

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

TÍTULO



Belo Jardim, Pernambuco
DATA

MINISTÉRIO DA EDUCAÇÃO – (MEC)
SECRETARIA DE ENSINO MÉDIO E TECNOLÓGICO – (SETEC)
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
PERNAMBUCO - *Campus* Belo Jardim
PRÓ-REITORIA DE ENSINO

TÍTULO

Autore(s): AUTOR

Orientador: ORIENTADOR

Trabalho de conclusão de Curso (TCC) apresentado como requisito parcial para obtenção do grau de Bacharel em Engenharia de Software.

Banca de Qualificação:

NOME

NOME

IFPE
INSTITUIÇÃO

Belo Jardim, Pernambuco, DATA.

Agradecimentos

Em construção! São muitas pessoas.

Resumo

É apresentada

Keywords: CHAVE 01, CHAVE 02, CHAVE N.

Abstract

ABSTRACT

Keywords: KEY 01, KEY 02.

Lista de Abreviaturas

| | |
|-------|--|
| 4x16 | Configuração em dois níveis hierárquicos e 4 domínios constituídos por 16 nós da rede de comunicação formada por 64. |
| ASIC | Application Specific Integrated Circuit |
| BGP-4 | Border Gateway Protocol Version 4 |
| FPGA | Matrizes de Blocos Lógicos Programáveis em Campo (<i>Field Programmable Gate Array</i>) |
| GA | Genetic Algorithm |
| WDM | Wavelength-division multiplexing |

Lista de Símbolos

| | |
|-------------------------------------|---|
| δ | Função Delta de Kronecker |
| δt | Passo incremental de tempo |
| λ | Grau de inclinação da função de ativação |
| $\mu_1; \mu_2; \mu_3; \mu_4; \mu_5$ | Constantes da equação de energia |
| ρ | Matriz de conexão dos enlaces |
| τ | Constante de tempo da equação de atualização da entrada do neurônio do modelo de Hopfield |

Sumário

| | |
|---|-----------|
| Resumo | ii |
| Lista de Abreviaturas | iv |
| Lista de Símbolos | v |
| 1 Introdução | 1 |
| 1.1 Estrutura da Dissertação | 1 |
| 2 Redes Neurais de Hopfield | 2 |
| 2.1 Neurônio Artificial | 2 |
| 3 Modelos de HNN para Redes de Comunicação | 5 |
| 3.1 HNN para roteamento | 5 |
| 3.1.1 Execução do algoritmo | 6 |
| 3.1.2 Modelo de Bastos-Filho et al. | 6 |
| 3.1.3 Paralelismo | 6 |
| 4 Redes Neurais de Hopfield Hierárquicas | 8 |
| 4.1 Pseudo Código | 9 |
| 5 Arranjo Experimental | 10 |
| 5.1 Implementações HNN e HHNN em Java | 10 |
| 5.2 Implementações HNN e HHNN em FPGA | 11 |
| 5.2.1 Decisões Arquiteturais do Modelo | 11 |
| 6 Resultados | 12 |
| 6.1 Resultados do modelo codificado em Java | 12 |
| 6.1.1 Análise estrutural | 12 |
| 6.1.2 Análise da convergência | 13 |
| 7 Conclusões e Trabalhos Futuros | 14 |
| 7.1 Trabalhos Futuros | 14 |
| Bibliografia | 15 |
| A Artigos Publicados | 17 |

Lista de Figuras

| | | |
|-----|---|---|
| 2.1 | Neurônio biológico. Extraído de [8]. | 2 |
| 2.2 | Um Neurônio Artificial. Extraído de [7]. | 3 |
| 4.1 | Resultado parcial do roteamento interdomínio entre os nós 0 e 15 (Cenário I). | 8 |

Lista de Tabelas

| | | |
|-----|--|----|
| 5.1 | Características estruturais das HHNNs correspondentes aos cenários topológicos das duas redes de comunicação utilizadas nos experimentos. | 10 |
| 5.2 | Tabela contendo os parâmetros da simulação em Java. | 10 |
| 5.3 | Constantes utilizadas na implementação de HHNN em VHDL. | 11 |
| 6.1 | Características estruturais das HHNN referentes ao cenários topológicos das duas redes neurais de comunicação utilizados nos experimentos. | 12 |
| 6.2 | Redução da quantidade de neurônios e sinapses obtidos com a utilização do modelo HHNN. | 12 |
| 6.3 | Resultado do roteamento utilizando uma HHNN composta por 5 HNNs numa rede com 16 nós e 4 domínios. | 13 |

