

Algoritmos Adaptativos e Evolutivos

Introdução

M. Cristina F. De Castro & Fernando C. C. De Castro

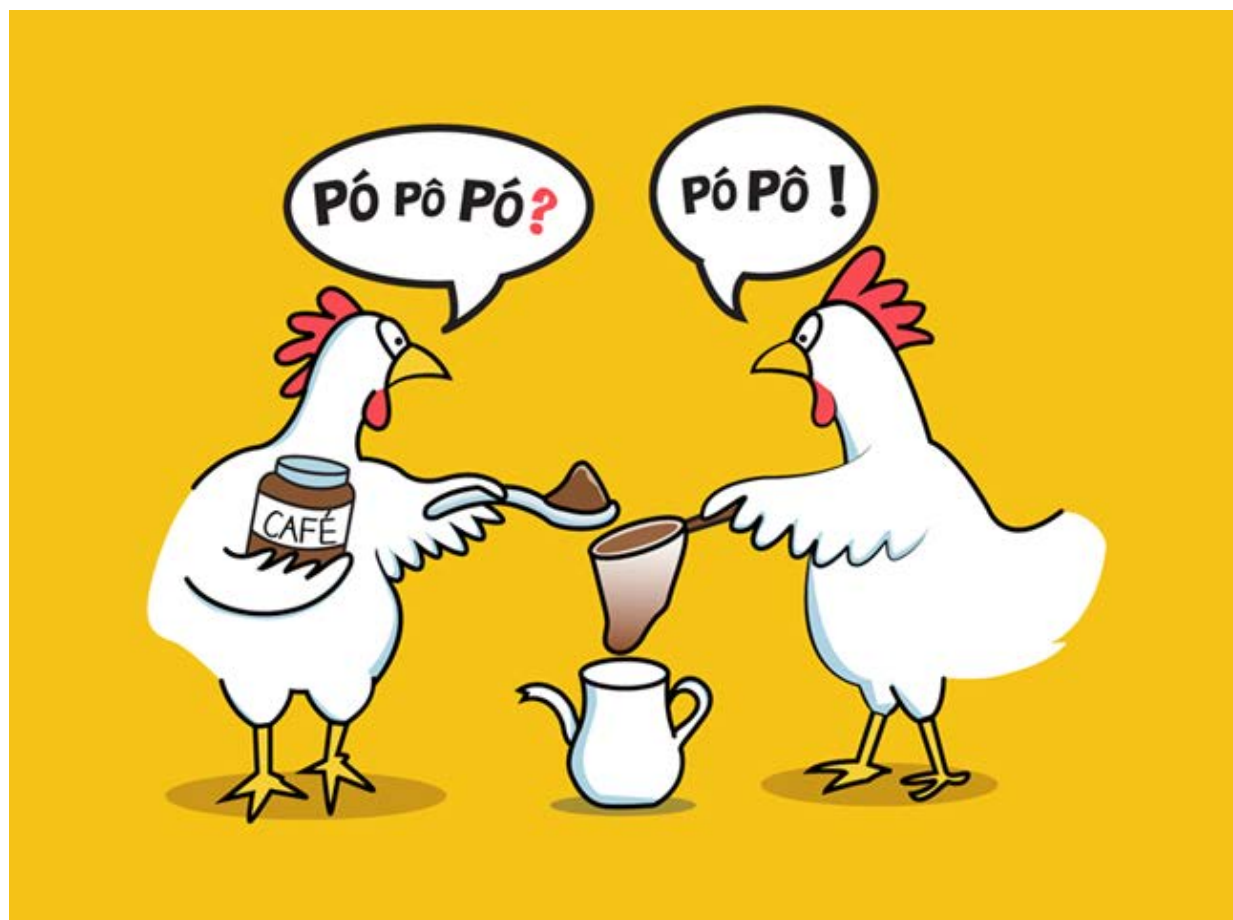


Centro de Pesquisa em Tecnologias Wireless

Dado um problema, como máquinas podem aprender soluções exatas, encontrar soluções a partir de informações imprecisas, ou encontrar soluções de modo evolutivo ?



Inteligência artificial – até onde pode ser aplicada?



Fowl Language: AI Decodes the Nuances of Chicken “Speech”

How machine learning can translate chicken chatter and improve farming

FONTE: <https://www.scientificamerican.com/article/fowl-language-ai-decodes-the-nuances-of-chicken-ldquo-speech-rdquo/>



Ethiopian flight 302: second new Boeing 737 MAX 8 to crash in four months

“It is notable that insurers considering driverless cars worry most about the period when highly autonomous vehicles will coexist with human drivers, the **uncertain interface between human and artificial intelligence**”.

FONTE: <https://www.theguardian.com/world/2019/mar/10/ethiopian-flight-302-second-new-boeing-737-max-8-to-crash-in-four-months>

“Objetivo da Ciência





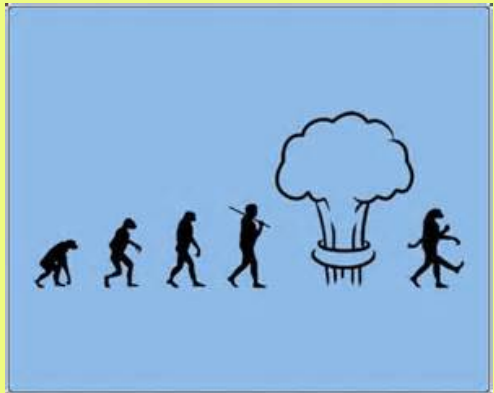
Compreender e controlar o mundo“

(Compreender e representar, modelar, a natureza)

- Modelamento da natureza – modelos matemáticos – como teoria das filas, formulação da entropia, etc, datam do começo do século XIX...
- Advento dos computadores representou revolução na história da ciência e da tecnologia.
- Criar **inteligência artificial e vida artificial** (desde Turing, Von Neumann, Wiener, ...)
 - Software inteligente, com capacidade de se auto-replicar, aprender adaptativamente, controlar (dentro de limites) o ambiente.
 - Interessados tanto em Biologia e Psicologia como em Eletrônica.
 - **Inspiração:** procuravam em sistemas naturais as metáforas para atingir seus objetivos (através do modelamento do cérebro, do aprendizado humano e da simulação da evolução biológica).

1980	→ Artificial Neural Networks (ANN)	Técnicas para processamento de informação que são complementares no projeto de Sistemas Inteligentes
	→ Evolutionary Computation (GAs)	
	→ Fuzzy Logic (FL)	

Aprendizagem, evolução, imprecisão... conceitos para nós intuitivos mas, e para as máquinas!?..

<u>Fuzzy Logic ou Fuzzy sets</u> (FL, FS)	<u>Artificial Neural Networks</u> (ANN)	<u>Genetic Algorithms</u> (GA)
<p>Solucionam um problema com parâmetros naturalmente imprecisos através de um conjunto de dados e através de uma lógica relacional que intrinsecamente contemplam imprecisão descritiva.</p>	<p>Solucionam um problema com base na aprendizagem dos processos dinâmicos que descrevem o problema.</p>	<p>Soluciona um problema com base na evolução de um conjunto (população) de soluções.</p>
<p>buscam mimetizar a forma pela qual o cérebro lida com informações imprecisas</p> 	<p>são modeladas de acordo com a arquitetura física do cérebro e buscam mimetizar as técnicas de formação dos “mapas de conhecimento” cerebrais</p> 	<p>buscam simular o processo de evolução natural (seleção, combinação, mutação ...)</p> 

Redes Neurais Artificiais

RNAs têm a analogia neurobiológica como fonte de inspiração.

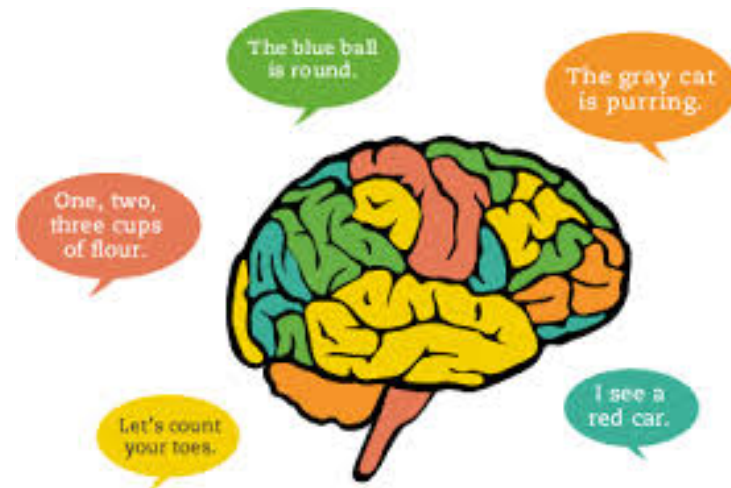


- São estruturas computacionais projetadas para mimetizar a maneira pela qual o cérebro desempenha uma particular tarefa de seu interesse.
- São compostas de múltiplas camadas de unidades de processamento, ou neurônios, conectadas por pesos sinápticos, ou sinapses.
- Adquirem conhecimento através de um processo de aprendizagem.

I HAVE NO IDEA
WHAT'S GOING
TO HAPPEN.



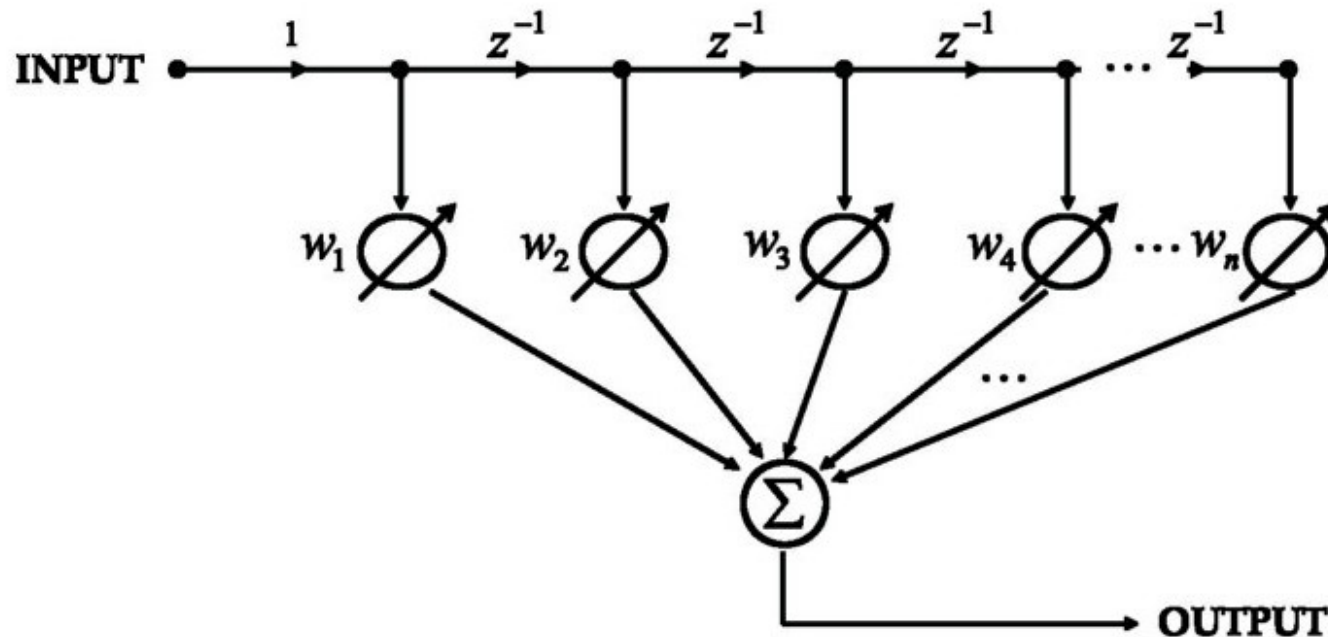
AND I LOVE IT.



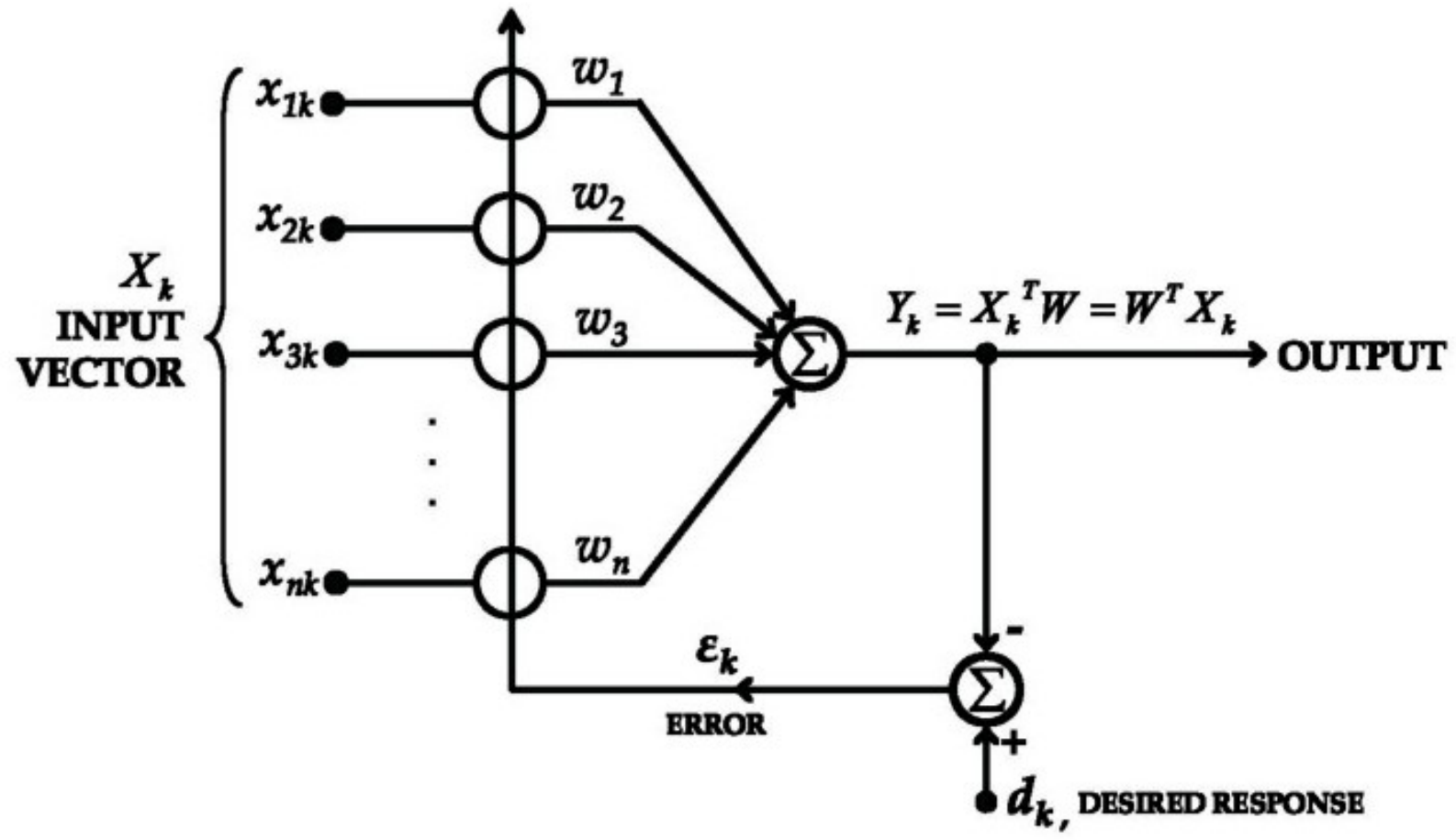
In response to the demands of the world, the baby's brain sculpts itself.

Redes Neurais Artificiais – como tudo começou?

No outono de 1959 Prof. Bernard Widrow e o aluno Ted Hoff discutiam em um quadro negro da universidade de Stanford sobre a implementação de filtros FIR adaptativos. Em determinado momento a conversa muda inesperadamente para uma discussão de como um processo de aprendizagem poderia ser realizado por um neurônio artificial utilizando técnicas numéricas de otimização baseada no gradiente de uma função que medisse o quanto o processo de aprendizagem está distante de obter o resultado desejado para o estímulo aplicado ao neurônio. Nascia o algoritmo LMS (*least mean square*).



