

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

REDES NEURAIS BASEADAS NO MÉTODO DE GRUPO
DE MANIPULAÇÃO DE DADOS: TREINAMENTO,
IMPLEMENTAÇÕES E APLICAÇÕES

ANDRÉ LUIZ SORDI BRAGA

ORIENTADOR: CARLOS H. LLANOS

COORIENTADOR: LEANDRO DOS SANTOS COELHO
Pontifícia Universidade Católica do Paraná — PUCPR

TESE DE DOUTORADO EM SISTEMAS MECATRÔNICOS

BRASÍLIA – DF – 26 de fevereiro de 2013

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

REDES NEURAIS BASEADAS NO MÉTODO DE GRUPO DE
MANIPULAÇÃO DE DADOS: TREINAMENTO, IMPLEMENTAÇÕES
E APLICAÇÕES

ANDRÉ LUIZ SORDI BRAGA

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA
DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM SISTEMAS
MECATRÔNICOS.

APROVADA POR:

Prof. Carlos H. Llanos (ENM – UnB)
(Orientador)

Prof. Guilherme Caribé de Carvalho (ENM – UnB)
(Examinador Interno)

Prof. Maurício Ayala Rincón (CIC – UnB)
(Examinador Externo)

Prof. Antônio de Pádua Braga
Departamento de Engenharia Eletrônica – UFMG
(Examinador Externo)

Prof. Michael Hübner
Cadeira de Sistemas Embarcados da Tecnologia da Informação
Ruhr-Universität Bochum – Alemanha
(Examinador Externo)

BRASÍLIA/DF, 26 DE FEVEREIRO DE 2013.

FICHA CATALOGRÁFICA

BRAGA, ANDRÉ LUIZ SORDI

Redes Neurais Baseadas no Método de Grupo de Manipulação de Dados:

Treinamento, Implementações e Aplicações. [Distrito Federal] 2013.

ENM.TD-03/13, 132 p., 297 mm (ENM/FT/UnB, Doutor, Sistemas Mecatrônicos).

Tese de Doutorado - Universidade de Brasília.

Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. FPGA

2. RNA

3. Redes Neurais Artificiais

4. GMDH

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

BRAGA, A.L.S. (2013). Redes Neurais Baseadas no Método de Grupo de Manipulação de Dados: Treinamento, Implementações e Aplicações. Tese de Doutorado em Sistemas Mecatrônicos, Publicação ENM.TD-03/13, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 132p.

CESSÃO DE DIREITOS

NOME DO AUTOR: André Luiz Sordi Braga.

TÍTULO DA TESE: Redes Neurais Baseadas no Método de Grupo de Manipulação de Dados: Treinamento, Implementações e Aplicações.

GRAU/ANO: Doutor/2013

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese pode ser reproduzida sem a autorização por escrito do autor.

André Luiz Sordi Braga
Quadra 202 lote 03 ap. 103 Ed. Heitor Villa Lobos, Águas Claras
71937-720 - Brasília - DF

RESUMO

O Método de Grupo para Manipulação de Dados (GMDH – *Group Method of Data Handling*) é um modelo de rede neural artificial (RNA) constituído de neurônios compostos por polinômios de baixa ordem. Os neurônios GMDH dispensam o uso das complexas funções de ativação, que demandam grandes esforços dos microprocessadores para que realizem seus cálculos. A simplicidade dos neurônios, em conjunto com outras características, tornam o GMDH uma opção interessante para aplicações de engenharia. Desde a proposta seminal do método, em 1966, diversos trabalhos têm sido desenvolvidos com o intuito de melhorar seus resultados em áreas como mineração de dados, predição, modelagem de sistemas, otimização e reconhecimento de padrões. Este trabalho introduziu uma abordagem teórica e experimental no estudo do GMDH, lidando com vários aspectos de seu processo de treinamento. Como resultado, produziu-se uma ferramenta de desenvolvimento chamada *GMDH Box*, que permite que o desenvolvedor escolha diferentes parâmetros de treinamento e variações do método, que podem ser aplicados em diferentes áreas. Nesse sentido, três aplicações foram escolhidas: (a) modelagem e predição de sistemas dinâmicos; (b) reconhecimento de padrões em bioinformática e; (c) sistemas embarcados baseados em FPGAs. A identificação de sistemas dinâmicos não lineares é uma tarefa difícil e existem vários relatos na literatura sobre a utilização de diferentes modelos de RNAs para aproximar essa classe de problemas. Este trabalho introduziu a aplicação da *modelagem paralela* de GMDH nesta área, comparando seus resultados com outros obtidos utilizando-se redes neurais MLP (*Multiple Layers Perceptron*). Na área de bioinformática, o método GMDH foi testado com sete sequências de proteínas cujos tamanhos variam de 14 a 54 resíduos. Os resultados demonstraram que as estruturas terciárias preditas adotam uma forma similar às das estruturas experimentais. Na área de sistemas embarcados, partes do método GMDH foram implementadas em *hardware*, mais precisamente em FPGAs (*Field Programmable Gate Arrays*). Um conjunto de experimentos foi realizado em um PC utilizando o FPGA como um coprocessador acessado através do protocolo TCP/IP. O fluxo de projeto demonstrou que é possível implementar-se a rede em *hardware*, que pode ser utilizada como um acelerador em sistemas embarcados. Experimentos demonstraram que a implementação é efetiva para encontrar-se soluções de boa qualidade para o problema em análise e representam os primeiros resultados da técnica inovadora da aplicação de GMDH em *hardware* para a solução do problema da predição de estruturas de proteínas. Nas três aplicações, obteve-se resultados interessantes, relacionados ao desempenho e à aplicabilidade do método, comparando-os com resultados obtidos a partir de abordagens mais conhecidas como o modelo MLP.

ABSTRACT

GMDH (Group Method of Data Handling) are artificial neural networks (ANNs) composed of neurons constituted of low order polynomials. GMDH neurons are exempt from the use of complex activation functions, which require big efforts from microprocessors to perform their calculations. The simplicity of the GMDH neurons model, along with other characteristics, make GMDH an interesting option for engineering applications. Since the seminal proposal of the GMDH method in 1966, several works have been developed in order to improve its results in different areas such as data mining, knowledge discovery, forecasting, systems modeling, optimization and pattern recognition. This work has introduced a theoretical and experimental approach in the study of GMDH, discussing and addressing several aspects of its training process. This work has yielded a development tool called *GMDH Box*, which allows the designer to choose several GMDH training parameters and variations, which can be applied on different areas. In this direction, three applications have been chosen: (a) modeling and prediction of dynamic systems; (b) pattern recognition in bioinformatics and; (c) embedded systems based on FPGAs. Identification and prediction of nonlinear dynamic systems are difficult tasks and there are several reports in the literature about the utilization of different models of ANNs to approximate this class of systems. This work has introduced the application of GMDH *parallel modeling* to this problem, comparing its results with the ones obtained with MLP (Multiple Layers Perceptron) ANNs. In the bioinformatics area, the GMDH method was tested with seven protein sequences whose sizes vary from 14 to 54 amino acid residues. Results show that the predicted tertiary structures adopt a fold similar to the experimental structures. In the embedded systems area, portions of the GMDH method have been implemented in hardware, more precisely in FPGAs (Field Programmable Gate Arrays). A set of experiments have been performed on a PC using the FPGA as a co-processor accessed through sockets over the TCP/IP protocol. The design flow employed demonstrated that it is possible to implement the network in hardware, which can be used as an accelerator in embedded systems. Experiments show that the proposed implementation is effective in finding good quality solutions for the example problem, which represents the early results of the novel technique of applying the GMDH algorithms in hardware for solving the problem of protein structures prediction. In the three applications, this work has detected several interesting results, related to performance and suitability, comparing the results with more well-known approaches such as the MLP model.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	NOTAS HISTÓRICAS	1
1.2	JUSTIFICATIVA DO TRABALHO	6
1.3	OBJETIVOS	10
1.3.1	Objetivos Gerais	10
1.3.2	Objetivos Específicos	11
1.4	RESULTADOS E PUBLICAÇÕES	13
1.5	ESTRUTURA DA TESE	15
2	O MÉTODO DE GRUPO DE MANIPULAÇÃO DE DADOS — GMDH	16
2.1	CONSIDERAÇÕES INICIAIS	16
2.2	PRINCÍPIOS BÁSICOS	16
2.2.1	Visão geral do algoritmo	16
2.2.2	O treinamento	18
2.2.3	Critério de Seleção	22
2.2.4	Critério de Parada	25
2.3	RESUMO DO PROCESSO DE TREINAMENTO	25
2.4	O ALGORITMO DE TREINAMENTO COMBI-GMDH	26
2.5	APROXIMADORES UNIVERSAIS	28
2.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO	29
3	TÓPICOS SOBRE O DESEMPENHO COMPUTACIONAL DOS AL- GORITMOS E INOVAÇÕES PROPOSTAS NO MÉTODO DE TREINA- MENTO	31
3.1	CONSIDERAÇÕES INICIAIS	31
3.2	ADIÇÃO E REMOÇÃO DE NEURÔNIOS	31
3.3	SELEÇÃO DE NEURÔNIOS E INTRODUÇÃO DO CRITÉRIO DO LIMIAR DINÂMICO	37
3.4	POLINÔMIOS DE ORDEM MAIOR DO QUE 1 NO C-GMDH	39

3.5	CRITÉRIOS DE PARADA	40
3.6	OTIMIZAÇÃO <i>LS</i> E SOLUÇÃO DE SISTEMAS MAL CONDICIONADOS	41
3.7	CONSIDERAÇÕES FINAIS DO CAPÍTULO	43
4	MODELAGEM DE SISTEMAS DINÂMICOS	45
4.1	CONSIDERAÇÕES INICIAIS	45
4.2	TIPOS DE ENTRADAS	45
4.3	FORMAS DE IDENTIFICAÇÃO	46
4.4	PREDIÇÃO DE SAÍDAS DOS SISTEMAS	48
4.5	CONFIGURAÇÕES DOS EXPERIMENTOS	49
4.6	SISTEMA 1	51
4.7	SISTEMA 2	53
4.8	SISTEMA 3	54
4.9	SISTEMA 4	54
4.10	RESULTADOS EXPERIMENTAIS	55
4.10.1	Resultados para o Sistema 1	57
4.10.2	Resultados para o Sistema 2	60
4.10.3	Resultados para o Sistema 3	62
4.10.4	Resultados para o Sistema 4	64
4.11	CONSIDERAÇÕES FINAIS DO CAPÍTULO	67
5	PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS USANDO GMDH	70
5.1	CONSIDERAÇÕES INICIAIS	70
5.2	ESTRUTURAS E REPRESENTAÇÃO DE PROTEÍNAS	72
5.3	O MÉTODO	74
5.4	RESULTADOS EXPERIMENTAIS	78
5.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO	86
6	IMPLEMENTAÇÃO DE GMDH EM <i>HARDWARE</i> USANDO FPGAs	90
6.1	CONSIDERAÇÕES INICIAIS	90
6.2	MÓDULOS IMPLEMENTADOS EM <i>HARDWARE</i>	92
6.2.1	Co-verificação de <i>software-hardware</i>	94
6.2.2	O Módulo de Solução de Sistemas Lineares (LSSM)	95
6.3	RESULTADOS EXPERIMENTAIS	96
6.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO	100

7	CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	102
7.1	CONCLUSÕES	102
7.2	TRABALHOS FUTUROS	106
	REFERÊNCIAS BIBLIOGRÁFICAS	108
	APPENDICES	108
A	RESULTADOS EXPERIMENTAIS DAS MODELAGENS DE SISTEMAS DINÂMICOS	109
B	OS VINTE AMINOÁCIDOS E SUAS ESTRUTURAS	119

LISTA DE TABELAS

4.1	Tipos de Entrada da Rede	46
4.2	Configurações das tarefas das redes.	50
4.3	Configurações das redes neurais.	52
5.1	Estados conformacionais.	78
5.2	Os vinte resíduos de aminoácidos.	79
5.3	Sequências alvo	79
5.4	RMSD dos carbonos alfa das estruturas preditas em relação às estruturas experimentais	83
5.5	Análises das disposições das estruturas secundárias das conformações preditas.	85
5.6	Valores numéricos do diagrama de Ramachandran para as estruturas experimentais e preditas.	87
6.1	RMSD entre os carbonos alfa das estruturas preditas e experimental e análise da estrutura secundária das conformações preditas.	97
6.2	Valores de Ramachandran obtidos a partir das conformações experimen- tal e preditas.	99
A.1	Resultados para o <i>Sistema 1</i> , variação a	109
A.2	Resultados para o <i>Sistema 1</i> , variação b	111
A.3	Resultados para o <i>Sistema 2</i>	113
A.4	Resultados para o <i>Sistema 3</i>	115
A.5	Resultados para o <i>Sistema 4</i>	117
B.1	Os vinte aminoácidos.	119

LISTA DE FIGURAS

2.1	Comparação entre (a) uma rede Perceptron de múltiplas camadas totalmente conectada e (b) uma rede GMDH	18
2.2	Neurônio GMDH com um pré-processador não linear na entrada	20
2.3	Exemplo gráfico do processo e treinamento de uma rede GMDH	27
2.4	Possível topologia de uma rede C-GMDH treinada.	28
3.1	Quantidades de neurônios em uma rede O-GMDH	32
3.2	Uma rede C-GMDH sem encaminhamento de nós	33
3.3	Quantidades de neurônios em uma rede C-GMDH	34
4.1	Comparação entre identificações <i>série-paralela</i> e <i>paralela</i>	47
4.2	Treinamento e execução de redes neurais para a predição de valores do sistema modelado.	49
4.3	Resultados dos experimentos com o <i>Sistema 1</i> , variação a	59
4.4	Resultados dos experimentos com o <i>Sistema 1</i> , variação b	61
4.5	Resultados dos experimentos com o <i>Sistema 2</i>	63
4.6	Resultados dos experimentos com o <i>Sistema 3</i>	65
4.7	Resultados dos experimentos com o <i>Sistema 4</i>	66
5.1	Estruturas dos aminoácidos	72
5.2	Formação de um Dipeptídeo	73
5.3	Representação esquemática de um modelo de peptídeo.	73
5.4	Representação esquemática da fragmentação de uma sequência proteica	75
5.5	Construindo as amostras de treinamento.	77
5.6	Predição dos ângulos de torção ϕ e ψ do aminoácido central de cada fragmento da sequência alvo k	80
5.7	Representação em fita das estruturas obtidas pelo método experimental e as estruturas preditas.	81
5.8	Valores mínimos dos RMSDs de cada experimento	82
5.9	Valores médios dos RMSDs de cada experimento	82
5.10	Desvios padrão entre os RMSDs dos ensaios de cada experimento	84

6.1	Diagrama do neurônio GMDH em <i>hardware</i>	93
6.2	Neurônio GMDH e módulo LSSM como periféricos de um processador MicroBlaze	95
6.3	LSSM – Módulo de solução de sistemas lineares	97
6.4	Representação em fita das estruturas experimental e preditas.	98
6.5	Diagramas de Ramachandran para as estruturas experimentais e preditas.	99

1 INTRODUÇÃO

1.1 NOTAS HISTÓRICAS

No ano de 1800, um grupo de vinte e quatro cientistas, liderados por Franz Xaver von Zach, inspirados pela Lei de Titus-Bode, buscavam um planeta existente entre as órbitas de Marte e Júpiter. A Lei de Titus-Bode, agora desacreditada, afirmava que existiria um padrão nos *semieixos maiores*¹ das órbitas dos planetas conhecidos, quebrado apenas pela lacuna entre Marte e Júpiter. Entretanto, foi o astrônomo Giuseppe Piazzi, da Academia de Palermo, Sicília, quem descobriu o corpo celeste no dia primeiro de janeiro de 1801, dando-lhe o nome de Ceres, em referência à deusa romana da agricultura [2].

Primeiramente, Piazzi pensou que se tratasse de um cometa. Ele observou o objeto até o dia 11 de fevereiro de 1801, quando ficou seriamente doente, mas já o tinha anunciando em 24 de janeiro de 1801 em cartas para dois outros astrônomos. As cartas chegaram ao seu destino apenas no final do mês de março, quando o objeto já se encontrava posicionado muito próximo do brilho do sol, impedindo que outros astrônomos confirmassem as observações. Era necessário aguardar até setembro, quando Ceres sairia da zona inobservável. Os poucos dados registrados por Piazzi, que foram divulgados publicamente, eram insuficientes para fornecerem um conhecimento preciso sobre sua trajetória, apenas servindo para indicar que o corpo seguia uma órbita circular entre Marte e Júpiter. As únicas predições que permitiram que von Zach localizasse novamente o objeto, no dia 31 de dezembro de 1801, foram as realizadas pelo jovem matemático Carl Friedrich Gauss, que verificara que os dados das observações sobre o objeto se enquadravam em uma órbita elíptica, da qual ele calculou os elementos utilizando o método dos mínimos quadrados [2, 3]. Como consequência disto, Ceres foi classificado como um planeta por um longo período, concluindo-se, depois, que representava o primeiro de uma nova classe de objetos denominados “asteroides”. Hoje é considerado um Planeta Anão [4].

Nascia, na virada do século XVIII para o século XIX, o mais importante método para a solução aproximada de sistemas sobredeterminados (aqueles que têm mais equações

¹*Semieixo maior*: é a metade do maior diâmetro de uma elipse [1].

do que incógnitas), como os existentes nos problemas de ajuste de curvas. Assim como no caso da descoberta do planeta Ceres, muitas observações científicas geram dados experimentais que precisam ser modelados, tanto para a realização de extrapolações como de interpolações.

Ajustes de curvas também podem ser realizados através de polinômios quadráticos ou de ordem superior através da regressão polinomial, que envolve o cálculo das derivadas parciais em relação aos coeficientes da expressão polinomial. As técnicas de interpolação podem utilizar um único polinômio (com ordem $m = n - 1$, onde n é o número de pontos observados), mais de um polinômio (linear, quadrático ou cúbico), além dos polinômios de Lagrange e de Newton [5].

Um dos primeiros trabalhos computacionais relacionados à aproximação de curvas baseada em polinômios foi realizado por Dennis Gabor. No ano de 1961, ele publicou a descrição de sua máquina, o Filtro Não-linear Universal, Preditor e Simulador [6], baseado no modelo de Gabor-Kolmogorov. Nas palavras de seu autor, o filtro era “uma máquina analógica com 18 entradas, capaz de computar, em aproximadamente 2,5 milissegundos, 94 termos de um polinômio” [7]. Gabor, que vinha desenvolvendo sua máquina desde 1954, propôs e construiu um filtro flexível que otimizava sua resposta através do aprendizado. Essa flexibilidade era fornecida pela presença de parâmetros que eram ajustados pelo critério do mínimo erro médio quadrado, calculado entre os resultados obtidos pelo filtro em relação a uma função alvo. O autor explica que isso era feito pela repetida alimentação da máquina com um registro do processo estocástico e pelo ajuste dos coeficientes variáveis, um por vez, após cada rodada, usando uma estratégia que assegurava que o erro decrementaria monotonicamente [6].

Em 1966, o acadêmico ucraniano Alexey Grigorevich Ivakhnenko apresentou seu trabalho: *Group Method of Data Handling — A Rival of the Method of Stochastic Approximation* (Método de Grupo de Manipulação de Dados — Um Rival do Método da Aproximação Estocástica), na revista *Soviet Automatic Control* [8]. O método de Ivakhnenko, também conhecido pela sigla GMDH (iniciais do nome do método), modela as entradas e saídas de sistemas complexos iterativamente, criando uma estrutura que se assemelha a uma **rede neural artificial** do tipo Perceptron de múltiplas camadas, sem deixar, contudo, de guardar importantes diferenças relacionadas à topologia e às estruturas internas dos elementos processadores. A principal característica distintiva em relação ao modelo MLP, é que uma rede GMDH pode ser tratada como um polinômio de uma determinada ordem. A cada iteração, durante o treinamento, a complexidade

desse polinômio é incrementada com novos termos, cujos coeficientes são calculados pelo método dos mínimos quadrados.

No contexto de redes neurais artificiais, Simon Haykin apresenta a seguinte definição sobre o tema [9]:

Uma rede neural é um processador distribuído massivamente paralelo, formado por unidades de processamento simples, que tem uma propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso.