Capítulo 1

Introdução

Roteamento é o processo de determinação do caminho ou rota no qual o dados entre o remetente e o destinatário devem fluir. Os algoritmos que calculam esse caminho são denominados Algoritmos de Roteamento. Um algoritmo de roteamento deve selecionar uma rota em tempo real e também deve ser adaptativo, se adequando às alterações da topologia da rede de comunicação, assim como deve manter uma QoS estipulada. O processo de roteamento tem significativo impacto no desempenho da rede. Alguns algoritmos usam heurísticas baseadas em métricas predefinidas, como por exemplo: menor caminho (SP - Shortest Path), Menor número de saltos (MH - *Minimum Hops*), *Least Resistance Weight* (LRW) [13], *Optical Signal to Noise Ratio Routing* (OSNR-R) [10], *Physical Impairments Aware Adaptive Weight Function* (PIAWF) [5], *Power Series Routing* (PSR) [9]. Dentre os algoritmos de roteamento usados em rede de comunicação que utilizam estas métricas, pode-se destacar o algoritmo de Dijkstra (Estado do Enlace) [6] e o algoritmo de Bellman-Ford (Vetor Distância) [3], [4].

1.1 Estrutura da Dissertação

O capítulo 2 é composto por uma breve introdução às Redes Neurais Artificiais e pela descrição do modelo de Redes Neurais de John Hopfield, conhecido como Redes Neurais de Hopfield (HNN). Também é feito neste capítulo um resumo sobre implementações de algoritmos inteligentes em plataformas paralelas - Matrizes de Blocos Lógicos Programáveis em Campo (FPGA, *Field Programmable Gate Array*) e *Application Specific Integrated Circuit* (ASIC).

O capítulo 3 é constituído pela descrição da abordagem de HNN desenvolvida para solucionar o problema de estabelecimento de rotas entre um par origem-destino em redes de comunicação.

Capítulo 2

Redes Neurais de Hopfield

O sistema nervoso é constituído por células nervosas, chamadas de neurônios. Como ilustrado na Figura 2.1, ...

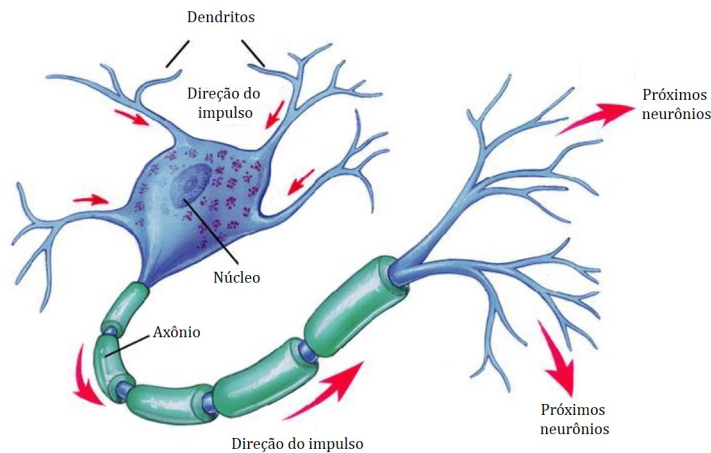


Figura 2.1: Neurônio biológico. Extraído de [8].

2.1 Neurônio Artificial

Neurônio Artificial (NA) é um modelo do Neurônio Biológico (NB). O NA implementa um mapeamento não-linear de \mathbb{R}^N usualmente para $[0,1]$ ou $[-1,1]$, dependendo da função de ativação usada, onde N é o número de sinais de entrada da Rede Neural Artificial (RNA). A Figura 2.2 apresenta uma ilustração de um NA com as notações convencionais.

O sinal de entrada net é usualmente computado como a soma do produto dos sinais de entrada pelo respectivo peso sináptico,

$$net = \sum_{i=1}^N z_i v_i \quad (2.1)$$

Os tipos de funções de ativação mais utilizados são [7]:

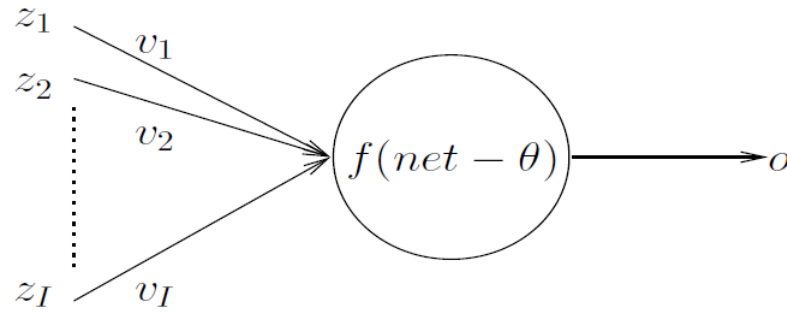


Figura 2.2: Um Neurônio Artificial. Extraído de [7].

