

# Modelos Neurais para Dispositivos Ativos Semicondutores de GaAs nas Faixas de Microondas/Óptica

**Paulo Henrique da Fonseca Silva<sup>1</sup>**

CEFET-PB/GTEMA – Av. 1º de Maio, 720 Jaguaribe

CEP: 58015-430 – João Pessoa, PB, Brasil

E-mail: [henrique@cefetpb.edu.br](mailto:henrique@cefetpb.edu.br)

**Márcio G. Passos e Humberto C. C. Fernandes**

UFRN/TECFOTON – Campus Universitário

CEP: 59072-970 – Natal, RN, Brasil

E-mail: [marcio\\_passos@yahoo.com.br](mailto:marcio_passos@yahoo.com.br) e [humbeccf@ct.ufrn.br](mailto:humbeccf@ct.ufrn.br)

**Resumo:** Neste artigo descrevem-se os conceitos básicos para a elaboração de modelos de dispositivos ativos semicondutores de arseneto de gálio (GaAs), usados em sistemas de comunicação em operação nas faixas de frequências de microondas/óptica, através de redes neurais artificiais sem realimentação (FNN's – feedforward neural networks). O enfoque é voltado para a computação realizada por uma FNN: processamento de sinais e aprendizado supervisionado através de algoritmos de treinamento fundamentados no método do gradiente. A partir de uma nomenclatura uniforme, a formulação matemática desta metodologia se aplica a qualquer FNN estática. São apresentadas algumas contribuições nesta área, bem como aplicações das FNN's como modelos para transistores de microondas e amplificadores ópticos. Os resultados obtidos demonstram a eficiência computacional e a capacidade de generalização dos modelos neurais propostos.

**Palavras Chave:** redes neurais sem realimentação, dispositivos ativos, modelos neurais.

## 1. Introdução

A expansão mundial do mercado de produtos e serviços de telecomunicações observada nas últimas décadas devido, por exemplo ao advento da Internet e da telefonia celular, só foi possível graças ao avanço tecnológico dos sistemas de comunicação. No estágio tecnológico atual, tornou-se comum o uso de ferramentas computacionais sofisticadas para projeto, análise e simulação dos dispositivos e circuitos componentes de um sistema de comunicação. A tecnologia através de redes neurais artificiais (ANN's – Artificial Neural Networks) é uma das ferramentas de Inteligência Computacional, que também tem contribuído para o desenvolvimento nesta área. Os modelos neurais para dispositivos de microondas/ópticos avançaram rapidamente no que se refere à aproximação, simulação e otimização de parâmetros físicos, elétricos e ópticos (ZHANG, GUPTA, 2000).

Na elaboração dos modelos neurais propostos, uma FNN é treinada com dados precisos (medidos ou simulados), que representam uma parte conhecida da região de operação de um dispositivo. Após a fase de treinamento, a rede FNN deve ser capaz de generali-

zar nesta região de operação. Desta forma, obtém-se o modelo neural para o dispositivo em questão, que pode ser então incorporado, com suas vantagens computacionais, a um programa de CAD (Computer Aided Design), por exemplo, para a simulação de circuitos não lineares.

Em geral, os modelos neurais associam a precisão dos simuladores eletromagnéticos (modelos físicos/métodos numéricos de onda completa) e a eficiência computacional, que é uma característica dos modelos empíricos, (SILVA, 2002).

Em particular, este artigo reúne os procedimentos utilizados para o projeto de uma FNN estática. Na seção 2 são apresentados alguns conceitos básicos sobre redes neurais. As seções 3 e 4 são devotadas à formulação da metodologia através das redes neurais MLP (multilayer perceptrons) e RBF (radial basis functions), respectivamente. A seção 5 descreve a aplicação de redes FNN's como modelos para dispositivos ativos semicondutores de GaAs: transistores de microondas e amplificadores ópticos. Os modelos propostos através de redes modulares SFNN (sample function neural network) também são descritos. Os resultados decorrentes da simulação computacional dos modelos neurais elaborados são apresentados na seção 6. Enfim, na seção 7 constam as conclusões sobre os resultados obtidos.

<sup>1</sup> Autor a quem toda correspondência deverá ser endereçada.

## Nomenclatura

- $a$  = potencial de ativação, V  
 $d$  = vetor resposta desejada, V  
 $e$  = sinal erro, V  
 $E$  = erro médio quadrático, J  
 $N$  = número total de exemplos de treinamento  
 $Nh$  = número de neurônios (nós) ocultos  
 $Ni$  = número de neurônios (nós) de entrada  
 $Ns$  = número de neurônios (nós) de saída  
 $S$  = matriz de dados de treinamento, V  
 $x$  = vetor de sinais de entrada, V  
 $y$  = sinal de saída de um neurônio, V  
 $w$  = peso sináptico ou parâmetro livre

## Símbolos gregos

- $\Delta$  = variação ou ajuste  
 $\varphi$  = função ativação  
 $\eta$  = taxa de aprendizado  
 $\sigma$  = largura de uma RBF Gaussiana  
 $\nabla$  = operador gradiente

## Índices

- $i$  relativo à camada de entrada  
 $j$  relativo à camada oculta  
 $k$  relativo à camada de saída  
 $n$  relativo ao número de iterações de treinamento  
 $t$  relativo ao número de épocas de treinamento  
 $T$  transposição matricial

## 2. Conceitos Básicos

### - Redes Neurais Artificiais

Uma rede neural artificial (rede neural, ou simplesmente rede) é uma máquina que pode ser construída com componentes eletrônicos ou simulada em computadores digitais. A unidade básica de processamento de informação de uma rede neural é o neurônio artificial (neurônio ou simplesmente nó). Cada neurônio possui um conjunto de conexões de entrada. Na configuração de uma rede neural os neurônios são reunidos em camadas, constituindo um processador de sinais paralelo e totalmente distribuído.