O procedimento de aprendizagem das redes neurais está relacionado com a teoria da filtragem adaptativa [9]. Em 1960, três importantes trabalhos nessa área foram apresentados ou consolidados: Rosenblat apresenta seu teorema de convergência do Perceptron, Widrow e Hoff introduziram o teorema do mínimo quadrado médio (LMS – *Least Mean-Square*) e, com ele, desenvolveram o Adaline (*Adaptive Linear Element* – elemento linear adaptativo) [10]. Neste contexto, o Perceptron e o Adaline são clássicos exemplos de unidades processadoras de redes neurais artificiais. Historicamente, o Perceptron foi desacreditado por Minsky e Papert em 1969, que provaram que o modelo não poderia fazer algumas generalizações globais baseadas em exemplos aprendidos localmente, o que lançou dúvidas sobre as capacidades funcionais das redes neurais em geral até meados da década de 1980. Após essa intermitência, vários sucessos foram alcançados com muitos modelos de RNAs, por exemplo, com os Perceptrons de múltiplas camadas. [9, 11].

Este contexto histórico mostra que os surgimentos das áreas da filtragem adaptativa e das redes neurais artificiais estão muito ligados [11], embora as pesquisas com RNAs tenham mantido ainda sua inspiração nos modelos biológicos. Um exemplo disso é o uso da função de ativação, existente no Perceptron e em outros tipos de neurônios, que se assemelha ao potencial de ativação dos neurônios naturais. O GMDH de Ivakhnenko, por outro lado, baseado nas pesquisas de Kolmogorov, Wiener e Gabor, introduziu uma abordagem conexionista aos modelos de regressão polinomial. O GMDH produz modelos equivalentes a polinômios de alta ordem que podem ser usados como aproximadores ou filtros não lineares. Durante o treinamento, a complexidade do polinômio é aumentada iterativamente através da concatenação de descrições polinomiais parciais, ajustadas pelos mínimos quadrados. As descrições parciais são avaliadas por um

critério de seleção que visa eliminar aquelas que não se enquadrem nos requisitos estabelecidos. O método é capaz de realizar não apenas a seleção das descrições parciais, mas também a seleção das próprias entradas fornecidas ao polinômio, não sendo incomum que, após o treinamento, a rede produzida deixe de utilizar as entradas que não contribuem para o modelo. Essa capacidade de identificar entradas que podem ser descartadas representa uma característica *sui generis* deste modelo [12, 13]. Por apresentar uma abordagem conexionista semelhante às redes Perceptron, o GMDH tem sido denominado também Rede Neural Polinomial. Outros autores preferem o termo Rede Neural do Tipo GMDH (GMDH-T-ANN — *GMDH Type Artificial Neural Network*).

GMDH tem sido utilizado com sucesso em diversas áreas como mineração de dados, descoberta de conhecimento, predição, modelagem de sistemas complexos, reconhecimento de padrões e otimização [14, 15, 16, 17]. O método tem sido amplamente aplicado como uma forma de extrair modelos matemáticos dos sistemas estudados. Por outro lado, esses modelos são estáticos, sendo perceptível na literatura a busca por implementações que realizem adaptações dinâmicas às variações dos sistemas devido a modificações ambientais ou ao tempo. A adaptação dinâmica é uma característica fortemente presente na filtragem adaptativa e, conseqüentemente, nas redes neurais artificiais. As buscas por melhores resultados durante o treinamento de redes GMDH e pela adaptação dinâmica dos modelos podem ser notadas através das variações do método original, algumas das quais introduzidas pelo seu próprio fundador [18].

Trabalhos recentes vêm demonstrando a aplicabilidade do GMDH, ao mesmo tempo que ressaltam a necessidade da adaptação. A seguir, apresentam-se alguns exemplos de aplicações:

- Mingzhu Zhang et al. [19] utilizam GMDH para a previsão de séries de dados obtidos da Biblioteca de Séries de Dados Temporais [20] e também apresentam uma discussão sobre os critérios de seleção.
- Sung Han Lee et al. [21] utilizam GMDH para o diagnóstico de acidentes nucleares por perda de líquido de resfriamento (LOCA - *Loss of Coolant Accidents*).
- Madandoust et al. [22] treinaram redes GMDH para a predição da resistência à compressão em estruturas de concreto utilizando dados coletados pelo método de velocidade do pulso ultrassônico (UPV - *Ultrasonic Pulse Velocity*). Os autores utilizaram algoritmos genéticos (GA - *Genetic Algorithms*) para alterar a

estrutura da rede após o treinamento com o intuito de melhorar o ajuste de suas saídas aos dados previamente coletados.

- Elattar et al. [23] também utilizaram algoritmos genéticos para alterar a estrutura da rede após o treinamento. Em sua abordagem, cada cromossomo representa a estrutura de uma rede GMDH completa; são necessárias duas redes treinadas para se empregar a técnica. O cálculo dos coeficientes das descrições parciais é feito através do método dos mínimos quadrados. As entradas atrasadas são ponderadas de acordo com sua distância em relação à entrada atual; entradas passadas mais próximas da entrada atual têm peso maior do que as mais distantes.
- Aplicações em medicina têm mostrado também o potencial do modelo GMDH. Por exemplo, Kondo et al. [24] utilizam redes GMDH com neurônios modificados para a análise de imagens tomográficas em um sistema de detecção de câncer de fígado. A modificação introduzida foi a aplicação de funções de ativação exponenciais às saídas dos polinômios. Os cálculos dos coeficientes são feitos através de regressão linear, que envolve a linearização da função de ativação. As imagens das tomografias são percorridas com janelas móveis de 7 por 7 pixels. As entradas das redes são formadas pelas estatísticas extraídas das frações de imagens obtidas com o janelamento e pelas coordenadas centrais das janelas em relação às imagens completas das tomografias. As saídas das redes podem ter valores: 1, se os pixels centrais das janelas corresponderem a partes da imagem do fígado, e 0, se não corresponderem. Após o processamento de toda uma tomografia, uma imagem é formada a partir da extração dos pixels pertencentes à região da imagem onde se encontra o fígado. Essa imagem gerada é, então, submetida à mesma rede GMDH, o que produz uma nova imagem. O processo é realizado por seis vezes consecutivas e a comparação entre as imagens geradas em diferentes etapas permite que o sistema separe as regiões onde provavelmente se encontram nódulos cancerosos.
- Igualmente, aplicações em economia e temas correlatos são encontradas na literatura. Por exemplo, Chaudhuri [25] utiliza uma versão do método GMDH, chamada Combinacional, para modelar e prever os valores de ganhos das ações de uma montadora de automóveis indiana. Os dados de entrada são os valores anuais anteriores dos preços dos títulos, do capital da empresa e do giro de estoque.
- Ainda na área de economia, Jun-Shuw Lin [26] apresenta um modelo baseado em GMDH para a previsão da quantidade de circuitos integrados funcionais em

testes após a fabricação. O modelo foi projetado para ser usado como auxiliar nas decisões executivas dos fabricantes e apresentou resultados superiores a outros modelos utilizados como comparação, incluindo redes neurais treinadas com o algoritmo *backpropagation*.

- Aplicações na área de Mecatrônica também podem ser encontradas na literatura. Em seu artigo, Bingwen Cheb et al. [27] apresentam uma abordagem para a calibração de sistemas de visão estéreo utilizando GMDH para evitar o uso de modelos de distorção explícitos. Três redes GMDH são treinadas para receberem as coordenadas bidimensionais dos pontos alvos das imagens e fornecerem as coordenadas tridimensionais no ambiente. O treinamento é feito através do algoritmo LM - *Levenberg-Marquardt*. O projeto também emprega a conexão de nós da camada de entrada para as camadas seguintes não adjacentes.
- Outros exemplos de aplicações na área de controle, robótica e automação podem ser vistas em Amanivard et al. [28], Bueno [29], Sakaguchi e Yamamoto [30], Puig et al. [31], Witczak et al. [32].

1.2 JUSTIFICATIVA DO TRABALHO

Os algoritmos do método GMDH seguem uma estratégia indutiva para a solução de problemas. Dessa forma, a complexidade do modelo é determinada durante o processamento dos dados fornecidos, contrastando com algumas estratégias aplicadas a redes neurais, onde a topologia é definida antes do treinamento. GMDH também possui a capacidade de selecionar as entradas que não contribuem para a saída do modelo, em outras palavras, os algoritmos GMDH são capazes de identificar as entradas que podem ser desprezadas, excluindo-as dos cálculos após o treinamento.

Ao prescindir da descrição humana na fase de treinamento, os métodos GMDH podem evitar a interferência de eventuais vieses

criados pelo desenvolvedor. Para se iniciar a criação de um modelo, as únicas informações necessárias são as entradas e as saídas do sistema a ser identificado. O processo iterativo singular do método realiza combinações das entradas, dando origem a diferentes descrições parciais do modelo a ser aproximado. Durante o treinamento, o GMDH também faz combinações das descrições criadas para verificar as que melhor aproximam a saída desejada. A cada iteração, a complexidade do modelo como um

todo é aumentada, em busca de uma solução que atenda aos critérios estabelecidos. Por outro lado, essa complexidade é sempre controlada pela eliminação das descrições que não forem capazes de contribuir para a aproximação.

Além disso, os neurônios GMDH não utilizam funções de ativação, o que alivia a complexidade dos cálculos durante o treinamento e durante a execução dos modelos, evitando potências com expoentes fracionários e funções implementadas por aproximações em série, como a função exponencial e os logaritmos naturais.

O método GMDH funciona pelo ajuste dos parâmetros de expressões polinomiais simples, normalmente de ordem um ou dois, o que simplifica a implementação da regressão, que pode ser realizada pelo método dos mínimos quadrados. Existem relatos de implementações que utilizam os mínimos quadrados médios, o que torna o desenvolvimento ainda mais simples, mas ao custo de um aumento do tempo de treinamento e da inclusão de mais parâmetros de configuração para melhorar a convergência do LMS [33].

Durante a execução, as redes GMDH baseiam-se apenas em operações de soma, subtração e multiplicação. Como as descrições parciais são polinômios, as potências são inteiras não negativas, permitindo que sejam realizadas por multiplicações sucessivas. Pela simplicidade das operações realizadas e pela menor quantidade de interações entre os neurônios, redes GMDH tendem a consumir menos recursos computacionais, o que pode ser muito atrativo para implementações em sistemas embarcados, especialmente os realizados em plataformas reconfiguráveis como os FPGAs (*Field Programmable Gate Arrays* — Arranjos de Portas Lógicas Programáveis em Campo).

O algoritmo ainda carrega outras características vantajosas, como o aprendizado indutivo e a formação de uma estrutura dinâmica determinada durante o processo de treinamento, permitindo assim a seleção das melhores variáveis e, conseqüentemente, a eliminação daquelas que têm pouca influência nos resultados [12]. Redes GMDH dispensam a interferência do desenvolvedor tanto na especificação do número de camadas e da quantidade de neurônios por camada quanto na determinação da influência das variáveis disponíveis [33].

Depois de treinada uma rede, um modelo matemático pode ser extraído diretamente de sua estrutura, pois as características não lineares são todas obtidas pelas expressões polinomiais e pela conexão dessas descrições em camadas. Todos esses aspectos são atrativos para a utilização prática em sistemas que necessitem realizar a modelagem

ou a predição de sistemas não lineares. Isso se contrapõe ao que acontece com as redes Perceptron e similares, onde o conceito de “caixa preta” é aplicado ao sistema gerado, mesmo após o treinamento, o que não permite a obtenção de um modelo matemático da rede neural artificial implementada.

Por outro lado, o aprendizado envolve a criação de uma quantidade de descrições parciais e a exclusão de parte delas, ou seja, aquelas descrições que menos se adéquam ao critério de seleção escolhido. A experiência mostra que esse processo pode apresentar um alto consumo de recursos computacionais, principalmente memória, de acordo com a variação do algoritmo utilizada e com a quantidade de entradas da rede. Caso o número de descrições aumente muito, o tempo de processamento também será considerável. Dessa forma, o aprendizado indutivo e a estrutura dinâmica, consideradas como vantagens, podem acoimar os benefícios do método.

Apesar de existir uma grande variedade de trabalhos em que redes GMDH foram utilizadas, os artigos não costumam apresentar detalhes sobre as implementações. Algumas ferramentas que utilizam o método podem ser encontradas na Internet [34], por exemplo podem ser citadas:

- Zelezny apresentou, em 1999, seu NeuroNet GMDH for Java [35], um aplicativo escrito em Java que realiza treinamentos para modelagem e para predição de valores e utiliza arquivos de texto contendo os dados de entrada em formato decimal com ponto. O treinamento é realizado através do método de Widrow-Hoff para o treinamento de neurônios Adaline, ou seja, pelo algoritmo LMS. Pham e Liu descreveram o treinamento GMDH por este algoritmo em 1994 [33]. O aplicativo utiliza uma taxa de aprendizado fixa, fornecida pelo usuário. A quantidade de neurônios remanescentes após a seleção também é fixa e definida pelo usuário, bem como o percentual de amostras utilizadas para treinamento. Saídas das redes processadas não são fornecidas, mas é possível gerar gráficos comparativos entre as saídas dos sistemas.
- A empresa Peak Consulting [36] fornece um programa escrito em VBA [37] para planilhas Excel [38] para a criação de modelos baseados em GMDH.
- O *software* NeuroShell Trader [39] é voltado para o mercado de ações. Com os dados dos ativos, a aplicação, de acordo com seus criadores, é capaz de gerar modelos de previsão utilizando diversas técnicas, entre elas, o GMDH.

- Tetko et al. [40] apresentam uma página da Internet com diversas aplicações que podem ser executadas diretamente no navegador. Uma delas é voltada para a predição de sistemas usando GMDH [41].
- A empresa Zaptron Systems oferece uma aplicação para modelagem de dados que utiliza diversas técnicas, dentre elas o GMDH combinado com lógica Fuzzy [42].
- O aplicativo GMDH Shell [43] utiliza GMDH para realizar previsões de séries temporais, classificações, regressões e ajuste de curvas. Os dados obtidos podem ser exportados para planilhas eletrônicas.

Outros relatos de aplicativos podem ser encontrados na Internet no sítio sobre GMDH da Academia Nacional de Ciências da Ucrânia [44]. Entretanto, nenhum desses programas é fornecido com o código fonte, impedindo que se conheça a forma como os diversos aspectos do GMDH foram neles implementados. Além disso, são soluções fechadas ou de difícil integração com outras ferramentas, o que efetivamente impede a sua utilização de forma ampla e geral. Uma nobre exceção é o trabalho de Pham e Liu [33], que apresenta uma descrição didática do método GMDH. No apêndice F do livro dos mesmos autores [45], encontra-se uma listagem de um programa em linguagem C que implementa o algoritmo.

O artigo de Pham e Liu, bem como o programa que apresentam em seu livro, contribuem muito para a compreensão de detalhes práticos para a realização do método GMDH. Sem depreciar o valor dessa importante contribuição, é possível avaliar que algumas de suas características podem ser aperfeiçoadas, por exemplo:

- *Aceita apenas uma única variável de entrada:* o programa foi escrito para fazer a predição e a modelagem de sistemas com uma variável de entrada (e seus atrasos) e uma saída (que pode ser realimentada) na forma da Eq. 1.1, onde x_{i-k} e y_{i-l} representam as entradas e as saídas atrasadas, respectivamente. O programa permite o uso de diferentes atrasos, mas, mesmo assim, essa característica representa uma limitação importante, pois muitos sistemas possuem mais de uma variável de entrada nos processos.

$$y_i = f(x_i, x_{i-1}, x_{i-2}, \dots, x_{i-n}, y_{i-1}, y_{i-2}, \dots, y_{i-m}) \quad (1.1)$$

- *Exige uma constante intervenção manual* — Em cada execução, o programa solicita ao operador as configurações dos atrasos e, antes de incluir uma nova camada, demanda intervenção manual para a definição do limiar a ser utilizado na remoção dos neurônios.
- *Utiliza o método dos mínimos quadrados médios para realizar o ajuste dos pesos* — Se por um lado esse algoritmo é de implementação muito simples, por outro, a análise da convergência e da estabilidade do LMS é bastante complicada [11]. Se a taxa de aprendizagem for muito pequena, a convergência é lenta e se for muito grande, pode causar a interrupção prematura do algoritmo. O uso do LMS inclui uma complexidade desnecessária no algoritmo GMDH, pelo menos no que se refere ao estudo da convergência das redes, que é um dos focos desse trabalho.

1.3 OBJETIVOS

1.3.1 Objetivos Gerais

Existe na bibliografia um grande número de relatos sobre a utilização do GMDH em diversas aplicações. Em muitos deles os pesquisadores introduziram modificações no método de Ivakhnenko, alegando que buscavam maior adequação aos sistemas com que trabalharam. Contudo, não foram encontrados na literatura detalhes claros sobre as razões das alterações propostas. O intuito principal deste trabalho é avaliar os diversos aspectos do método de Ivakhnenko, levando em considerações os seguintes objetivos gerais:

1. **Realizar uma implementação aberta e flexível** para o treinamento de redes GMDH, que possa ser facilmente integrada a outros sistemas. As ferramentas disponibilizadas *on-line* não são de ampla utilização ou configuração, sendo que muitas apenas utilizam GMDH para seus propósitos específicos.
2. **Avaliar experimentalmente** os diversos aspectos computacionais do método, buscando identificar suas características, especialmente em problemas voltados para aplicações de engenharia.
3. **Propor melhorias**, incluindo **novos critérios e funcionalidades**, além de definir formas de explorá-los.

4. **Estudar a possibilidade de desenvolver aceleradores em *hardware*.** Neste sentido, pretende-se avaliar o uso de sistemas reconfiguráveis baseados em FPGAs tanto para o treinamento como para a aplicação de redes GMDH.
5. **Utilizar a ferramenta em sistemas reais** onde as capacidades de aproximação, predição ou reconhecimento de padrões presentes nas redes neurais artificiais sejam necessárias, permitindo que avaliem-se os resultados obtidos em condições práticas.

1.3.2 Objetivos Específicos

Para alcançar os objetivos gerais explicitados, este trabalho visa desenvolver as atividades descritas a seguir.

1. **Elaborar um ferramental para treinamento e execução de redes do tipo GMDH.**

Para tanto foi utilizada a linguagem de programação do ambiente do **MATLAB**, da empresa Mathworks [46]. Essa linguagem, em conjunto com seu ambiente de execução, é apropriada para a realização das operações matriciais necessárias, utilizando uma representação de dados flexível, extensamente adaptável e apropriada para a realização dos cálculos. Além disso, o projeto foi testado também no ambiente GNU Octave, uma ferramenta de *Software* Livre [47] que fornece compatibilidade com o MATLAB.

A ferramenta desenvolvida neste projeto, denominada ***GMDH Box***, está disponível ao público [48] sob uma licença de código aberto [49]. Não existem relatos de outras implementações tão flexíveis e abertas quanto esta.

2. **Avaliar o método em termos computacionais práticos, tais como consumo de memória e tempo de processamento.**
3. **Estudar características e introduzir parâmetros de controle no método para atender a alguns critérios, como:**
 - *Métodos e critérios de seleção de neurônios:* Analisando a bibliografia, percebe-se que esse é ainda um problema em aberto, merecendo a máxima atenção. A seleção dos neurônios durante o treinamento influencia profundamente a convergência do processo, tanto em termos matemáticos quanto computacionais. Por exemplo, uma seleção muito rigorosa limita a busca no

espaço de soluções enquanto que uma seleção pouco exigente desencadeia um alto consumo de recursos computacionais, especialmente de memória, o que torna o processo lento ou mesmo inexecutável.

- *Crítérios de parada*: Não há na literatura extensos detalhes práticos sobre a interrupção do treinamento. Em tese, as redes devem ser treinadas enquanto o seu desempenho puder ser melhorado. Entretanto, observou-se que, em muitos casos, a manutenção desse processo por longos períodos não fornece melhorias expressivas no desempenho. Além disso, como o treinamento envolve a criação de novas camadas, a manutenção do processo produz redes muito complexas. É necessário que se criem novas formas de avaliação para se determinar quando os treinamentos devem ser interrompidos, sejam esses mecanismos relacionados à complexidade dos modelos, à acurácia alcançada, ao tempo, ou a outros fatores que podem variar de acordo com a aplicação alvo.