- Função Linear: A função linear produz uma saída linearmente modulada. λ determina a inclinação da reta;

$$f(net - \theta) = \lambda(net - \theta) \quad (2.2)$$

- Função Degrau: A função degrau produz uma saída composta por um de dois valores, dependendo de um limiar. Usualmente, um valor binário é utilizado $\gamma_1 = 1$ ou $\gamma_2 = 0$; ou um valor bipolar $\gamma_1 = 1$ ou $\gamma_2 = -1$;

$$f(net - \theta) = \begin{cases} \gamma_1 & \text{se } net \geq \theta \\ \gamma_2 & \text{se } net < \theta \end{cases} \quad (2.3)$$

- Função Rampa: A função rampa é uma combinação das funções linear e degrau.

$$f(net - \theta) = \begin{cases} \gamma & \text{se } net - \theta \geq \epsilon \\ net - \theta & \text{se } -\epsilon < net - \theta < \epsilon \\ \gamma & \text{se } net - \theta \leq -\epsilon \end{cases} \quad (2.4)$$

- Função Sigmoide Logística: A função sigmoide é uma versão contínua da função rampa. Os valores variam continuamente de 0 a 1.

$$f(net - \theta) = \frac{1}{1 + e^{-\lambda(net - \theta)}} \quad (2.5)$$

- Função Tangente Hiperbólica: A função tangente hiperbólica é uma versão contínua da função rampa. Os valores variam continuamente de -1 a 1.

$$f(net - \theta) = \frac{e^{\lambda(net - \theta)} - e^{-\lambda(net - \theta)}}{e^{\lambda(net - \theta)} + e^{-\lambda(net - \theta)}} \quad (2.6)$$

As HNNs são formadas por neurônios de McCulloch e Pitts (MCP). Esta rede possui uma arquitetura do tipo recorrente na qual cada neurônio possui um somatório do con-

junto de entradas ponderadas net_i , uma saída V_i que é o resultado da aplicação de uma função de ativação sobre net_i . A saída V_i após um atraso de tempo, é aplicada à entrada dos outros neurônios ponderada por um peso sináptico T_{ij} somado a uma polarização externa (*bias*) I_i , modificando posteriormente a entrada do neurônio, e alterando assim o estado da rede neural. As saídas dos neurônios podem ser calculadas usando a Equação (2.7).

$$V_i = g_i(net_i) = \frac{1}{1 + e^{-\lambda_i net_i}}. \quad (2.7)$$

em que $net_i(t+1) = \sum_{\substack{j=1 \\ j \neq i}}^n T_{ij} V_j(t) + I_i(t)$.

Capítulo 3

Modelos de HNN para Redes de Comunicação

Neste capítulo são descritos os principais modelos de redes neurais de Hopfield utilizados em redes de comunicação...

3.1 HNN para roteamento

A utilização de HNN para solucionar o problema do menor caminho a partir de um determinado par origem-destino foi iniciado por Rauch e Winarske [12].

A função de energia da HNN de roteamento é descrita como:

$$\begin{aligned} E = & \frac{\mu_1}{2} \sum_{\substack{x=1 \\ (x,i) \neq (d,s)}}^N \sum_{\substack{i=1 \\ i \neq x}}^N C_{xi} V_{xi} + \frac{\mu_2}{2} \sum_{\substack{x=1 \\ (x,i) \neq (d,s)}}^N \sum_{\substack{i=1 \\ i \neq x}}^N \rho_{xi} V_{xi} \\ & + \frac{\mu_3}{2} \sum_{x=1}^N \left\{ \sum_{\substack{i=1 \\ i \neq x}}^N V_{xi} - \sum_{\substack{i=1 \\ i \neq x}}^N V_{ix} \right\}^2 \\ & + \frac{\mu_2}{2} \sum_{x=1}^N \sum_{\substack{i=1 \\ i \neq x}}^N V_{xi} (1 - V_{xi}) + \frac{\mu_5}{2} (1 - V_{ds}), \end{aligned} \quad (3.1)$$

onde C_{xi} é a matriz de custo, ρ_{xi} é a matriz topológica e μ_1 , μ_2 , μ_3 , μ_4 e μ_5 são constantes. μ_1 minimiza o custo total de um caminho considerando o custo dos enlaces existentes. μ_2 previne a inclusão de enlaces não existentes no caminho escolhido. μ_3 é zero para cada nó contido em um caminho válido. μ_4 força a HNN convergir para um estado estável. Por fim, μ_5 é introduzido para assegurar que os nós origem e destino estejam contidos na solução.