Uma característica importante de uma rede neural é a sua natureza adaptativa. Esta propriedade é atribuída às conexões entre os neurônios da rede. A cada conexão está associado um peso (amplitude, ou parâmetro livre). Este peso amplifica, atenua, e/ou muda um sinal de informação pré-sináptico - na entrada de uma conexão. Durante o processo de aprendizado, quando a rede é ativada com exemplos de treinamento, seus pesos (ou parâmetros livres) são ajustados de acordo com uma estratégia de aprendizado.

Estas características de uma ANN são responsáveis não só por sua capacidade de aprendizado a partir

de exemplos, mas principalmente, por sua habilidade para generalizar respostas, quando ativada com novos exemplos - uma propriedade muito difícil de se obter, a partir de ferramentas de computação convencional.

Devido a sua natureza paralela, uma rede neural realiza o processamento de sinais de informação em alta velocidade; além disso, são tolerantes a falhas de seus elementos individuais de processamento.

As redes neurais possuem uma vocação natural para armazenar o conhecimento experimental, tornando-o disponível para uso futuro. Seu funcionamento é semelhante ao do cérebro humano em dois aspectos: i) o conhecimento é adquirido pela rede neural por um processo de aprendizado; ii) as amplitudes das conexões entre os neurônios artificiais são utilizadas para armazenar o conhecimento, (HAYKIN, 1994, p. 2).

De um modo geral, o projeto de uma rede neural consiste em três elementos principais: configuração - como a rede neural é organizada em camadas e como estas estão conectadas; aprendizado - como a informação é armazenada na rede; evocação - como a informação armazenada é recuperada da rede.

### - Representação do Conhecimento

Em funcionamento, uma rede neural está inserida em um ambiente, do qual são obtidos os sinais de informação (exemplos) que ativam a rede, Fig. (1). Neste contexto, o conhecimento se refere à informação armazenada ou um modelo utilizado, por uma pessoa ou máquina para interpretar, prever, e responder apropriadamente ao ambiente exterior.

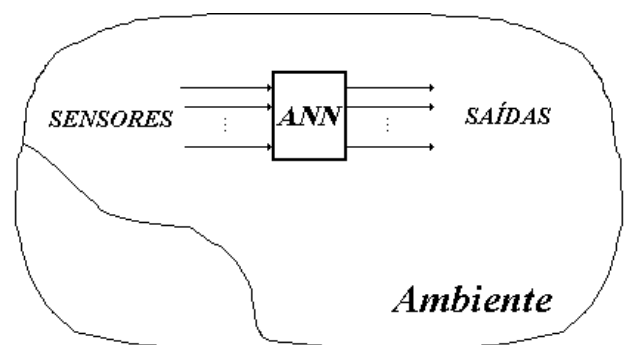


Figura 1. Ambiente de funcionamento de uma rede neural.

O problema de representação do conhecimento consiste na solução das seguintes questões, (HAYKIN, 1994, p. 22): Qual é a informação conhecida a priori sobre o estado do ambiente? Como esta informação pode ser codificada e utilizada?

A tarefa principal de uma rede neural é justamente o aprendizado de um modelo do ambiente, no qual ela está inserida, bem como manter este modelo consistente em ambiente real.

Em qualquer aplicação os dados obtidos formam um conjunto de exemplos, que são usados durante o treinamento. Estes exemplos de treinamento devem ser representativos do ambiente - problema de aprendizado submetido à rede neural. Os exemplos 'falam por si' - nenhuma informação adicional é necessária para o treinamento da rede.

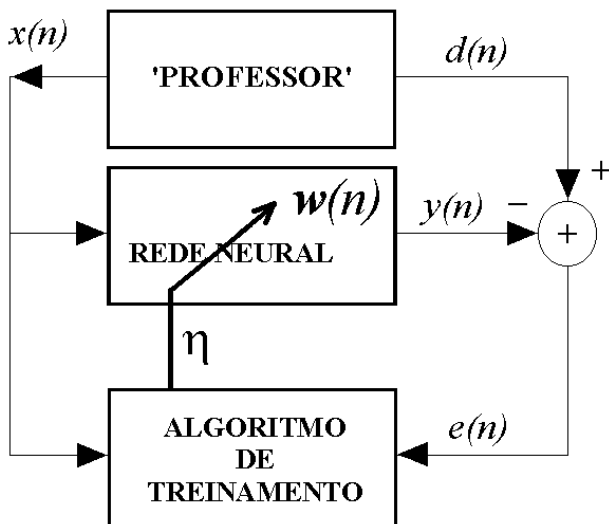
Cada exemplo de treinamento consiste de um par de vetores de sinais de entrada/respostas desejadas. O conjunto de dados de treinamento é definido por:

$$S(n) = (\mathbf{x}(n), \mathbf{d}(n)), \quad n = 1, 2, \dots, N \quad (1)$$

Em que cada iteração  $n$  corresponde ao instante de apresentação de um exemplo;  $N$  indica o número total de exemplos. Convém ressaltar que os dados de treinamento são normalizados.

#### - Aprendizado Supervisionado

Na simulação computacional do processo de aprendizado supervisionado de correção do erro, a adaptação dos pesos de uma rede neural é feita por um algoritmo de treinamento. O sinal erro (a diferença entre a resposta desejada e a saída da rede) é obtido através da ativação da rede neural. É como se a rede estivesse na presença de um 'professor', que fornecesse os exemplos de treinamento, Fig. (2). O objetivo deste método é a minimização do sinal erro.



**Figura 2.** Representação do processo de aprendizado supervisionado.

#### - Modelo Não-Linear de um Neurônio Artificial

O modelo de neurônio mais usado atualmente é o perceptron de Rosenblatt (1958), cuja operação é baseada no modelo não-linear de um neurônio artificial proposto por McCulloch-Pitts (1943). Neste modelo, um neurônio

é constituído por um conjunto de conexões e um elemento de processamento, que inclui um somador e uma função ativação linear ou não-linear, Fig. (3).