Filtro FIR



Filtro FIR adaptativo

Algoritmo LMS:

$$y_k = X_k^T W_k = W_k^T X_k$$

$$\epsilon_k = d_k - X_k^T W_k$$

$$W_{k+1} = W_k + \mu(-\hat{\nabla}_k) \quad (1) \longrightarrow \text{Regra Delta}$$

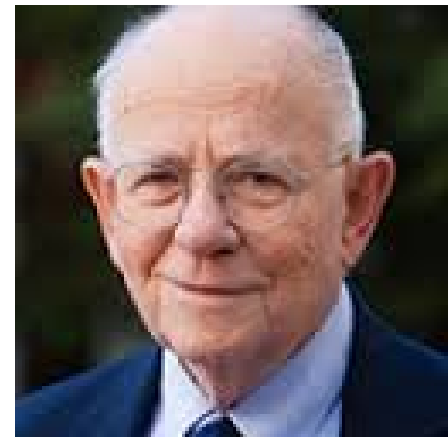
$$\text{MSE} = E[\epsilon^2]$$

$$\hat{\nabla}_k = \frac{\partial \epsilon_k^2}{\partial W_k} = 2\epsilon_k \frac{\partial \epsilon_k}{\partial W_k} = -2\epsilon_k X_k \quad (2)$$

$$(2) \longrightarrow (1)$$

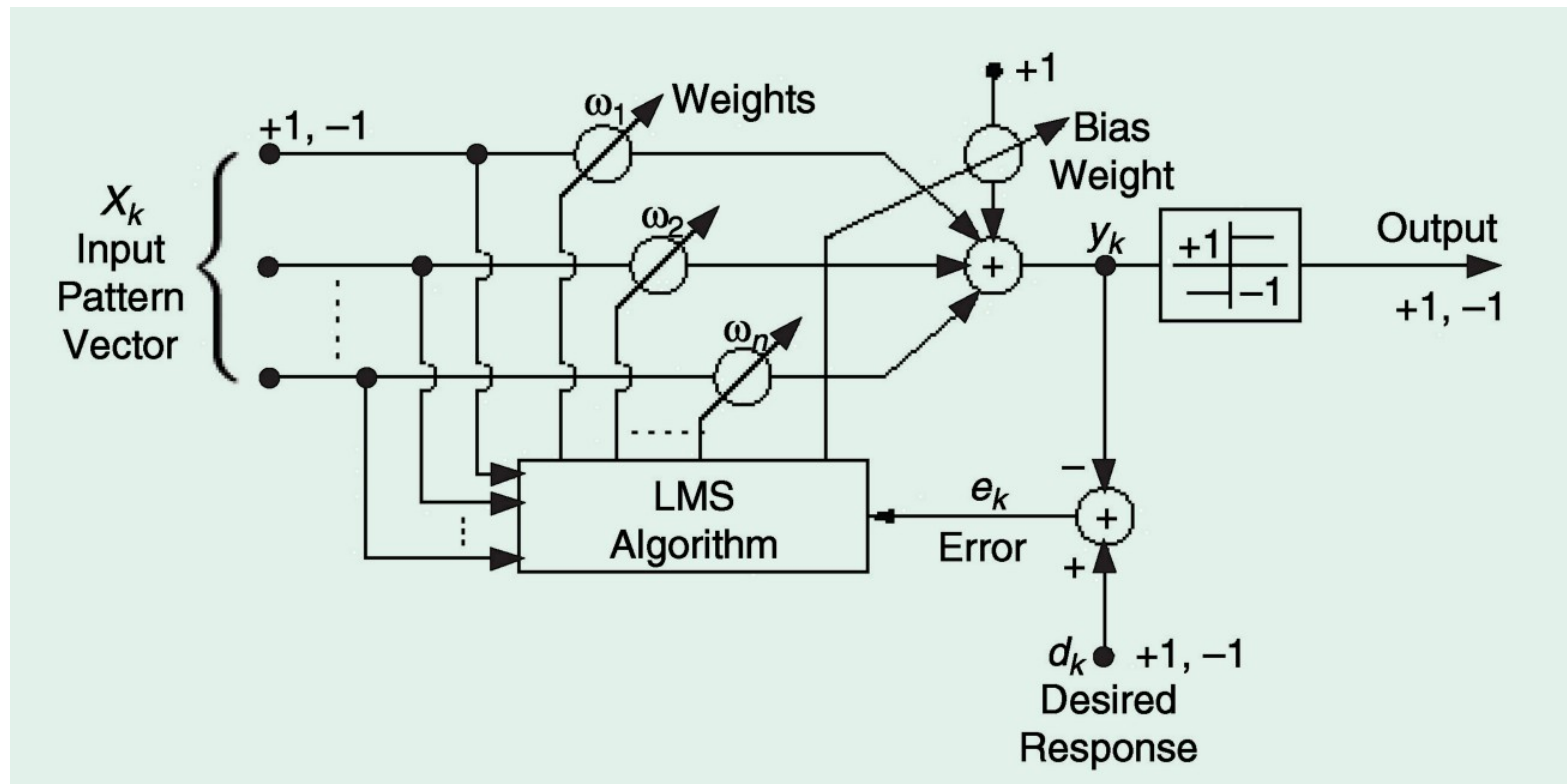
$$W_{k+1} = W_k + 2\mu\epsilon_k X_k \quad (3)$$

$$\epsilon_k = d_k - X_k^T W_k \quad (4)$$



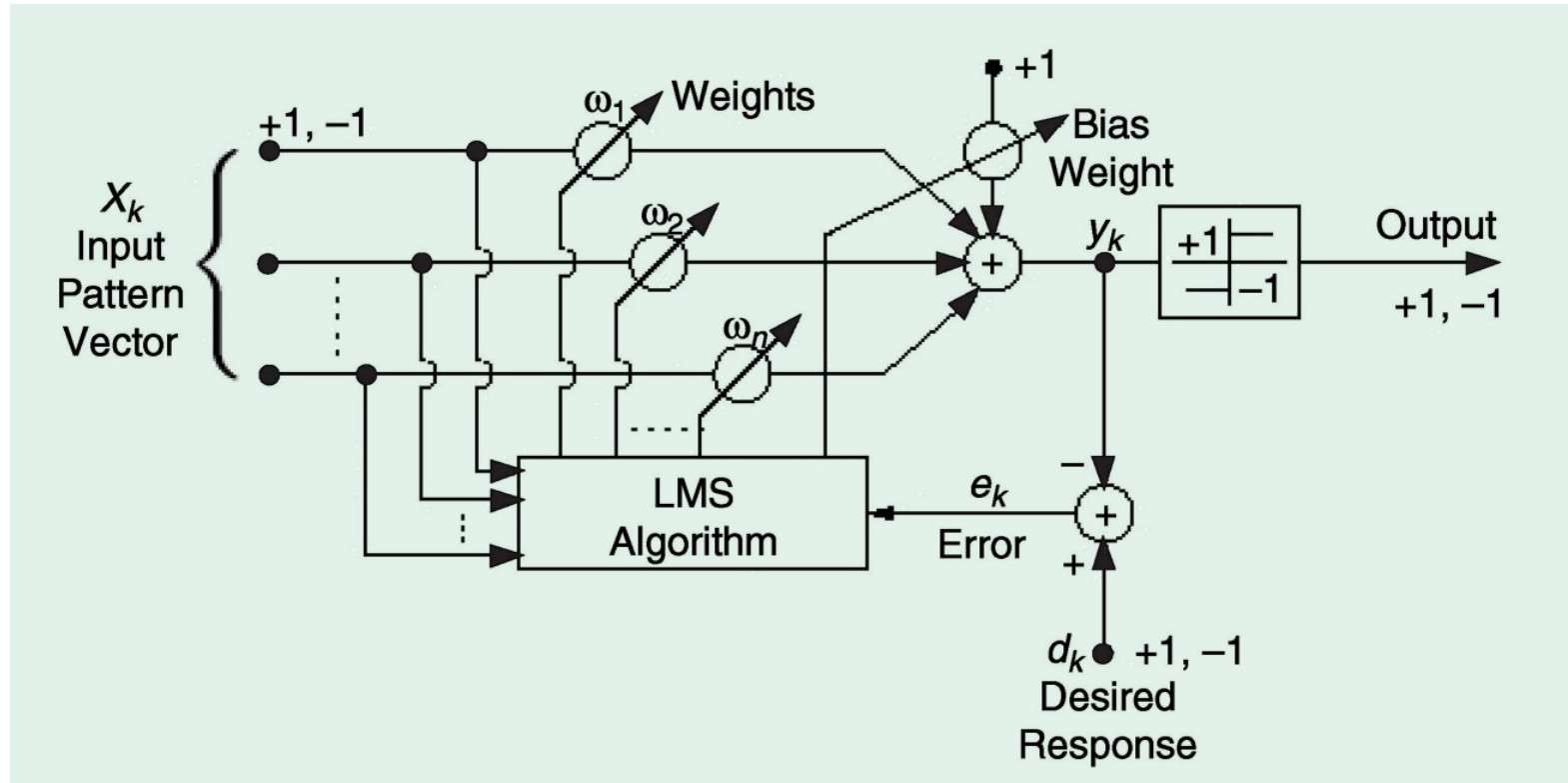
Bernard Widrow

Tendo escrito o algoritmo LMS no quadro negro, em menos de meia hora Ted Hoff já estava trabalhando em um computador analógico localizado do outro lado do corredor do escritório de Widrow, efetuando o processo de treino de um combinador linear que implementava um neurônio linear para classificação de padrões. Nascia o ADALINE (*Adaptive Linear Neuron*).



ADALINE (*Adaptive Linear Neuron*)

Especificamente, um padrão representado na forma de vetor \underline{X}_k era aplicado aos nós de entrada do Adaline na iteração k , e, dada a resposta $d_k = \{+1, -1\}$ desejada à \underline{X}_k (que representa a classe a que pertence o padrão \underline{X}_k aplicado na entrada na iteração k), os parâmetros livres \underline{W}_k são ajustados pela regra delta no sentido de minimizar o erro e_k a cada iteração k . O Adaline, portanto, é treinado com um conjunto de vários vetores \underline{X}_k e respectivas respostas desejadas d_k



ADALINE (*Adaptive Linear Neuron*)

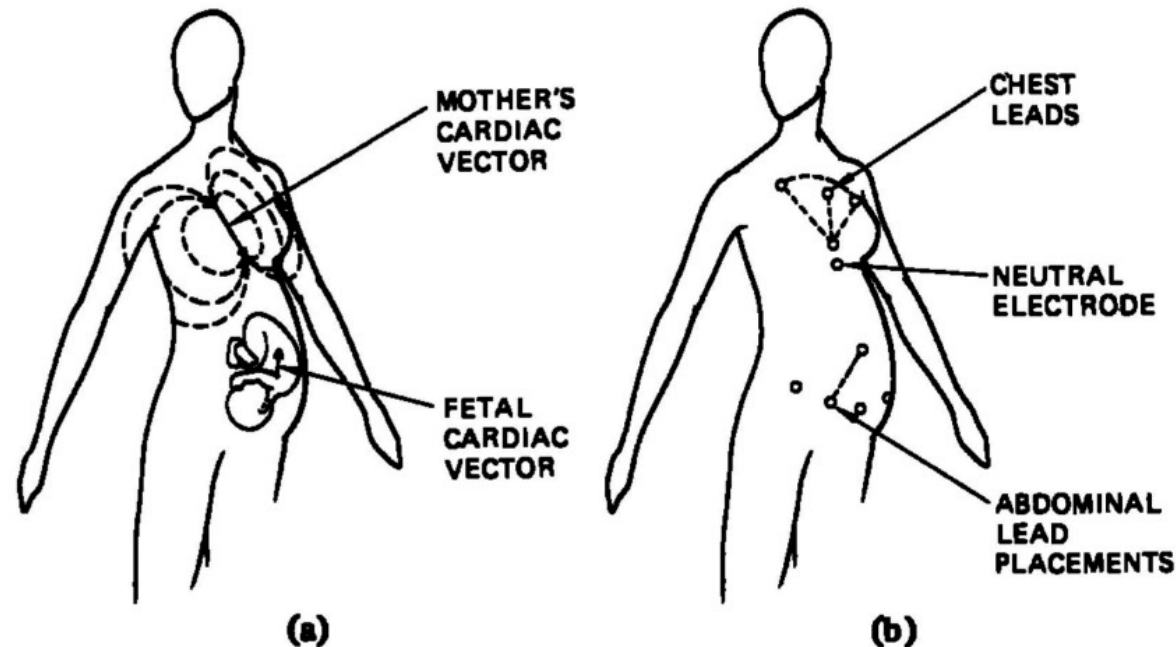
A figura mostra o Adaline analógico usado por Ted Hoff após a dedução do LMS para a validação do mesmo como algoritmo de aprendizado. Cada um dos 16 knobs controla um dos 16 coeficientes W_k do combinador linear do Adaline, cujas 16 entradas estão mostradas abaixo do medidor. O medidor mede o erro e_k na k -ésima iteração de forma que o operador humano possa implementar a regra delta manualmente: se, ao aplicar um pequeno ajuste a um knob o erro diminui, então o sentido do ajuste é o correto e deve-se ajustar o knob até minimizar o erro. Caso contrário, se o erro aumenta, deve-se aplicar o mesmo ajuste ao knob mas variando o knob no sentido contrário até o mínimo do erro. Os resultados obtidos por Widrow e Hoff foram muito promissores, e se tornaram a base para o aprendizado de redes neurais através de técnicas de gradiente.



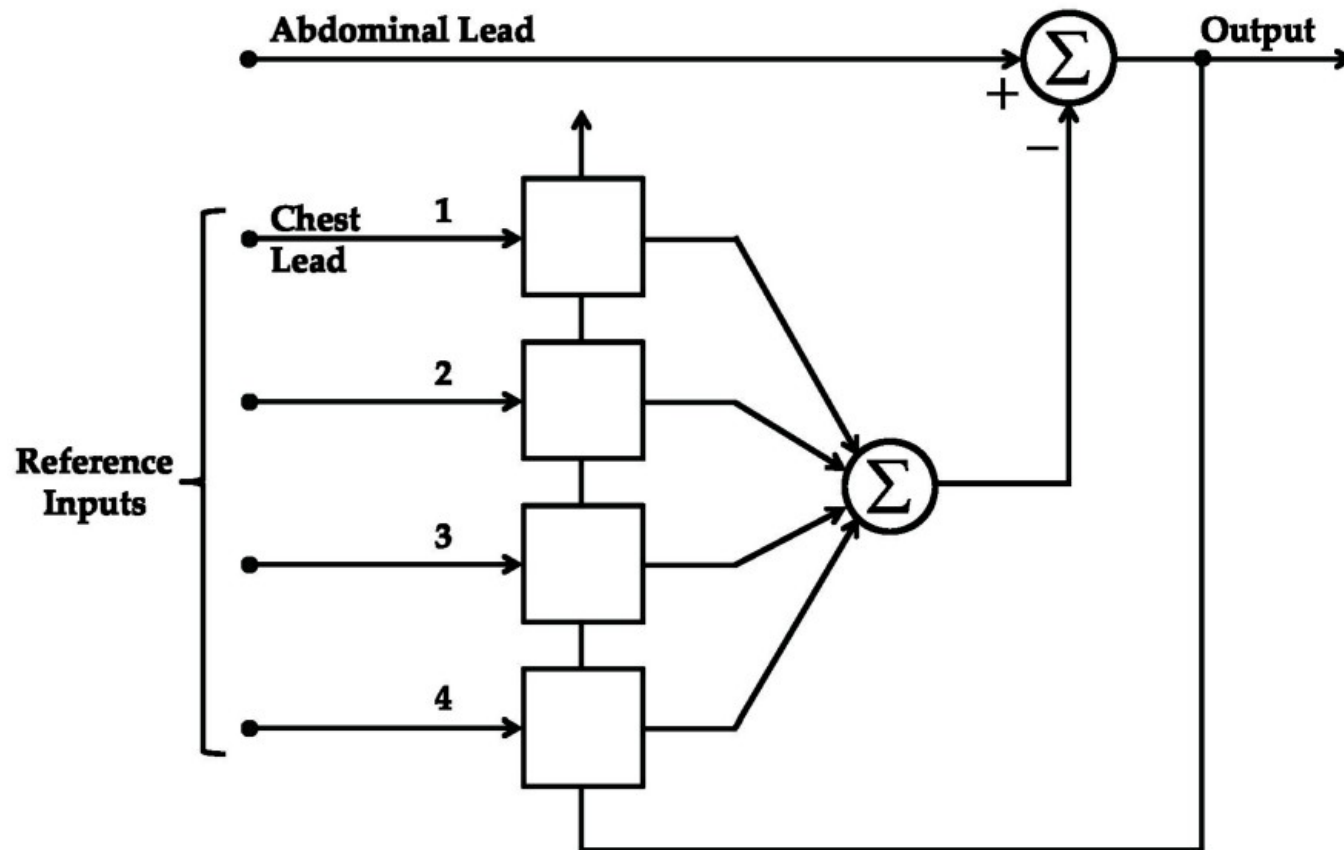
ADALINE analógico

O LMS se mostrou versátil não somente como um algoritmo de aprendizado para o Adaline, mas também como um algoritmo geral para ajuste dos parâmetros livres de sistemas adaptativos, sejam de banda larga ou de banda estreita. Em 1963 nasce a filha Deborah de Widrow, no Stanford Hospital. O médico residente de plantão na maternidade convidou o professor Widrow a visitar a enfermaria, onde ele tinha um osciloscópio Tektronix conectado por meio de uma chave seletora rotativa à dúzia de salas de trabalho. Ele estava monitorando eletrocardiogramas (ECGs) fetais. Era possível ver o ECG de Deborah antes de ela nascer.

