_{i=1}^n x_i \cdot w_i\right)$$

- d) Se este nível de atividade exceder um certo limite, a unidade produz uma determinada resposta de saída.



Operação de uma RNA



Função de ativação

Um neurônio soma todos os pesos das entradas e passa o resultado para uma função de ativação (função de transferência) que será responsável por determinar a forma e a intensidade de alterações dos valores transmitidos de um neurônio a outro.

Exemplos de funções de ativação:

Função Degrau

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases}$$

Função de ativação

Função Linear

$$f(x) = \begin{cases} 1, & x \geq \frac{1}{2} \\ x, & -\frac{1}{2} < x < \frac{1}{2} \\ 0, & x \leq -\frac{1}{2} \end{cases}$$

Função Sigmoidal

$$f(x) = \frac{1}{1 + \exp(-\alpha x)}$$



Aprendizado de RNAs

As RNA's podem ser distinguidas com base na direção na qual o sinal flui:

- **Feedforward:** sinais se propagam em uma direção a partir da unidade de entrada, passando pela camada intermediária até a saída;
- **Feedback:** os sinais de entrada podem propagar da saída de qualquer neurônio para a entrada em um outro neurônio.

A habilidade de aprender de uma RNA ocorre através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a RNA atinge uma solução generalizada para uma classe de problemas.



Aprendizado de RNAs

Algoritmo de aprendizado

A maioria dos modelos de Redes Neurais possui alguma regra de treinamento, onde os pesos das conexões são ajustados de acordo com os padrões apresentados, ou seja, “aprendem” através de exemplos. Desta forma uma RNA é capaz de extrair regras básicas a partir de dados reais.

Paradigmas de aprendizado:

- **Supervisionado:** utiliza-se um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Não-supervisionado:** não existe um agente externo.



Aprendizado de RNAs

Existem diversos algoritmos de aprendizado específicos para determinados modelos de RNA's, diferenciando-se entre si pelo modo como os pesos são modificados.

O algoritmo de Backpropagation

Durante o treinamento com este algoritmo, a rede opera em uma sequência de dois passos. Primeiro, um padrão é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No 2º passo, a saída obtida é comparada à saída desejada para esse padrão particular.

Se não estiver correta, o erro é calculado e propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados conforme o erro é retropropagado. Com isso o erro vai sendo progressivamente diminuído.



Aprendizado de RNAs

Etapas para o desenvolvimento de aplicações de RNA

1- Coleta de dados

Os dados devem ser significativos e cobrir amplamente o domínio do problema e também as exceções.

2- Separação em conjuntos

Dados de treinamento e dados de teste (verificar performance sob condições reais de utilização).



Aprendizado de RNAs

3- Configuração da rede

- Seleção do paradigma neural
- Definição da topologia da rede
- Determinação de parâmetros do algoritmo de treinamento e funções de ativação

4- Treinamento

- Emprego do algoritmo de treinamento
- Ajuste do peso das conexões

5- Teste

Determinar a performance da rede, medida nesta fase, é uma boa indicação de sua performance real.



Aplicações com RNAs

Diversas são as aplicações com redes neurais artificiais:



Exemplo de aplicação para resolução de problema de PREVISÃO

O órgão gestor de uma cidade está estudando a possibilidade de construir um dique de proteção da cidade contra eventuais inundações decorrentes de chuvas. Para estabelecer a altura do dique em função de uma análise custo x benefício, o órgão precisa dispor de uma função que forneça a probabilidade para cada vazão possível de ocorrer. Para desenvolver esta função vazão versus probabilidade, o órgão elaborou uma tabela com os dados de sete valores de vazões com suas probabilidades acumuladas (probabilidade de ocorrer um evento um valor menor ou igual).

Exemplo de aplicação para resolução de problema de PREVISÃO

Exemplo	Vazão (q) (m ³ /s)	Probabilidade (x100) (valor ≤ q)
1	1,50	0,02
2	2,00	0,08
3	2,50	0,18
4	3,00	0,50
5	3,50	0,65
6	4,00	0,88
7	4,50	0,96



Exemplo de aplicação para resolução de problema de PREVISÃO

O objetivo é o desenvolvimento de uma rede PERCEPTRON Multicamadas para realizar a aproximação da função probabilidade tendo como entrada a vazão. Considerar 2 (dois) neurônios na camada escondida e utilizar a função SIGMOIDAL como função de saída (função de ativação).