Em conjunto, o somador e os pesos efetuam uma combinação linear dos sinais de entrada do neurônio  $y_i$ , cujo resultado corresponde ao nível de atividade interna ou potencial de ativação, Eq. (2). Por sua vez, o sinal de saída do neurônio,  $y_j$ , é a resposta da função ativação a este potencial, Eq. (3).

$$a_j(n) = \sum_{i=0}^{N_i} w_{ji}(n) y_i(n) \quad (2)$$

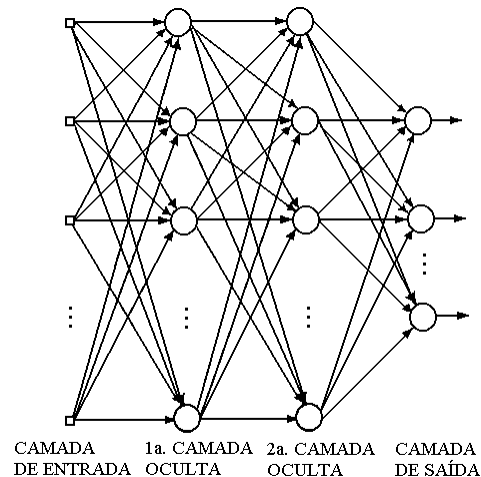
$$y_j(n) = \phi_j(a_j(n)) \quad (3)$$

em que  $\phi_j$  corresponde ao potencial de polarização do neurônio.

**Figura. 3** Modelo não-linear de um neurônio.

#### - Redes Neurais sem Realimentação

Na configuração de uma FNN os neurônios artificiais são reunidos em camadas. Cada neurônio de uma camada está conectado a todos os neurônios da camada anterior, conforme a configuração ilustrada na Fig. (4). A propagação dos sinais ocorre no sentido direto: da camada de entrada para a camada de saída, passando através das camadas ocultas da rede FNN.



**Figura 4.** Configuração de uma rede neural sem realimentação com quatro camadas.

A camada de entrada de uma FNN é composta de neurônios sensores, que captam os sinais de entrada do ambiente adequando-os ao processamento na rede neural. A computação realizada na camada de entrada é simbolizada por  $x_i \rightarrow y_i$ . Os neurônios ocultos são considerados unidades associativas, responsáveis pela representação das características presentes nos sinais de entrada. Os neurônios de saída, por sua vez, geram os sinais de saída da rede.

### 3. Metodologia Através de Redes MLP's

Esta seção descreve o tipo mais usado de FNN: a rede neural MLP, com apenas uma camada oculta, cuja configuração é ilustrada na Fig. (5). A rede MLP tem suas raízes no *perceptron* de Rosenblatt (1958). Descrita pela primeira vez por Werbos (1974), despertou bastante interesse da comunidade científica a partir dos estudos reportados por Rumelhart et al, (1986). Este e outros trabalhos publicados, em meados da década de 1980, representaram um marco para o avanço atual das redes neurais.

#### - Computação no Sentido Direto

Em termos matemáticos, o processamento de sinais realizado pela rede MLP, Fig. (5), é descrito através das seguintes expressões de computação no sentido direto, Eq. (4)-(7).

$$a_j(n) = \sum_{i=0}^{N_i} w_{ji}(n) y_i(n), \quad j = 1, 2, \dots, N_h \quad (4)$$

$$y_j(n) = \varphi_j(a_j(n)) \quad (5)$$

$$a_k(n) = \sum_{j=0}^{N_h} w_{kj}(n) y_j(n), \quad k = 1, 2, \dots, N_s \quad (6)$$

$$y_k(n) = \varphi_k(a_k(n)) \quad (7)$$

Em que  $w_{ji}$  é o peso da conexão entre os neurônios  $i$  e  $j$ ;  $w_{kj}$  é o peso da conexão entre os neurônios  $j$  e  $k$ ;  $y_i$  denotam os sinais de saída de cada neurônio nas camadas de entrada, oculta e de saída, respectivamente;  $\varphi_j$  e  $\varphi_k$  correspondem às funções ativação, respectivamente, nas camadas oculta e de saída.

Os neurônios ocultos da rede MLP são definidos com uma função ativação *sigmóide*. No modelo proposto, a camada de saída é linear e para a camada oculta define-se a função *logística* como função ativação não linear, Eq. (8).

$$\varphi_j(a_j(n)) = \frac{1}{1 + \exp(-a_j(n))} \quad (8)$$

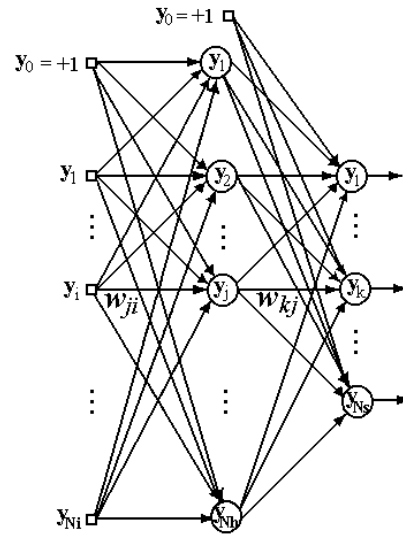


Figura 5. Rede MLP com uma camada oculta

#### - Medidas de Desempenho de Treinamento

As medidas de desempenho de treinamento são definidas como uma função quadrática do sinal erro. No jargão das redes neurais, estas medidas recebem várias denominações, tais como: função erro, função objetivo ou custo, superfície de erro, etc.

Define-se o sinal erro para o  $k$ -ésimo neurônio da camada saída da rede MLP através da Eq. (9).

$$e_k(n) = d_k(n) - y_k(n) \quad (9)$$

A partir da informação do sinal erro, obtém-se o erro médio quadrático instantâneo, Eq. (10). Esta medida de desempenho leva em conta o sinal erro de cada neurônio na camada de saída da rede.

$$E(n) = \frac{1}{2N_s} \sum_{k=1}^{N_s} [e_k(n)]^2 \quad (10)$$

Outra medida muito utilizada é o erro médio quadrático (MSE - *mean square error*), cujo cálculo leva em conta a apresentação do conjunto de treinamento completo, Eq. (11).

$$MSE(t) = \frac{1}{N} \sum_{n=1}^N E(n) \quad (11)$$

Em que  $t$  é o índice que indica o número de épocas de treinamento. Uma época corresponde ao tempo necessário para a apresentação de todos os exemplos de treinamento à rede neural.