3.1.1 Execução do algoritmo

O pseudocódigo do algoritmo de Ali e Kamoun [1] e abordagens similares é apresentado no Algoritmo 1.

Algoritmo 1: Pseudocódigo do algoritmo *Hopfield Neural Networks*.

início

Recebe parâmetros (A,B,C, $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5$);

Recebe C_{xi} ;

Recebe p_{xi} ;

Recebe origem e destino;

Insere ruído em U_{xi} ;

repita

| Atualiza os neurônios (U_{xi} e V_{xi});

até $\Delta V_{xi} < limiar$;

Calcula V_{xi} (binarização de V_{xi});

Retorna o caminho.

3.1.2 Modelo de Bastos-Filho et al.

Bastos-Filho et al. [2] propuseram uma simples equação de diferença finita e discreta em substituição à equação diferencial proposta por Ali e Kamoun [1], como descrito a seguir:

$$U_{xi}[k+1] = U_{xi}[k] + AU_{xi}[k-1] + BU_{xi}[k-2] + C \left(\sum_{y=1}^N \sum_{\substack{j=1 \\ j \neq y}}^N T_{xi,yj} V_{yj}[k] + CI_{xi}[k] \right) \quad (3.2)$$

3.1.3 Paralelismo

Os níveis de paralelismo variam desde paralelizar os cálculos dos bits até paralelizar a execução de diferentes treinamentos ao mesmo tempo [11]. Os tipos de paralelismo são:

- Paralelismo dos Neurônios: Uma vez que as entradas são inseridas nos neurônios, o processamento de cada um pode ser feito em paralelo.
- Paralelismo na Saída: O produto presente no somatório da saída de um neurônio pode ser feito em paralelo e a soma deles pode ser feita com alto grau de paralelismo;
- Paralelismo de Camada: Em redes multicamadas, diferentes camadas podem ser processadas em paralelo;
- Paralelismo entre Redes Neurais: Num sistema constituído por mais de uma rede neural, essas redes neurais podem ser processadas em paralelo;

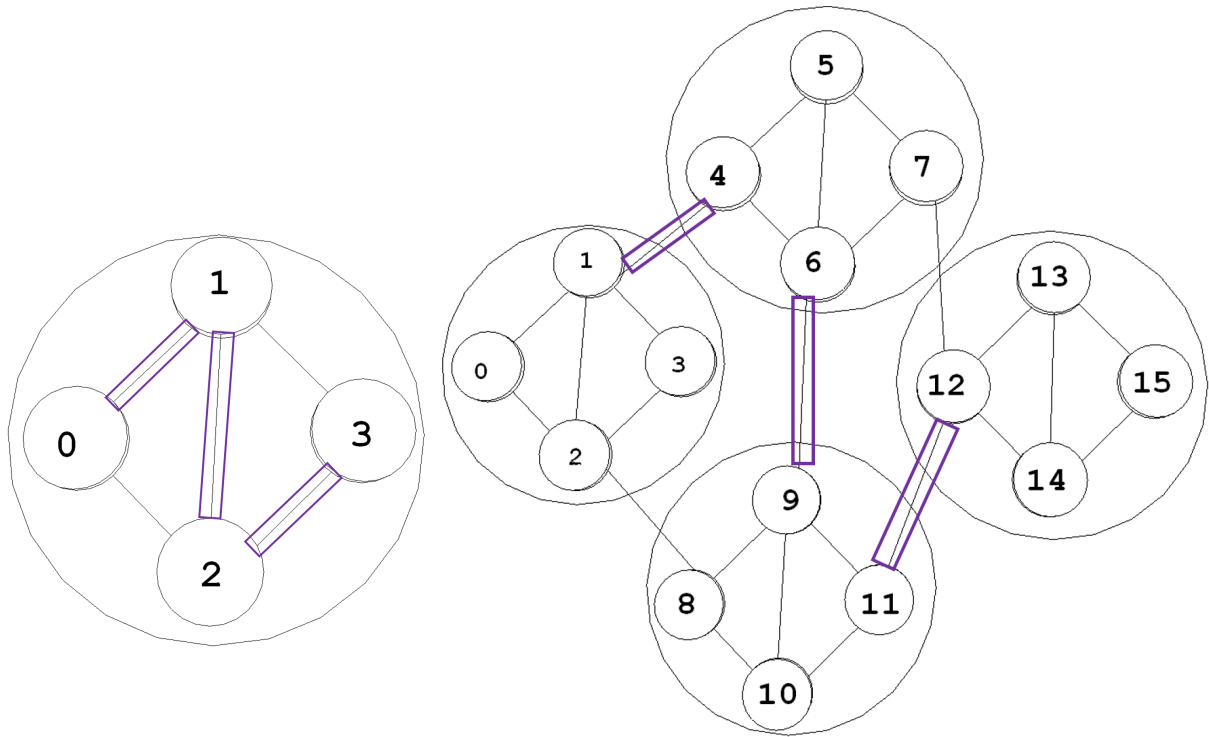
- Paralelismo no Treinamento: Várias sessões de treinamento numa rede neural podem ser executada paralelamente.