Eletrodos abdominais captam sinais do coração do bebê, mas, forte interferência do coração da mãe presente em seu abdômen torna o ECG fetal difícil de interpretar ((a) na figura abaixo). O coração do bebê bate independentemente do coração da mãe. Um filtro passa-alta não pode separar esses sinais, já que a segunda harmônica do sinal da mãe é quase sempre a fundamental do sinal do bebê. O médico perguntou a Widrow: “o que pode ser feito?”.

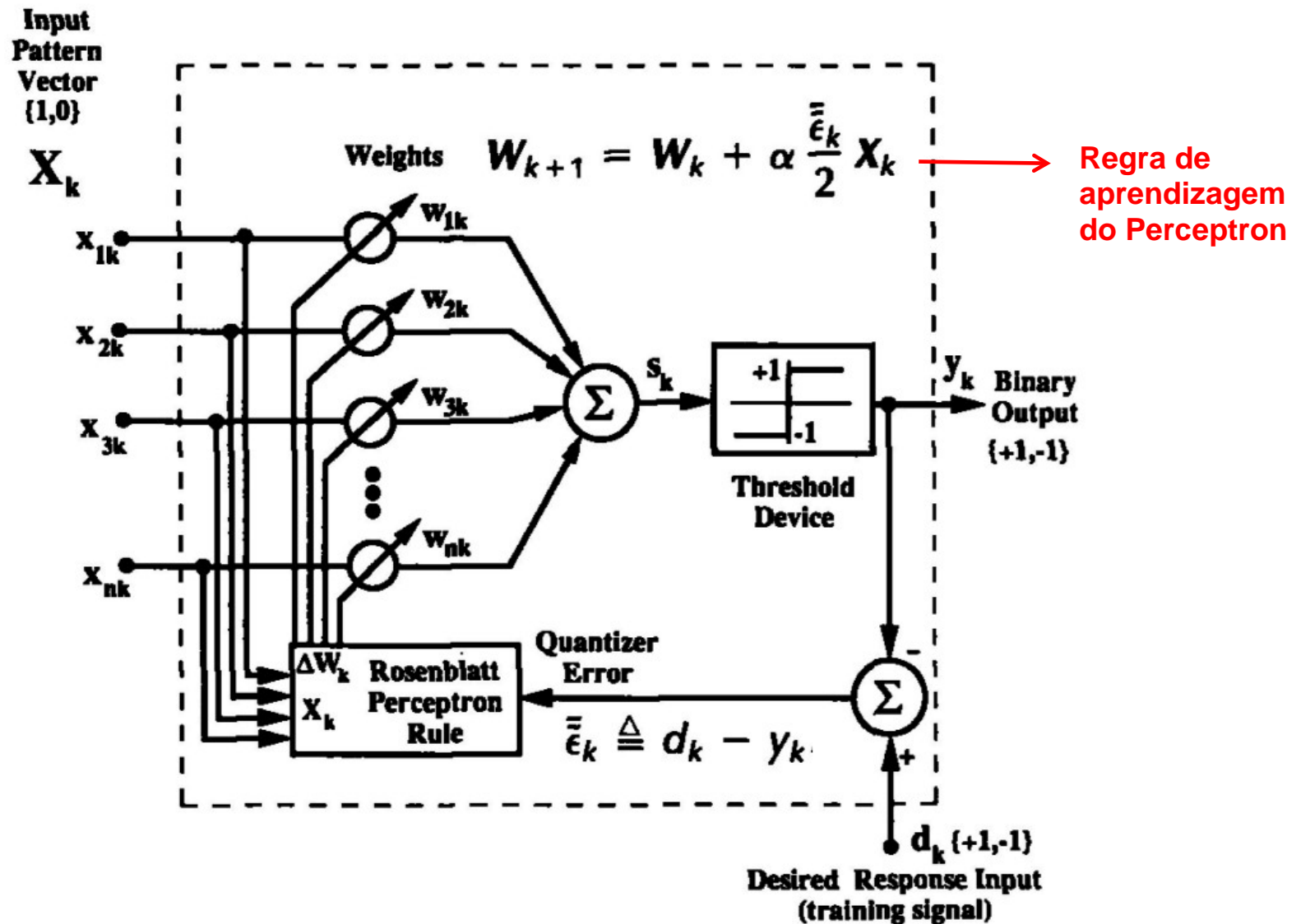


Parecia um problema de impossível solução. Widrow sugeriu que o problema do ECG fetal poderia ser solucionado através da seguinte abordagem: Colocar eletrodos de tórax na mãe para captar o sinal eletrocardiográfico, alto e claro ((b) na figura anterior). Processar este sinal com um filtro adaptativo ajustado pelo LMS e subtrair do sinal abdominal para eliminar a interferência materna, conforme mostra a figura abaixo. Isso envolve cancelamento de ruído adaptativo de banda larga (todo o espectro do sinal materno deve ser cancelado – não é apenas o cancelamento de , por exemplo, ruído de banda estreita de 60Hz). O resultado deste experimento – o cancelamento com êxito da interferência do sinal cardíaco materno – demonstrou que o cancelamento de ruído de banda larga era possível através do LMS.

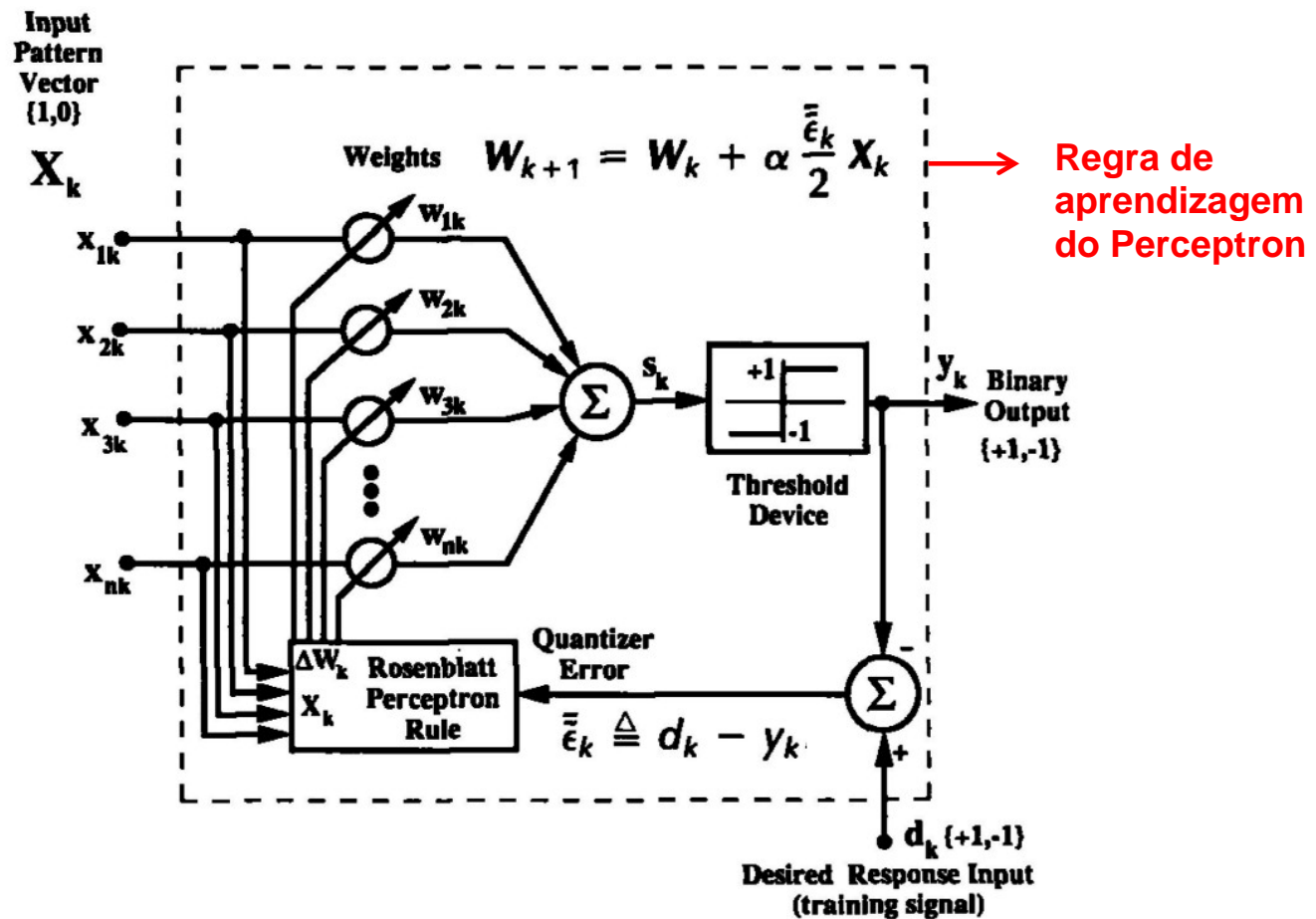


O Perceptron de Rosenblatt – o primo do Adaline:

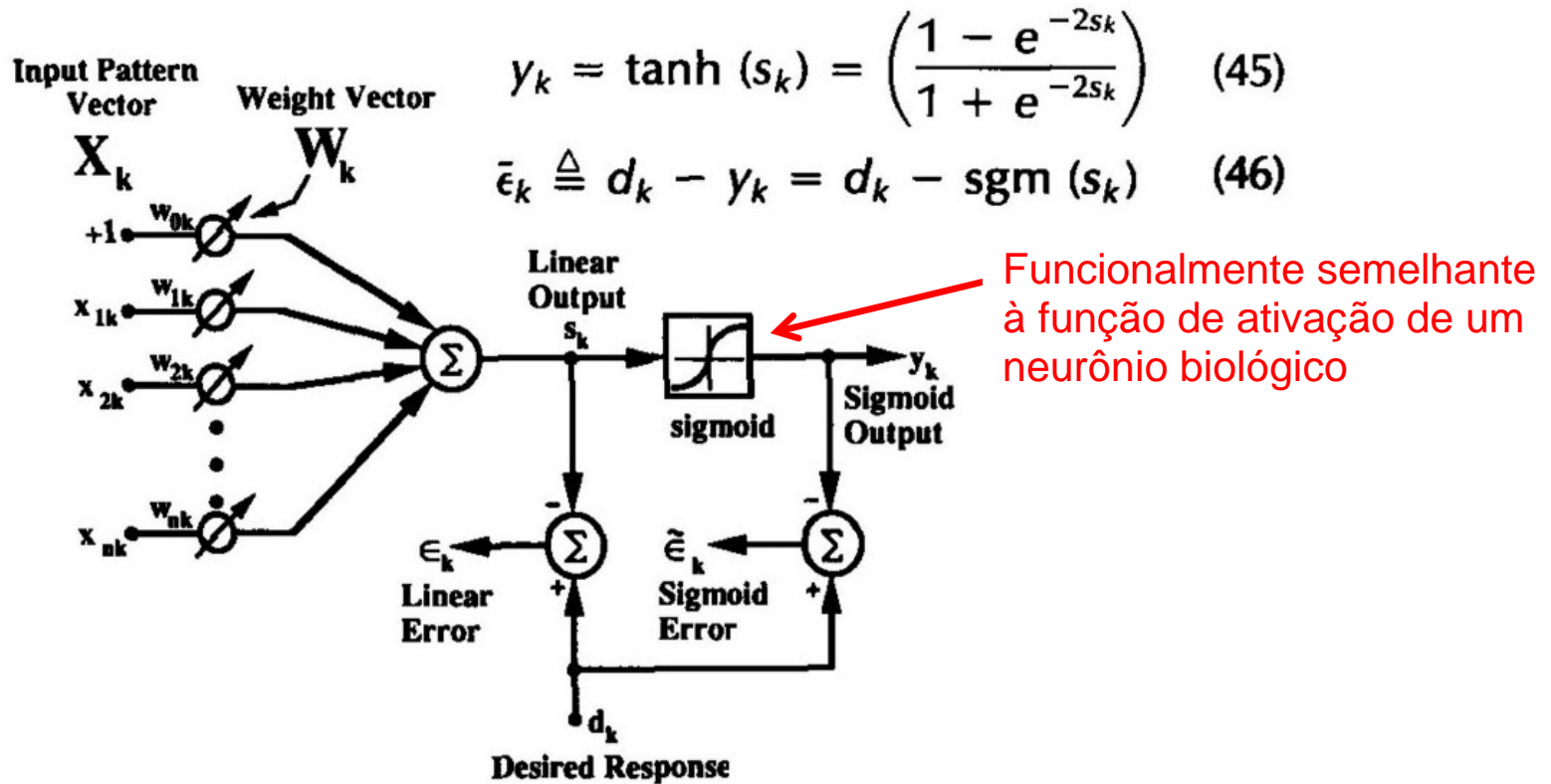
O Perceptron e sua regra de aprendizagem foi apresentada por Rosenblatt em 1960. Embora de arquitetura semelhante ao Adaline, o comportamento do Perceptron é bem distinto do Adaline.



A diferença entre o Adaline e o Perceptron começa no fato de a regra de aprendizagem do Perceptron não adaptar na iteração k o vetor de coeficientes \underline{W}_k se a decisão de saída y_k está correta, isto é, se $e_k = 0$. No entanto, se a decisão na saída y_k discorda da resposta binária desejada d_k , o ajuste adaptativo é efetuado adicionando o vetor de entrada \underline{X}_k ao vetor de coeficientes \underline{W}_k quando o erro e_k é positivo ou subtrai o vetor de entrada \underline{X}_k do vetor de coeficientes \underline{W}_k quando o erro e_k é negativo. Outras diferenças significativas: (I) As entradas do Adaline podem receber qualquer valor entre $[-1,1]$ enquanto no Perceptron elas devem ser binárias $\{0,1\}$. (II) Não há coeficiente para ajuste de *bias* no Perceptron. (III) O Adaline determina o erro a partir do sinal na entrada do *threshold device* enquanto o Perceptron determina o erro a partir do sinal na saída do *threshold device*.



O Perceptron/Adaline com não-linearidade sigmoial – a solução para o problema da descontinuidade no *threshold device* e a consequente indeterminação do gradiente



Adaline with sigmoidal nonlinearity.