Quando se apresentam os sete exemplos de treino, tem-se um ciclo (época). Para treinar a rede são necessários muitos ciclos. Após cada ciclo se pode calcular o valor do erro médio quadrático de tal maneira que se fizermos um gráfico onde na horizontal se colocam os ciclos e na vertical o erro médio quadrático, tem-se a evolução do erro ao longo dos ciclos durante o processo de treinamento. Esta curva é construída e denominada de curva de aprendizagem.



Exemplo de aplicação para resolução de problema de PREVISÃO

Os pesos sinápticos das conexões da rede (determinados aleatoriamente no intervalo de $[-1,1]$ são:

$$w_{10}^1 = 0,1$$

$$w_{20}^1 = -0,7$$

$$w_{10}^2 = -0,6$$

$$w_{12}^2 = -0,8$$

$$w_{11}^1 = -0,3$$

$$w_{21}^1 = 0,4$$

$$w_{11}^2 = 0,1$$

Onde:

w_{ij} ,

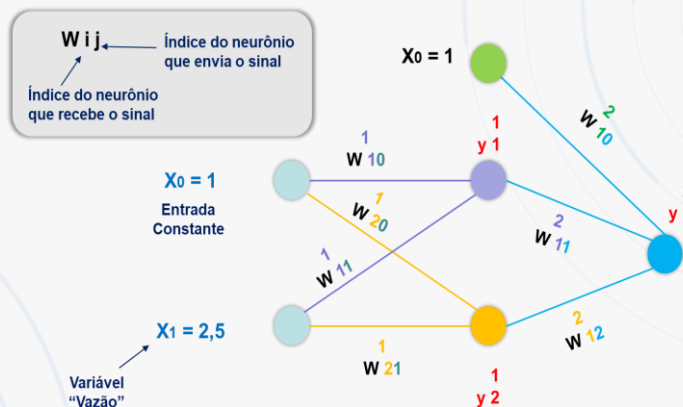
"i" é o índice do neurônio que recebe o sinal

"j" é o índice do neurônio que envia o sinal

α (ALFA): taxa de aprendizagem



Exemplo de aplicação para resolução de problema de PREVISÃO



Exemplo de aplicação para resolução de problema de PREVISÃO

Boa Prática de Arquitetura na Construção de uma RNA

Quando há uma Rede Neural com uma entrada constante, 1 entrada só, é uma boa prática criar sinteticamente uma segunda entrada com valor inicial constante igual a 1.

Além disso, pesquisas também indicam que é importante, na camada escondida (ou intermediária) acrescentar um neurônio a mais, com valor constante igual a 1. Esse neurônio é tecnicamente chamado de **BIAS**.



1ª Etapa – Propagação do sinal

Fase 1: Cálculo da entrada líquida para os neurônios da camada escondida (intermediária).

$$net_1^1 = \sum_{j=0}^1 w_{ij} \cdot x_j = w_{10}^1 \cdot x_0 + w_{11}^1 \cdot x_1$$

$$net_1^1 = (0,1) \cdot 1 + (-0,3) \cdot 2,5 = -0,65$$

$$net_2^1 = \sum_{j=0}^1 w_{ij} \cdot x_j = w_{20}^1 \cdot x_0 + w_{21}^1 \cdot x_1$$

$$net_2^1 = (-0,7) \cdot 1 + (0,4) \cdot 2,5 = 0,30$$



1ª Etapa – Propagação do sinal

Fase 2: Cálculo da função de saída para os neurônios da camada escondida (intermediária)

$$y_1^1 = f^1(net_1^1) = \frac{1}{1+e^{-(net_1^1)}} = \frac{1}{1+e^{-(-0,65)}} = 0,34299$$

$$y_2^1 = f^1(net_2^1) = \frac{1}{1+e^{-(net_2^1)}} = \frac{1}{1+e^{-(0,30)}} = 0,57444$$

O número de Euler (e) é, aproximadamente: 2,718.