Vale salientar que alguns dos tipos de paralelismo não são passíveis de serem implementados em determinados modelos de FPGA. Em alguns casos, quando dois ou mais tipos de paralelismo são utilizados em conjunto, pode levar ao esgotamento dos recursos do dispositivo.

Capítulo 4

Redes Neurais de Hopfield Hierárquicas

A Figura 4.1 ilustra o cenário I de roteamento interdomínio...



(a) Resultado do cálculo da rota interdomínio. (b) Resultado do cálculo da rota interdomínio mostrado na camada inferior da rede de comunicação.

Figura 4.1: Resultado parcial do roteamento interdomínio entre os nós 0 e 15 (Cenário I).

4.1 Pseudo Código

O pseudocódigo do modelo ...

Algoritmo 2: Pseudocódigo do algoritmo de roteamento utilizando Redes Neurais de Hopfield Hierárquicas.

```

início
  obterMelhorRota(parOrigemDestino)
    se os nós origem e destino estiverem no mesmo nível hierárquico então
    |   calcularRota(parOrigemDestino);
    fim
  calcularRota(par)
    parAtual =
      obterParentesQueEstejamNoMesmoDominio(parOrigemDestino);
    hnn = obterHNN(obterDominio(parAtual));
    hnn.calcularRotaHNN(parAtual);
    se se o nível do par for maior do que nível do parAtual então
    |   pares = gerarNovosPares(par, parAtual, hnn.listaNeuroniosAtivos);
    |   para todo pares faça
    |   |   calcularRota(par);
    |   fim
    fim
  calcularRotaHNN(parOrigemDestino)
    Recebe parâmetros (A,B,C, $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5$ );
    Recebe  $C_{xi}$ ;
    Recebe  $p_{xi}$ ;
    Recebe origem e destino;
    Insere ruído em  $U_{xi}$ ;
    repita
    |   Atualiza os neurônios ( $U_{xi}$  e  $V_{xi}$ );
    até  $\Delta V_{xi} < \text{limiar}$ ;
    Calcula  $V_{xi}$  (binarização de  $V_{xi}$ );
    Retorna o caminho.
fim

```

Capítulo 5

Arranjo Experimental

Este capítulo tem o objetivo de descrever como a HHNN foi validada... A Tabela 5.1 contém as características estruturais das HNNs correspondentes ao cenários topológicos das duas redes de comunicação utilizados nos experimentos.

| Topologias | | | | | | |
|------------|-------------|--------------|--------------|-----------|--|---|
| Qtd. Nós | Qtd. Níveis | Configuração | Qtd Domínios | Qtd. HNNs | Qtd. Neurônios | Qtd. Conexões |
| 16 | 1 | 1x16 | 4 | 1 | $16 \times 15 = 240$ | $240 \times 240 = 57600$ |
| 16 | 2 | 4x4 | 4 | 5 | $5 \times (4 \times 3) = 60$ | $5 \times (12 \times 12) = 720$ |
| 64 | 1 | 1x64 | 1 | 1 | $64 \times 63 = 4032$ | $4032 \times 4032 = 16257024$ |
| 64 | 2 | 4x16 | 4 | 5 | $(4 \times 3) + 4 \times (16 \times 15) = 972$ | $12 \times 12 + 4 \times (240 \times 240) = 230544$ |
| 64 | 2 | 8x8 | 8 | 9 | $9 \times (8 \times 7) = 504$ | $9 \times (56 \times 56) = 28224$ |
| 64 | 3 | 4x4x4 | 16 | 21 | $21 \times (4 \times 3) = 252$ | $21 \times (12 \times 12) = 3024$ |

Tabela 5.1: Características estruturais das HHNNs correspondentes aos cenários topológicos das duas redes de comunicação utilizadas nos experimentos.