– Algoritmo Backpropagation (BP)

O algoritmo *backpropagation* é uma aplicação particular do método do gradiente para a otimização dos pesos da rede MLP. Neste método, a informação do vetor gradiente da função erro,  $\nabla E(n)$ , indica a direção de pesquisa no espaço de pesos. O gradiente em relação a um peso qualquer da rede,  $w(n)$ , é definido na Eq. (12):

$$\nabla E(w(n)) = \frac{\partial E(n)}{\partial w(n)} \quad (12)$$

Especificamente, os pesos são ajustados no sentido oposto ao do vetor gradiente da função erro. A adaptação é efetuada de forma iterativa através da seguinte relação recursiva:

$$w(n) = w(n-1) - \eta \nabla E(w(n-1)) \quad (13)$$

em que,  $0 < \eta < 1$  é a taxa de aprendizado, uma constante que regula os valores de ajuste dos pesos.

A solução da Eq. (13) aplicada ao MLP resulta nas seguintes relações recursivas para ajuste dos pesos, Eq. (14)-(15),

$$w_{kj}(n) = w_{kj}(n-1) + \eta e_k \phi'_k y_j \Big|_{(n-1)} \quad (14)$$

$$w_{ji}(n) = w_{ji}(n-1) + \eta e_k \phi'_k w_{kj} \phi'_j y_i \Big|_{(n-1)} \quad (15)$$

em que,  $\phi'_j$  e  $\phi'_k$  indicam as derivadas de primeira ordem das funções ativação do MLP.

Inicialmente, atribuem-se valores aleatórios aos pesos, dentro de uma faixa pré-estabelecida, que servem como ponto de partida para o treinamento com o algoritmo BP.

#### 4. Metodologia Através de Redes RBF's

– Rede RBF Gaussiana

Uma rede neural de funções de base radiais consiste em uma camada de entrada, uma camada oculta de funções RBF's, e uma camada de saída linear. A configuração ilustrada na Fig. (6), com apenas um nó de saída é a mais simples para uma rede RBF. As relações de computação no sentido direto, Eq. (16)-(18), descrevem o processamento de sinais na rede RBF Gaussiana.

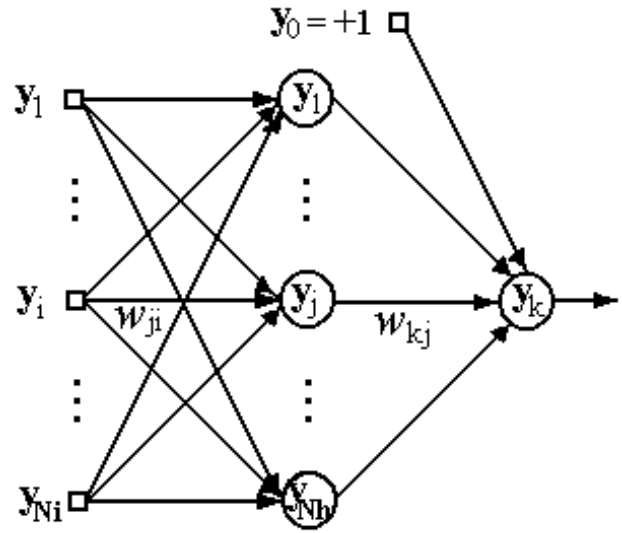


Figura 6. Rede RBF com um neurônio de saída

$$a_j(n) = \|y_i(n) - w_j(n)\| = \sqrt{\sum_{i=1}^{Ni} (y_i(n) - w_{ji}(n))^2} \quad (16)$$

$$y_j(n) = \phi_j(a_j(n)) = \exp(-a_j(n)/\sigma_j^2(n)) \quad (17)$$

$$y_k(n) = \sum_{j=0}^{Nh} w_{kj}(n) y_j(n), \quad k = 1 \quad (18)$$

em que,  $w_j$  é o vetor de centros e  $\sigma_j$  é a 'largura' da  $j$ -ésima função RBF Gaussiana.

A estratégia de seleção supervisionada dos centros é a mais usada para o treinamento de uma rede RBF. Nesta estratégia, os parâmetros livres da rede (centros, larguras e pesos) são ajustados através de um processo de aprendizado supervisionado de correção do erro, (LOWE, 1989). O objetivo é a minimização da função erro definida na Eq. (11). Através do método do gradiente, obtêm-se as seguintes relações recursivas para a otimização dos parâmetros livres da rede RBF, Eq. (19)-(21).

$$\Delta w_{kj}(n) = \eta e_k y_j \Big|_{(n-1)} \quad (19)$$

$$\Delta w_j(n) = 2\eta e_k w_{kj} y_j (y_i - w_j) / \sigma_j^2 \Big|_{(n-1)} \quad (20)$$

$$\Delta \sigma_j^2(n) = \eta e_k w_{kj} y_j \|y_i - w_j\|^2 / \sigma_j^2 \Big|_{(n-1)} \quad (21)$$

– Rede de Funções Sample

A rede de funções *sample* (SFNN – *sample functions neural network*) é uma contribuição original, que tem sido aplicada com sucesso na área de engenharia de microondas, (SILVA, 2002). Sua configuração é idêntica a da rede RBF, Fig. (6). A escolha da função ativação do tipo *sample*,  $sample(x) = sen(x)/x$  para os neurônios ocultos da SFNN teve motivação nas aplicações desta função para reconstrução de sinais. O processamento de sinais numa SFNN é descrito através da seguinte computação no sentido direto, Eq. (22)-(24).

$$a_j(n) = \left\| y_i(n) - w_j(n) \right\|^2 \quad (22)$$

$$y_j(n) = \varphi_j(a_j(n)) = sample(\sigma_j(n) \cdot a_j(n)) \quad (23)$$

$$y_k(n) = \sum_{j=0}^{Nh} w_{kj}(n) y_j(n), \quad k = 1 \quad (24)$$

Com a aplicação do método do gradiente, resultam as seguintes relações recursivas para a otimização dos parâmetros livres da rede SFNN, Eq. (25)-(27).