The instantaneous gradient estimate obtained during presentation of the k th input vector X_k is given by

$$\hat{\nabla}_k = \frac{\partial (\bar{\epsilon}_k)^2}{\partial W_k} = 2\bar{\epsilon}_k \frac{\partial \bar{\epsilon}_k}{\partial W_k} \quad (47)$$

Differentiating Eq. (46) yields

$$\frac{\partial \bar{\epsilon}_k}{\partial \mathbf{W}_k} = -\frac{\partial \text{sgm}(s_k)}{\partial \mathbf{W}_k} = -\text{sgm}'(s_k) \frac{\partial s_k}{\partial \mathbf{W}_k}. \quad (48)$$

We may note that

$$s_k = \mathbf{X}_k^T \mathbf{W}_k. \quad (49)$$

Therefore,

$$\frac{\partial s_k}{\partial \mathbf{W}_k} = \mathbf{X}_k. \quad (50)$$

Substituting into Eq. (48) gives

$$\frac{\partial \bar{\epsilon}_k}{\partial \mathbf{W}_k} = -\text{sgm}'(s_k) \mathbf{X}_k. \quad (51)$$

Inserting this into Eq. (47) yields

$$\hat{\nabla}_k = -2\bar{\epsilon}_k \text{sgm}'(s_k) \mathbf{X}_k. \quad (52)$$

Using this gradient estimate with the method of steepest descent provides a means for minimizing the mean-square error even after the summed signal s_k goes through the non-linear sigmoid. The algorithm is

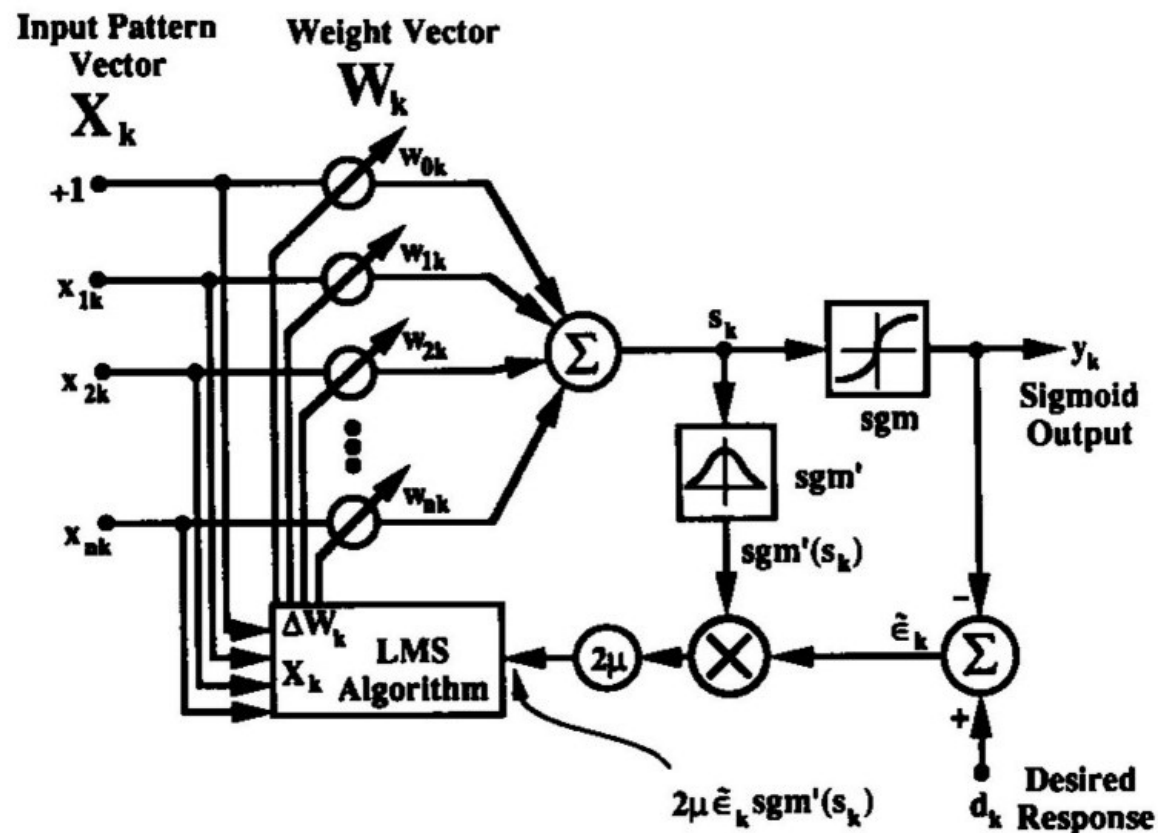
$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(-\hat{\nabla}_k) \quad (53)$$

$$= \mathbf{W}_k + 2\mu\bar{\epsilon}_k \text{sgm}'(s_k) \mathbf{X}_k. \quad (54)$$

Algorithm (54) is the **backpropagation algorithm** for the sigmoid Adaline element.

Rumelhart em 1986 generalizou o algoritmo backprop para Perceptrons de múltiplas camadas (Multilayer Perceptron – MLP), que é a base do *deep learning* (a ser visto mais adiante na disciplina).

$$\begin{aligned}
 \mathbf{W}_{k+1} &= \mathbf{W}_k + \mu(-\hat{\nabla}_k) \\
 &= \mathbf{W}_k + 2\mu\tilde{\epsilon}_k \text{sgm}'(s_k) \mathbf{X}_k
 \end{aligned}$$



Implementation of backpropagation for the sigmoid Adaline element.

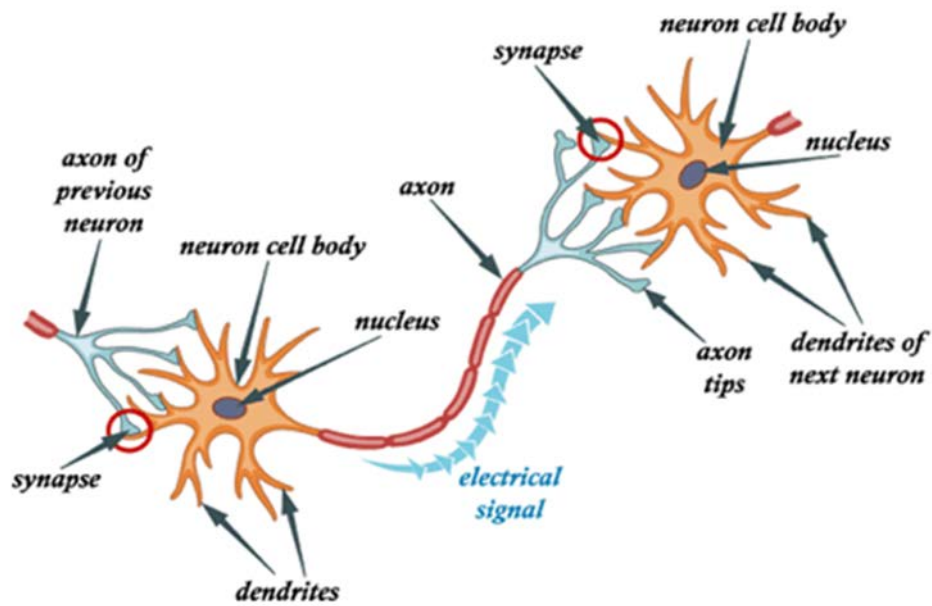
If the sigmoid is chosen to be the hyperbolic tangent function (45), then the derivative $\text{sgm}'(s_k)$ is given by

$$\text{sgm}'(s_k) = \frac{\partial(\tanh(s_k))}{\partial s_k} = 1 - (\tanh(s_k))^2 = 1 - y_k^2. \quad (55)$$

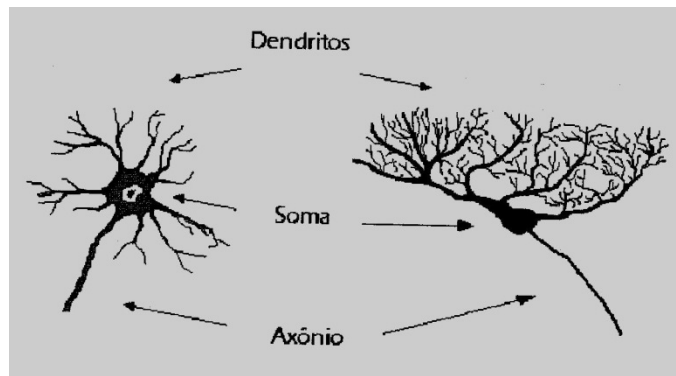
Accordingly, Eq. (54) becomes

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu\bar{\epsilon}_k(1 - y_k^2)\mathbf{X}_k. \quad (56)$$

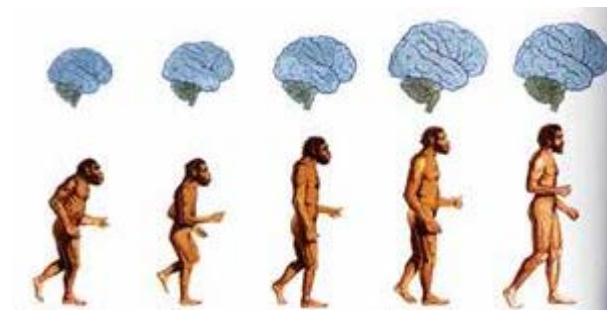
Inspiração biológica p/ o modelamento do neurônio artificial



Neurônio biológico

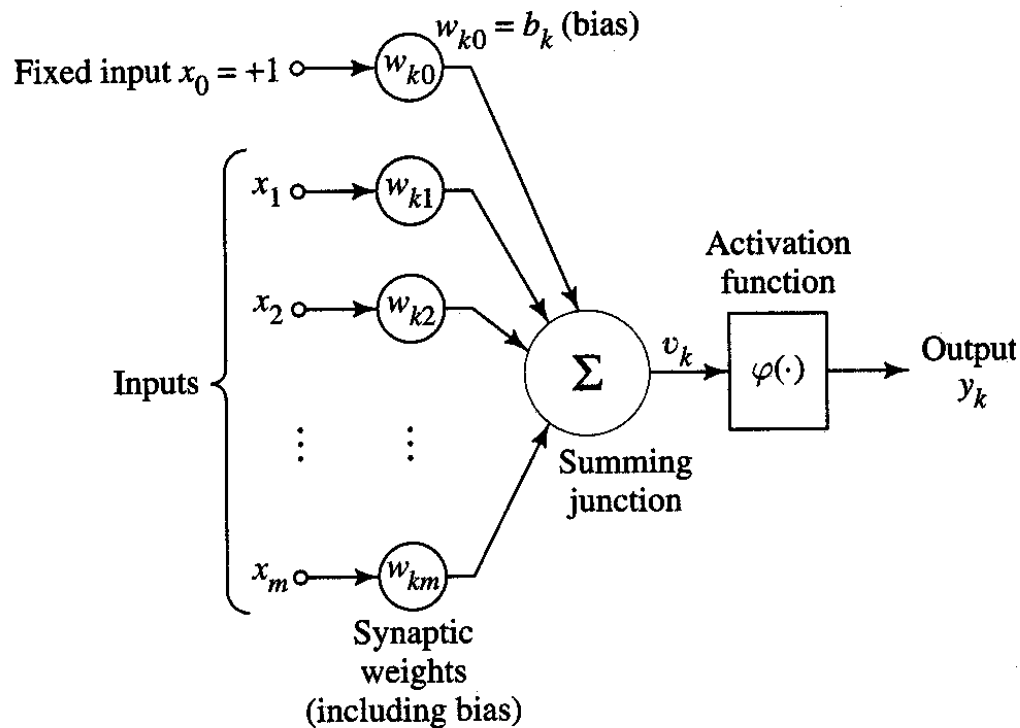
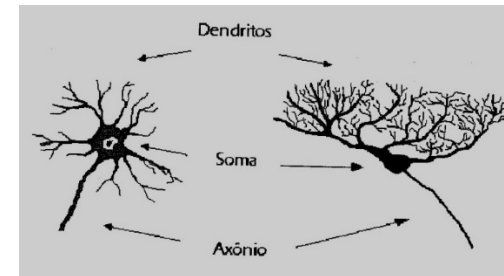


Rede neural biológica



Modelamento do neurônio artificial

Neurônios biológicos



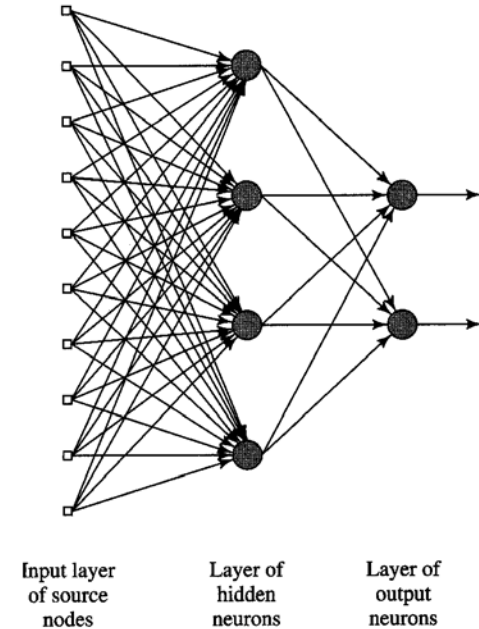
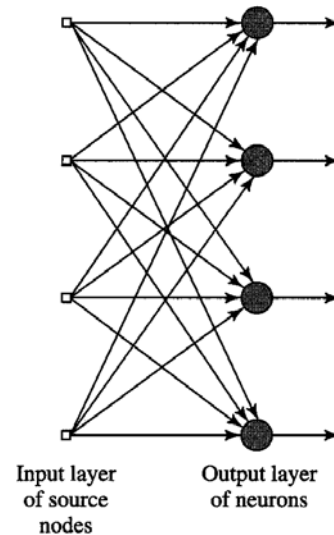
Modelo de um neurônio artificial:

Combinador linear (representando as sinapses/dendritos) seguidos de transmitância não-linear (representando a função de ativação)



Arquiteturas de RNAs

RNAs práticas são formadas por múltiplas camadas de neurônios densamente conectados por pesos sinápticos, em diferentes estruturas.



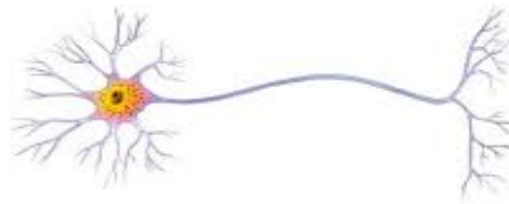
o que buscamos? ➡ modelar processos subjacentes presentes nos dados, quando não conhecemos o modelamento analítico ou estocástico do problema (quanto mais não-lineares e não-estacionários forem os processos, melhor!!!)

como é possível? ➡ por meio de estruturas complexas, altamente densas em conexões e construída em múltiplas camadas de neurônios fortemente não-lineares

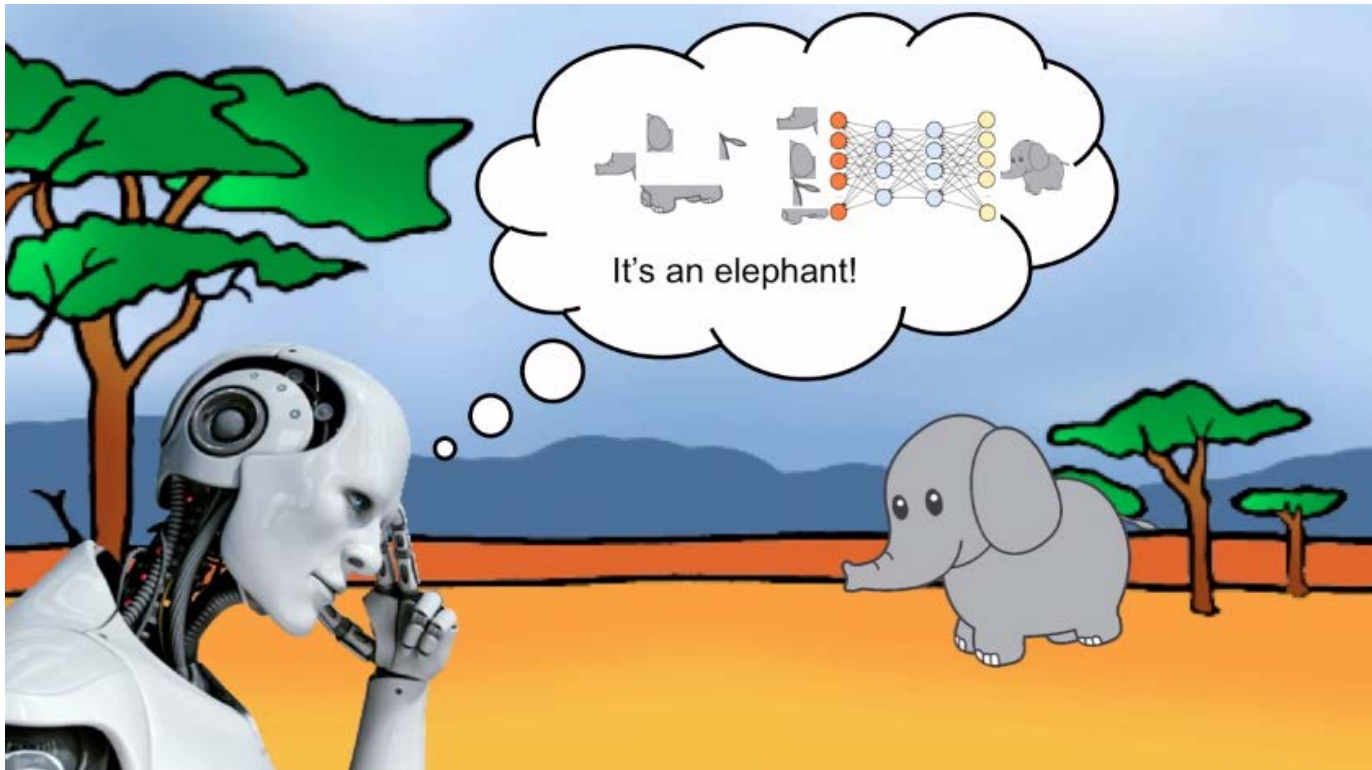
Processos de Aprendizagem



- RNAs têm a capacidade de obter conhecimento a partir do ambiente através de um processo de aprendizagem.
- O conhecimento obtido pelas RNAs é armazenado nos parâmetros livres da rede (pesos sinápticos e parâmetros que definem a(s) função(ões) de ativação dos neurônios).
- O procedimento utilizado para o processo de aprendizagem é chamado **Algoritmo de Aprendizagem** e tem por função modificar de forma adaptativa os parâmetros livres da rede para atingir um objetivo desejado.