5.1 Implementações HNN e HHNN em Java

Como prova de conceito da ...

| Parâmetro | Valor |
|-----------------|-----------|
| μ_1 | 950 |
| μ_2 | 2500 |
| μ_3 | 2500 |
| μ_4 | 475 |
| μ_5 | 2500 |
| A | 10^{-3} |
| B | 10^{-3} |
| C | 10^{-3} |
| δV_{th} | 10^{-5} |
| λ | 1 |

Tabela 5.2: Tabela contendo os parâmetros da simulação em Java.

5.2 Implementações HNN e HHNN em FPGA

Com o objetivo de validar e avaliar a eficiência das ...

5.2.1 Decisões Arquiteturais do Modelo

Paralelismo

Além do paralelismo ...

Matriz de Sinapses

Para realizar o cálculo da matriz de sinapses $T_{xi,yj}$, ...

Os tipos de sinais definidos no modelo e as constantes utilizadas são mostrados nas Tabelas 5.3 ...

| Nome | Tipo | Objetivo |
|----------------------|----------------------------|---|
| yLBits | INTEGER | Define a quantidade de bits da parte decimal utilizada na saída dos neurônios. |
| LBits | INTEGER | Define a quantidade de bits da parte decimal da função de energia. |
| RBits | INTEGER | Define a quantidade de bits da parte fracionária de diversos sinais e tipos. |
| nodes | INTEGER | Define a quantidade de nós nas HNNs . |
| neurons | INTEGER | Define a quantidade de neurônios em cada HNN. |
| total_nodes | INTEGER | Define a quantidade total de nós na topologia de rede. |
| nodes_level_1 | INTEGER | Define a quantidade total de nós na camada 1 da topologia de rede. |
| nodes_level_2 | INTEGER | Define a quantidade total de nós na camada 2 da topologia de rede. |
| nodes_level_3 | INTEGER | Define a quantidade total de nós na camada 3 da topologia de rede. |
| levels | INTEGER | Define a quantidade de níveis hierárquicos. |
| qtd_interfaces | INTEGER | Define a quantidade de Interfaces no somador. |
| qtd_hnns | INTEGER | Define a quantidade de HHNs. |
| topology_map_level_2 | topology_map_level_2_array | Define o mapeamento entre os nós dos neurônios de diferentes níveis (Neste caso, níveis 1 e 2). |
| topology_map_level_3 | topology_map_level_2_array | Define o mapeamento entre os nós dos neurônios de diferentes níveis (Neste caso, níveis 2 e 3). |

Tabela 5.3: Constantes utilizadas na implementação de HHNN em VHDL.

Capítulo 6

Resultados

Após a conclusão das implementações dos modelos codificados em Java e VHDL das Redes Neurais de Hopfield Hierárquicas para roteamento de redes de comunicação, ...

6.1 Resultados do modelo codificado em Java

...

6.1.1 Análise estrutural

A Tabela 6.1 contém as ...

| Topologias | | | |
|--------------|-----------|----------------|---------------|
| Configuração | Qtd. HNNs | Qtd. Neurônios | Qtd. Conexões |
| 1x16 | 1 | 240 | 57600 |
| 4x4 | 5 | 60 | 720 |
| 1x64 | 1 | 4032 | 16257024 |
| 4x16 | 5 | 972 | 230544 |
| 8x8 | 9 | 504 | 28224 |
| 4x4x4 | 21 | 252 | 3024 |

Tabela 6.1: Características estruturais das HHNN referentes ao cenários topológicos das duas redes neurais de comunicação utilizados nos experimentos.

| Configuração | Redução Estrutural | |
|--------------|--------------------|----------|
| | Neurônios | Conexões |
| 4x4 | 75,00 % | 98,75 % |
| 4x16 | 75,89 % | 98,58 % |
| 8x8 | 87,50 % | 99,83 % |
| 4x4x4 | 93,75 % | 99,98 % |

Tabela 6.2: Redução da quantidade de neurônios e sinapses obtidos com a utilização do modelo HHNN.