$$\Delta w_{kj}(n) = \eta e_k y_j \Big|_{(n-1)} \quad (25)$$

$$\Delta w_j = 2\eta e_k w_{kj} \sigma_j \frac{y_j - \cos(a_j)}{a_j} (y_i - w_j) \Big|_{(n-1)} \quad (26)$$

$$\Delta \sigma_j(n) = \eta e_k w_{kj} (\cos(a_j) - y_j) / \sigma_j \Big|_{(n-1)} \quad (27)$$

## 5. Modelo Neural Proposto

Um programa de CAD para simulação de circuitos de microondas/ópticos incorpora vários modelos de dispositivos ativos não lineares. Durante a simulação, estes modelos são avaliados de uma forma repetitiva. Por isso, simuladores de circuitos, como o SPICE, usado para análise transitória ou como o MBH (método do balanço harmônico), usado para análise em regime permanente, requerem modelos de dispositivos eficientes e precisos.

Contudo, eficiência computacional e precisão não andam juntas nos modelos convencionais. Por exemplo, os modelos empíricos, formulados em termos de expressões analíticas conhecidas, são eficientes, mas geralmente suas respostas são apenas aproximadas.

Por outro lado, os modelos EM's (eletromagnéticos), que levam em conta a física dos dispositivos, recaem em métodos numéricos precisos, mas com elevado custo computacional.

Neste contexto, as redes neurais surgiram como ferramentas numéricas alternativas e bastante atrativas. A precisão de um modelo neural se equipara a de um modelo físico, enquanto que, sua eficiência computacional é tão boa quanto a de um modelo empírico.

Os modelos neurais propostos neste artigo encontram aplicações para dispositivos não lineares de duas portas, em operação nas faixas de microondas/óptica. Um dispositivo susceptível a este tipo de modelagem possui uma característica em particular: sua resposta ou sinal de saída depende de um sinal de entrada e de um sinal de controle. Admite-se que a resposta não linear seja ditada principalmente pelo sinal de entrada, sendo sua magnitude regulada pelo sinal de controle, Fig. (7). Em particular são consideradas aplicações para transistores de microondas e amplificadores ópticos.

Os dados de treinamento de um dado dispositivo são obtidos a partir de suas curvas características. Estes dados são fornecidos pelo fabricante do dispositivo ou obtidos através de métodos numéricos precisos. As curvas características delimitam a região de operação do dispositivo entre os valores mínimo e máximo dos sinais de entrada e de controle. Esta região de operação é chamada de região de interesse, Fig. (7).

Na aplicação das FNN's como modelos de dispositivos, além da interpolação dos dados de treinamento, espera-se que o modelo neural forneça respostas precisas para toda região de interesse, mesmo onde se tem pouco ou nenhum conhecimento sobre o funcionamento do dispositivo.

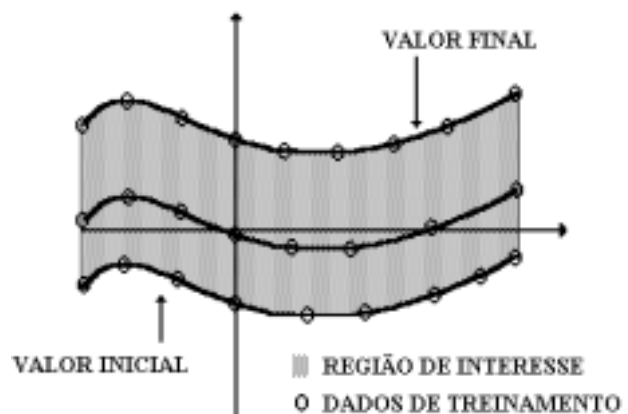


Figura 7. Região de interesse

Neste sentido, alguns aspectos como a escolha dos dados de treinamento, do número de neurônios ocultos e do algoritmo de treinamento a ser utilizado, influenciam a capacidade de generalização do modelo neural resultante.

Uma rede MLP com uma camada oculta, treinada com o algoritmo BP, é a metodologia mais utilizada na elaboração de um modelo neural. Contudo, para dispositivos de microondas/ópticos esta metodologia nem sempre é suficiente. Ela falha devido a limitações do algoritmo BP, principalmente quanto à convergência, que, além de não ser garantida, é bastante lenta. Outro problema encontrado é a escolha adequada do número de neurônios da camada oculta do MLP, que tanto influencia a convergência do algoritmo BP, quanto a capacidade de generalização da rede.

Para contornar estes problemas, propôs-se um modelo, neural através de uma rede modular SFNN, composta de duas redes especialistas e de uma rede de saída, Fig. (8). As saídas das redes especialistas são aplicadas às entradas da rede de saída, que fornecem a resposta deste modelo neural.

As redes especialistas são treinadas a partir das curvas características de valor inicial e de valor final do sinal de controle, que delimitam a região de operação do dispositivo, Fig. (7). Para o treinamento da rede de saída, o conjunto de treinamento completo é utilizado.

Esta configuração de rede modular visa à divisão de um problema de aprendizado dado em problemas menores e mais simples de serem resolvidos. Assim, ao invés de se utilizar apenas uma SFNN para mapear toda a região de interesse, esta tarefa é dividida entre as três SFNN's da rede modular.



Figura 8. Configuração da rede modular SFNN

O uso da rede modular SFNN traz alguns benefícios interessantes tais como: redução do número de neurônios ocultos para as redes especialistas, pois cada uma precisa aproximar apenas uma curva característica; melhor capacidade de generalização da rede de saída, uma vez que, as redes especialistas lhes fornecem informação adicional, através de duas entradas extras. O preço pago é a necessidade de se treinar três SFNN's para apenas um modelo neural.

O algoritmo *resilient propagation* (RPROP), proposto inicialmente para o treinamento de redes MLP's (RIEDMILLER; BRAUN, 1993) foi adaptado para o treinamento das redes SFNN's. O RPROP é robusto quanto à escolha dos parâmetros de treinamento e acelera o treinamento das redes SFNN's. Estas duas vantagens facilitou também a escolha, através de tentativa e erro, do número adequado de neurônios ocultos para cada SFNN.