4. Implementar variações algoritmo.

O GMDH, por ser um método de otimização, está sujeito a diversas condições, como a qualidade dos dados utilizados, os critérios de exploração do espaço de soluções e ainda a precisão e a eficiência das plataformas computacionais onde é executado.

Existem trabalhos relatando modificações no método original realizadas por diferentes pesquisadores, dentre eles o próprio Ivakhnenko, mas não foram encontrados, no material pesquisado, resultados comparativos que elucidem e corroborem as variações.

Em muitos artigos, são sugeridas novas formas de expressões matemáticas para a formação das descrições parciais. Algumas formas, por exemplo, utilizam exponenciais ou logaritmos, o que poderia abrandar uma das vantagens das redes GMDH: a utilização de funções matemáticas de simples processamento em diversas plataformas. Este trabalho está focado no uso de descrições parciais polinomiais, com o objetivo de preservar a simplicidade do método.

5. Explorar a implementação em *hardware*.

Uma das possibilidades exploradas por esse trabalho é a implementação de redes GMDH em *hardware* reconfigurável — mais especificamente os FPGAs — tendo como motivação algumas características do método, quais sejam: (a) utilização de funções polinomiais, que se baseiam em multiplicações e potências de números inteiros, que podem ser realizadas através de multiplicações sucessivas em *hard-*

ware, e; (b) isenção do uso de funções de ativação, que dependem de funções matemáticas de difícil implementação em *hardware*.

1.4 RESULTADOS E PUBLICAÇÕES

1. Ferramenta *GMDH Box*

Durante as pesquisas com o método, não foi encontrada uma ferramenta que fornecesse a flexibilidade para a realização dos experimentos, das avaliações e dos estudos necessários. Os *softwares* disponíveis ou são ferramentas comerciais que não dão acesso a configurações do processo de treinamento e execução, ou se tratam de implementações simplificadas, que não se adequam às extensas avaliações do método. Apesar de existirem diversas publicações que reportam o uso do método, o ferramental utilizado para os experimentos não é identificado pelos pesquisadores, levando a crer que realizaram implementações customizadas para as tarefas apresentadas. Nesse sentido, a primeira contribuição deste projeto foi a implementação da ferramenta de treinamento e de execução *GMDH Box*. Pelo que foi pesquisado neste trabalho, não há outro sistema similar para treinamento de redes GMDH.

2. Novas parametrizações do método

Encontram-se na literatura relatos de utilização de GMDH em sistemas diversos, entretanto, percebeu-se que muitos detalhes sobre aspectos computacionais do método não são claramente explicitados na bibliografia. Através de extensa quantidade de experimentos, foi possível estabelecer uma série de critérios para a aplicação do método. Foram introduzidos na ferramenta *GMDH Box* diferentes critérios de parada do treinamento, de seleção dos neurônios, de aumento da complexidade das descrições parciais e de realização das conexões entre os neurônios.

Além das contribuições ao método em si, foi necessário resolver questões técnicas relativas ao processamento dos dados. Introduziu-se, na ferramenta, funções para a realização otimizada do cálculo de combinações de elementos, para aliviar problemas com instabilidades causadas por sistemas de equações mal condicionados, dentre outros. Também foi necessário empreender um considerável esforço para viabilizar o treinamento intensamente dinâmico do GMDH, que envolve a criação e a remoção de neurônios e a organização de suas conexões.

Essas contribuições são apresentadas no capítulo 3.

3. Aplicação em modelagem paralela

Uma das características atrativas das redes neurais artificiais é sua capacidade de aprender o comportamento de sistemas. Isso é importante, por exemplo, na identificação de sistemas não lineares. Este trabalho introduz a aplicação de redes GMDH para a modelagem desse tipo de sistema utilizando uma estrutura paralela. Isso significa que os modelos GMDH foram validados utilizando realimentação a partir dos dados produzidos pelas próprias redes neurais, em contrapartida às implementações seriais-paralelas, que empregam a realimentação das redes com dados obtidos das saídas dos sistemas a que modelam. Apesar de ainda ser uma tarefa em progresso, este trabalho demonstrou que redes GMDH podem ser aplicadas para esse fim.

Essa aplicação foi primeiramente apresentada em um congresso internacional, no artigo *Comparing Artificial Neural Network Implementations for Prediction and Modeling of Dynamical Systems* [50], e foi selecionado para aparecer como capítulo do livro *ABCM Symposium Series in Mechatronics - Vol. 5* [51].

Um aprimoramento deste trabalho foi realizado e encontra-se descrito no capítulo 4.

4. Reconhecimento de estruturas de proteínas

Outra aplicação inovadora do método GMDH foi a sua utilização em um *framework* para o reconhecimento de estruturas estereoquímicas de proteínas. Essa aplicação foi publicada em revista científica internacional, sob o título *A GMDH Polynomial Neural Network-based Method to Predict Approximate Three-dimensional Structures of Polypeptides* [52]. Os detalhes desses experimentos estão descritos no capítulo 5.

5. Implementação em *hardware*

Esta pesquisa também introduziu a primeira implementação de GMDH baseada em *hardware*. Numa incursão inicial, um neurônio GMDH implementado em *hardware* reconfigurável foi utilizado em uma plataforma de co-verificação de *hardware/software*, utilizando comunicação em rede TCP/IP para ligar o algoritmo de treinamento e a plataforma de *hardware* onde o neurônio foi implementado e estava em execução. Essa implementação foi utilizada com sucesso numa versão simplificada do problema de reconhecimento de estruturas de proteínas. Detalhes sobre essa realização encontram-se no capítulo 6. Essa pesquisa foi publicada em uma conferência internacional sob o título *Hardware Implementation*

of GMDH-Type Artificial Neural Networks and its use to Predict Approximate Three-dimensional Structures of Proteins [53].

1.5 ESTRUTURA DA TESE

Este documento está estruturado da seguinte forma:

- O Capítulo 2 contém a descrição do método GMDH, explicitando a forma dos cálculos dos pesos e introduzindo os conceitos necessários.
- O Capítulo 3 apresenta os detalhes relacionados à implementação do método, bem como as inovações introduzidas por este trabalho.
- O Capítulo 4 contém os resultados dos experimentos com o uso de GMDH para a identificação de sistemas não lineares.
- O Capítulo 5 mostra os resultados da aplicação de GMDH na predição de estruturas estereoquímicas de proteínas.
- O Capítulo 6 descreve a utilização de GMDH em *hardware* através de uma plataforma de co-verificação de *hardware/software*.
- O Capítulo 7 apresenta as conclusões deste projeto, bem como as perspectivas para trabalhos futuros.

2 O MÉTODO DE GRUPO DE MANIPULAÇÃO DE DADOS — GMDH

2.1 CONSIDERAÇÕES INICIAIS

O Método de Grupo de Manipulação de Dados (*Group Method of Data Handling* — GMDH) baseia-se em uma abordagem indutiva para a estimação de modelos do tipo *caixa preta* com relações desconhecidas entre as suas variáveis [54], gerando uma estrutura ótima do modelo através de gerações sucessivas de descrições parciais (DPs) dos dados [55].

Desenvolvido no final da década de 1960 pelo Professor Alexey G. Ivakhnenko, o método é uma das primeiras abordagens relacionadas ao projeto sistematizado de relações não lineares, tendo sido aplicado com sucesso em diversas áreas como mineração de dados, descoberta de conhecimento (*knowledge discovery*), predição e modelagem de sistemas, reconhecimento de padrões e otimização [14, 15, 17, 16].

2.2 PRINCÍPIOS BÁSICOS

2.2.1 Visão geral do algoritmo

A criação de uma rede GMDH é feita iterativamente, onde cada iteração possui duas fases: (a) treinamento e (b) seleção. Para cada uma dessas fases utiliza-se um conjunto diferente de dados. Em outras palavras, o conjunto de dados coletados a partir das entradas e das saídas do sistema real é dividido em duas partes, ficando uma destinada ao treinamento da rede e outra à seleção dos neurônios.

A fase de treinamento compreende a criação de uma camada da rede neural, formada por diversas descrições polinomiais parciais do problema. Cada descrição parcial é contida por um neurônio da camada e seus coeficientes são calculados pelo método dos mínimos quadrados, ainda durante a fase de treinamento, utilizando-se exclusivamente os dados do conjunto de treinamento.

Na fase de seleção, os dados de seleção são apresentados às descrições parciais. As saídas das descrições são avaliadas segundo um critério de seleção e aqueles neurônios que não atenderem ao critério são eliminados.

Depois do treinamento e da seleção dos neurônios da primeira camada, um novo conjunto de descrições parciais é criado. Esse conjunto define uma nova camada da rede e utiliza como entradas as saídas da camada anterior. A regra básica é que esse processo deve se repetir até que uma camada nova não consiga melhorar o desempenho da rede.

Os algoritmos GMDH iniciam-se com a criação de polinômios de graus pequenos que tentam aproximar, mesmo que de forma superficial, os sistemas dos quais os dados foram extraídos. Os graus dos polinômios são aumentados iterativamente, buscando-se a criação de modelos cada vez mais apurados. Os polinômios normalmente seguem a forma de Kolmogorov-Gabor [6], que é uma versão discreta da função de Volterra [56], como apresentado na Eq. 2.1 [57, 58].

$$q = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} x_i x_j x_k + \dots \quad (2.1)$$

Para desenvolver uma descrição completa de um sistema (ver Eq. 2.2), várias camadas de descrições parciais são criadas de maneira a formarem uma rede inspirada nas redes neurais do tipo Perceptron de camadas múltiplas (*Multilayer Perceptron* - MLP). Uma rede GMDH tem as propriedades de uma função polinomial de grau elevado, como exemplificado pela Eq. 2.1 [58, 12].

$$q = f(x_1, x_2, x_3, \dots, x_n) \quad (2.2)$$

A Fig. 2.1 apresenta uma rede do tipo MLP totalmente conectada e uma rede GMDH comum. Em ambas as imagens apresentadas na figura, os nós quadrangulares são os nós de entrada da rede, os quais se localizam na camada de entrada. Os nós circulares, localizados nas camadas “ocultas” ou na camada de saída, são denominados *Elementos Processadores* (EPs) ou neurônios. O formato mais conhecido das redes MLP consiste em neurônios que são totalmente conectados, ou seja, os neurônios de uma camada são conectados a todos os neurônios da camada seguinte. Por outro lado, a forma original

das redes GMDH apresenta neurônios formados, cada um, por uma descrição parcial e conectados apenas a dois outros neurônios da camada anterior.

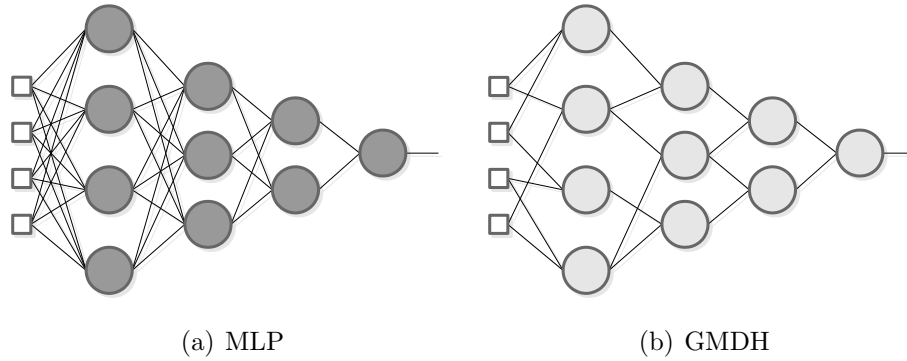


Figura 2.1: Comparação entre (a) uma rede Perceptron de múltiplas camadas totalmente conectada e (b) uma rede GMDH

Além das diferenças citadas, as redes do tipo MLP podem fornecer mais de uma saída, enquanto as redes GMDH fornecem uma única saída, devido a características relacionadas à sua forma de treinamento. Se mais de um mapeamento entre entradas e saídas for necessário, diferentes redes GMDH devem ser criadas para cada mapeamento.

2.2.2 O treinamento

De acordo com o número de entradas da rede, o algoritmo de treinamento instancia uma quantidade de descrições parciais que constituem os neurônios da primeira camada (cada neurônio possui uma descrição parcial). A quantidade de descrições parciais é definida pelo número de combinações das n entradas da rede, tomadas duas a duas, como demonstrado na Eq. 2.3. Cada descrição parcial toma duas entradas, produzindo uma única saída [12].

$$C_2^m = \frac{n!}{2!(n-2)!} = \frac{n^2 - n}{2} \quad (2.3)$$

A primeira camada é formada com $m = C_2^m$ neurônios (descrições parciais), onde n é o número de entradas da rede (ver Eq. 2.4).

$$\begin{aligned}
y_{11} &= f_{11}(x_1, x_2) \\
y_{12} &= f_{12}(x_1, x_3) \\
&\vdots \\
y_{1m} &= f_{1m}(x_{n-1}, x_n)
\end{aligned} \tag{2.4}$$

Na Eq. 2.4, y_{1k} representa a saída do neurônio k da camada 1 e f_{1k} é a descrição parcial contida por aquele neurônio, onde $k = [1, 2, \dots, m]$ e onde $[x_1, \dots, x_n]$ são as n entradas da rede.

Após o treinamento da primeira camada, um critério de seleção é utilizado para que sejam excluídos alguns neurônios, mantendo-se apenas aqueles que melhor se adequem ao critério. O conjunto com \hat{m} descrições parciais (neurônios) é o subconjunto das m descrições parciais originais que melhor se adequam ao critério de seleção (Eqs. 2.5 e 2.6). Mais detalhes sobre o processo de seleção encontram-se na seção 2.2.3.

$$[\hat{y}_{11}, \hat{y}_{12}, \dots, \hat{y}_{1\hat{m}}] \subset [y_{11}, y_{12}, \dots, y_{1m}] \tag{2.5}$$

$$[\hat{f}_{11}, \hat{f}_{12}, \dots, \hat{f}_{1\hat{m}}] \subset [f_{11}, f_{12}, \dots, f_{1m}] \tag{2.6}$$

onde

$$\hat{m} < m$$

Após a seleção dos melhores neurônios da primeira camada, uma segunda camada é adicionada. O número de neurônios nesta nova camada é $p = C_2^{\hat{m}}$, onde \hat{m} é a quantidade de neurônios na primeira camada após o processo e seleção (ver Eq. 2.7).

$$\begin{aligned}
y_{21} &= f_{21}(\hat{y}_{11}, \hat{y}_{12}) \\
y_{22} &= f_{22}(\hat{y}_{11}, \hat{y}_{13}) \\
&\vdots \\
y_{2p} &= f_{2p}(\hat{y}_{1(\hat{m}-1)}, \hat{y}_{1\hat{m}})
\end{aligned} \tag{2.7}$$

O processo de adição de camadas e seleção de neurônios é repetido até que um dos critérios de parada seja alcançado. Os critérios de parada são descritos na seção 2.2.4

[59]. Um exemplo do treinamento, explicitando os passos de criação e de seleção de neurônios, é dado pela Fig. 2.3 e descrito na seção 2.3.

Diferentemente do que acontece nas redes MLP, os neurônios GMDH dispensam o uso de funções de ativação. Uma descrição parcial típica usada por esses elementos processadores é representada na Eq. 2.8, onde x_i e x_j são as duas entradas do neurônio e $[a_0, \dots, a_5]$ são seus coeficientes.

$$y = a_0 + a_1x_i + a_2x_j + a_3x_i^2 + a_4x_j^2 + a_5x_ix_j \quad (2.8)$$

Com o intuito de determinar os coeficientes, um sistema de equações é criado para cada descrição parcial e resolvido pelo método dos mínimos quadrados [12]. Cada descrição parcial é uma função na forma da Eq. 2.8, que pode ser reescrita na forma matricial, dada pela Eq. 2.9, onde \mathbf{x} representa as **entradas pré-processadas** x_i e x_j do neurônio, expressadas por $\mathbf{x} = [1, x_i, x_j, x_i^2, x_j^2, (x_ix_j)]$ [33], e \mathbf{a} é um vetor coluna com os coeficientes a serem calculados: $\mathbf{a} = [a_0, a_1, a_2, a_3, a_4, a_5]^T$. A Fig. 2.2 mostra uma visão conceitual de um neurônio GMDH com um pré-processador não-linear que transforma as duas entradas do neurônio nos cinco valores livres do polinômio.

$$y = \mathbf{x}\mathbf{a} \quad (2.9)$$

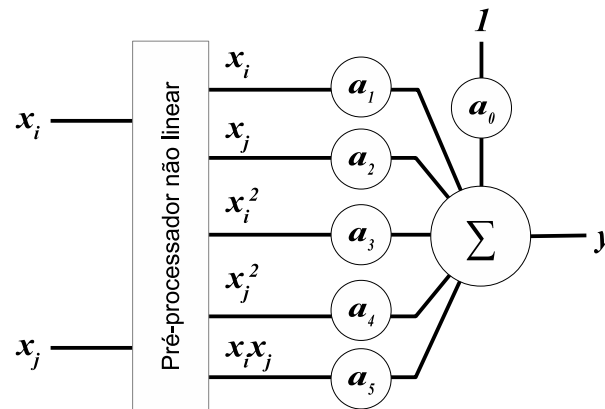


Figura 2.2: Neurônio GMDH representado com um pré-processador não linear na entrada. Os valores pré-processados x_i , x_j , x_i^2 , x_j^2 e x_ix_j correspondem às variáveis livres dos polinômios, enquanto $[a_0, \dots, a_5]$ são os coeficientes. Baseada em Pham e Liu [33].

Seja s o número de amostras adquiridas de um sistema de n entradas dado pela Eq. 2.2, como representado na Eq. 2.10. Considere-se, ainda, k o índice de um

neurônio específico entre os m neurônios criados na camada 1. A descrição parcial deste neurônio utiliza as entradas $x_i(t)$ e $x_j(t)$ de cada amostra, onde $t = [1, 2, \dots, s]$, $i = [1, 2, \dots, n-1]$, $j = [2, 3, \dots, n]$ e $j > i$. A saída no tempo t , $y_k(t)$, deste neurônio específico é dada pela Eq. 2.11 (o índice da camada a que o neurônio pertence está omitido para simplificar a representação).

$$\begin{aligned}
q(1) &= f(x_1(1), x_2(1), \dots, x_n(1)) \\
q(2) &= f(x_1(2), x_2(2), \dots, x_n(2)) \\
q(3) &= f(x_1(3), x_2(3), \dots, x_n(3)) \\
&\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
q(s) &= f(x_1(s), x_2(s), \dots, x_n(s))
\end{aligned} \tag{2.10}$$

$$y_k(t) = a_{k0} + a_{k1}x_i(t) + a_{k2}x_j(t) + a_{k3}x_i^2(t) + a_{k4}x_j^2(t) + a_{k5}x_i(t)x_j(t) \tag{2.11}$$

Fazendo-se $y_k(t) = q(t)$ e reescrevendo-se a Eq. 2.11 na forma da Eq. 2.9, a Eq. 2.12 é obtida, onde $\mathbf{x}(t)$ é o vetor linha (1 x 6) com as entradas da amostra t pré-processadas e \mathbf{a}_k é um vetor coluna (6 x 1) contendo os seis coeficientes do neurônio.

$$q(t) = \mathbf{x}(t)\mathbf{a}_k \tag{2.12}$$

Faça-se \mathbf{H} uma matriz que contenha as duas entradas pré-processadas do neurônio, x_i e x_j , de todas as amostras disponíveis. Considere-se, ainda, \mathbf{q} um vetor coluna com todas as saídas desejadas do sistema para cada amostra, como definido nas Equações 2.13 and Eq. 2.14.

$$\mathbf{H} = \begin{bmatrix} 1 & x_i(1) & x_j(1) & x_i^2(1) & x_j^2(1) & (x_i(1)x_j(1)) \\ 1 & x_i(2) & x_j(2) & x_i^2(2) & x_j^2(2) & (x_i(2)x_j(2)) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_i(s) & x_j(s) & x_i^2(s) & x_j^2(s) & (x_i(s)x_j(s)) \end{bmatrix} \tag{2.13}$$

$$\mathbf{q} = [q(1), q(2), q(3), \dots, q(s)]^T \quad (2.14)$$

A relação entre os pesos \mathbf{a}_k , as s entradas de todas as amostras e as saídas do neurônio é expressada pela Eq. 2.15, onde as matrizes \mathbf{q} , \mathbf{H} e \mathbf{a}_k têm dimensões $s \times 1$, $s \times 6$, e 6×1 , respectivamente [12].

$$\mathbf{q} = \mathbf{H}\mathbf{a}_k \quad (2.15)$$

As equações normais são formadas pela pré-multiplicação dos dois lados da equação pela transposta de \mathbf{H} (Eq. 2.16). A matriz $\mathbf{H}^T\mathbf{H}$ tem dimensões 6×6 e a solução das equações, apresentadas na Eq. 2.17, é encontrada invertendo-se essa matriz [12].

$$\mathbf{H}^T\mathbf{q} = (\mathbf{H}^T\mathbf{H})\mathbf{a}_k \quad (2.16)$$

$$\mathbf{a}_k = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{q} \quad (2.17)$$

Após o cálculo dos coeficientes de todas as equações parciais, aquelas que obtiverem os piores resultados nos cálculos dos critérios de seleção serão removidas da camada. O passo seguinte é criar uma nova camada com o número de neurônios definidos pela Eq. 2.3, substituindo n pelo número de neurônios remanescentes na camada atual.