Processos de Aprendizagem



- RNAs têm a capacidade de, através da informação de uma resposta desejada, aproximar um sinal alvo durante o processo de aprendizagem.
- Esta aproximação é obtida através do ajuste, de forma sistemática, de um conjunto de parâmetros livres, característico de cada rede neural.
- O conjunto de parâmetros livres provê um mecanismo para armazenar o **conteúdo de informação subjacente** presente nos dados que são apresentados à rede, na fase de treinamento.

Algoritmos Genéticos

Um Algoritmo Genético é um modelo que:

- Trata uma população de soluções de um determinado problema;
- Aplicando determinadas operações aos membros da população (indivíduos), de forma que:
 - A população resultante seja composta por indivíduos mais aptos do que a população inicial,
 - ou até que se selecione apenas o indivíduo mais apto.

Algoritmos Genéticos foram criados e formalmente descritos por John Holland, nos anos 60.



John Holland

- Cada um dos indivíduos \in à população representa uma solução completa do problema que se deseja tratar e tem validada a sua aptidão na execução da tarefa;
- A avaliação é feita não apenas de forma absoluta (evaluation), mas também de forma relativa aos demais indivíduos da população (fitness).
- São alocadas propriedades seletivas e reprodutivas, de tal forma que os indivíduos que representem melhores soluções p/ o problema alvo tenham mais chances de persistir ou de se reproduzir no conjunto de soluções.

População (de Genomas ou Cromossomos)

As soluções p/ o problema alvo são codificadas em estruturas que possam ser armazenadas e operadas computacionalmente:

As estruturas codificadas são chamadas genomas ou cromossomos, e são sequências de 0s e 1s.

- Um algoritmo genético cria, então, uma população de cromossomos tipicamente aleatória;
- Aplica operações de crossover, mutation & selection para gerar novas soluções, no espaço de busca.

- Note que um GA não calcula uma solução p/ o problema que está sendo tratado.
- Simplesmente seleciona e encoraja as “melhores soluções” (os melhores indivíduos), que surgem após certas operações aleatórias.
- Ou seja: não precisamos saber como resolver o problema alvo... apenas avaliamos a qualidade da solução gerada através de uma função de adequação do indivíduo à tarefa (fitness function).
- A convergência p/a solução é matematicamente comprovada (e depende da qualidade da implementação).

Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...

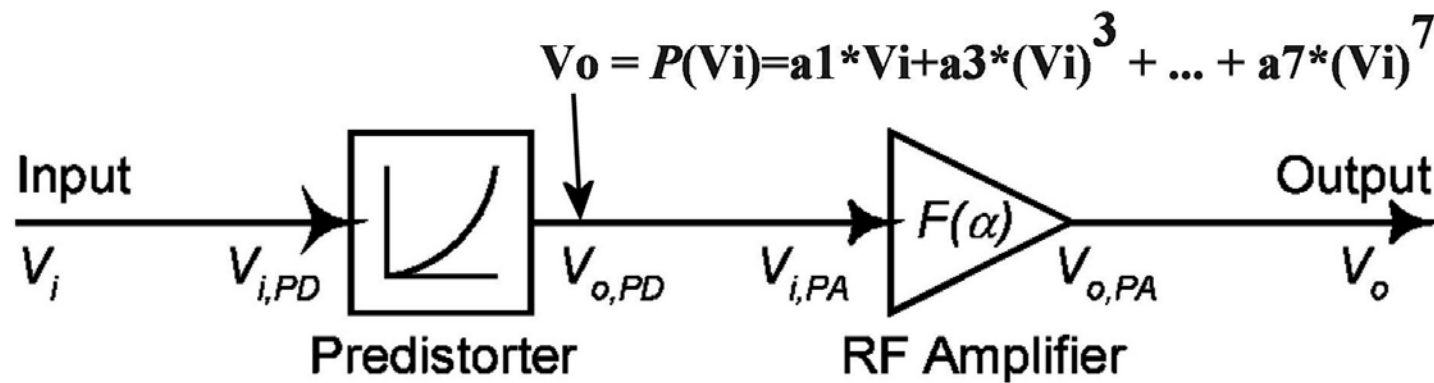
Por que GAs funcionam ?

- O processo de cruzamento e seleção conduz a sempre melhores soluções.
- Como as piores soluções são descartadas, a população resultante será sempre melhor, em um sentido global.
- Este processo é chamado pressão seletiva.

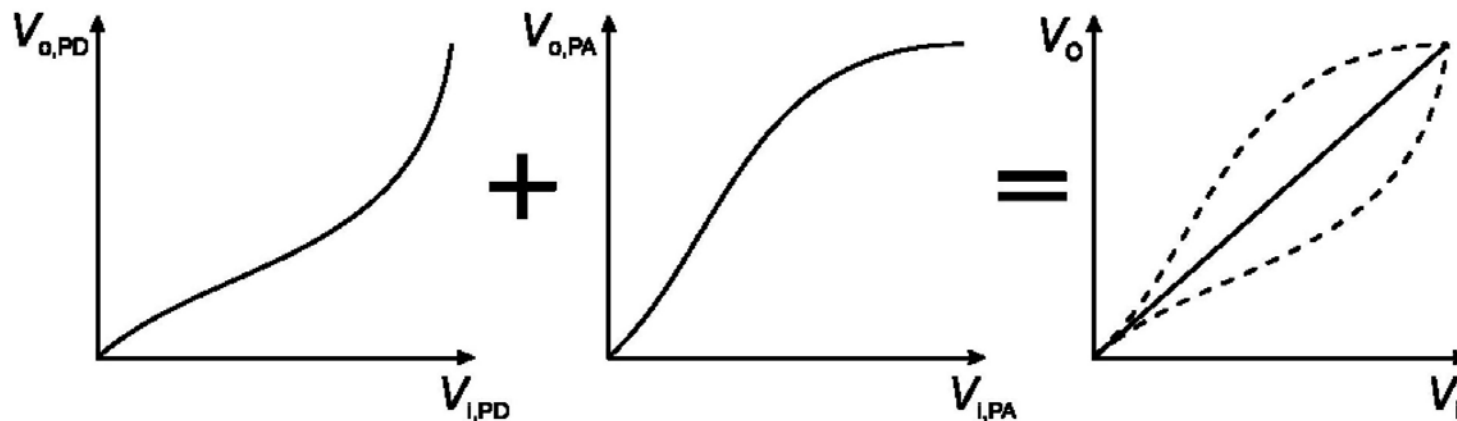
“Only the fittest of the fittest shall survive”

Bob Marley - COULD YOU BE LOVED

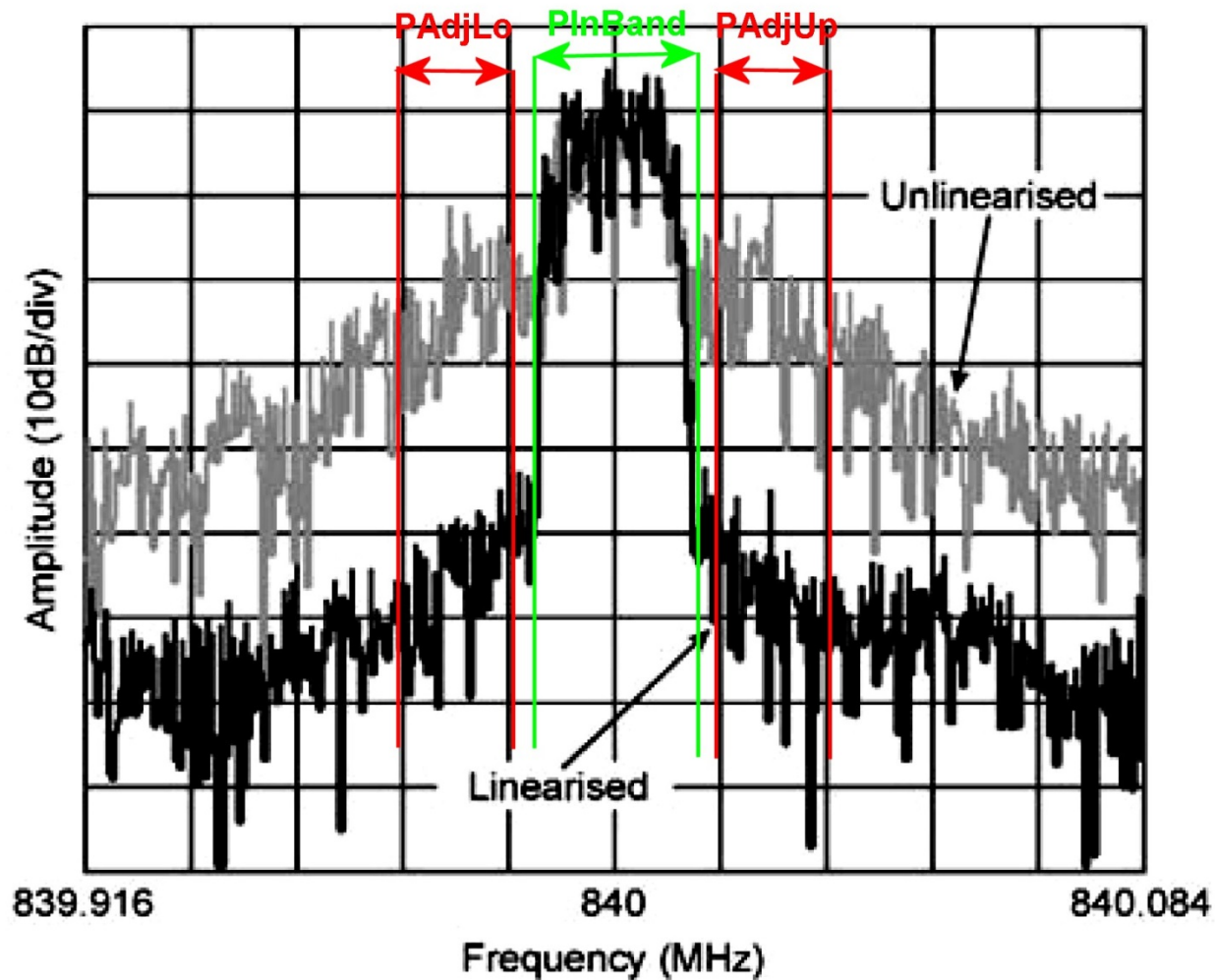
Exemplo - Linearização de HPA de RF através de GA binário



Objetivo: Determinar os coeficientes a_1, a_3, \dots, a_7 do polinômio $P(V_i)$ de forma a que $P(V_i)$ seja a função inversa do polinômio $\alpha(V_i)$ que descreve a não-linearidade do HPA, linearizando, portanto, a transmitância conjunta do Predistorter + HPA:

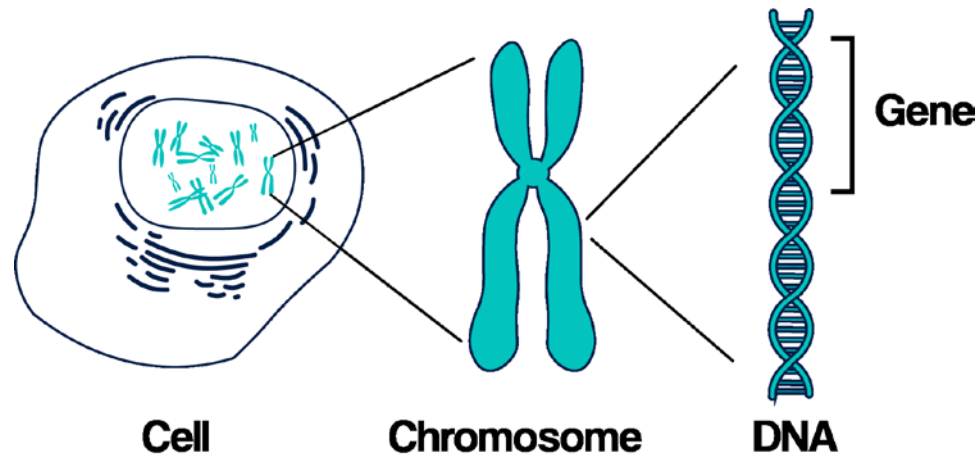


Indicador do grau de linearização (*fitness* para o GA): $Fitness = 1 - ACPR$, onde o $ACPR$ (*Adjacent Channel Power Ratio*) é obtido do espectro de saída do HPA conforme abaixo:

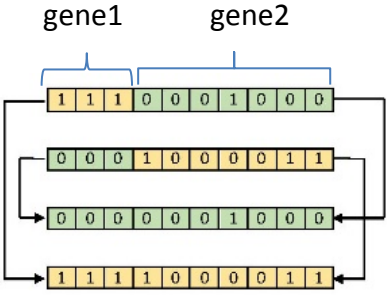


$$ACPR = \frac{P_{AdjLo} + P_{AdjUp}}{P_{InBand}}$$

Indivíduo, cromossomos e genes constituintes:

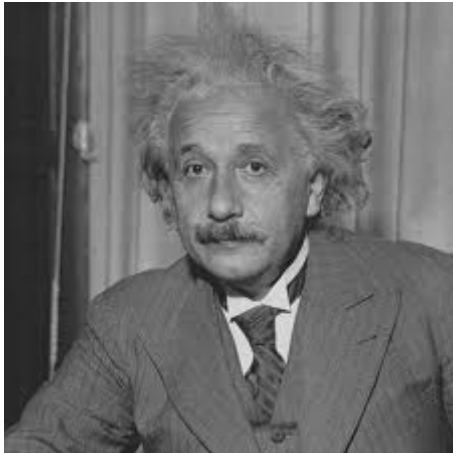


- Cada indivíduo da população é um polinômio de 4 coeficientes (a_1, a_3, \dots, a_7).
- Cada coeficiente é representado por uma palavra binária de 10 bits sequenciais (=cromossomo).
- Cada palavra binária de 10 bits sequenciais (cromossomo) é subdividida em duas sub-sequencias de bits (=dois genes).