## 6. Aplicações

### - Transistores GaAs MESFET's

Os transistores de microondas do tipo GaAs MESFET (*Metal Semiconductor Field Effect Transistor*) são dispositivos complexos internamente. Os modelos empíricos/analíticos são incapazes de descrevê-los de forma precisa e sob todas as condições. Simulações precisas só são possíveis através do uso de um modelo físico/método numérico (SILVA et al., 2001).

Nesta seção, propõe-se a aplicação de uma rede MLP como modelo de um transistor do tipo GaAs MESFET. O MLP utilizado possui 3 nós de entrada, uma camada oculta com 10 neurônios e um neurônio de saída linear. Para o treinamento com o algoritmo RPROP foram utilizados 33 exemplos para representar a característica  $I_{DS} = f(V_{GS}, V_{DS})$  deste transistor.

Na Figura (9), é feita uma comparação entre as respostas do modelo neural MLP e de outros modelos empíricos, com os resultados do modelo físico/EM de análise em 2-D (SILVA et al, 2002). Verifica-se que o modelo neural MLP apresenta a melhor aproximação para os dados de análise 2-D.

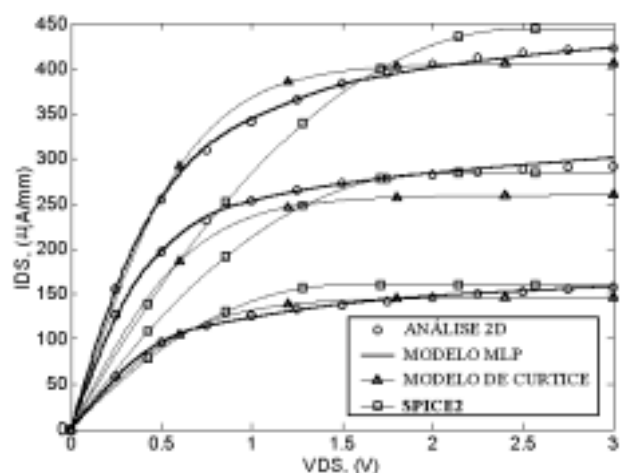
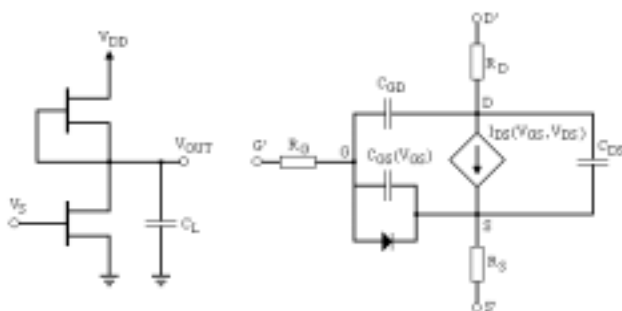


Figura 9. Curvas características de um transistor de microondas GaAs MESFET

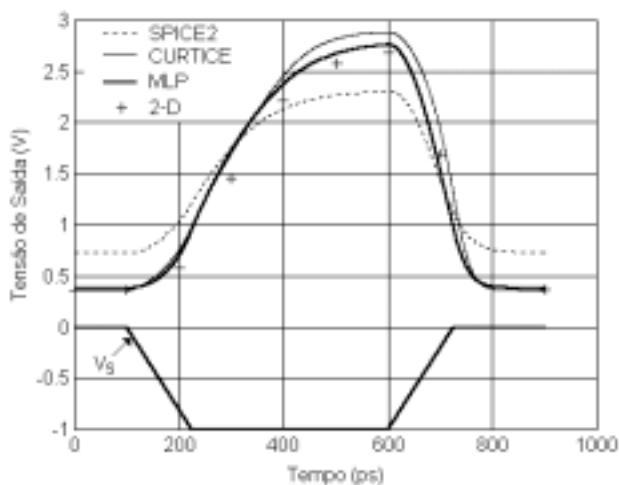
No teste da capacidade de generalização do modelo MLP, simulou-se a resposta transitória do circuito de entrada de uma porta lógica com dois transistores GaAs MESFET, Fig. (10). Para tanto se utilizou o

circuito não linear equivalente proposto por Curtice (1980). Os resultados obtidos nesta análise transitória demonstraram a validade do modelo MLP. A Figura (11) apresenta as respostas para uma entrada pulsada numa taxa de 1 Gbit/s. A tensão de saída do circuito porta lógica, obtida a partir do modelo de SCHICHMAN; HODGES (1968), incorporado ao simulador SPICE2, apresenta erros no tempo de subida, no ganho e no atraso de propagação. Para o modelo de Curtice (1980), os resultados são razoáveis para um modelo empírico. O modelo MLP é o que melhor aproxima a resposta transitória do modelo preciso de análise numérica em 2-D.

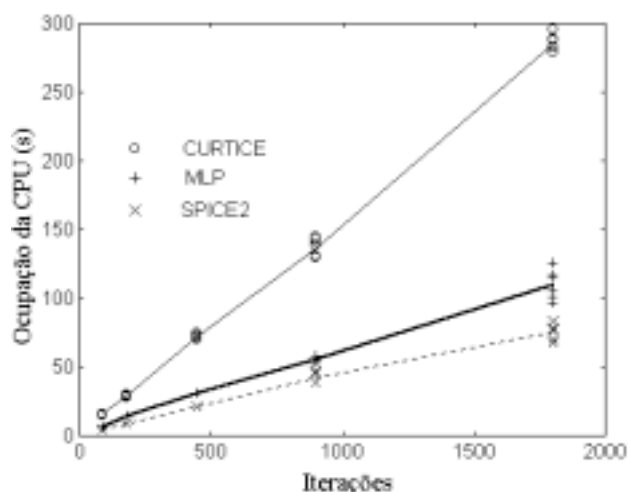


**Figura 10.** Circuito porta lógica e circuito equivalente para o GaAs MESFET

A eficiência computacional de cada um dos modelos simulados foi medida em função do número de iterações do método de Newton-Raphson, usado na análise transitória do circuito porta lógica. Os resultados obtidos são mostrados na Fig. (12). Neste estudo comparativo, verificou-se que, o custo computacional do modelo MLP é inferior ao do modelo de Curtice, e, ligeiramente superior ao do modelo de Schichman-Hodges.