6.1.2 Análise da convergência

Após terem sido analisadas as

| Pair | HNN 00 | | HNN 01 | | HNN 02 | | HNN 03 | | HNN 04 | | Total | |
|-------|--------------------|------------------|----------------------|-----------------|-------------------|------------------|-------------------|------------------|-------------------|------------------|-------------------|------------------|
| | Tempo (μs) | Iterações | Tempo (μs) | Iterações | Tempo (μs) | Iterações | Tempo (μs) | Iterações | Tempo (μs) | Iterações | Tempo (μs) | Iterações |
| 00-01 | 00,00 (00,00) | 00,00 (00,00) | 3790,58 (2543,21) | 60,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 4076,18 (1,74) | 60,00 (0,00) |
| 00-02 | 00,00 (00,00) | 00,00 (00,00) | 787,89 (378,75) | 66,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 935,28 (0,62) | 66,00 (0,00) |
| 00-03 | 00,00 (00,00) | 00,00 (00,00) | 577,63 (95,48) | 77,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 761,06 (0,37) | 77,00 (0,00) |
| 00-04 | 466,38 (87,69) | 60,00 (0,27) | 464,52 (62,15) | 60,00 (0,00) | 470,43 (90,76) | 61,70 (2,13) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 1724,46 (0,47) | 181,70 (2,13) |
| 00-05 | 411,75 (46,88) | 60,00 (0,00) | 411,75 (50,76) | 60,00 (0,00) | 415,47 (57,25) | 60,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 1576,67 (0,37) | 180,00 (0,00) |
| 00-06 | 378,58 (42,98) | 60,00 (0,00) | 374,12 (38,11) | 60,00 (0,00) | 410,20 (49,21) | 66,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 1518,87 (0,43) | 186,00 (0,00) |
| 00-07 | 379,25 (59,99) | 60,00 (0,00) | 373,58 (34,74) | 60,00 (0,00) | 452,68 (51,03) | 77,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 00,00 (00,00) | 1577,06 (0,45) | 197,00 (0,00) |
| 00-08 | 398,56 (30,11) | 66,00 (0,00) | 373,49 (39,02) | 60,00 (0,00) | 397,91 (39,11) | 66,00 (0,00) | 374,85 (57,75) | 60,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 2014,35 (0,46) | 252,00 (0,00) |
| 00-09 | 403,57 (53,29) | 66,00 (0,05) | 367,85 (36,19) | 60,00 (0,22) | 410,20 (48,00) | 66,00 (0,09) | 380,28 (38,12) | 62,17 (1,51) | 00,00 (00,00) | 00,00 (00,00) | 1951,00 (0,45) | 254,17 (1,51) |
| 00-10 | 531,70 (754,27) | 66,00 (0,00) | 376,39 (34,22) | 60,00 (0,00) | 399,40 (47,99) | 66,00 (0,00) | 393,15 (29,18) | 66,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 2104,78 (0,87) | 258,00 (0,00) |
| 00-11 | 410,66 (63,45) | 66,00 (0,00) | 394,78 (90,78) | 60,00 (0,00) | 421,01 (83,35) | 66,00 (0,00) | 461,62 (62,06) | 76,00 (0,00) | 00,00 (00,00) | 00,00 (00,00) | 2168,22 (0,59) | 268,00 (0,00) |
| 00-12 | 433,70 (243,78) | 77,00 (0,04) | 303,30 (63,65) | 60,00 (0,09) | 325,98 (38,09) | 66,00 (0,05) | 373,00 (47,17) | 76,00 (0,23) | 308,86 (50,48) | 61,93 (1,57) | 2187,04 (0,72) | 340,93 (1,57) |
| 00-13 | 353,38 (53,39) | 77,00 (0,00) | 276,90 (36,06) | 60,00 (0,00) | 301,10 (29,69) | 66,00 (0,00) | 346,12 (42,26) | 76,00 (0,00) | 273,00 (32,30) | 60,00 (0,00) | 1933,01 (0,44) | 339,00 (0,00) |
| 00-14 | 361,28 (18,38) | 77,00 (0,00) | 280,79 (24,40) | 60,00 (0,00) | 314,27 (39,54) | 66,00 (0,00) | 356,05 (62,61) | 76,00 (0,00) | 299,43 (3,22) | 66,00 (0,00) | 2010,97 (0,42) | 345,00 (0,00) |
| 00-15 | 401,36 (173,95) | 77,00 (0,00) | 287,75 (60,73) | 60,00 (0,00) | 310,18 (44,03) | 66,00 (0,00) | 355,97 (42,62) | 76,00 (0,00) | 373,94 (85,69) | 77,00 (0,00) | 2183,63 (0,61) | 356,00 (0,00) |

Tabela 6.3: Resultado do roteamento utilizando uma HHNN composta por 5 HNNs numa rede com 16 nós e 4 domínios.