Cromossomos	Operação
1010101010 1110001000 0001000011 0100001111 1101100011 0011101010 ...	<ul style="list-style-type: none"> • População de indivíduos (polinômios) é inicializada com valores aleatórios em cada cromossomo (coeficiente do polinômio), mas de valores pertinentes ao problema em questão. • Cada linha de números binários ao lado representa um dos 4 cromossomos (coeficientes) de cada indivíduo (polinômio) na população. É usual converter o número em ponto flutuante p/a a sua representação inteira normalizada de modo que o menor número inteiro corresponda ao menor número em ponto flutuante e o maior número inteiro corresponda ao maior número em ponto flutuante. • <i>Fitness</i>: Função que caracteriza a linearização do HPA ($Fitness=1 - ACPR$) para cada indivíduo (polinômio). • Desejamos reproduzir os indivíduos (polinômio) com maior <i>fitness</i>.
1010101010 1110001000 0001000011 0100001111 ...	<ul style="list-style-type: none"> • Avaliação do fitness da população (seleção de potenciais reprodutores) <ul style="list-style-type: none"> – Avalia-se o <i>fitness</i> de cada indivíduo (polinômio) da população inteira e reordena-se os indivíduos (polinômios) em ordem crescente de <i>fitness</i>. – Apenas indivíduos (polinômios) com <i>fitness</i> maior que um determinado <i>threshold</i> determinado experimentalmente são selecionados para cruzamento.
	<ul style="list-style-type: none"> • Cruzamento (reprodução) <ul style="list-style-type: none"> – Os indivíduos selecionados p/ cruzamento são aleatoriamente selecionados em pares, tendo os indivíduos de maior <i>fitness</i> a maior probabilidade de seleção (<i>roulette wheel</i>). Cada par de indivíduos (polinômios) selecionados gera dois novos filhos (polinômios) através da troca de material genético entre cada correspondente cromossomo (coeficiente). Cada par de cromossomos (coeficientes) respectivos a um par de indivíduos (polinômios) troca genes entre si, isto é, troca duas sub-sequências binárias formadas de cada coeficiente, conforme mostrado ao lado. O tamanho de cada gene é aleatório.
0000001000 1111000011 1010001111 0100101010 ...	<ul style="list-style-type: none"> • Prole <ul style="list-style-type: none"> – Os polinômios-filho têm alta probabilidade de linearizarem o HPA, porque ambos os polinômios-pais têm cromossomos que resultaram em um alto <i>fitness</i>. – As novas sequências binárias dos filhos contêm porções das sequências binárias de ambos os pais.

<pre> 0001101000 110000011 1010100111 0100101010 ... </pre>	<ul style="list-style-type: none"> • Mutaç�o <ul style="list-style-type: none"> – Para evitar o “elitismo” na seleç�o, inverte-se aleatoriamente em uma pequena parcela (~10%) de indiv�duos (polin�mios) da populaç�o alguns poucos bits dos seus cromossomos (coeficientes).
<pre> 1110001000 000100011 0100001111 1010101010 0000001000 1111000011 1010001111 0100101010 </pre>	<ul style="list-style-type: none"> • Nova Populaç�o <ul style="list-style-type: none"> – Cada dois novos filhos substituir�o os dois pais, portanto n�o h� altera�o no n�mero de indiv�duos da populaç�o – Volta-se � etapa “Avalia�o do fitness da populaç�o” e o processo � executado recorrentemente at� que o Fitness=1 – ACPR de um indiv�duo (polin�mio) da populaç�o seja pr�ximo de 1.0 de modo que o ACPR � pr�ximo de zero, indicando que o polin�mio lineariza o HPA.

Lógica Fuzzy

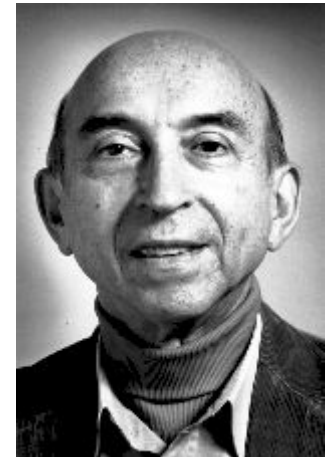


“À medida que as proposições matemáticas se referem à realidade, elas não são certas; e, à medida que são certas, elas não se referem à realidade.”

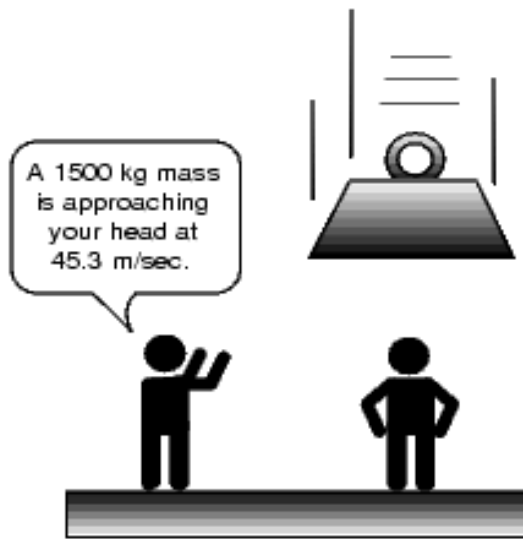
Albert Einstein, 1928

“Dito informalmente, a essência do princípio da incompatibilidade é que, à medida que a complexidade de um sistema aumenta, nossa habilidade de fazer afirmações precisas e significativas sobre o seu comportamento diminui até um limite além do qual precisão e relevância se tornam características quase mutuamente exclusivas.”

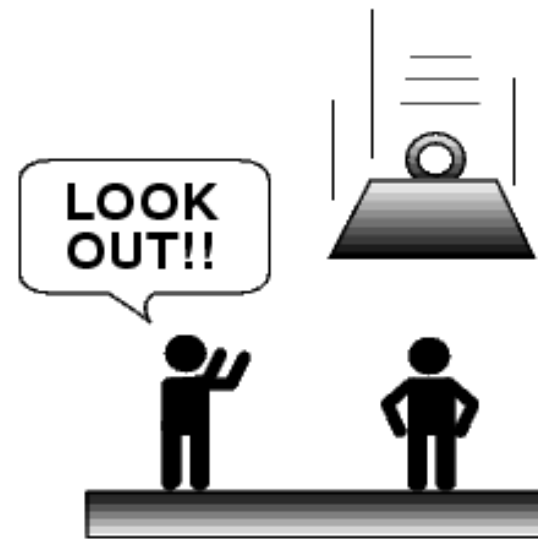
Lotfi Zadeh, 1973



Precision and Significance in the Real World



Precision



Significance

Lógica Fuzzy (Fuzzy Sets, Lógica Nebulosa) foi concebida e formalmente descrita por Lotf Zadeh, nos anos 60.

- Generalização da Teoria dos Conjuntos
- Representa e trata “noções linguísticas” → objetos com limites não definidos
- Lógica multi-valores x Lógica clássica binária

Problema da Dicotomia

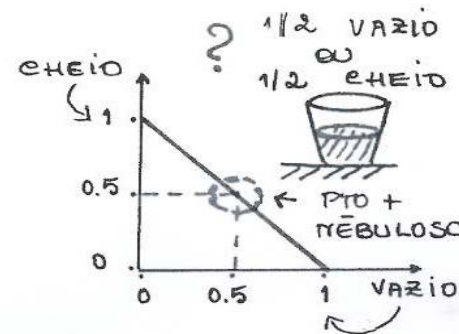
Uma semente não constitui um monte de sementes, nem 2 ou 3... por outro lado, 100 milhões de sementes constituem um monte de sementes!

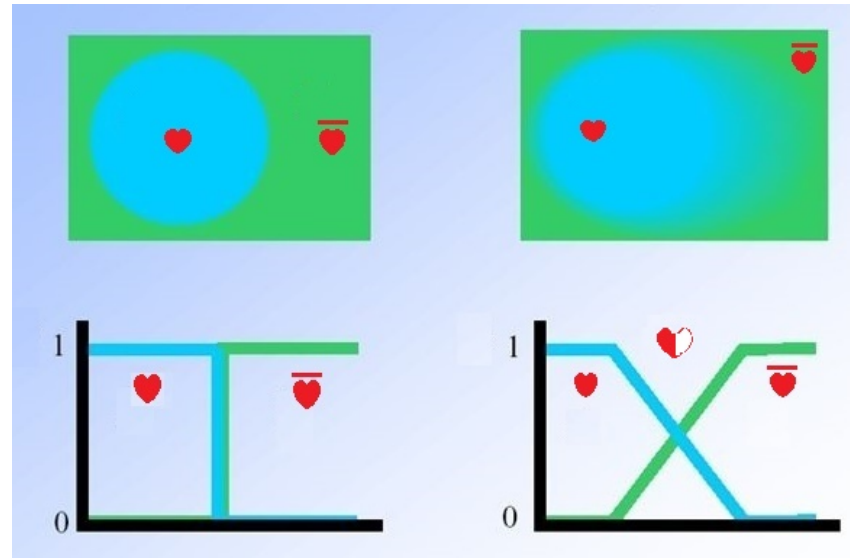
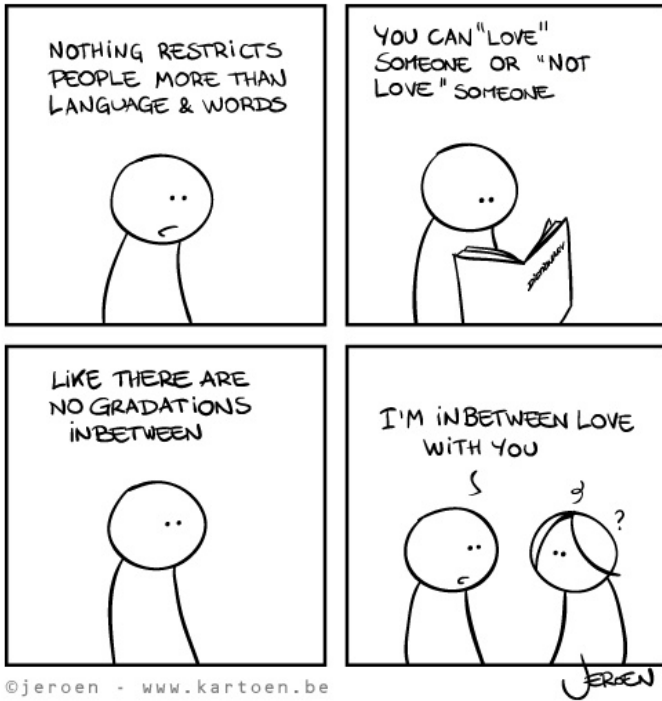
Qual é o limite apropriado?

325647 sementes não constituem um monte de sementes, mas 325648 constituem?

- dogma sim/não ... V/F ... 0/1
- Uma sentença pode, portanto, ser V ou F, ou não ter um valor determinado como V ou F, mas, sim, INDETERMINADO.
- Podemos atribuir um número infinito de valores contidos entre V ou F, a tais sentenças de sentido indeterminado.

Como fazer uma máquina lidar com imprecisão e incerteza?





"in love or not in love"

"in-between love"



Engradado A:

Cuidado! Garrafas com 50% de probabilidade de conter veneno



Engradado B:

Cuidado! Garrafas com grau de pertinência 0.5 (em $[0,1]$) relativo ao conceito "conter veneno".

Qual a escolha? Garrafas de A ou Garrafas de B?

Teoria clássica dos conjuntos:

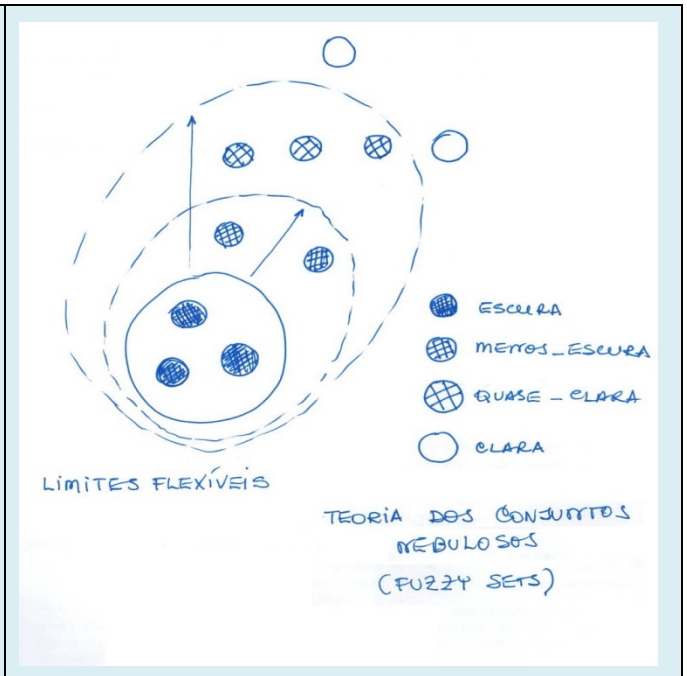
- Trata coleções de objetos que compartilham alguma característica de propriedade, tais como: números reais, roseiras floridas em um jardim, frutas em uma cesta, filés na chapa...
- Conjuntos introduzem a noção de dicotomia, que impõe uma noção binária: sim ou não, tudo ou nada, pertencente ou não pertencente a uma categoria.
- Mas, na realidade, existem situações como: roseiras muito floridas em um jardim, frutas maduras em uma cesta, filés mal passados, *baixa temperatura, pequeno erro, rápida resposta de um sistema, ...*

Teoria de conjuntos nebulosos:

- Trata classes cujos limites não são bem delimitados, compreendendo: aritmética nebulosa, programação matemática nebulosa, grafos nebulosos e a “fuzzificação” de teorias clássicas.
- A Teoria de conjuntos nebulosos inclui a lógica nebulosa.

Lógica nebulosa:

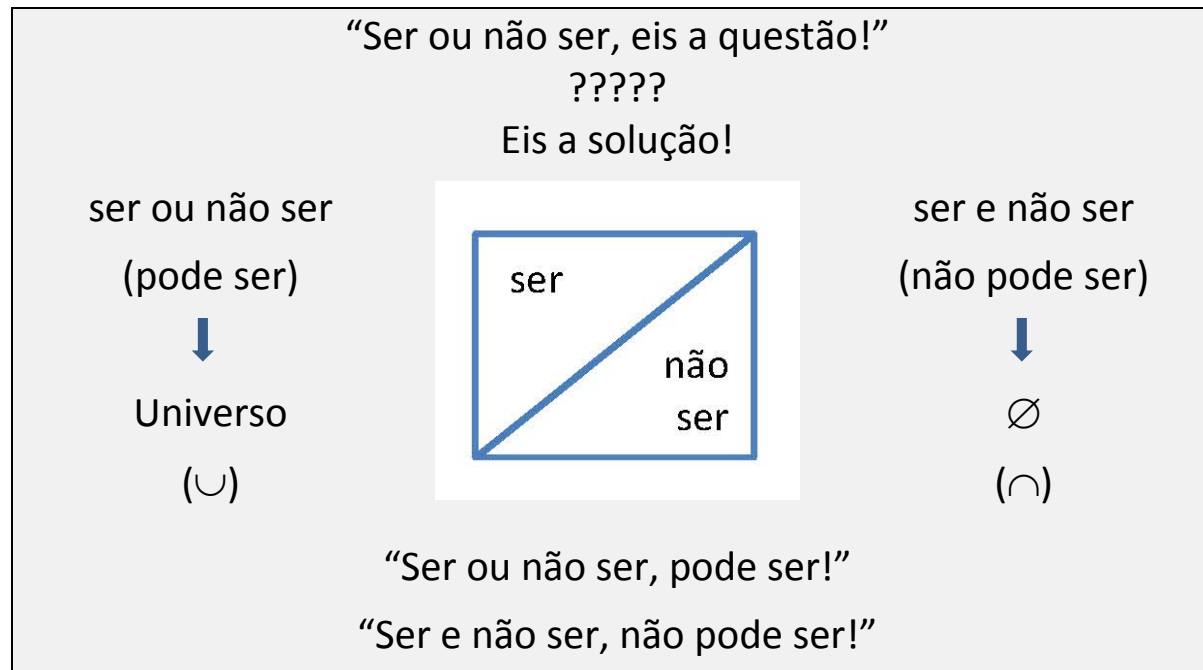
- Sistema lógico que formaliza o raciocínio aproximado, compreendendo: variáveis linguísticas, regras se-então...



Zadeh:

- FS são ferramentas que possibilitam “computar” (operar) com palavras (linguística), quando a informação disponível é muito imprecisa para justificar o uso de números”.
- FS permitem: tratar tais problemas e obter soluções de baixo custo e mais associadas com a realidade.
- FS utilizam variáveis linguísticas + regras linguísticas.
- Variáveis linguísticas, uma vez convertidas em funções adequadas (de pertinência), permitem processar numericamente conceitos de natureza imprecisa.