**Figura 11.** Respostas pulsadas a 1,0 Gbit/s



**Fig. 12** Custo computacional dos modelos

### - Amplificadores Ópticos

Com o advento das comunicações por fibras ópticas, as aplicações do laser aumentaram também nesta área. Os amplificadores ópticos com laser semiconductor são usados, por exemplo, como amplificadores de linha, pré-amplificadores, amplificadores de banda-estreita com sintonia elétrica.

Nas aplicações em estações repetidoras, os amplificadores ópticos possibilitam *links* extensos de comunicação por fibra, que são inviáveis sem a recuperação dos sinais transmitidos. Antes do advento destes amplificadores, os sinais chegavam aos repetidores, onde eram convertidos em sinais elétricos, restaurados, sincronizados e retransmitidos. Todo este processo é eliminado com o uso dos amplificadores ópticos (IIZUKA, 2002).

O coeficiente de ganho de um amplificador a laser semiconductor,  $\gamma$ , é uma função do nível de energia dos fótons,  $h\nu$ , bem como, da concentração de portadores injetados,  $\Delta n$ . Nesta aplicação, um modelo através da rede modular SFNN deve realizar o mapeamento  $\gamma = f(h\nu, \Delta n)$ . Os dados precisos para o treinamento e teste foram obtidos a partir de soluções das equações integrais que regem o funcionamento destes amplificadores, (SALEH; TEICH, 1991, p. 613).

### - Amplificador Laser de GaAs

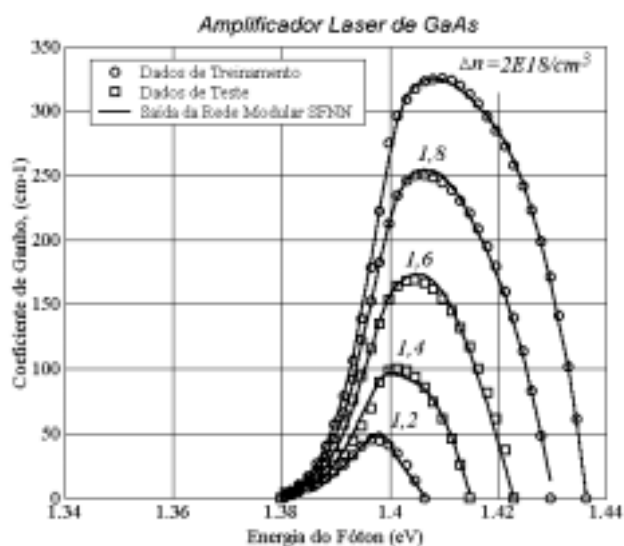
Na elaboração do modelo para o amplificador de GaAs, as SFNN's da rede modular foram treinadas separadamente. As informações relevantes para o treinamento estão contidas na Tab. (1). A simulação dos modelos foi realizada em um microcomputador pessoal Celeron 566 MHz com 80 Mb de memória. Obviamente, o tempo total de treinamento cresce com o tamanho do conjunto de treinamento.



**Tabela 1.** Informações de treinamento da rede modular SFNN para o amplificador de GaAs.

Rede:	Especialista 1
Aproximação da Curva ( $\Delta n$ ):	$1,2E18 \text{ cm}^{-3}$
Neurônios da Camada Oculta:	4
Épocas de Treinamento:	10000
MSE Final:	$1,88E-06$
Tempo Total de Treinamento:	3,72 minutos
Rede:	Especialista 2
Aproximação da Curva ( $\Delta n$ ):	$2E18 \text{ cm}^{-3}$
Neurônios da Camada Oculta:	4
Épocas de Treinamento:	10000
MSE Final:	$6,23E-05$
Tempo Total de Treinamento:	10,57 minutos
Rede:	Saída
Neurônios da Camada Oculta:	20
Épocas de Treinamento:	5000
MSE Final:	$6,49E-05$
Tempo Total de Treinamento:	13,21 minutos

Os resultados de simulação para o coeficiente de ganho do amplificador de GaAs, ilustrados na Fig. (13), indicam boa concordância entre a resposta do modelo através da rede modular SFNN e o método de cálculo convencional. Para verificação da capacidade de generalização, este modelo neural foi utilizado para estimar os valores de pico do coeficiente de ganho, em termos da concentração de portadores injetados. A estimativa obtida, Fig. (14), indica uma boa aproximação para a resposta precisa do modelo de equações integrais.



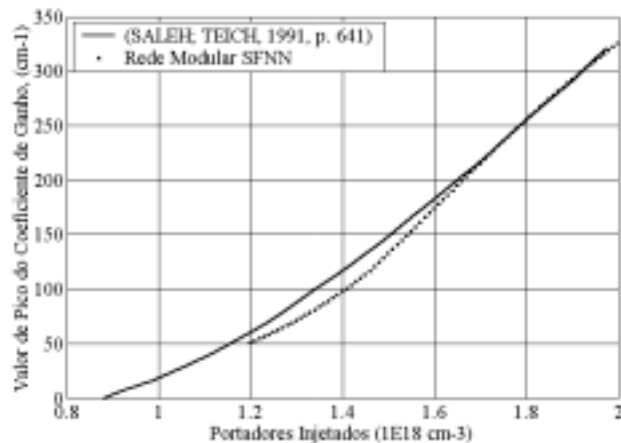
**Figura 13.** Resposta da rede modular SFNN – amplificador laser semiconductor de GaAs.

- Amplificador Laser de InGaAsP

Na elaboração do modelo do amplificador de InGaAsP, as redes SFNN's também foram treinadas

de forma separada e as informações relevantes para o treinamento estão contidas na Tab. (2).