Capítulo 7

Conclusões e Trabalhos Futuros

Nesta dissertação foram descritas ...

7.1 Trabalhos Futuros

Para o algoritmo de roteamento baseado em HNN proposto, como melhorias pode-se citar:

Bibliografia

- [1] M. K. M. Ali e F. Kamoun. “Neural networks for shortest path computation and routing in computer networks”. Em: *Neural Networks, IEEE Transactions on* 4.6 (nov. de 1993), pp. 941–954. ISSN: 1045-9227. DOI: 10.1109/72.286889.
- [2] C. J. A. Bastos-Filho, R. A. Santana e A. L. I. Oliveira. “A Novel Approach for a Routing Algorithm Based on a Discrete Time Hopfield Neural Network”. Em: *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*. Abr. de 2007, pp. 363–369. DOI: 10.1109/FOCI.2007.371497.
- [3] R. Bellman. “On a Routing Problem”. Em: *Quarterly of Applied Mathematics* 16.1 (1958), pp. 87–90. URL: <http://wisl.ece.cornell.edu/ECE794/Jan29/bellman1958.pdf>.
- [4] E. Bonabeau et al. “Routing in Telecommunications Networks With "smart" Ant-Like Agents”. Em: *In Proceedings of IATA'98, Second Int. Workshop on Intelligent Agents for Telecommunication Applications. Lectures Notes in AI*. Springer Verlag, 1998.
- [5] D. A. R. Chaves et al. “Novel physical impairments aware adaptive weight function for routing in all optical networks”. Em: *Microwave and Optoelectronics Conference, 2007. IMOC 2007. SBMO/IEEE MTT-S International*. 2007, pp. 194–198. DOI: 10.1109/IMOC.2007.4404244.
- [6] E. Dijkstra. “A Note on Two Problems in Connection with Graphs”. Em: *Numerische Mathematik* 1 (1959), pp. 269–271.
- [7] A. P. Engelbrecht. *Computational Intelligence: An Introduction*. 2nd. Wiley Publishing, 2007. ISBN: 0470035617.
- [8] Portal São Francisco. *Sistema Nervoso, Estrutura, Características, Função Sistema Nervoso*. Disponível em: <http://www.portalsaofrancisco.com.br/alfa/corpo-humano-sistema-nervoso/sistema-nervoso-1.php>. Acesso em 28/10/2014.
- [9] J. F. Martins-Filho et al. “Intelligent and fast IRWA algorithm based on power series and Particle Swarm Optimization”. Em: *Transparent Optical Networks, 2008. ICTON 2008. 10th Anniversary International Conference on*. Vol. 3. Jun. de 2008, pp. 158–161. DOI: 10.1109/ICTON.2008.4598679.

- [10] J. F. Martins-Filho et al. “Novel routing algorithm for transparent optical networks based on noise figure and amplifier saturation”. Em: *Microwave and Optoelectronics Conference, 2003. IMOC 2003. Proceedings of the 2003 SBMO/IEEE MTT-S International*. Vol. 2. 2003, 919–923 vol.2.
- [11] A. R. OMONDI e J. C. RAJAPAKSE. *FPGA Implementations of Neural Networks*. 2nd. Springer, 2006. ISBN: 0387284850.
- [12] H. E. Rauch e T. Winarske. “Neural networks for routing communication traffic”. Em: *Control Systems Magazine, IEEE* 8.2 (abr. de 1988), pp. 26–31. ISSN: 0272-1708. DOI: 10.1109/37.1870.
- [13] Bo Wen, R. Shenai e K. Sivalingam. “Routing, wavelength and time-slot-assignment algorithms for wavelength-routed optical WDM/TDM networks”. Em: *Lightwave Technology, Journal of* 23.9 (set. de 2005), pp. 2598–2609. ISSN: 0733-8724. DOI: 10.1109/JLT.2005.854039.

Apêndice A

Artigos Publicados

- H. O. da Silva e C. J. A. Bastos-Filho. Inter-Domain Routing for Communication Networks Using Hierarchical Hopfield Neural Networks. LA-CCI - 2014: *The Latin American Congress on Computational Intelligence*, San Carlos de Bariloche, Argentina, Jun. 2014.