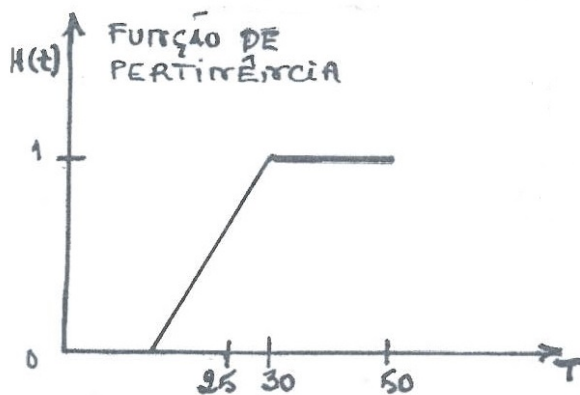
computando com palavras, um exemplo



Regras Linguísticas



- Fuzzy sets são caracterizados por funções de pertinência.
- O valor da função de pertinência descreve o grau de pertinência da variável linguística em questão, com relação ao conceito que ela descreve (o quão relacionada a variável é com o conceito).



O valor de $H(t)$ descreve o grau de pertinência de t em H .

t : variável linguística

T : universo de discurso de t , $T=[0,50]$

*Note que aqui consideramos o conceito de Alta Temperatura (HT) em um contexto ambiental, com temperaturas distribuídas no intervalo $[0, 50^\circ\text{C}]$.

Caracterização do FS H de “Alta Temperatura (HT)” no universo de discurso $T=[0,50]$

Temperatura em questão $t \in T$	Relação com o conceito “Alta Temperatura = HT”	Valor de $H(t)$ relacionado à compatibilidade de t com o conceito de HT
0°C	Não é entendido como HT	Grau de Pertinência = 0
$t \geq 30^\circ\text{C}$	Certamente é entendido como HT	Grau de Pertinência = 1
$30^\circ\text{C} < t \leq 50^\circ\text{C}$	Certamente é entendido como HT	Grau de Pertinência = 1
$12.5^\circ\text{C} \leq t < 30^\circ\text{C}$	≠s relações com o conceito	≠s valores em $[0,1]$

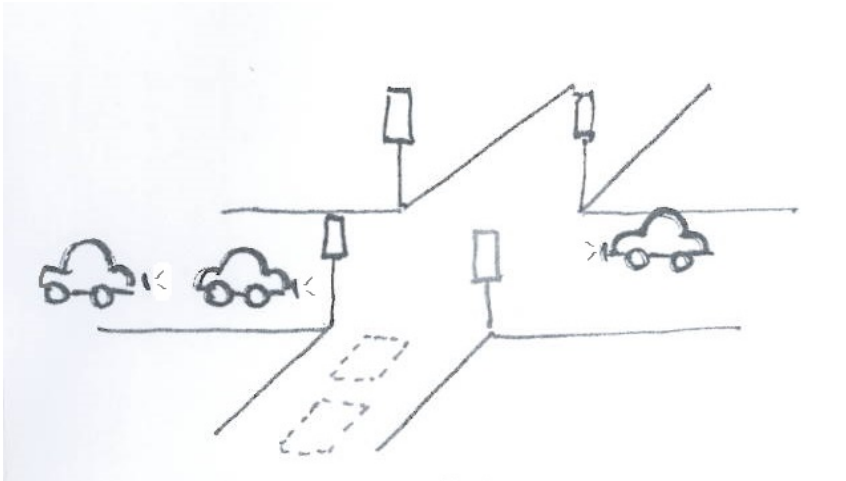
Tráfego: controle de semáforos



luzes, faróis,
sinaleiras,
semáforos, ...

- Tarefa complexa, diversos fatores a serem considerados.
- Qualquer modelo (determinístico ou estocástico) usado necessita da estimação de um substancial nº de parâmetros.
- Ambiente de controle altamente não-estacionário ou seja, um modelo projetado p/ uma situação específica não representará igual desempenho sob condições variáveis (tipo predominante de veículos, proximidade de outros cruzamentos próximos, hospitais, escolas, bombeiros, ladeiras ou avenidas planas, ...)

Abordagem Fuzzy



Encapsular todo o domínio do conhecimento disponível e organizar de forma tratável um conjunto de regras **If - Then** que construirão o protocolo de controle adequado.

Variáveis linguísticas:

entrada – tamanho da fila, chegada de novos veículos...

saída – tempo de permanência no verde

Labels das variáveis linguísticas (termos linguísticos):

longo, muito curto, não tão longo, não tão longo e não tão curto...
(tamanho da fila)

muitos, poucos, não tão poucos, não muitos...(chegada de novos veículos)

muito breve, breve, médio, longo... (tempo de permanência no verde)

Regra linguística:

If \exists muitos novos veículos chegando à fila já formada e o tamanho da fila é longo,

Then o tempo de permanência no verde é longo.

REDES NEURAIS PARA CONTROLE DE TAXA EM CODIFICADORES MPEG

O problema do controle de taxa em MPEG2 e a solução proposta:

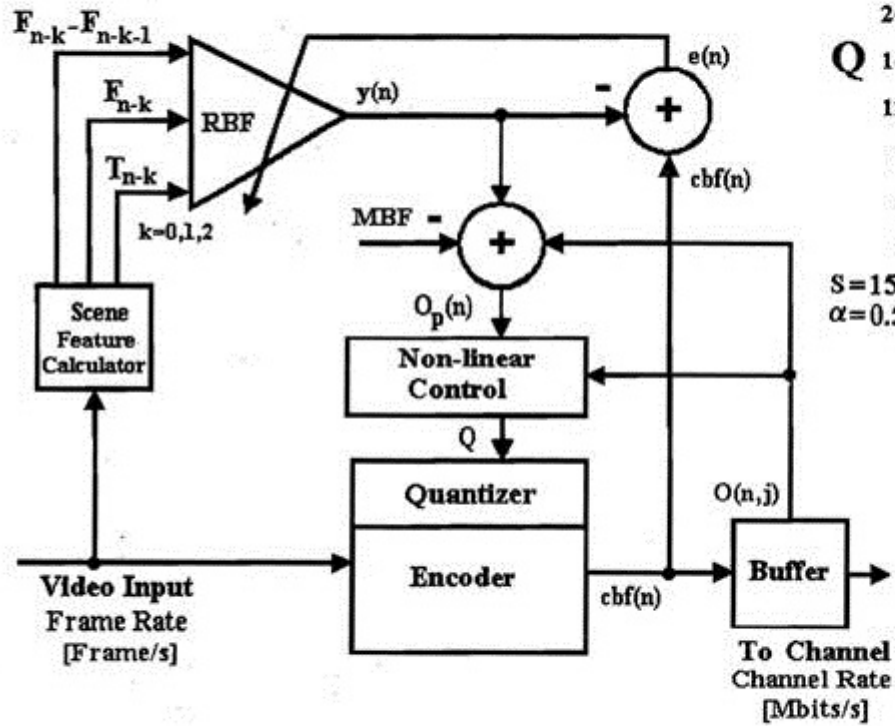


Figura 1: Controlador neural CBR para codificadores MPEG2.

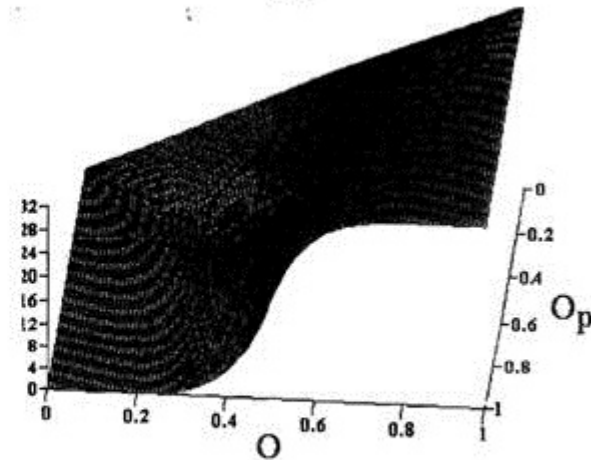
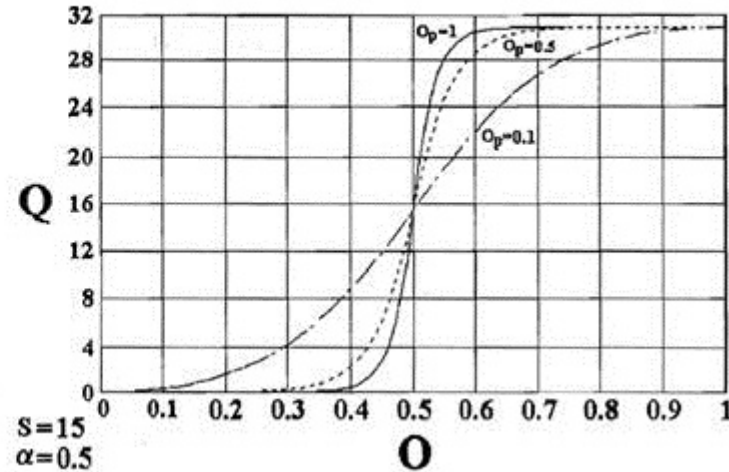


Figura 2: Curva de Controle.

O treinamento de uma rede Neural RBF:

$$w_k(n+1) = w_k(n) + \mu_w e(n) \varphi(u(n), t_k(n))$$

$$t_k(n+1) = t_k(n) + 2\mu_t e(n) w_k(n) \varphi(u(n), t_k(n)) \frac{u(n) - t_k(n)}{\sigma_k^2(n)}$$

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n) \varphi(u(n), t_k(n)) \frac{\|u(n) - t_k(n)\|^2}{\sigma_k^2(n)}$$

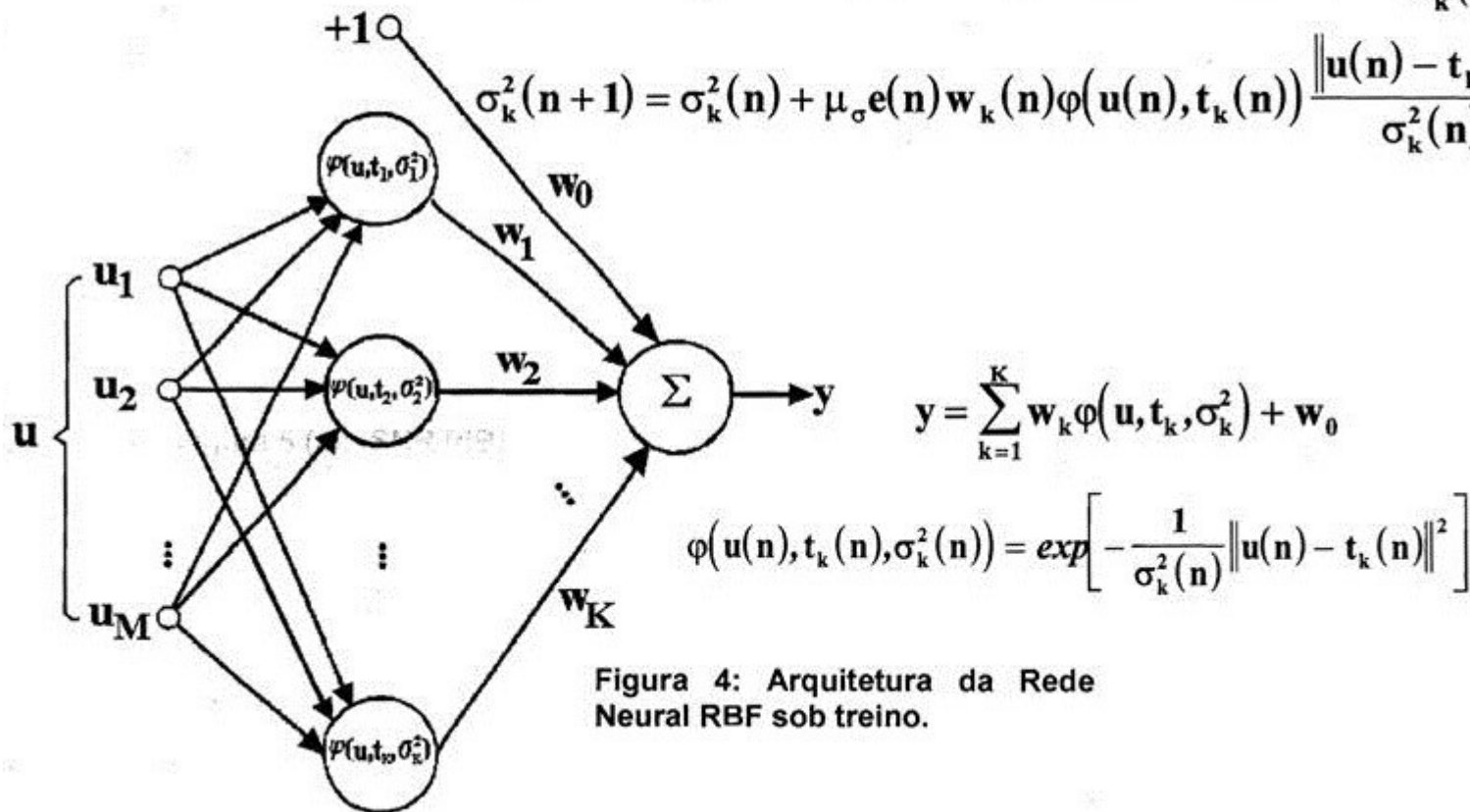


Figura 4: Arquitetura da Rede Neural RBF sob treino.

REDES NEURAIS PARA CONTROLE DE TAXA EM CODIFICADORES MPEG

Resultados obtidos para a seqüência "Kiel" a uma taxa de 3.5 Mbps no canal:

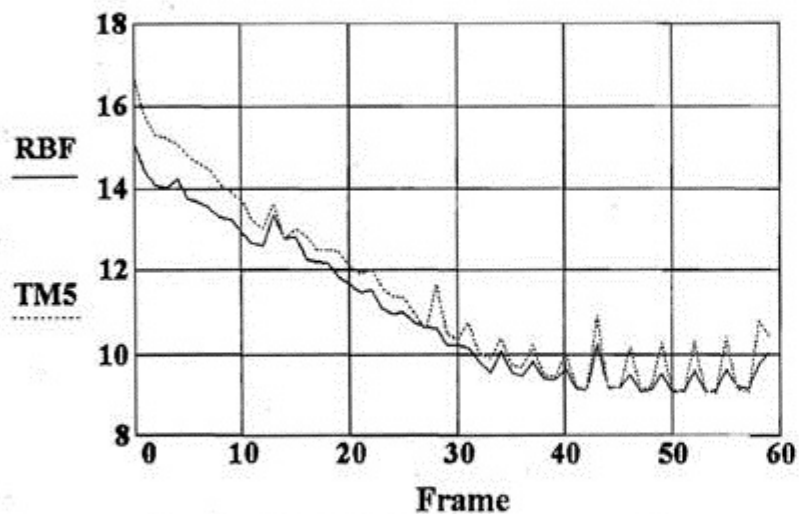


Figura 5 (a): SNR [dB] .

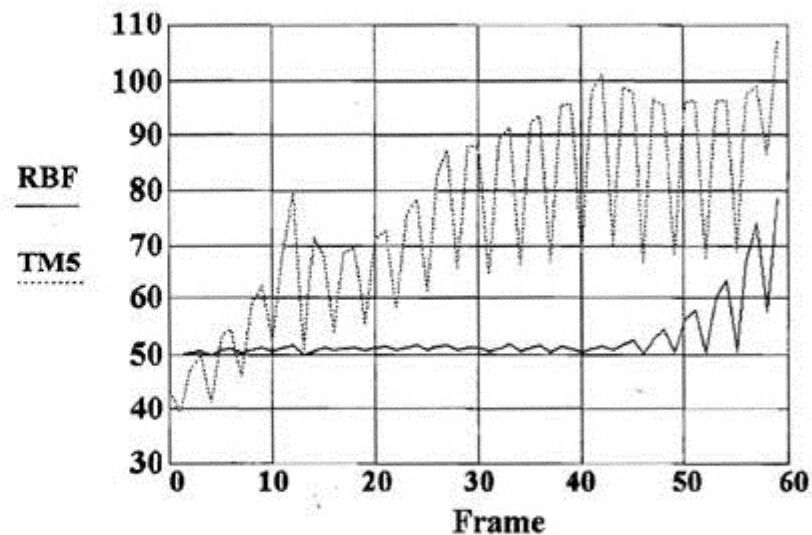


Figura 5 (b): Ocupação do *buffer* [%] .