2.2.3 Critério de Seleção

Com os coeficientes obtidos durante o treinamento de uma camada, o processo de seleção remove as descrições que se mostrarem menos úteis ou “prejudiciais” [12]. Em seguida, uma nova camada é adicionada. As descrições parciais da nova camada tomam como entradas as saídas dos neurônios das camadas anteriores.

Esse procedimento pode ser comparado com o princípio de seleção de fenótipos de plantas e animais. Por exemplo, após o cultivo de uma primeira geração de plantas, aquelas que melhor se adequam às características desejadas são fertilizadas de maneira cruzada e aquelas que menos se conformam aos critérios são descartadas. As sementes resultantes são cultivadas e as plantas daí obtidas são selecionadas novamente, fertilizadas

entre si e suas sementes cultivadas. Esse processo pode continuar até que uma geração de plantas produza indivíduos que se adequem aos critérios predefinidos [12, 13].

Nos algoritmos GMDH, a seleção é útil para que se removam as descrições parciais que aproximem excessivamente os dados de treinamento, isto é, que não descrevam precisamente as relações entre as entradas e as saídas [13]. Através da remoção dos “indivíduos” indesejáveis, a seleção também reduz o número de “descendentes”, diminuindo assim o tamanho da rede e, conseqüentemente, o consumo de recursos computacionais para realizar seu processamento.

A utilização dos procedimentos de seleção aproxima as redes GMDH das técnicas bioinspiradas, tais como algoritmos genéticos e otimização por enxames de partículas, dentre outros, e mostra outra diferença fundamental entre GMDH e outros modelos de RNAs.

Para realizar-se a seleção de neurônios nos algoritmos GMDH, as amostras de dados expressadas na Eq. 2.10 devem ser divididas em dois grupos: (a) amostras de treinamento e (b) amostras de seleção. Os dados de seleção normalmente são compostos de um terço ($1/3$) do total de amostras. Nos experimentos apresentados neste trabalho, os conjuntos de seleção foram obtidos normalmente da terceira parte ao final da sequência de amostras: $t \in (s - \lceil s/3 \rceil, s]$, onde o operador $\lceil \cdot \rceil$ significa “o inteiro mais próximo”. Entretanto, é possível alterar-se a ordem das amostras colhidas com o intuito de criar-se diferentes conjuntos de treinamento e de seleção.

Com a divisão dos dados, o treinamento, ou seja, os procedimentos necessários para o cálculo dos pesos dos neurônios (coeficientes das descrições parciais), é realizado utilizando-se apenas as amostras de treinamento. Isso significa que, nas Equações 2.13 e 2.14, apenas as amostras pertencentes ao intervalo $[1, \lceil s/3 \rceil]$ serão utilizadas. Conseqüentemente, os coeficientes calculados pela Eq. 2.17 levarão em conta apenas os valores de entrada e saída desse intervalo.

Alterar as amostras do conjunto de treinamento modifica os sistemas de equações que são utilizados para calcular os coeficientes das descrições parciais. Por conseqüência, os dados utilizados durante o processo de seleção também são modificados. O conhecimento sobre os dados coletados contribui para uma separação das amostras que represente bem o comportamento do sistema a ser identificado. Por exemplo, se um sistema tem seu comportamento modificado durante o tempo, é importante que os

conjuntos de dados de treinamento contêm amostras que abrangem todos os comportamentos possíveis. Obviamente, o conjunto de dados de seleção também deve conter dados que representam os comportamentos do sistema, para que o critério seja corretamente avaliado.

Após o cálculo dos coeficientes, as amostras de seleção são apresentadas aos neurônios treinados e suas saídas são calculadas para essas amostras. Como esses dados não foram utilizados para a definição dos coeficientes, eles podem ser aproveitados para avaliar quais neurônios conseguiram extrair melhor a relação entre as entradas e saídas, evitando-se a sobreaproximação.

Para a avaliação do desempenho dos neurônios, uma das medidas mais utilizadas é o erro médio quadrado (MSE — *Mean Squared Error*) entre as saídas dos neurônios e as saídas desejadas para cada amostra. Essa medida é conveniente devido ao fato de que cada expressão polinomial, durante o treinamento, tem seus coeficientes ajustados visando minimizar a medida do erro global, calculado pela soma dos quadrados dos erros. As principais vantagens do uso do MSE são que (a) os erros quadrados são sempre positivos (vantajoso porque erros positivos não são cancelados por erros negativos) e (b) os resíduos maiores têm mais peso no erro global [5]. Neste trabalho, utilizou-se o MSE como medida de desempenho em todos os experimentos, variando-se apenas o critério de avaliação utilizado sobre essa medida. O capítulo 3 apresenta uma discussão sobre os critérios de avaliação.

O MSE é calculado pela Eq. 2.18 sobre os dados de seleção, após o cálculo dos coeficientes das descrições parciais, onde y_{ln} designa as saídas do neurônio n da camada l ; u é a quantidade de amostras de seleção e $q(i)$ é a saída desejada da rede para a amostra i .

$$\mathbf{MSE}(y_{ln}) = \frac{1}{u} \sum_{i=1}^u (y_{ln}(i) - q(i))^2 \quad (2.18)$$

Para a exclusão dos neurônios, os métodos mais comuns apresentados na literatura são (a) remoção por *limiar fixo* e (b) por *número de neurônios remanescentes*. O primeiro método remove os neurônios que obtiveram valores de MSE superiores a um valor fixo. O método de remoção por número de neurônios remanescentes, por sua vez, exclui os neurônios com maiores valores de MSE mas mantém um número mínimo determinado de neurônios na camada.

No contexto da pesquisa alvo desta tese, uma nova medida para a seleção foi introduzida, o *limiar dinâmico*, obtido pelo cálculo da mediana dos valores de MSE de todos os neurônios da camada. A seção 3.3 apresenta mais detalhes sobre esse método.

2.2.4 Critério de Parada

Em linhas gerais, o processo de treinamento consiste em: (a) a adição de camadas, (b) o cálculo dos pesos e (c) a remoção dos neurônios com pior desempenho. A condição de parada da rede é a impossibilidade de se melhorar o seu desempenho pela adição de novas camadas. Isso acontece basicamente em duas situações: (a) quando a camada recém treinada contém apenas um neurônio após a seleção ou (b) quando o menor valor de MSE obtido após a seleção é maior do que o menor MSE da camada anterior.

Se a última camada contém apenas um único neurônio, esse neurônio passa a fornecer a saída da rede. Por outro lado, se o menor MSE da última camada for maior do que o da camada precedente, ela será removida. A camada anterior, que passa a ser a última camada, invariavelmente conterà mais de uma descrição parcial. Neste caso, o neurônio que produziu o menor MSE é preservado e todos os demais daquela camada são excluídos. O último estágio do treinamento consiste na remoção de todos os neurônios das camadas anteriores que não estão de alguma forma conectados à saída da rede [33].

2.3 RESUMO DO PROCESSO DE TREINAMENTO

Os passos descritos a seguir sumarizam os processos de criação e de treinamento das redes neurais artificiais do tipo GMDH:

1. **Criação da primeira camada da rede:** Na primeira camada, cada neurônio é uma descrição parcial do sistema, definida pela Eq. 2.8. O número de neurônios é dado pela Eq. 2.3. A Fig. 2.3(1) mostra uma camada criada para uma rede com quatro entradas.
2. **Cálculo dos coeficientes das descrições parciais:** Para calcular os coeficientes, usa-se a Eq. 2.17.
3. **Avaliação do desempenho e remoção dos neurônios:** Utilizando-se os dados de seleção, calcula-se o MSE de cada neurônio e aqueles que fornecerem

os piores resultados, segundo o critério utilizado, devem ser removidos. A Fig. 2.3(2) retrata os neurônios a serem removidos utilizando uma cor mais clara do que a dos outros neurônios.

4. **Criação de uma nova camada:** A nova camada toma como entradas as saídas dos neurônios da camada anterior. O número de neurônios nesta camada também é dado pela Eq. 2.3. A Fig. 2.3(2) mostra uma nova camada após a seleção dos neurônios da primeira. Antes de criar-se uma nova camada, contudo, deve-se realizar a verificação explicitada no item 6.
5. **Treinamento e seleção dos neurônios na nova camada:** A nova camada passa pelo mesmo processo realizado na camada anterior. Após a seleção, inclui-se uma nova camada, como visto na Fig. 2.3(3).
6. **Verificar o primeiro critério de parada:** Se qualquer das camadas em treinamento contiver apenas um único neurônio, após a seleção, interrompe-se o processo. É o caso apresentado na Fig. 2.3(4).
7. **Verificar o segundo critério de parada — Avaliar qualidade da camada:** Se, em qualquer camada, exceto a primeira, o desempenho for inferior ao melhor desempenho da camada anterior, ou seja, se o menor MSE for numericamente maior que o menor MSE da camada anterior, também se interrompe o treinamento. Neste caso, a camada é removida e o melhor neurônio da camada anterior passa a ser a saída da rede. Os outros neurônios da camada anterior são eliminados.
8. **Podar os neurônios que não contribuem para a saída:** Todos os neurônios de todas as camadas anteriores que não contribuem para a saída da rede são removidos, como mostram as Figuras 2.3(5) e 2.3(6).

2.4 O ALGORITMO DE TREINAMENTO COMBI-GMDH

Encontram-se na literatura relatos sobre diversas variações do algoritmo GMDH original. Isso se deve ao fato de que os sistemas a serem modelados diferem em suas características, demandando diferentes metodologias de exploração para que sejam aproximados satisfatoriamente. Pesquisadores sugerem a experimentação com vários algoritmos de treinamento para que se possa determinar o que melhor descreve o objeto em estudo [18].

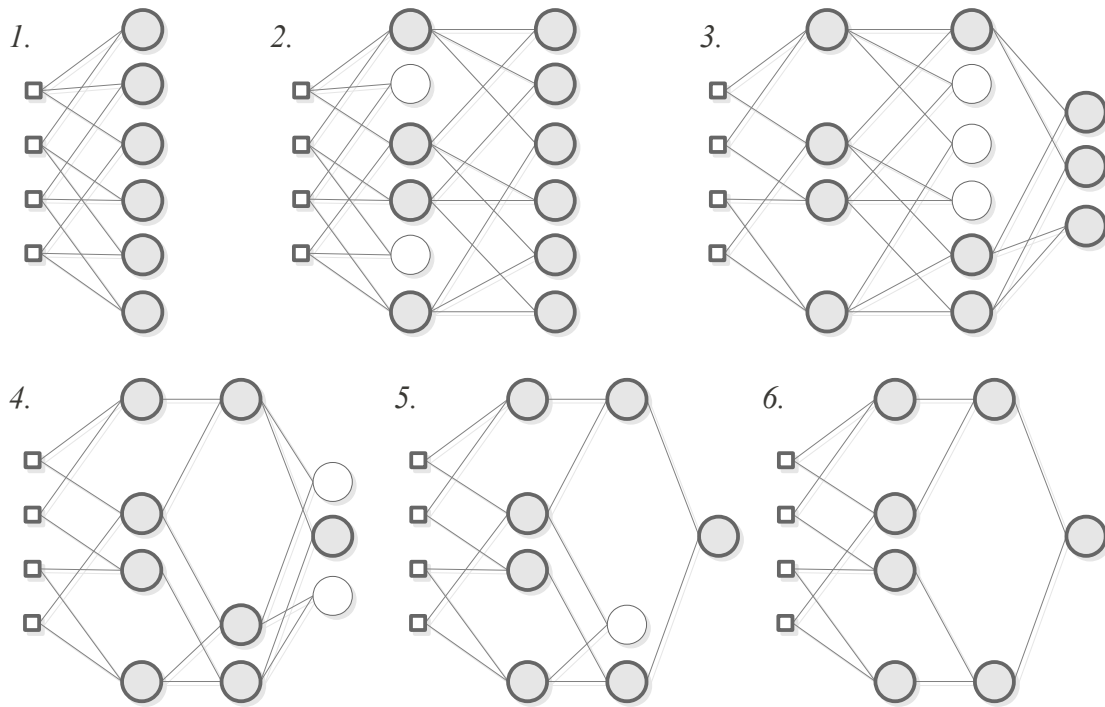


Figura 2.3: Exemplo gráfico do processo e treinamento de uma rede GMDH

Uma das variações importantes do algoritmo é o método *combinacional*. No restante deste trabalho, a expressão O-GMDH será usada para referenciar a versão original do algoritmo e a expressão C-GMDH será usada para referenciar o método combinacional. A expressão GMDH continuará a ser usada para referenciar o método de uma forma geral.

O algoritmo C-GMDH segue a estrutura básica do algoritmo original, ou seja, camadas são criadas e treinadas e seus neurônios selecionados de acordo com o critério escolhido. As regras que determinam a interrupção do processo de treinamento também se aplicam. As diferenças fundamentais estão nas formas das descrições parciais, assim como nas regras para criação e conexão dos neurônios.

As redes C-GMDH como propostas por Ivakhnenko [18] utilizam polinômios de primeira ordem. O número de entradas dos neurônios é igual ao número da camada a que pertencem. Com isso, quanto maior for o número da camada, maior será a quantidade de entradas que seus neurônios receberão. O aumento na quantidade de entradas causa um incremento no número de termos das descrições parciais contidas nos neurônios. Em outras palavras, as descrições parciais se tornam mais complexas à medida que novas camadas são incluídas. Os polinômios C-GMDH seguem a forma de Kolmogorov-Gabor (Eq. 2.1). Por exemplo, um neurônio encontrado na camada três de uma rede C-GMDH, tem a forma da Eq. 2.19.

$$y = a_0 + a_1x_i + a_2x_j + a_3x_k + a_4x_ix_j + a_5x_ix_k + a_6x_jx_k + a_7x_ix_jx_k \quad (2.19)$$

No algoritmo O-GMDH, a quantidade de neurônios em cada camada corresponde à combinação das saídas da camada anterior, tomadas em grupos de duas. No C-GMDH, o número de neurônios da camada é igual ao número de combinações de n entradas (fornecidas pelos neurônios da camada anterior) tomados em grupos de l entradas, onde l é igual ao índice da nova camada (ver Eq. 2.20).

$$C_l^n = \frac{n!}{l!(n-l)!} \quad (2.20)$$

Outra característica do método C-GMDH é a possibilidade de se conectarem nós de entrada ou neurônios de qualquer camada a neurônios pertencentes a camadas não adjacentes. Por exemplo, na Fig. 2.4, a *entrada 1* está conectada ao *neurônio 1* da primeira camada oculta, mas também está conectada ao *neurônio 3* da segunda camada oculta e ao neurônio de saída.

A Fig. 2.4 mostra uma possível topologia de uma rede C-GMDH após o treinamento. Neste trabalho, os nós triangulares em cores escuras sob os neurônios são chamados de *expedidores*. Eles são criados durante o treinamento da rede e somente são excluídos, por seleção, ao final do treinamento, quando todos os nós que não contribuem para a saída da rede são removidos.

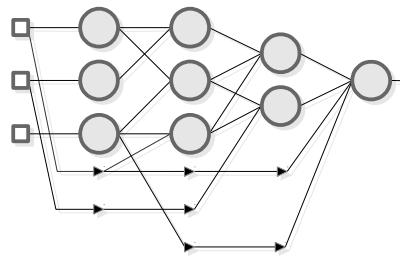


Figura 2.4: Possível topologia de uma rede C-GMDH treinada.

2.5 APROXIMADORES UNIVERSAIS

Cybenko demonstrou que uma combinação linear finita de funções limitadas por uma função discriminante pode aproximar qualquer função contínua de variáveis reais com

suporte em um hipercubo unitário [141]. Em outras palavras, qualquer função que tenha suas saídas em um subconjunto fechado da imagem com intervalo $]0, 1]$ pode ser representada pelo somatório de funções discriminantes. Cibenko também mostra que as funções de ativação sigmóides, utilizadas comumente em redes neurais artificiais, são discriminantes. Levando em consideração que essas funções têm suporte no intervalo $]0, 1[$, o autor demonstra que o somatório das saídas de um número finito de neurônios com função de ativação sigmoide pode representar qualquer função contínua com imagem no intervalo $]0, 1]$ com uma precisão arbitrária, desde que se não apliquem restrições ao número de neurônios e às dimensões dos pesos. Na prática, isso pode ser feito através de uma rede neural com uma camada oculta de neurônios com função de ativação sigmoide e uma camada de saída de neurônios com funções de ativação lineares [141].

As conclusões de Cybenko não podem ser aplicadas a redes GMDH, pois seus neurônios não apresentam funções com suporte no hipercubo unitário. Além disso, o treinamento de redes GMDH produz redes que realizam combinações não-lineares dos neurônios. Dessa forma, segundo o critério apresentado, não se pode afirmar que redes GMDH sejam aproximadores universais. Por outro lado, a grande quantidade de aplicações de sucesso baseadas no modelo, em diversas áreas do conhecimento, justificam o constante aprofundamento de sua compreensão.

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentadas as informações gerais sobre o treinamento de redes neurais artificiais do tipo GMDH ou, em inglês, *GMDH-T ANNs (GMDH-Type Artificial Neural Networks)*. Essas redes são treinadas utilizando-se o Método de Grupo para Manipulação de Dados (*Group Method of Data Handling - GMDH*), criado pelo Professor A. G. Ivakhnenko no final da década de 1960.

O método foi desenvolvido para realizar o incremento iterativo dos graus e dos números de termos de polinômios, com o intuito de aproximar-se sistemas não lineares complexos. Sua principal característica consiste na criação e avaliação de termos adicionais dos polinômios, seguindo-se por passos de exclusão dos termos que menos contribuem para a aproximação. O polinômio é tratado como uma rede de polinômios menores, ou descrições parciais, organizadas em camadas, de forma semelhante às redes do tipo perceptron de múltiplas camadas. O processo de treinamento consiste em incluir camadas

de neurônios (descrições parciais) que são todos treinados e selecionados, seguindo-se por novas inclusões de camadas, procedimento esse que se repete até que as condições de parada do algoritmo são alcançadas.

Existem diversas variações do modelo. Neste trabalho foram utilizados os métodos original, denominado aqui de O-GMDH, e o método combinacional, referido neste texto como C-GMDH. O método original utiliza sempre polinômios de grau dois com duas variáveis independentes. Os neurônios são organizados tomando-se as entradas duas a duas, na criação da camada, mas alguns deles são excluídos após o treinamento, num processo chamado seleção.

O método combinacional utiliza polinômios com uma ou mais variáveis independentes, de acordo com o número da camada a que pertençam. O método C-GMDH consome mais recursos computacionais do que o O-GMDH. Por outro lado, ele expande a busca exaustiva sobre o espaço de solução [18].

As duas versões do método apresentadas neste capítulo, O-GMDH e C-GMDH, foram utilizadas em diversos experimentos, apresentados nos próximos capítulos deste trabalho. Além disso, diferentes inovações foram feitas no método, visando resolver questões de desempenho, tempo de processamento, melhoria das aproximações e solução de questões de instabilidades. Essas inovações são apresentadas no capítulo 3.