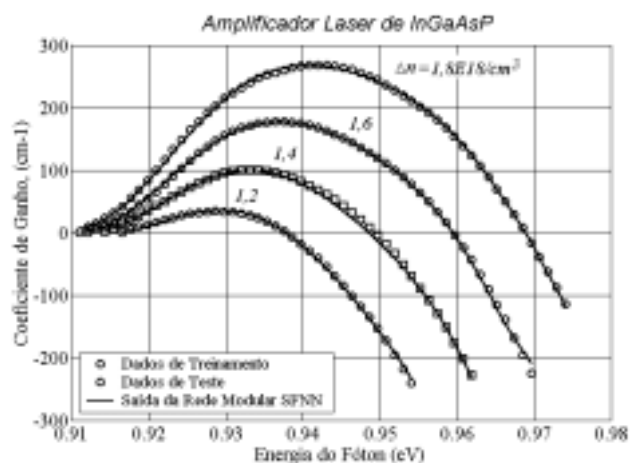
Para o amplificador InGaAsP, os resultados de teste, Fig. (15), demonstraram que a resposta do modelo neural está em excelente concordância com o modelo físico preciso, com generalização para as curvas de teste.



**Figura 14.** Valor de pico do coeficiente de ganho

**Tabela 2.** Informações de treinamento da rede modular SFNN para o amplificador de InGaAsP.

Rede:	Especialista 1
Aproximação da Curva ( $\Delta n$ ):	$1,2E18 \text{ cm}^{-3}$
Neurônios da Camada Oculta:	4
Épocas de Treinamento:	10000
MSE Final:	$2,35E-06$
Tempo Total de Treinamento:	6,85 minutos
Rede:	Especialista 2
Aproximação da Curva ( $\Delta n$ ):	$1,8E18 \text{ cm}^{-3}$
Neurônios da Camada Oculta:	4
Épocas de Treinamento:	10000
MSE Final:	$5,48E-05$
Tempo Total de Treinamento:	10,48 minutos
Rede:	Saída
Neurônios da Camada Oculta:	15
Épocas de Treinamento:	10000
MSE Final:	$1,8E-04$
Tempo Total de Treinamento:	31,58 minutos



**Figura 15.** Resposta da rede modular SFNN - amplificador laser semiconductor de InGaAsP.

## 7. Conclusões

Neste artigo, formulou-se uma metodologia para a elaboração de modelos neurais através de FNN's para dispositivos ativos de GaAs. A proposta da rede modular SFNN foi uma contribuição que, aplicada como modelo neural, apresentou uma maior capacidade de generalização. Algumas vantagens desta metodologia incluem eficiência computacional e reduzida ocupação de memória. Os resultados obtidos demonstraram que as FNN's são eficientes e precisas nas aplicações como modelos neurais. A versatilidade destes modelos possibilita a aplicação desta metodologia a uma ampla classe de dispositivos de microondas/ópticos.

## 8. Referências

- CURTICE, W. R. A MESFET Model for Use in the Design of GaAs Integrated Circuits. **IEEE Trans. on Microwave Theory and Techniques**, v. 28, p. 448-456, 1980.
- HAYKIN, S., **Neural Networks: A Comprehensive Foundation**. New York: Macmillan College Publishing Company, 1994. 696 p.
- IIZUKA, K. **Elements of photonics**. John Wiley and Sons, 2002. cap. 3, p. 833-893.
- LOWE, B. Adaptive Radial Basis Function Nonlinearity, and the Problem of Generalization. In: INTERNATIONAL CONFERENCE OF ARTIFICIAL NEURAL NETWORKS, Londres. **Proceedings**. Londres: IST-IEE, 1989. p. 171-175.
- MCCULLOCH, W. S., PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115-133, 1943.
- RIEDMILLER, M.; BRAUN, H. A Direct Adaptive Method for Faster Back-Propagation Learning: The RPROP Algorithm. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, San Francisco. **Proceedings**, San Francisco: IEEE-ICNN, 1993. v.1, p. 586-591.
- ROSENBLATT, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. **Psychological Review**, v. 65, p. 386-408, 1958.
- RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. Learning Internal Representations by Error Backpropagation. **Parallel Distributed Processing**, Cambridge M.I.T. Press, v. 1, p. 318-362, 1986.
- SALEH, B. E. A., TEICH, M. C. **Fundamentals of Photonics**. John Wiley and Sons, INC., 1991. cap. 16, p. 592-643.
- SCHICHMAN, H., HODGES, D. A. Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits. **IEEE J. Solid-State Circuits**, v. 3, p. 285-289, 1968.
- SILVA, P. H. DA F.; MELO, M. A. B.; NETO, A. D. D. Fast and Accurate Neural Network GaAs MESFET Model for Time-Domain Circuit Simulation In: IMOC - INTERNATIONAL MICROWAVE AND OPTOELECTRONICS CONFERENCE, Belém. **Proceedings**. Belém: SBMO/ IEEE MTT-S, 2001.
- SILVA, P. H. DA F. **Modelos Através de Redes Neurais Artificiais sem Realimentação para Dispositivos e Circuitos de RF/Microondas**. 2002. 166 f. Tese (Doutorado em Engenharia Elétrica) – Departamento de Engenharia Elétrica, UFCG, Campina Grande.
- SILVA, P. H. DA F.; MELO, M. A. B.; NETO, A. D. D. Using Neural Network Models for Efficient GaAs MESFET Time Domain Nonlinear Circuit Analysis. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, Honolulu. **Proceedings**. Honolulu: WCCI-IEEE, 2002. v. 3, p. 2282-2287.
- WERBOS, P. J. **Beyond Regression: New Tools for Prediction and Analysis in the behavioral sciences**. 1974. Phd Dissertation (Committee on Applied Mathematics), Harvard University, Cambridge, MA.
- ZHANG, Q. J., GUPTA, K. C., **Neural Networks for RF and Microwave Design**, 1.ed. Norwood, MA: Artech House, Inc., 2000. 369 p.

### Responsabilidade de autoria

As informações contidas neste artigo são de inteira responsabilidade de seus autores. As opiniões nele emitidas não representam, necessariamente, pontos de vista da Instituição e/ou do Conselho Editorial.