1ª Etapa – Propagação do sinal

Fase 3: Cálculo da entrada líquida para o neurônio da camada de saída.

$$net_1^2 = \sum_{j=0}^2 w_{ij} \cdot f^1(net_j^1) = w_{10}^2 \cdot f^1(net_0^1) + w_{11}^2 \cdot f^1(net_1^1) + w_{12}^2 \cdot f^1(net_2^1)$$

$$net_1^2 = (-0,6) \cdot 1 + (0,1) \cdot 0,34299 + (-0,8) \cdot 0,57444 = -1,02526$$

Fase 4: Cálculo da função de saída para o neurônio da camada de saída.

$$y_1^2 = f^2(net_1^2) = \frac{1}{1+e^{-(net_1^2)}} = \frac{1}{1+e^{-(-1,02526)}} = 0,26401$$



1ª Etapa – Propagação do sinal

Fase 5: Cálculo do ERRO para o neurônio da camada de saída.

$$\epsilon_i = d_i - y_1$$

$$\epsilon_i = d_1 - y_1^2 = (0,18) - (0,26401) = -0,08401$$

O valor 0,18 (saída prevista) foi estabelecida por um especialista humano com domínio do problema.



2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada

Fase 1: Cálculo da sensibilidade para o neurônio da camada de saída.

Observação: As sensibilidades necessárias para ajuste dos pesos são calculadas da última para a 1ª camada. A sensibilidade de cada neurônio de saída (δ_i^2) é calculada pelo produto da derivada da função de ativação ($f^2(net_i^2)$) pelo respectivo erro de saída (e_i).

$$\delta_1^2 = y_1^2 \cdot (1 - y_1^2) \cdot e_1$$

$$\delta_1^2 = 0,26401 \cdot (1 - 0,26401) \cdot (-0,08401) = -0,01632$$



2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada

Fase 2: Cálculo da sensibilidade para os neurônios da camada escondida.

$$\delta_1^1 = f^1(net_1^1) \cdot w_{11}^2 \cdot \delta_1^2$$

$$\delta_1^1 = 0,34299 \cdot (1 - 0,34299) \cdot 0,1 \cdot (-0,01632) = -0,00037$$

$$\delta_2^1 = f^1(net_2^1) \cdot w_{12}^2 \cdot \delta_1^2$$

$$\delta_2^1 = 0,57444 \cdot (1 - 0,57444) \cdot (-0,8) \cdot (-0,01632) = -0,00319$$



2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada

Fase 3: Ajustes dos pesos que ligam a camada de saída a camada escondida.

α (ALFA): taxa de aprendizagem: o valor da taxa de aprendizagem da RNA pode variar entre 0 e 1. Quanto maior a constante, maior a mudança dos pesos. Quanto menor a constante, mais tênue será a mudança.

$$w_{ij}^2(novo) = w_{ij}^2(antigo) + \alpha \cdot \delta_i^2 \cdot f^1(net_j^1)$$

$$w_{10}^2(novo) = w_{10}^2(antigo) + \alpha \cdot \delta_1^2 \cdot f^1(net_1^1)$$

$$w_{10}^2(novo) = -0,6 + 0,7 \cdot (-0,01632) \cdot 1 = -0,61143$$

$$w_{11}^2(novo) = 0,1 + 0,7 \cdot (-0,01632) \cdot 0,34299 = 0,09608$$

$$w_{12}^2(novo) = -0,8 + 0,7 \cdot (-0,01632) \cdot 0,57444 = -0,80656$$



2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada

Fase 4: Ajustes dos pesos que ligam a camada escondida a camada de entrada.

$$w_{10}^1(novo) = 0,1 + 0,7. (-0,00037). 1 = 0,09974$$

$$w_{11}^1(novo) = -0,3 + 0,7. (-0,00037). 2,5 = -0,30064$$

$$w_{20}^1(novo) = -0,7 + 0,7. (0,00319). 1 = -0,69777$$

$$w_{21}^1(novo) = 0,4 + 0,7. (0,00319). 2,5 = 0,40559$$



2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada

Quando se apresentam os 7 EXEMPLOS de treino, tem-se um ciclo (época). Para treinar a rede são necessários muitos ciclos. Após cada ciclo se pode calcular o valor do ERRO MÉDIO quadrático verificando a evolução de erro ao longo dos ciclos durante o processo de treinamento. Esta curva é denominada CURVA DE APRENDIZAGEM.