3 TÓPICOS SOBRE O DESEMPENHO COMPUTACIONAL DOS ALGORITMOS E INOVAÇÕES PROPOSTAS NO MÉTODO DE TREINAMENTO

3.1 CONSIDERAÇÕES INICIAIS

No capítulo 2 foi detalhada a estrutura e os métodos de treinamento das redes GMDH. Destaque-se que dois modelos de redes foram desenvolvidos: (a) o original, chamado aqui de O-GMDH, e (b) o combinacional, denominado neste trabalho de C-GMDH. O treinamento de ambos envolve a criação de camadas, o cálculo dos coeficientes e a seleção dos neurônios, ressaltando-se que as redes C-GMDH apresentam mais descrições parciais por camada antes da fase de seleção.

Neste capítulo são discutidos aspectos pontuais do método GMDH detectados durante a realização deste trabalho. Como consequência, foram introduzidas diversas configurações ou parametrizações, não apresentadas na literatura, e desenvolvidas a partir da experiência adquirida com o treinamento e a execução de várias redes, nos diversos experimentos realizados neste trabalho. Além das inovações em termos do algoritmo em si, encontram-se detalhes de implementação, relacionados ao consumo de recursos computacionais, que foram introduzidos buscando viabilizar a implementação prática das redes. Os ajustes apropriados desses parâmetros permitem que se alcance a convergência nos modelos e evitam o consumo excessivo de recursos computacionais, aumentando, assim, a rapidez da convergência. Todos os pontos apresentados neste capítulo foram avaliados e implementados na ferramenta *GMDH Box* desenvolvida neste projeto.

3.2 ADIÇÃO E REMOÇÃO DE NEURÔNIOS

No método O-GMDH, a quantidade de neurônios instanciados na criação da camada é determinada pela combinação das entradas da camada, tomadas 2 a 2 (Eq. 2.3). A Fig. 3.1 mostra que existe uma relação quadrática entre a quantidade de entradas da camada e a quantidade de neurônios criados durante o treinamento.

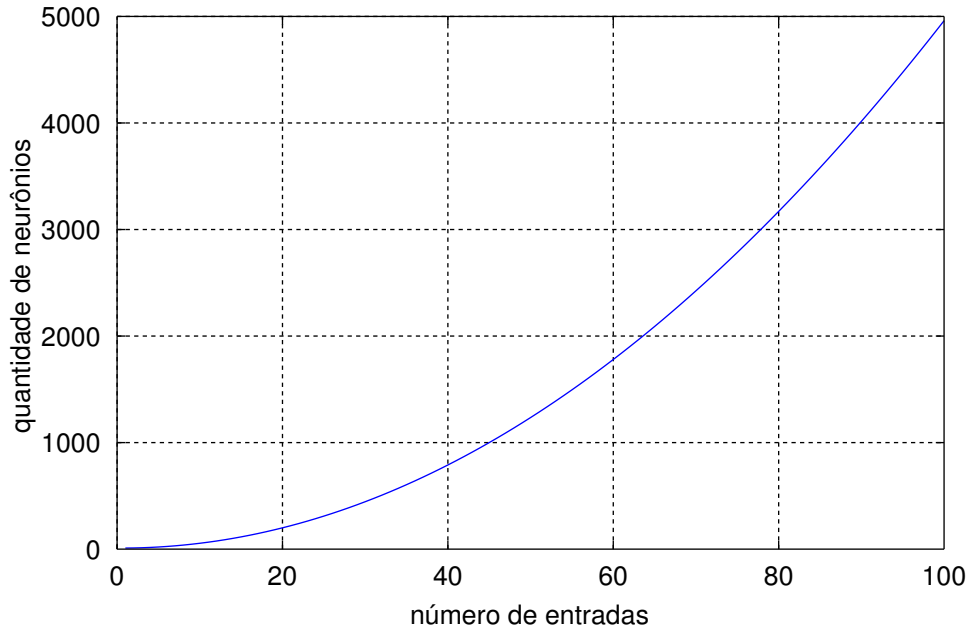


Figura 3.1: Quantidades de neurônios em uma rede O-GMDH

Entretanto, no método C-GMDH, a relação entre o número de entradas e a quantidade de neurônios varia de acordo com a camada. Como discutido na seção 2.4, o número de entradas dos neurônios é igual ao número da camada (l), fazendo com que a quantidade de neurônios criados seja igual à combinação do número de saídas da camada anterior (número de neurônios após a seleção) tomados em grupos de l elementos (veja Eq. 2.20).

A Fig. 3.2 contém um exemplo de rede C-GMDH em treinamento, configurada para não realizar a seleção de neurônios após o cálculo dos coeficientes das descrições parciais. Por questões de simplicidade, o exemplo não inclui o encaminhamento de nós, o que reduz a quantidade de elementos processadores em cada nova camada. Para maior entendimento do processo, é preciso reforçar a compreensão de que no C-GMDH a quantidade de entradas aceitas por cada neurônio varia entre as camadas: elementos da *camada 1* aceitam uma entrada, elementos da *camada 2* aceitam duas entradas e assim por diante.

Considere-se que o sistema a que a rede da Fig. 3.2 busca identificar aceite quatro entradas. Dessa forma, a primeira camada oculta (*camada 1*) criada durante o treinamento terá $C_1^4 = 4$ descrições parciais. Após o cálculo dos coeficientes de cada descrição, como nenhuma delas é removida, já que o processo de seleção não está habilitado, a *camada 2* receberá as 4 saídas da *camada 1* como entrada. Desta feita, com

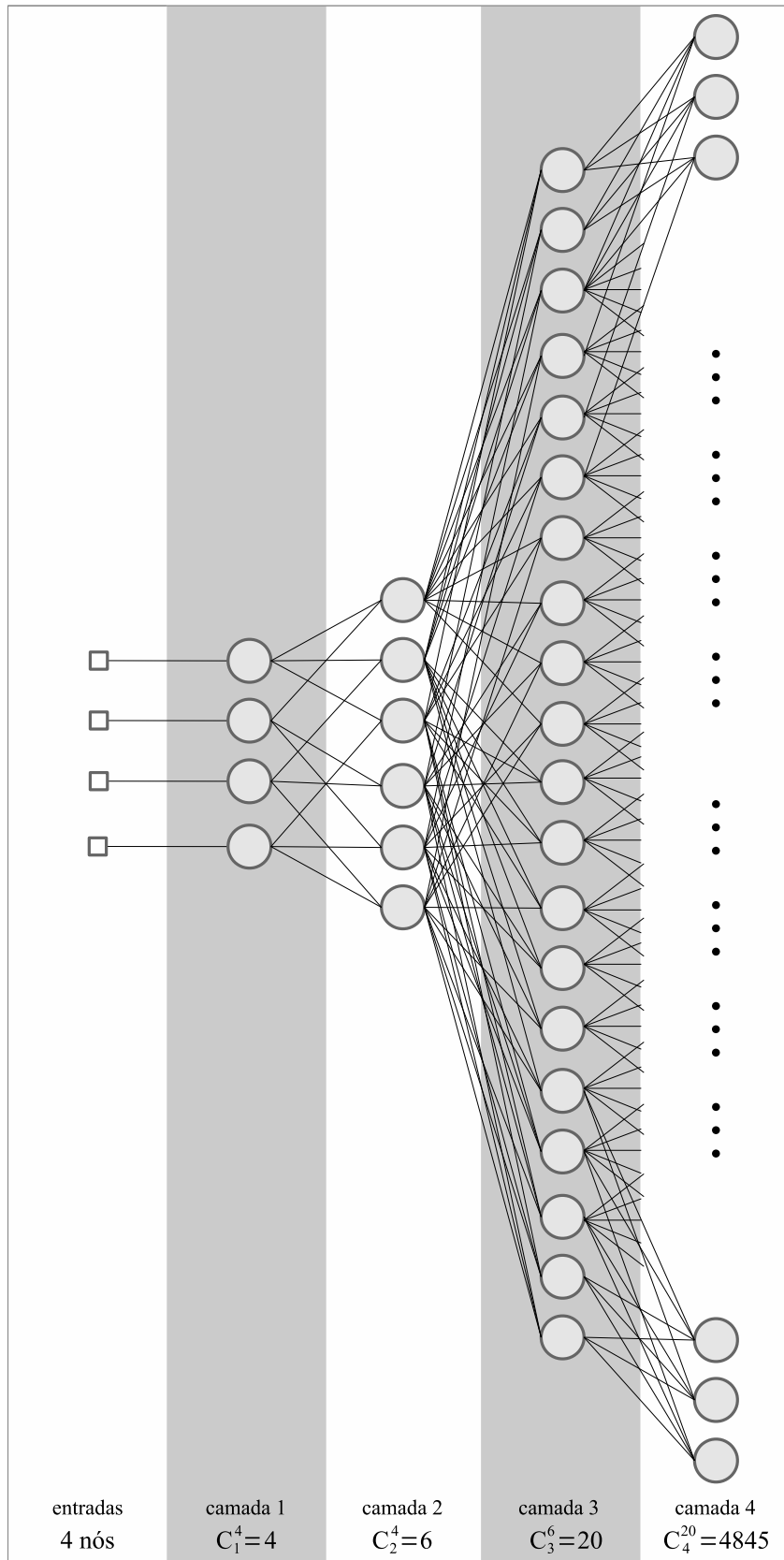


Figura 3.2: Exemplo de rede C-GMDH em treinamento, sem o encaminhamento de nós e sem a seleção de neurônios. Com 4 entradas, a primeira camada oculta possui $C_1^4 = 4$ elementos processadores, a segunda camada possui $C_2^4 = 6$ elementos, a terceira camada possui $C_3^6 = 20$ neurônios, a quarta possui $C_4^{20} = 4.845$ neurônios. Uma quinta camada possuiria $C_5^{4845} = 22.201.944.189.472.720$ neurônios.

neurônios que recebem 2 entradas, a quantidade de descrições criadas na *camada 2* é de $C_2^4 = 6$. A *camada 3*, com neurônios de 3 entradas e com 6 entradas disponíveis, vindas da camada anterior, a quantidade de descrições é igual a $C_3^6 = 20$. Seguindo o processo, a *camada 4* terá $C_4^{20} = 4845$ neurônios. Se o treinamento continuasse, a *camada 5* deveria conter $C_5^{4845} = 22.201.944.189.472.720$ neurônios, o que provavelmente não é computável. Daí, percebe-se a importância do processo de seleção.

Percebe-se, pela Fig. 3.3, que existe uma relação binomial entre a quantidade de entradas dos neurônios, determinada pelo número da camada a que pertençam, e a quantidade de entradas disponíveis para a camada, ou seja, a quantidade de valores fornecidos pela camada anterior após a seleção. Por exemplo, os neurônios da *camada 10* em uma rede C-GMDH possuem dez entradas; se a *camada 9*, previamente treinada e selecionada, contiver 17 neurônios, a *camada 10* terá $C_{10}^{17} = 19.448$ neurônios (ponto A na Fig. 3.3).

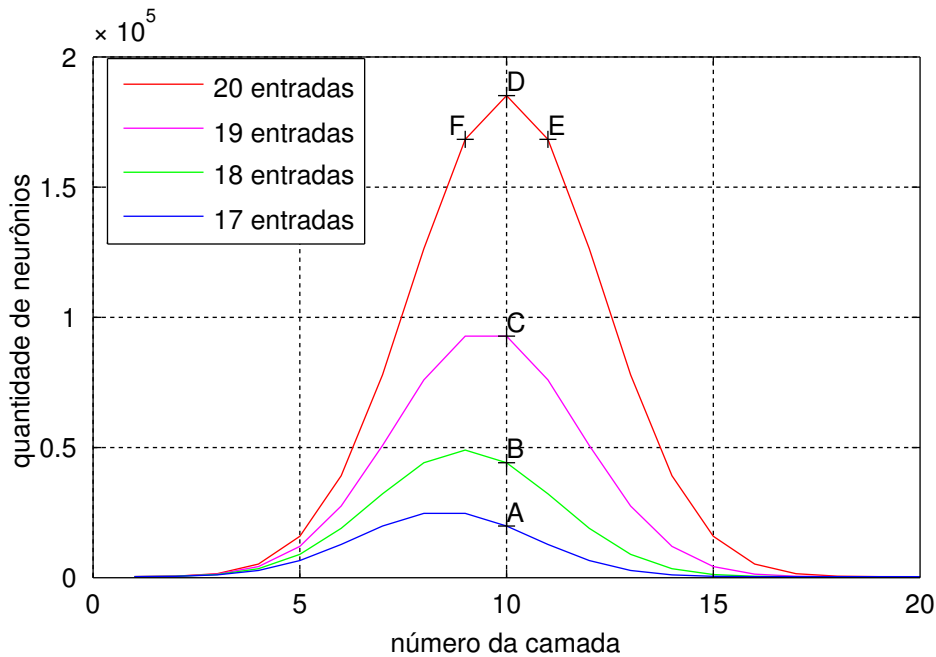


Figura 3.3: Quantidades de neurônios em uma rede C-GMDH. O eixo x representa o número da camada e o eixo y a quantidade de neurônios criados. A quantidade de entradas de cada neurônio em uma camada C-GMDH é igual ao número da camada. Cada curva mostra a quantidade de neurônios criada nas camadas para uma quantidade de entradas fixa. Por exemplo, a curva em vermelho representa uma quantidade fixa de 20 entradas. Em cada camada, essa quantidade fixa de entradas gera uma quantidade diferente de neurônios; na *camada 9* (ponto F do gráfico), 20 entradas causam a criação de 167.960 descrições; na *camada 10* (ponto D), 20 entradas causam a criação de 184.756 descrições; na *camada 11* (ponto E), as 20 entradas determinam a criação de 167.960 descrições, um valor igual ao da *camada 9*, devido às propriedades da função combinacional.

Se a *camada 9* tiver 18 neurônios, a *camada 10* terá $C_{10}^{18} = 43.758$ neurônios (Fig. 3.3,

B). Com 19 neurônios na *camada 9*, a *camada 10* terá $C_{10}^{19} = 92.378$ descrições parciais (Fig. 3.3, C); e com 20 neurônios, ou seja, 20 entradas vindas da *camada 9*, a *camada 10*, quando criada, terá $C_{10}^{20} = 184.756$ neurônios (Fig. 3.3, D).

Por outro lado, se a *camada 10*, após o treinamento e a seleção, tiver 20 neurônios, a *camada 11* terá $C_{11}^{20} = 167.960$ (Fig. 3.3, E), que é igual à quantidade de neurônios na *camada 9* caso esta tenha 20 entradas $C_9^{20} = 167.960$ (Fig. 3.3, F).

A grande quantidade de descrições parciais criadas durante o treinamento é importante para que se aumente a busca de resultados no espaço de soluções, mas uma vez determinadas as que melhor representam a solução final, as outras podem ser excluídas no processo de seleção. Contudo, com fins investigativos, foi acrescentada no *GMDH Box* a possibilidade de que os neurônios não sejam selecionados após o treinamento da camada. Dessa forma, a rede contará com mais descrições parciais durante o treinamento, o que aumenta a busca no espaço de soluções do sistema. A importância dessa característica reside na possibilidade de avaliar se o processo de seleção está comprometendo a capacidade de aproximação da rede, ou seja, se as descrições parciais escolhidas para a remoção na verdade contribuiriam positivamente para a diminuição dos erros.

Por outro lado, a não seleção de neurônios aumenta o consumo de recursos computacionais e, conseqüentemente, o tempo de treinamento. Isso acontece porque a quantidade de neurônios criados depende da quantidade de entradas disponíveis para a nova camada, vindas da camada anterior. Sem o recurso de exclusão de neurônios após a seleção, o tamanho de cada nova camada será maior do que seria caso houvesse seleção. Na prática, durante o desenvolvimento e aperfeiçoamento da ferramenta *GMDH Box*, os experimentos com O-GMDH sem o processo de seleção não produziam melhores resultados do que os que aplicavam o processo. Com C-GMDH, como visto no exemplo da Fig. 3.2, os experimentos em geral não chegaram ao término por esgotamento de memória, quando o processo de seleção foi inibido.

Diferentemente de outros tipos de RNAs, como as redes Perceptron de múltiplas camadas, a topologia das redes GMDH é definida dinamicamente durante o seu treinamento. A quantidade de neurônios que serão criados em uma nova camada só pode ser determinada quando a camada que a precede tiver sido treinada e seus neurônios tiverem sido selecionados segundo o critério escolhido. Por outro lado, é possível inserir um bom controle sobre tamanho da camada após a seleção, o que será discutido em detalhes neste trabalho.

Uma forma de se evitar grandes quantidades de descrições em uma nova camada é limitar o número de entradas (z) nos neurônios no método C-GMDH, alterando a regra de que esse número deve ser igual o número da camada. Pode-se fornecer um parâmetro que determina a quantidade máxima (m) de entradas. O efeito dessa configuração é que os neurônios de camadas $l \geq m$ possuirão sempre uma quantidade de entradas igual a m , o que afeta o cálculo das combinações de entradas; neurônios de camadas $l < m$ continuarão a ter l entradas, como sintetizado na Eq. 3.1.

$$z = \begin{cases} l, & \text{se } l < m \\ m & \end{cases} \quad (3.1)$$

A Fig. 3.3 ressalta o fato de que a maior quantidade possível de descrições parciais (número de neurônios) numa nova camada de uma rede C-GMDH é alcançada quando o número de entradas de cada neurônio da camada m é igual à metade do número de entradas fornecidas pela camada anterior (camada $m - 1$). Isso ocorre por efeito do cálculo das combinações das entradas (ver Eq. 2.20). Neste caso, a forma mais efetiva de se reduzir o tamanho da rede (referente ao número total de neurônios) é limitar o número de entradas para uma camada m , provenientes da camada $m - 1$, sendo este valor inferior ao dobro do número da camada m , mas isso só tem sentido se for possível limitar o número de camadas da rede, como descrito na seção 3.5.

Independentemente da técnica de criação e de seleção de neurônios utilizada, o último passo do treinamento das redes GMDH consiste na remoção de todos os neurônios que não contribuem para a saída do modelo. A Fig. 2.3, nos seus passos 5 e 6, mostra um neurônio da camada 3 que, após a seleção realizada na última camada, não contribui para a saída do modelo e é excluído. Nesse caso, quanto maior a quantidade de neurônios criados durante o treinamento, maior poderá ser o número dos que, ao final, serão removidos por não contribuírem para a saída da rede. O *GMDH Box* permite a definição de diferentes critérios de seleção de neurônios, como pode ser visto na seção 3.3, dentre eles, é possível estabelecer um limite para a quantidade máxima de neurônios remanescentes na camada após a seleção.

3.3 SELEÇÃO DE NEURÔNIOS E INTRODUÇÃO DO CRITÉRIO DO LIMIAR DINÂMICO

A Seção 2.2.3 menciona a inclusão de um novo critério de seleção, o *limiar dinâmico*, que consiste no cálculo da mediana dos MSEs dos neurônios e a exclusão daqueles que apresentaram desempenho abaixo desse limiar, ou seja, que possuem um valor de MSE acima da mediana. Na literatura, encontram-se várias referências sobre a definição de um limiar apropriado, mas não há muita discussão sobre como se definir esse valor, provavelmente porque ele é muito dependente do problema com que se está lidando. Entretanto, a utilização da mediana como limiar de seleção apresentou resultados empíricos muito satisfatórios nos diversos experimentos realizados no decorrer do período utilizado para desenvolver este trabalho.

O MSE de cada neurônio é uma medida que valoriza a distância entre as saídas desejadas para aquele nó e as saídas obtidas. É preciso lembrar que esse valor é calculado após o treinamento do neurônio, ou seja, é um critério de avaliação de desempenho, sendo que quanto maior for o valor do MSE de uma descrição parcial, ou seja, quanto maiores tiverem sido os erros por ela gerados sobre todo o conjunto de seleção, menos adequada ela apresenta-se para representar o problema. Entretanto, o valor do MSE pode estar localizado em faixas numéricas bem diferentes entre uma camada e outra.