REDES NEURAIS PARA CONTROLE DE TAXA EM CODIFICADORES MPEG

Resultados obtidos para a seqüência "Mobile" a uma taxa de 3.5 Mbps no canal:

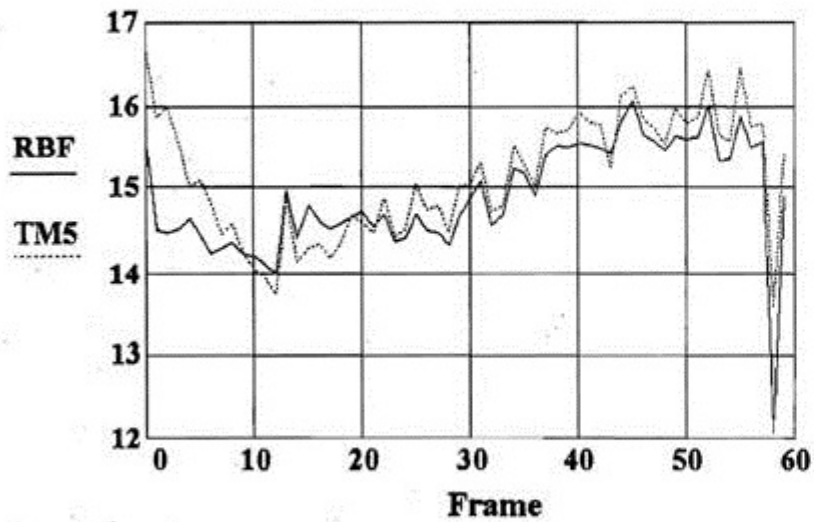


Figura 6 (a): SNR [dB] .

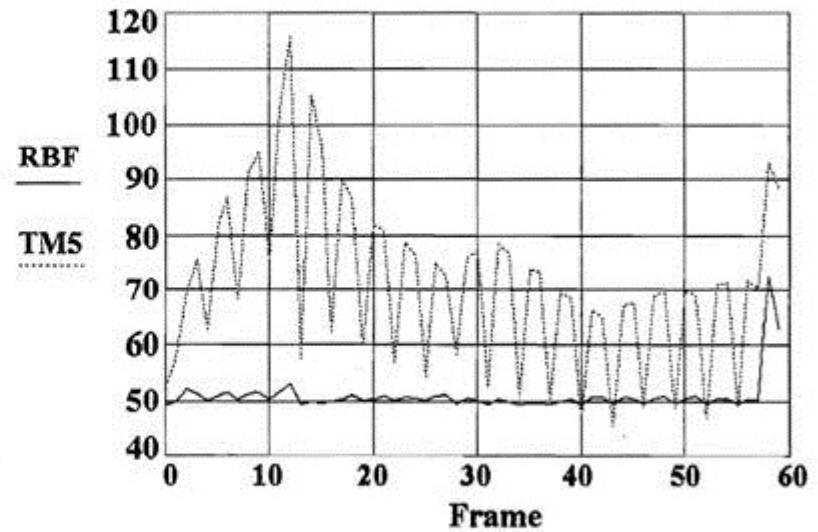
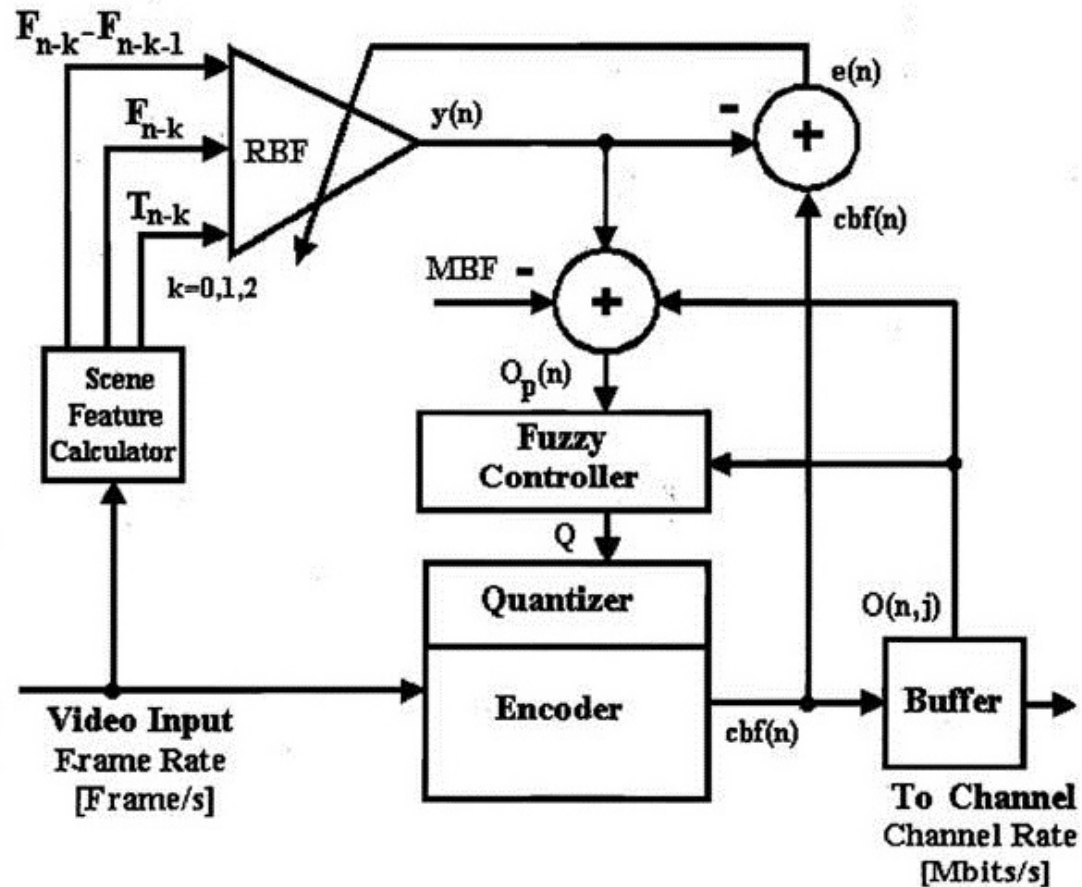


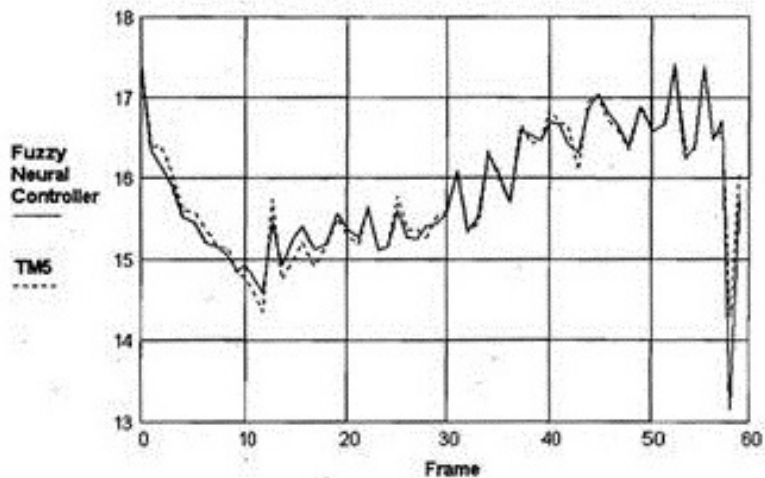
Figura 6 (b): Ocupação do *buffer* [%] .

The Fuzzy Neural Predictive Video Bit Rate Controller

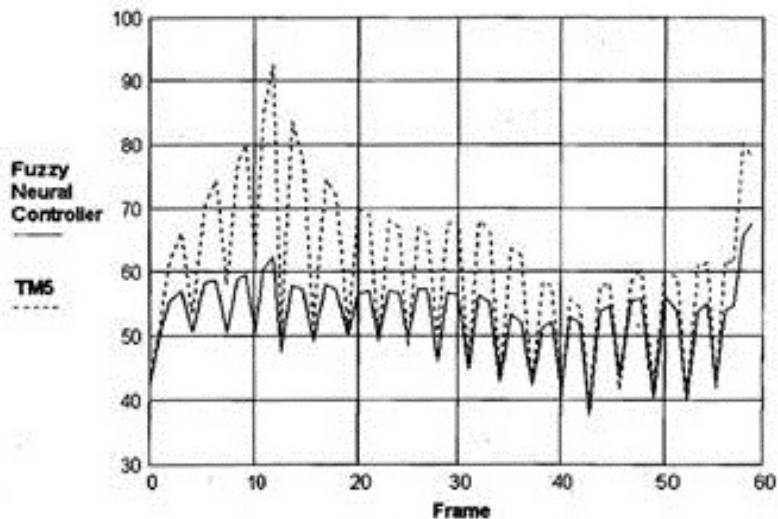


MBF= taxa de canal/taxa de frames
[Mbits/frame]

Mobile Sequence

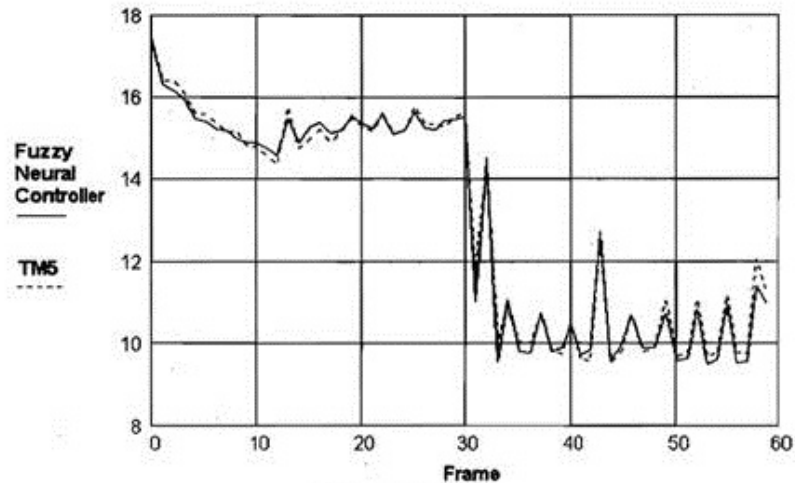


SNR [dB]

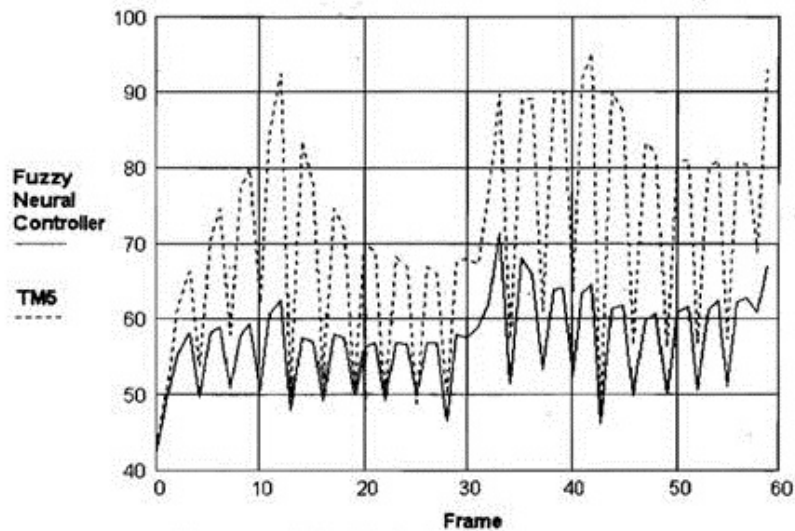


Percent Buffer Occupancy

Mobile & Kiel Sequence

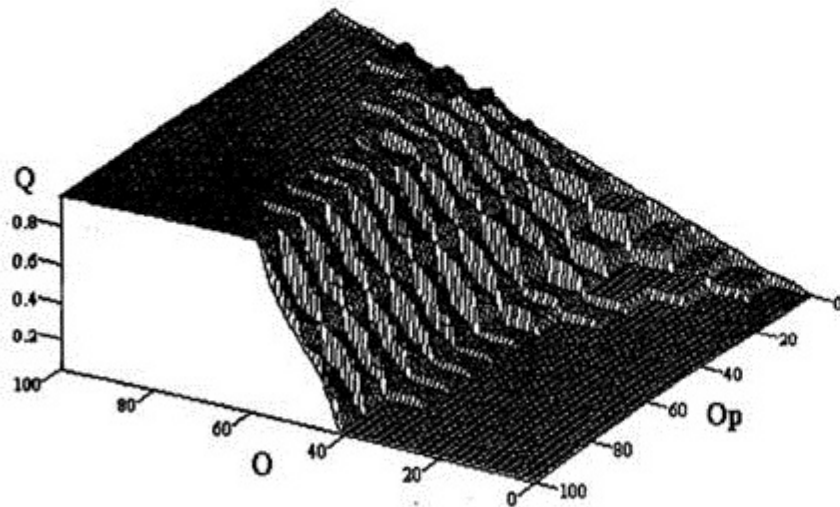


SNR [dB]

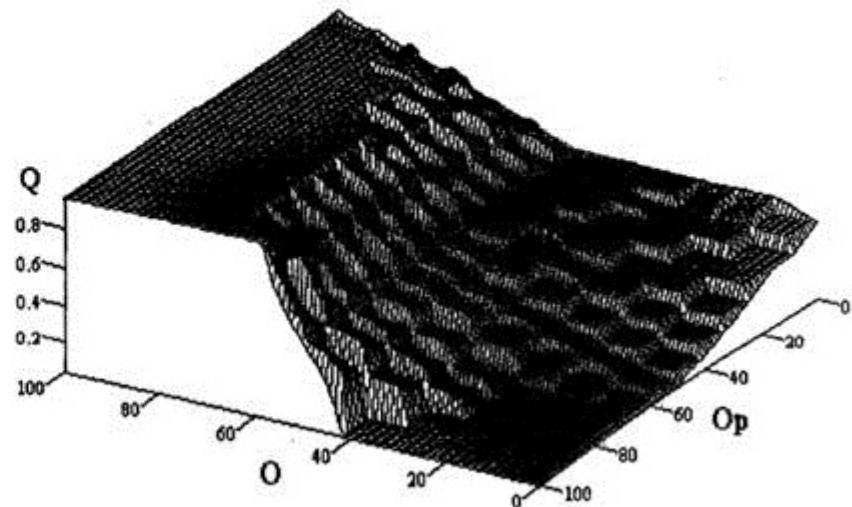


Percent Buffer Occupancy

<u>Resulting fuzzy rules:</u>	<u>Aggregation rule:</u>	<u>Defuzzification:</u>
connective and → min t-norm decision-making logic → Larsen conjunction	union operator → max s-norm	center of area method



Fuzzy Control Surface 1



Fuzzy Control Surface 2

Referências Bibliográficas:

- MALOBERTI; DAVIES, A Short History of Circuits and Systems. River, 2016.
- B. WIDROW, Thinking About Thinking: The Discovery of the LMS Algorithm. IEEE SIGNAL PROCESSING MAGAZINE, JANUARY 2005
- WIDROW;LEHR, 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. PROCEEDINGS OF THE IEEE, VOL. 78, NO. 9, SEPTEMBER 1990.
- HAYKIN, S. Adaptive Filter Theory. Prentice Hall, 4a ed., 2001.
- HAYKIN, S. Neural Networks and Learning Machines. Pearson, 3a ed., 2009.
- HAUPT; HAUPT, Practical Genetic Algorithms. Wiley, 2004.
- HAUPT; WERNER, Genetic Algorithms in Electromagnetics. IEEE Press, 2007.
- CHAMBERS, L. Practical handbook of genetic algorithms. 2001.
- PEDRYCZ; GOMIDE, An Introduction to Fuzzy Sets. MIT Press, 1998.
- PEDRYCZ, W. An introduction to Fuzzy Sets: analysis and design control. 1998.