Um limiar fixo possivelmente não será adequado a todas as faixas de valores, fazendo com que poucos neurônios sejam excluídos, caso o limiar seja muito baixo, ou que neurônios em excesso sejam desconsiderados, caso seja muito alto. Neste contexto, o significado de “baixo” ou “alto” está muito ligado aos recursos computacionais disponíveis e à relação entre a acurácia necessária ao processo e a representatividade dos dados de treinamento/seleção sobre o problema em estudo. Não é incomum ocorrerem casos em que todos os neurônios apresentem MSE acima do limiar, o que causará a interrupção do treinamento ou, no caso da primeira camada, a falha total do processo. Por outro lado, se poucos neurônios são excluídos, ou pior, se nenhum neurônio for excluído, o treinamento pode consumir muitos recursos, demorando excessivamente para finalizar ou mesmo sendo interrompido por falta de memória do computador.

A utilização do *limiar dinâmico* leva em consideração o comportamento geral da camada, garantido que os nós que apresentaram desempenho excessivamente distante dos produzidos por seus pares sejam excluídos. Por outro lado, o uso da mediana para o cálculo deste limiar evita que os resultados de neurônios que sobreaproximaram o

sistema determinem a exclusão de neurônios que não se aproximaram tanto da solução mas que tenham resultados compatíveis com a maioria de seus pares. Isso poderia acontecer, por exemplo, se o limiar fosse determinado pela média dos MSEs, pois a média pode ser alterada sensivelmente apenas por um único valor de um grupo com poucos elementos, caso esse valor seja muito maior ou muito menor que os outros. Destaque-se o fato de que aqui a avaliação é feita sobre os valores dos desempenhos dos neurônios e não sobre as amostras de seleção, ou seja, a média ou a mediana são calculadas sobre os MSEs produzidos entre as descrições parciais e as saídas desejadas. A quantidade de valores, portanto, utilizados no cálculo do limiar (média, mediana ou qualquer outro) corresponde à quantidade de neurônios da camada recém treinada.

O critério de *limiar fixo* é inconveniente por exigir constante intervenção do desenvolvedor. Dessa forma, os critérios de *limiar dinâmico* e do *número de neurônios remanescentes* são os que efetivamente foram aplicados, sendo o primeiro normalmente usado no método O-GMDH e o segundo, no C-GMDH, embora seja possível escolher qualquer um dos critérios em ambos os métodos.

Para reduzir a quantidade de descrições, é possível introduzir um **limite** no número de neurônios remanescentes também no método de *limiar dinâmico*. Isso significa que, após a seleção, se ainda existir uma quantidade de neurônios superior ao limite, os remanescentes são ordenados de acordo com o seus MSEs e são excluídos, a partir do que tiver o maior valor, até que se reduza à quantidade limite permitida.

Na prática, incluir um **limite** (número máximo de neurônios remanescentes) no método do *limiar dinâmico* faz com que a seleção aconteça por *limiar dinâmico* em algumas camadas e por *número de neurônios remanescentes* em outras camadas. Mais precisamente, nas camadas onde a quantidade de neurônios restantes após a seleção for inferior ao limite introduzido, nada mais é feito, ou seja, vale a regra básica do *limiar dinâmico*. Nas camadas em que a quantidade de neurônios, após a seleção por *limiar dinâmico*, for maior do que o limite, os neurônios excedentes são removidos, o que equivale à regra de seleção por *número de neurônios remanescentes*.

É sempre útil utilizar um limite para evitar um grande consumo de recursos computacionais. Para o método O-GMDH, estabeleceu-se, empiricamente, um limite de 30 neurônios por camada, após a seleção. A literatura sugere um limite de 5 neurônios após a seleção da camada quando se usa o algoritmo C-GMDH [18]. Vale salientar que o método C-GMDH utiliza a técnica de *encaminhamento* de nós, o que implica na in-

clusão dos nós *expedidores* depois da seleção, o que aumenta a quantidade de entradas da camada seguinte (ver seção 2.4 e Fig. 2.4).

3.4 POLINÔMIOS DE ORDEM MAIOR DO QUE 1 NO C-GMDH

Na seção 2.4 foi apresentado o método C-GMDH, no qual os neurônios possuem diferentes quantidades de entradas, de acordo com a camada a que pertençam, no entanto, seus polinômios são sempre de grau 1; a não linearidade dos modelos é alcançada pela multiplicação entre si das variáveis das descrições parciais e pela inclusão de novas camadas. Com o intuito de aumentar a busca no espaço de soluções realizado por cada camada, este trabalho introduziu uma variação no método C-GMDH, possibilitando a troca dos polinômios de primeira ordem por polinômios de ordem mais elevada.

Com essa modificação, os neurônios terão grau equivalente ao número de sua camada. Por exemplo, neurônios da *primeira camada* possuem dois termos, como visto na Eq. 3.2, onde o primeiro termo é um *bias*. A partir da *segunda camada*, devido ao incremento no número de entradas dos neurônios, suas expressões apresentarão termos adicionais. Por exemplo, neurônios na *camada 2* usam polinômios definidos pela Eq. 3.3. Um neurônio da *camada 3*, contendo três entradas, terá sua descrição parcial definida pelo polinômio expresso na Eq. 3.4.

$$y = a_0 + a_1x_1 \tag{3.2}$$

$$y = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2 + a_4x_2^2 + a_5x_1x_2 \tag{3.3}$$

$$y = a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + a_4x_2 + a_5x_2^2 + a_6x_2^3 + a_7x_3 + a_8x_3^2 + a_9x_3^3 + \tag{3.4}$$

$$a_{10}x_1x_2 + a_{11}x_1x_3 + a_{12}x_2x_3 + a_{13}x_1x_2x_3$$

Com o aumento do grau dos polinômios no C-GMDH, as descrições se tornam mais complexas devido à maior quantidade de termos e à inclusão dos cálculos de potência. Essa complexidade exige mais recursos computacionais para o processamento de cada descrição, principalmente memória, pois as entradas, quando pré-processadas, produzirão matrizes maiores. Para limitar esse efeito, o *GMDH Box* inclui a capacidade restringir o grau dos polinômios a um valor máximo.

Independente do grau máximo do polinômio, quanto mais entradas tiverem os neurônios C-GMDH, mais termos terão suas descrições. O *GMDH Box* também permite a configuração de um valor máximo para o número de entradas, como visto na seção 3.2, Eq. 3.1.

3.5 CRITÉRIOS DE PARADA

A regra geral do treinamentos das redes do tipo GMDH determina que a adição de novas camadas deva continuar até que o valor de desempenho comece a divergir ou que não seja possível mais adicionar camadas. A divergência significa que uma nova camada piorou o resultado da rede e por isso deve ser excluída e o neurônio com menor MSE na camada anterior fornecerá a saída da rede. A impossibilidade de se adicionar camadas está relacionada ao número de neurônios remanescentes, que devem ser em quantidade mínima igual ao número de entradas dos neurônios da camada posterior. No algoritmo O-GMDH, este número é igual a dois e no C-GMDH ele é igual ao número da camada (ver seções 2.3 e 2.4).

Muitas vezes as redes desenvolvem uma grande quantidade de camadas durante o treinamento. No O-GMDH, esse crescimento descontrolado não se apresentou muitas vezes nos experimentos realizados por causa da menor quantidade de neurônios por camada criados durante treinamento e, especialmente, a quantidade remanescente após a seleção. A adição de muitas camadas em redes C-GMDH é mais frequente porque o número de nós nas camadas costuma ser maior do que no modelo O-GMDH, especialmente quando o encaminhamento de saídas está habilitado. Por outro lado, as camadas C-GMDH com índices mais elevados possuem uma quantidade muito grande neurônios, quase sempre inviabilizando o treinamento. Para permitir o treinamento, a solução é limitar o número de camadas nas redes C-GMDH.

Digno de nota é o fato de que, nos experimentos com O-GMDH, as redes com muitas camadas não forneceram resultados muito mais precisos do que redes menores. Na maioria dos casos em que redes com muitas camadas foram desenvolvidas durante o treinamento, cada nova camada melhorava o desempenho de uma forma muito suave em comparação com outros casos, onde a convergência costuma ser mais acentuada.

Para dirimir esse problema, os seguintes critérios de parada foram introduzidos no algoritmo:

1. **Número máximo de camadas:**

A forma mais simples de se interromper o treinamento é limitar a quantidade de camadas. Quando o número é alcançado, a camada é treinada e o seu melhor neurônio se torna a saída da rede. Os outros neurônios da camada são excluídos, bem como os neurônios das camadas anteriores que não contribuam para a saída da rede.

2. **Gradiente mínimo:**

Este critério, quando ativo, determina que uma nova camada, após a seleção, tenha introduzido um mínimo de melhoria no resultado, ou seja, tenha diminuído pelo menos em uma determinada quantidade o erro produzido entre a saída da rede e o alvo de treinamento. Se essa melhoria for inferior ao mínimo, o treinamento é interrompido, mantendo-se, contudo, a camada recém treinada.

3. **Desempenho mínimo:**

É o desempenho desejado pelo desenvolvedor. Se o erro produzido pela rede for igual ou inferior a esse valor, o treinamento se interrompe.

3.6 OTIMIZAÇÃO *LS* E SOLUÇÃO DE SISTEMAS MAL CONDICIONADOS

Como visto na seção 2.2.2, o cálculo dos coeficientes dos polinômios é realizado resolvendo-se um sistema de equações lineares. A teoria dos mínimos quadrados sugere a inversão de uma matriz para resolver o sistema, como explicitado na Eq. 2.16, entretanto, pode-se utilizar qualquer método para esse fim. É preciso considerar, contudo, que não se conhece de antemão as características dos sistemas em que as redes serão utilizadas ou, ainda mais, as características das matrizes que serão criadas. Para resolver sistemas com matrizes das quais não se conhece as características, métodos baseados na eliminação gaussiana são considerados adequados, como por exemplo, a decomposição LU [60].

Os experimentos demonstraram que, em alguns casos, existem pequenas diferenças nos valores dos coeficientes calculados com o uso da inversão das matrizes em relação a outros métodos. Entretanto, essas pequenas diferenças fizeram-se insignificantes nas análises estatísticas das respostas. Optou-se, dessa forma, por evitar-se a inversão das matrizes devido ao custo computacional dessas operações.

No *GMDH Box*, utiliza-se duas abordagens para o cálculo dos coeficientes. Numa delas, deixa-se a cargo do ambiente, MATLAB ou GNU Octave, a escolha do método. O Octave, por exemplo, usa as seguintes técnicas para a solução de sistemas lineares [61].

1. Se a matriz for esparsa e triangular, superior ou inferior, resolve-se o sistema por eliminação gaussiana e passa-se para o passo 4.
2. Se a matriz for quadrada e Hermitiana com uma diagonal positiva, tenta-se realizar uma fatoração de Cholesky.
3. Se a fatoração de Cholesky falhar ou se a matriz não for Hermitiana com uma diagonal real positiva, mas ainda assim for quadrada, realiza-se uma fatoração LU.
4. Se a matriz não for quadrada ou se qualquer uma das soluções acima identificar uma matriz singular ou próxima de uma singular, encontra-se a solução por mínimos quadrados (LS).

Para evitar-se problemas com matrizes singulares, a outra abordagem utilizada no *GMDH Box* é a utilização da decomposição das matrizes em valores singulares (SVD - *Single Value Decomposition*) [60]. Em não raros casos, as matrizes geradas durante o treinamento são singulares ou numericamente muito próximas de matrizes singulares, o que gera instabilidades no processo. Essas matrizes geram sistemas mal-condicionados, onde pequenas mudanças nos coeficientes geram grandes alterações nos resultados [62]. Para esses casos, métodos como a decomposição LU não são adequados, sendo vantajoso utilizar o SVD.

Durante um período do desenvolvimento do *GMDH Box* foram encontradas sérias dificuldades nos treinamentos porque os processos utilizados pelos ambientes matemáticos para a solução dos sistemas de equações não incluíam automaticamente o SVD. Para resolver esse problema, incluiu-se uma função de treinamento de camadas que explicitamente realiza a decomposição e obtém os coeficientes a partir das matrizes decompostas. Com essa função de treinamento, uma grande quantidade dos problemas de instabilidade foi resolvida, sem prejuízo algum aos resultados positivos já obtidos.

Para realizar a solução de sistemas de equações lineares, incluindo as decomposições necessárias, o GNU Octave e o MATLAB utilizam o pacote de rotinas LAPACK,

fornecido pelas instituições Universidade do Tennessee, Universidade da Califórnia – Berkeley, Universidade do Colorado – Denver e pela organização sem fins lucrativos NAG – *Numerical Algorithms Group* [61, 63, 64].

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo representa o resultado da experiência adquirida durante o desenvolvimento da ferramenta *GMDH Box* e durante a realização de diversos experimentos com o método. Muitos dos pontos apresentados se relacionam à convergência dos treinamentos influenciada por aspectos computacionais. A introdução de limites para a criação de neurônios e mesmo para sua exclusão têm grande influência no consumo de recursos das máquinas, bem como na convergência. Dentre as funcionalidades descritas, pode-se destacar a introdução do **critério de parada** e do **critério de seleção dos neurônios**.

O **critério de parada** evita o superdimensionamento das redes. Na prática, se uma rede se tornar muito complexa, ela não será útil e, por outro lado, se já tiver alcançado o desempenho desejado pelo desenvolvedor, não há necessidade de se continuar o ajuste.

Já o **critério de seleção** dos neurônios é com certeza uma das mais importantes contribuições realizadas. Os relatos apresentados na literatura consultada não dão informações suficientes sobre os critérios de seleção aplicados. Mais ainda, na única implementação do algoritmo que se conseguiu utilizar, a apresentada por Pham e Liu [45], o programa depende da intervenção do operador para realizar a seleção. Neste trabalho foram introduzidos diferentes métodos para a escolha automática dos melhores neurônios.

Além da criação da ferramenta, as possibilidades de parametrização incluídas no método, permitiram a realização de experimentos com características bem distintas. Na literatura, o método quase sempre é apresentado em uma visão teórica ou com poucos exemplos. Artigos que introduzem alguma modificação no método original, não apresentam extensas avaliações em diferentes tipos de problemas, limitando-se a demonstrar resultados sobre uma aplicação específica.

O que se buscou neste trabalho foi dar uma base concisa para a utilização do método,

o que foi alcançado, como pode ser visto nos resultados dos experimentos apresentados nos próximos capítulos.

4 MODELAGEM DE SISTEMAS DINÂMICOS

4.1 CONSIDERAÇÕES INICIAIS

A identificação e a predição de sistemas não lineares é uma tarefa complexa. Existem diversos relatos na literatura sobre a utilização de diferentes modelos de redes neurais artificiais para realizar a aproximação dessa classe de sistemas. Este capítulo apresenta um estudo sobre a utilização das redes GMDH para esse fim, comparando-as com redes Perceptron de múltiplas camadas. Neste caso, foram analisados cinco exemplos de sistemas não lineares, propostos por Narendra e Parthasarathy [65], que simulam o funcionamento de plantas industriais não lineares. Cada planta é descrita em termos de sua modelagem e das entradas utilizadas para treinar e para validar as redes neurais.

Diferentes esquemas de apresentação dos dados de entrada podem ser utilizados para treinar as redes neurais, de acordo com as características dos sistemas a serem identificados. Neste estudo, três tipos de entrada foram utilizados: (a) *entradas senoidais*, (b) *entradas estocásticas* e (c) *entradas senoidais estocásticas*. No que se refere à utilização das redes neurais já treinadas, a identificação pode ser *paralela* ou *série-paralela*. Duas tarefas básicas foram realizadas: *modelagem* e *predição* dos sistemas, para as quais os três tipos de entradas foram avaliadas. Nas tarefas de predição, aplicou-se a identificação *série-paralela* e nas tarefas de modelagem, a identificação *paralela*. Os detalhes sobre os tipos de entradas, sobre as formas de identificação e sobre as tarefas realizadas são apresentados nas próximas seções, seguidos das especificações dos sistemas e dos resultados dos experimentos.

4.2 TIPOS DE ENTRADAS

Três diferentes configurações de entradas foram utilizadas para realizar o treinamento das redes: (a) *entradas senoidais*, (b) *entradas estocásticas* e (c) *entradas senoidais estocásticas*. Para a validação dos dados, foi criado um quarto grupo de entradas: *entradas de validação*. A Tabela 4.1 descreve cada um dos conjuntos. Detalhes sobre a geração desses dados são fornecidos nas seções que descrevem os sistemas.

Tabela 4.1: Tipos de entrada das redes utilizados para treinamento e para verificação dos resultados.

Tipo	Descrição
Entradas Senoidais	Entradas senoidais semelhantes às que seriam utilizadas durante o funcionamento das plantas.
Entradas Estocásticas	Valores aleatórios pertencentes ao intervalo entre os valores mínimo e máximo das entradas senoidais.
Entradas Senoidais Estocásticas	Sinal senoidal produzido a partir de um sinal temporal aleatório.
Entradas de Validação	Utilizadas para avaliar o desempenho das redes neurais. Contam com números mais elevados de dados do que as entradas de treinamento. Úteis para verificar as capacidades de interpolação e extrapolação das redes treinadas.

4.3 FORMAS DE IDENTIFICAÇÃO

Narendra e Parthasarathy propõem duas formas de identificação de sistemas de entradas e saídas únicas (SISO - *Single-Input Single-Output*), a saber: a *série-paralela* e a *paralela*. A diferença entre elas está na forma como suas saídas são realimentadas nos sistemas e nos modelos dos sistemas. Na forma *série-paralela*, as saídas das plantas produzidas nos tempos $t \leq k$, onde k é o tempo atual, ou seja, as saídas passadas e a saída presente, são utilizadas para realimentar as plantas e seus modelos para se obterem as saídas no instante de tempo seguinte, como visto na Eq. 4.1. Na forma *paralela*, o modelo do sistema é realimentado com suas próprias saídas, como apresentado na Eq. 4.2. Em ambas as equações 4.1 e 4.2, $u(\cdot)$ e $y(\cdot)$ representam a entrada e a saída do sistema real, respectivamente, $\hat{y}(\cdot)$ representa a saída do modelo do sistema, $\hat{\alpha}$ e $\hat{\beta}$ são os parâmetros ajustáveis do modelo, n é a quantidade de saídas passadas e presente utilizadas e m é a quantidade de entradas passadas e presente utilizadas.

$$\hat{y}(k+1) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k)y(k-i) + \sum_{j=0}^{m-1} \hat{\beta}_j(k)u(k-j) \quad (4.1)$$

$$\hat{y}(k+1) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k)\hat{y}(k-i) + \sum_{j=0}^{m-1} \hat{\beta}_j(k)u(k-j) \quad (4.2)$$

A Eq. 4.3 dá um exemplo de um sistema que utiliza a entrada atual (tempo k) e as

saídas atual e anterior (tempos k e $k - 1$) para fornecer uma saída no tempo $k + 1$. A Fig. 4.1 mostra essa planta – representada na imagem pelas letras “P” – ligada de duas formas diferentes a RNAs que tentam fazer a sua identificação – representadas pela letra “N”. Os blocos com rótulo z^{-1} representam atrasos unitários no sinal. Na prática significa uma memória que armazena a saída no instante de tempo anterior. A Fig. 4.1(a) mostra a planta da Eq. 4.3 numa identificação por rede neural em ligação do tipo *série-paralela* e a Fig. 4.1(b) exhibe essa mesma planta identificada por uma RNA ligada na forma *paralela*.

$$y(k + 1) = \alpha_0(k)y(k) + \alpha_1(k)y(k - 1) + \beta(k)u(k) \quad (4.3)$$

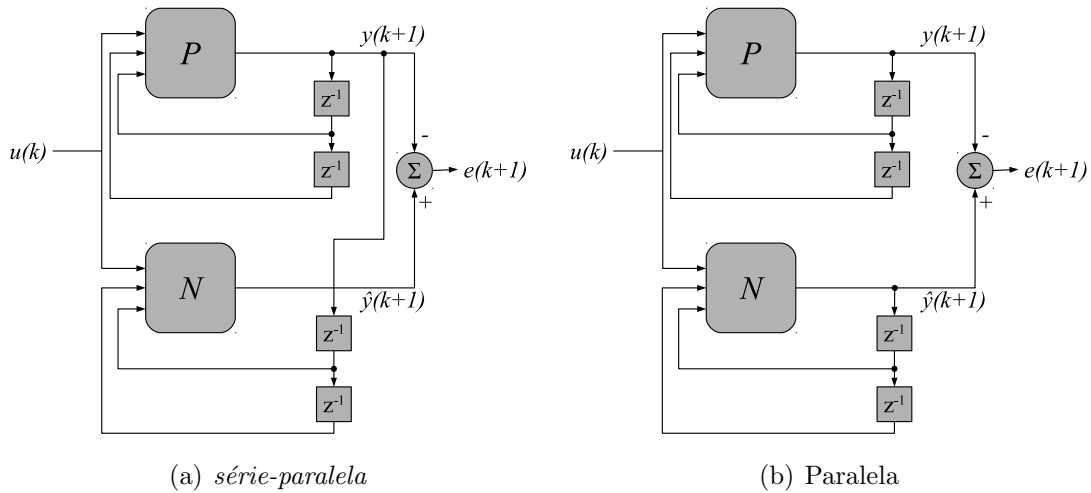


Figura 4.1: Comparação entre identificações *série-paralela* e *paralela*

[Comparação entre identificações (a) *série-paralela* e (b) *paralela*. “P” representa a planta a ser modelada e “N” o modelo da planta baseado em RNA. Os blocos com rótulo z^{-1} representam um atraso unitário no valor do sinal.]

As equações 4.4 e 4.5 mostram, respectivamente, as formas gerais das modelagens *série-paralela* e *paralela* por RNA, onde $N[\cdot]$ representa uma rede neural treinada e n e m são, respectivamente, as quantidades de saídas e entradas (passadas e presentes) utilizadas no modelo.

$$\hat{y}(k + 1) = N[y(k), \dots, y(k - n + 1), u(k), \dots, u(k - m + 1)] \quad (4.4)$$

$$\hat{y}(k + 1) = N[\hat{y}(k), \dots, \hat{y}(k - n + 1), u(k), \dots, u(k - m + 1)] \quad (4.5)$$

A identificação de sistemas utilizando redes neurais artificiais consiste no ajuste de seus parâmetros de acordo com uma função de otimização baseada nos erros entre os modelos e os sistemas reais [65]. Os algoritmos de treinamento *backpropagation* utilizados nas redes MLP costumam otimizar o erro em termos do *erro médio quadrado* [11]. O método dos mínimos quadrados utilizado para calcular os pesos dos neurônios GMDH também otimiza os valores dos coeficientes pelo erro médio quadrado [12]. Dessa forma, em todos os experimentos, a medida utilizada para verificar o desempenho dos neurônios durante o treinamento foi o MSE (ver Eq. 2.18) entre as saídas dos modelos (alvos de treinamento) e as saídas das plantas. Em tempo de treinamento, as redes são alimentadas com as entradas e as saídas das plantas. Somente durante a execução das identificações paralelas é que se aplica a realimentação das saídas da própria rede neural.

4.4 PREDIÇÃO DE SAÍDAS DOS SISTEMAS

Para realizar a predição de saídas das plantas, pode-se aplicar uma rede neural treinada, utilizando-se a realimentação de suas próprias saídas, como é feito na identificação *paralela*. Conhecendo a função do sinal, aplica-se uma sequência de entradas e colhem-se as saídas das redes, ao mesmo tempo em que essas saídas são realimentadas no modelo, mantendo-se esse procedimento pelo período necessário para que se obtenham as saídas futuras estimadas do sistema.

Neste trabalho, utilizou-se uma abordagem diferente da apresentada no parágrafo anterior para a predição de saídas futuras. Os valores aplicados às entradas das redes neurais foram atrasados em vinte unidades de tempo durante o treinamento. Durante a execução, espera-se que, utilizando-se as entradas e saídas atuais do sistema, a rede produza saídas equivalentes às saídas do sistema em 20 unidades de tempo no futuro. A Eq. 4.6 apresenta um exemplo do treinamento de uma rede dessa forma e a Fig. 4.2(a) mostra o diagrama representativo deste treinamento. A execução da rede treinada é apresentada na Eq. 4.7 e na Fig. 4.2(b). Os blocos com rótulo z^{-20} representam um atraso de 20 unidades de tempo que podem ser implementados, por exemplo, como memórias do tipo FIFO (*First in First Out* – Primeiro a Entrar, Primeiro a Sair) com 20 posições.

Este tipo de treinamento, proposto neste trabalho, é baseado no fato de que os ajustes dos pesos das redes neurais são realizados de acordo com uma função custo, que, por

sua vez, independe da relação entre as entradas e as saídas nos sistemas identificados, baseando-se apenas no erro existente entre a saídas dos sistemas e as saídas da redes neurais [65, 11, 12]. Deixou-se a cargo das tarefas experimentais a verificação se essa abordagem seria bem sucedida.

$$\hat{y}(k+1) = N[y(k-20), y(k-21), u(k-20)] \quad (4.6)$$

$$\hat{y}(k+20) = N[y(k), y(k-1), u(k)] \quad (4.7)$$

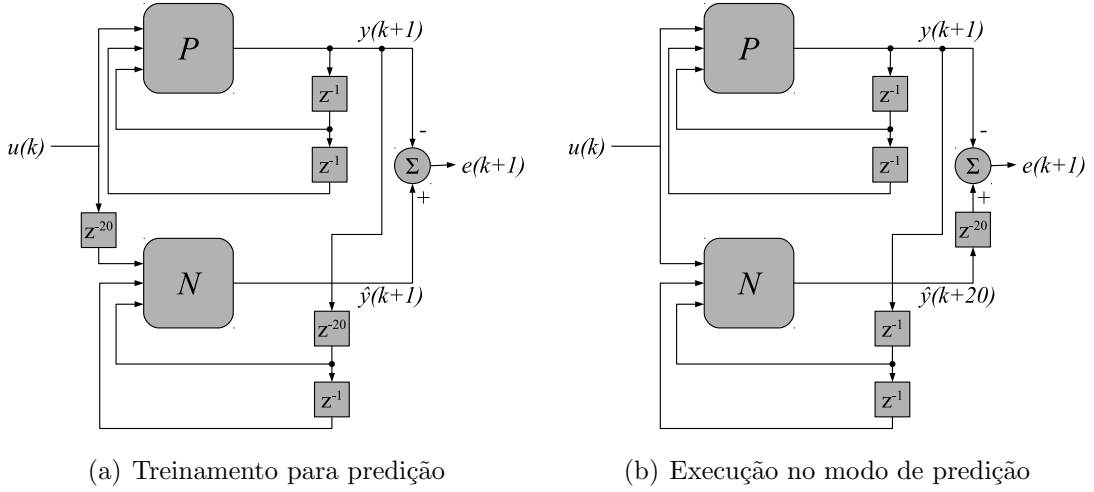


Figura 4.2: Treinamento e execução de redes neurais para a predição de valores do sistema modelado.

4.5 CONFIGURAÇÕES DOS EXPERIMENTOS

Cada rede neural foi treinada e avaliada em 9 diferentes tarefas, apresentadas na Tabela 4.2. As tarefas de 1 a 6 realizam trabalhos de predição de saídas futuras dos sistemas aplicando a identificação *série-paralela*, ora com atraso de uma unidade de tempo, ora com 20 unidades de tempo. As tarefas 7, 8 e 9 se destinam à modelagem dos sistemas, baseadas na identificação paralela.

Para cada configuração de tarefa (Tab. 4.2), foram avaliadas oito configurações diferentes de redes GMDH e oito de redes MLP. A Tabela 4.3 apresenta os parâmetros utilizados nos treinamentos. As diferentes configurações das redes GMDH permitidas pelo *GMDH Box* foram exploradas nestes experimentos, dando origem a redes dos tipos O-GMDH e C-GMDH, incluindo, contudo, novas variações nesses tipos. O

Tabela 4.2: Configurações das tarefas a serem desempenhadas pelas redes neurais.

	Tarefa	Identificação	Entrada	Atraso
1	Predição	Série-Paralela	Senoidal	1
2	Predição	Série-Paralela	Senoidal	20
3	Predição	Série-Paralela	Estocástica	1
4	Predição	Série-Paralela	Estocástica	20
5	Predição	Série-Paralela	Senoidal Estocástica	1
6	Predição	Série-Paralela	Senoidal Estocástica	20
7	Modelagem	Paralela	Senoidal	1
8	Modelagem	Paralela	Estocástica	1
9	Modelagem	Paralela	Senoidal Estocástica	1

desempenho de cada rede GMDH foi comparado com o desempenho de uma rede MLP nos experimentos. As configurações das redes são apresentadas abaixo:

- Nas *configurações A e B*, as redes GMDH são do tipo O-GMDH clássico, com um máximo de 5 e 30 neurônios por camada, respectivamente. As redes MLP comparadas têm uma camada oculta cada, com 5 e 10 neurônios, respectivamente.
- As *configurações C e D* acrescentam uma variação no O-GMDH, incluindo o encaminhamento de neurônios para camadas não adjacentes, mas, diferentemente das redes C-GMDH, os neurônios possuem sempre duas entradas e suas descrições parciais são de grau 2. As redes MLP correspondentes possuem duas camadas ocultas com 5 e 2 neurônios (*configuração C*) e 10 e 5 neurônios (*configuração D*).
- A *configuração E* determina a criação de redes C-GMDH sem encaminhamento, ou seja, as descrições parciais são de grau 1, os neurônios recebem quantidades de entradas diferentes em cada camada, mas não há conexão entre nós de uma camada com nós de outras camadas não adjacentes. A rede MLP nesta configuração apresenta uma camada oculta com 20 neurônios.
- Na *configuração F*, utiliza-se uma variação de C-GMDH onde os graus dos polinômios das descrições parciais crescem de uma camada para a outra. Por outro lado, nessa configuração não há o encaminhamento de nós. Para comparação, foram utilizadas redes MLP com uma camada oculta de 40 neurônios.
- A *configuração G* fornece redes C-GMDH como propostas por Ivakhnenko, ou seja, com descrições parciais sempre de primeiro grau, com neurônios recebendo

quantidades de entradas diferentes em cada camada e com encaminhamento de nós entre camadas não adjacentes. Experimentos com essas redes foram comparados com MLPs de duas camadas ocultas com 20 e 10 neurônios.

- Por fim, a *configuração H* introduz uma variação na C-GMDH, dando origem a redes com as mesmas características das apresentadas na *configuração G*, a não ser pelo que se refere aos graus dos polinômios, que, neste caso, aumentam de uma camada para outra. Neste caso, as redes MLP possuem duas camadas ocultas com 40 e 20 neurônios.

Os resultados obtidos com redes GMDH configuradas para realizarem o encaminhamento entre camadas não adjacentes foram comparados com os resultados de redes MLP com duas camadas ocultas. Por exemplo, a *configuração 4* (Tab. 4.3) define redes GMDH com até 5 camadas e até 10 neurônios por camada (após a seleção), sempre com duas entradas por neurônio e grau 2, mas com encaminhamento de saídas. Os resultados dessas redes são comparados com redes MLP com 3 camadas, contendo 10 neurônios na primeira camada oculta, 5 na segunda camada oculta e um neurônio na camada de saída.

As configurações utilizadas nos experimentos foram definidas empiricamente, tomando como base a experiência adquirida durante o desenvolvimento do *GMDH Box*. Procurou-se, contudo, manter uma relação entre as complexidades das redes GMDH e MLP através dos números de camadas e de neurônios. Por exemplo, redes GMDH com encaminhamento de neurônios foram comparadas com redes MLP com duas camadas ocultas. Importa ressaltar que essa flexibilidade na criação de redes GMDH, incluindo a possibilidade de se criarem diferentes variações nos métodos, é uma característica unicamente apresentada pela ferramenta desenvolvida neste estudo.

4.6 SISTEMA 1

A planta representada pelo *Sistema 1* pode apresentar dois comportamentos diferentes, determinados pelas equações 4.8 (*Modelo 1*) e 4.9 (*Modelo 2*). Os conjuntos de dados de entrada para o treinamento e seleção baseados no *Modelo 1* são determinados pela Eq. 4.10 e os baseados no *Modelo 2* são determinados pela Eq. 4.11, onde $\text{rnd}(n, b_1, b_2)$ é uma função que produz n números aleatórios no intervalo $[b_1, b_2]$. A Eq. 4.12 define os

Tabela 4.3: Configurações das redes neurais a serem utilizadas nos experimentos.

	Configuração →	A	B	C	D	E	F	G	H
GMDH	Num. Máx. de Camadas	5	5	5	5	5	5	5	4
	Num. Máx. de Neurônios	5	30	5	10	5	30	5	10
	Num. Min. de Entradas	2	2	2	2	1	1	1	1
	Num. Máx. de Entradas	2	2	2	2	-	-	-	-
	Grau Máximo	2	2	2	2	1	-	1	-
	Encaminhamento	Não	Não	Sim	Sim	Não	Não	Sim	Sim
MLP	Camadas	2	2	3	3	2	2	3	3
	Neurônios na Camada 1	5	10	5	10	20	40	20	40
	Neurônios na Camada 2	-	-	2	5	-	-	10	20
	Neur. na Cam. de Saída	1	1	1	1	1	1	1	1
	Época Máxima	1000	1000	1000	1000	1000	1000	1000	1000

valores fornecidos para $u_m(\cdot)$ para produzir o sinal senoidal de treinamento. A Eq. 4.13 produz 10.001 valores aleatórios ordenados crescentemente no intervalo $[0, 10000]$ usado em $u_m(\cdot)$ para gerar os dados de treinamento *senoidais estocásticos*. O conjunto i_v definido pela Eq. 4.14 é usado para validar a saída das redes neurais.

$$y_1(k+1) = 0.3 y_1(k) + 0.6 y_1(k-1) + 0.6 \text{ sen}[\pi u_1(k)] + 0.3 \text{ sen}[3\pi u_1(k)] + 0.1 \text{ sen}[5\pi u_1(k)] \quad (4.8)$$

$$y_2(k+1) = 0.3 y_2(k) + 0.6 y_2(k-1) + u_2^3(k) + 0.3 u_2^2(k) - 0.4 u_2(k) \quad (4.9)$$

$$u_1(k) = \begin{cases} \text{sen}[2\pi i_t(k)/250] & \text{para entradas originais;} \\ \text{rnd}(10001, -1, 1) & \text{para entradas estocásticas;} \\ \text{sen}[2\pi i_t^r(k)/250] & \text{para entradas senoidais estocásticas;} \\ \text{sen}[2\pi i_v(k)/250] & \text{para entradas de validação} \end{cases} \quad (4.10)$$

$$u_2(k) = \begin{cases} \text{sen}[2\pi i_t(k)/250] + \text{sen}[2\pi i_t(k)/25] & \text{para entradas originais;} \\ \text{rnd}(10001, -1, 1) & \text{para entradas estocásticas;} \\ \text{sen}[2\pi i_t^r(k)/250] + \text{sen}[2\pi i_t^r(k)/25] & \text{para entradas senoidais estocásticas;} \\ \text{sen}[2\pi i_v(k)/250] + \text{sen}[2\pi i_v(k)/25] & \text{para entradas de validação} \end{cases} \quad (4.11)$$

$$i_t = [0, 0.1, 0.2, \dots, 1000] \quad (4.12)$$

$$i_t^r = \text{sort} [\text{rnd}(10001, 0, 1000)] \quad (4.13)$$

$$i_v = [0, 0.05, 0.10, 0.15, \dots, 1000] \quad (4.14)$$

4.7 SISTEMA 2

No *Sistema 2*, a planta a ser modelada é representada pela Eq. 4.15. As entradas de treinamento e a validação são definidas pela Eq. 4.16. As Equações 4.17, 4.18 e 4.19 definem, respectivamente, as entradas *originais*, *senoidais estocásticas* e de *validação* produzidas por $u(\cdot)$.

$$y(k+1) = \frac{y(k)y(k-1)[y(k) + 2.5]}{1 + y^2(k) + y^2(k-1)} + u(k) \quad (4.15)$$

$$u(k) = \begin{cases} \text{sen} [2\pi i_t(k)/25] & \text{para entradas originais;} \\ \text{rnd}(1001, -2, 2) & \text{para entradas estocásticas;} \\ \text{sen} [2\pi i_t^r(k)/25] & \text{para entradas senoidais estocásticas;} \\ \text{sen} [2\pi i_v(k)/25] & \text{para entradas de validação} \end{cases} \quad (4.16)$$

$$i_t = [0, 0.1, 0.2, \dots, 100] \quad (4.17)$$

$$i_t^r = \text{sort} [\text{rnd}(1001, 0, 100)] \quad (4.18)$$

$$i_v = [0, 0.01, 0.02, 0.03, \dots, 100] \quad (4.19)$$

4.8 SISTEMA 3

No *Sistema 3*, o comportamento da planta é definido pela Eq. 4.20 e os dados de entrada pela Eq. 4.21. As equações 4.22, 4.23 e 4.24 descrevem os valores usados nas funções senoidais para produzir os dados de treinamento e validação.

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k) \quad (4.20)$$

$$u(k) = \begin{cases} \text{sen}[2\pi i_t(k)/25] + \text{sen}[2\pi i_t(k)/10] & \text{para entradas originais;} \\ \text{rnd}(1001, -2, 2) & \text{para entradas estocásticas;} \\ \text{sen}[2\pi i_t^r(k)/25] + \text{sen}[2\pi i_t^r(k)/10] & \text{para entradas senoidais estocásticas;} \\ \text{sen}[2\pi i_v(k)/25] + \text{sen}[2\pi i_v(k)/10] & \text{para entradas de validação} \end{cases} \quad (4.21)$$

$$i_t = [0, 0.1, 0.2, \dots, 100] \quad (4.22)$$

$$i_t^r = \text{sort}[\text{rnd}(1001, 0, 100)] \quad (4.23)$$

$$i_v = [0, 0.01, 0.02, 0.03, \dots, 100] \quad (4.24)$$

4.9 SISTEMA 4

Apesar de a planta descrita pela Eq. 4.25 se comportar de forma semelhante aos sistemas descritos anteriormente, este exemplo apresenta uma pequena diferença no esquema de entradas. A Eq. 4.26 define os valores de entrada para treinamento e validação entretanto, o comportamento das funções senoidais se modifica de acordo com o tempo, de acordo com a Eq. 4.27. As Equações 4.28, 4.29 e 4.30 determinam os valores usados na Eq. 4.26 para produzir os valores senoidais de referência.

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)u(k-1)[y(k-2)-1] + u(k)}{1 + y^2(k-2) + y^2(k-1)} \quad (4.25)$$

$$u(k) = \begin{cases} f[k, i_t(k)] & \text{para entradas originais;} \\ \text{rnd}(801, -2, 2) & \text{para entradas estocásticas;} \\ f[k, i_t^r(k)] & \text{para entradas senoidais estocásticas;} \\ f[k, i_v(k)] & \text{para entradas de validação} \end{cases} \quad (4.26)$$

$$f(k, g(k)) = \begin{cases} \text{sen}[2\pi g(k)/250] & \text{se } k \leq 500; \\ 0.8 \text{ sen}[2\pi g(k)/250] + 0.2 \text{ sen}[2\pi g(k)/25] & \text{se } k > 500 \end{cases} \quad (4.27)$$

$$i_t = [0, 1, 2, 3, \dots, 800] \quad (4.28)$$

$$i_t^r = \text{sort}[\text{rnd}(801, 0, 800)] \quad (4.29)$$

$$i_v = [0, 0.1, 0.2, 0.3, \dots, 800] \quad (4.30)$$

4.10 RESULTADOS EXPERIMENTAIS

Para validar as redes treinadas, foram analisados os *erros relativos* produzidos entre as saídas das redes (N) e as respostas dos sistemas (P), ambos, redes e sistemas, estimulados com as mesmas entradas. Os *erros relativos* foram calculados conforme a Eq. 4.31, onde \mathbf{y} é o vetor com as saídas dos sistemas, $\hat{\mathbf{y}}$ é o vetor com as saídas das RNAs. Foram considerados bem sucedidos (acertos) os resultados que obtiveram *erros relativos* inferiores a 10%. O número de acertos dividido pelo número total de amostras denota o valor do desempenho para cada experimento em particular. O desempenho ou *percentual de acertos* de cada rede foi calculado segundo a Eq. 4.32, onde os vetores \mathbf{y} e $\hat{\mathbf{y}}$ são os vetores com as saídas dos sistemas e da redes, respectivamente, s representa a quantidade de amostras e $X(i)$ representa um acerto caso o *erro relativo* da amostra

seja menor do que 0,1 (10%). Também foram registrados os tempos entre os inícios e os términos dos treinamentos.

$$Er(i) = \left| \frac{\mathbf{y}(i) - \hat{\mathbf{y}}(i)}{d} \right|, \quad \text{onde } d = \begin{cases} \mathbf{y}(i), & \text{se } \mathbf{y}(i) \neq 0 \\ 1 \end{cases} \quad (4.31)$$

$$Xp(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{s} \sum_{i=1}^s X(i), \quad \text{onde } X(i) = \begin{cases} 1, & \text{se } Er(i) \leq 0.1 \\ 0 \end{cases} \quad (4.32)$$

Os valores produzidos pelos sistemas representados pelas Equações 4.8, 4.9, 4.15 e 4.20 são funções das entradas atuais e de saídas atuais e passadas. O sistema 4 (Eq. 4.25) é uma função de entradas passadas, da entrada atual e de saídas passadas. As redes treinadas para modelagem dos sistemas e para predição de uma amostra utilizaram as configurações das entradas conforme descrito nas seções 4.6, 4.7, 4.8 e 4.9.

Para realizar a predição em 20 unidades de tempo, as redes foram treinadas com entradas atrasadas em 20 amostras. Mantendo-se os alvos sem atraso, planejou-se que as redes, depois de treinadas, realizassem a predição das saídas dos sistemas com 20 amostras de antecedência. Aplicando-se o atraso de 20 amostras, as Equações 4.8, 4.9, 4.15, 4.20 e 4.25 são reescritas, respectivamente, na forma das Equações 4.33, 4.34, 4.35, 4.36 e 4.37.

Durante a validação das redes treinadas para predição de 20 amostras, as entradas fornecidas são as mesmas utilizadas para a validação das redes treinadas para modelagem. Contudo, as saídas esperadas das redes são as aproximações das saídas dos sistemas com vinte amostras de antecedência. A validação neste caso também é feita analisando-se o percentual de acertos entre as saídas das redes e as saídas dos sistemas com vinte amostras de antecedência.

As seções seguintes apresentam as análises dos resultados obtidos, que foram tabulados e estão apresentados no apêndice A.

$$y_1(k+1) = 0.3 y_1(k-20) + 0.6 y_1(k-21) + 0.6 \text{ sen}[\pi u_1(k-20)] + \quad (4.33) \\ 0.3 \text{ sen}[3\pi u_1(k-20)] + 0.1 \text{ sen}[5\pi u_1(k-20)]$$

$$y_2(k+1) = 0.3 y_2(k-20) + 0.6 y_2(k-21) + u_2^3(k-20) + 0.3 u_2^2(k-20) - 0.4 u_2(k-20) \quad (4.34)$$

$$y(k+1) = \frac{y(k-20)y(k-21)[y(k-20) + 2.5]}{1 + y^2(k-20) + y^2(k-21)} + u(k-20) \quad (4.35)$$

$$y(k+1) = \frac{y(k-20)}{1 + y^2(k-20)} + u^3(k-20) \quad (4.36)$$

$$y(k+1) = \frac{y(k-20)y(k-21)y(k-22)u(k-21)[y(k-22) - 1] + u(k-20)}{1 + y^2(k-22) + y^2(k-21)} \quad (4.37)$$

4.10.1 Resultados para o Sistema 1

A Tab. A.1 mostra os resultados para os 72 experimentos realizados com o *Sistema 1*, variação **a**. A Fig. 4.3 compara o desempenho das implementações MLP e GMDH. A Fig. 4.3(a) mostra, no eixo y , os percentuais de acertos. Os resultados das redes GMDH são plotados em azul e das redes MLP em vermelho. O eixo x mostra o número do experimento, conforme a primeira coluna da Tab. A.1. Cada grupo de oito experimentos corresponde a uma das nove tarefas definidas na Tab. 4.2, realizadas com as redes configuradas de acordo com a Tab. 4.3. Esses grupos estão realçados com cores de fundo alternadas no gráfico. As configurações das redes utilizadas nos experimentos (letras de A a H) aparecem sobre o eixo x . A Fig. 4.3(b) mostra os tempos de treinamento das redes em cada experimento, onde o eixo x representa o número do experimento e o eixo y , o tempo, em segundos e em escala logarítmica. As configurações das redes (letras de A a H) também aparecem sobre o eixo x .

Observando-se a Fig. 4.3(a), percebe-se que as redes GMDH apresentam resultados equivalentes aos das redes MLP para as tarefas predição na configuração *série-paralela* (tarefas de 1 a 6). Uma exceção é o terceiro grupo de experimentos (17 a 24), que utiliza entradas estocásticas, onde quatro redes MLP alcançaram resultados próximos de 100% e todas as outras redes, MLPs ou GMDHs, tiveram desempenhos abaixo ou

próximos de 80%. A menos da rede MLP com configuração **D**, os experimentos de predição com 20 amostras de antecedência e entrada senoidal (tarefa 2) produziram percentuais de acertos superiores a 80%.

Os experimentos de predição da tarefa 4, onde, assim como na tarefa 3, as entradas são estocásticas, produziram saídas com percentuais de acertos próximos ou inferiores a 20%. Já as entradas *senoidais estocásticas* utilizadas nos treinamentos das redes nas tarefas 5 e 6 geraram identificações com percentuais de acertos bem semelhantes aos obtidos com as redes treinadas com as entradas *senoidais*. Percebe-se, ainda, que as entradas *senoidais estocásticas* favoreceram uma melhor identificação para as redes MLP, exercendo, contudo, menor influência sobre as redes GMDH, mas ainda assim fornecendo uma pequena melhora em alguns casos.

Os tempos de treinamento das redes GMDH foram, em grande parte dos casos, bem inferiores aos tempos de treinamento das redes MLP. Por outro lado, as redes C-GMDH com encaminhamento (G) e com encaminhamento e polinômios com graus crescentes (H) tiveram tempos de processamento bem mais altos, quase sempre se equiparando aos tempos das MLPs e, em alguns casos, superando-os.

As tarefas de modelagem, ou seja, aquelas em que as redes recebem a realimentação de suas próprias saídas, foram de difícil identificação por ambos os modelos de RNA. Dos 24 experimentos (49 a 72), apenas cinco redes MLP alcançaram desempenhos próximos ou iguais a 100%. Duas redes MLP alcançaram acertos em torno de 90% das saídas. A rede GMDH com melhor resultado dentre as tarefas 7, 8 e 9 foi a criada no experimento 72, com 16,61% de acertos.

A Tab. A.2 e a Fig. 4.4 mostram os resultados para os 72 experimentos realizados com o *Sistema 1*, variação **b**. A tarefa de predição com uma amostra de antecedência mostrou resultados excelentes em todas as configurações e modelos de redes neurais. A tarefa 2, que realiza a predição com 20 amostras de antecedência teve percentuais de acertos bem acima dos 80% obtidos com a variação **a** do Sistema 1, a não ser pelas redes C-GMDH com encaminhamento (G e H) e pela rede MLP H.

As tarefas com entradas estocásticas não produziram bons resultados, entretanto, a tarefa 3 foi mais favorável na variação **b** do que na variação **a** do *Sistema 1*. A tarefa 4 não produziu nenhum resultado aproveitável. As tarefas 5 e 6, que usam entradas *senoidais estocásticas*, tiveram resultados ligeiramente superiores aos obtidos

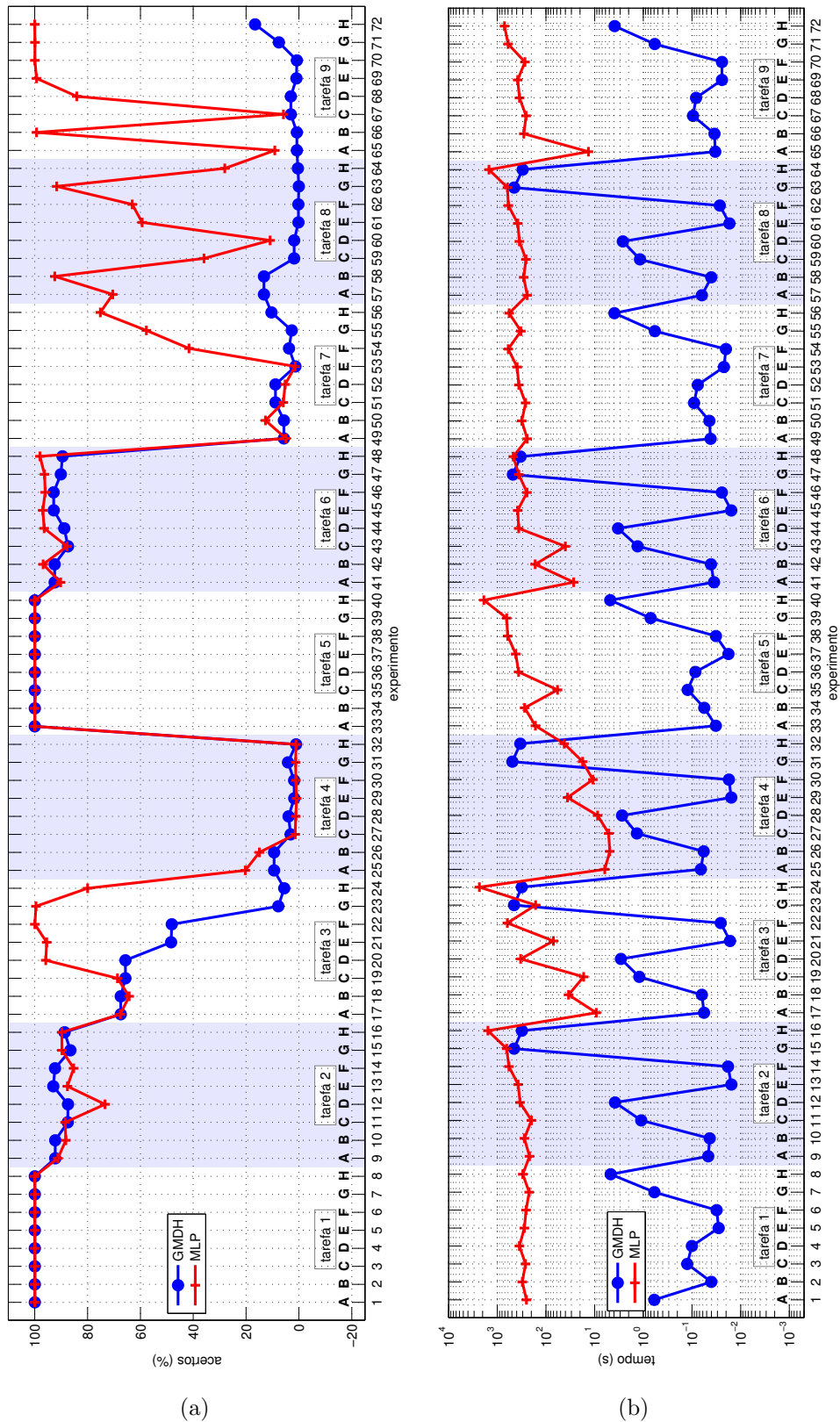


Figura 4.3: Resultados dos experimentos com o *Sistema 1*, variação a. (a) Percentuais de acerto. (b) Tempos de treinamento (eixo dos tempos em escala logarítmica). Os eixo x de ambos os gráficos mostram o número do experimento, como visto na Tab. A.1. Cada tarefa contém oito experimentos, com configurações de rede variando de A a H. Os fundos com cores alternadas destacam as tarefas (ver Tab. 4.2).

na variação **a**. O mais significativo foi o fato de que os experimentos com predição de 20 amostras tiveram resultados iguais a 100% ou muito próximos desse percentual.

As tarefas 7, 8 e 9, que realizam a modelagem com entradas *paralelas*, tiveram maior número de experimentos bem sucedidos com as redes MLP e forneceram também percentuais de acertos bem altos com as redes C-GMDH com polinômios de graus crescentes (configuração H).

Os tempos de treinamento das redes seguiram padrões e patamares semelhantes os obtidos nos treinamentos da variação **a** do *Sistema 1*. Importantes exceções foram os tempos de treinamento das rede C-GMDH dos experimento 56 e 72, que foram bem inferiores aos das respectivas redes MLP, além do fato de que essas redes C-GMDH conseguiram realizar com sucesso a modelagem dos sistemas.

4.10.2 Resultados para o Sistema 2

Os resultados dos experimentos realizados com o Sistema 2 são apresentados na Tabela A.3. A Fig. 4.5(a) mostra os percentuais de acertos (eixo y) por experimento (eixo x). A Fig. 4.5(b) retrata os tempos de treinamento em escala logarítmica. Cada conjunto de oito experimentos corresponde a uma tarefa (ver Tab. 4.2). Cores de fundo alternadas separam os experimentos entre as tarefas correspondentes. Os identificadores das configurações das redes neurais aparecem sobre o eixo x (letras de A a H).

Analisando os resultados, vê-se que as tarefas de predição com 20 amostras de antecedência (tarefas 2, 4 e 6) deram origem a redes com baixos percentuais de acertos. As melhores predições foram feitas na tarefa 6, que usa entradas *senoidais estocásticas*, onde as redes MLP de configurações B, E, F e H alcançaram acertos em torno de 80%. Os melhores resultados entre as GMDH na tarefa 6 estiveram por volta dos 60% e foram obtidos pelas redes C-GMDH. As tarefas de predição de uma amostra com entradas *senoidais* e *senoidais estocásticas* (tarefas 1 e 5) tiveram percentual de acertos próximo ou superior a 90%.

As predições de 20 amostras realizadas na tarefa 2, que usa entradas *senoidais*, tiveram percentuais de acertos inferiores a 60% em todas as redes. A predição de uma amostra com entradas *estocásticas* (tarefa 3) favoreceu quatro das redes MLP, que, nos experimentos de 21 a 24, acertaram as saídas em mais de 90% das amostras. Nenhuma das

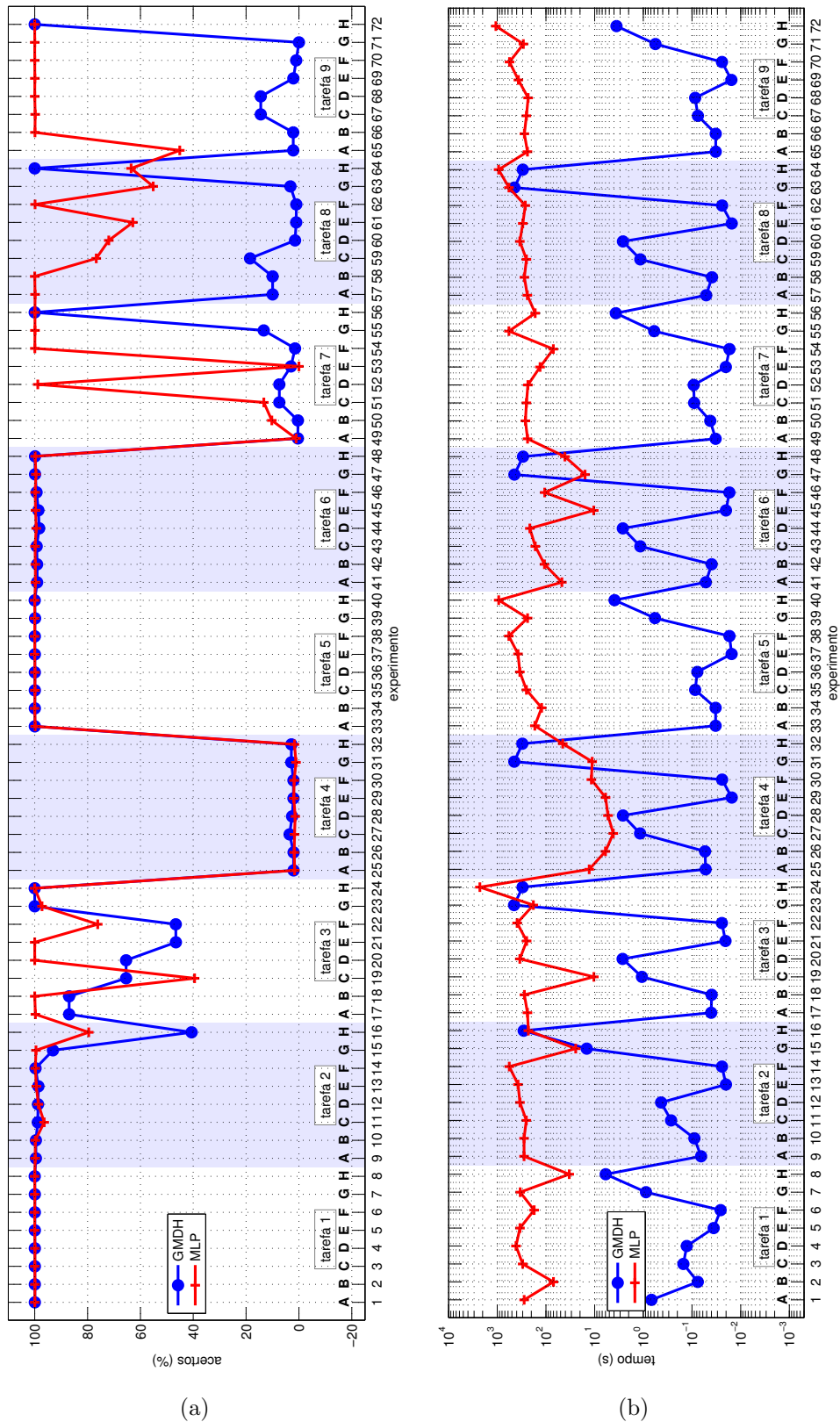


Figura 4.4: Resultados dos experimentos com o *Sistema 1*, variação **b**. (a) Percentuais de acerto. (b) Tempos de treinamento (eixo dos tempos em escala logarítmica). Os eixo x de ambos os gráficos mostram o número do experimento, como visto na Tab. A.2. Cada tarefa contém oito experimentos, com configurações de rede variando de A a H. Os fundos com cores alternadas destacam as tarefas (ver Tab. 4.2).

redes treinadas para a tarefa 4 (experimentos de 25 a 32) conseguiu alcançar 1% de acerto.

Nas tarefas de modelagem, as redes GMDH não foram capazes de identificar o sistema. As redes MLP conseguiram percentuais de acertos maiores do que 90% em 14 dos 24 experimentos de números entre 49 e 72.

Os tempos de treinamento das redes GMDH foram quase sempre menores do que os das redes MLP, a não ser nos treinamentos das redes C-GMDH, que demoraram mais do que os treinamentos MLP correspondentes, a menos do experimento 24, em que as duas redes foram treinadas em tempos muito próximos.

4.10.3 Resultados para o Sistema 3

A Tabela A.4 contém os dados obtidos com os experimentos realizados com o Sistema 3. Esses valores foram mostrados na Fig. 4.6. Seguindo o padrão de apresentação dos experimentos anteriores, os eixos x dos gráficos representam os experimentos, que foram agrupados de oito em oito, através de cores de fundo alternadas, onde cada grupo representa uma das tarefas definidas na Tab. 4.2. Cada experimento do grupo corresponde a uma configuração GMDH e a uma configuração de MLP, cujos resultados foram representados com linhas azuis e vermelhas, respectivamente. Os identificadores das configurações das redes neurais aparecem sobre o eixo x (letras de A a H).

É visível na Fig. 4.6(a) que as redes GMDH apenas foram capazes de realizar predições com uma amostra de antecedência, mais especificamente nas tarefas 1 e 5. As entradas estocásticas, aplicadas nos experimentos das tarefas 3 e 4 não permitiram uma boa identificação pelas redes GMDH. As redes MLP se comportaram de forma semelhante nos mesmos experimentos, apresentando, contudo, boa identificação nos experimentos 17, 19, 20, 21 e 24, correspondentes à tarefa 3.

Os experimentos de 49 a 72 objetivaram a modelagem do sistema. Nestes casos, as redes MLP, apesar de mostrarem também alguma dificuldade, produziram alguns percentuais de acertos maiores do que 90%, enquanto que as redes GMDH não produziram resultados estatisticamente válidos.

A Fig. 4.6(b) retrata os tempos de treinamento, que mantiveram comportamento semel-

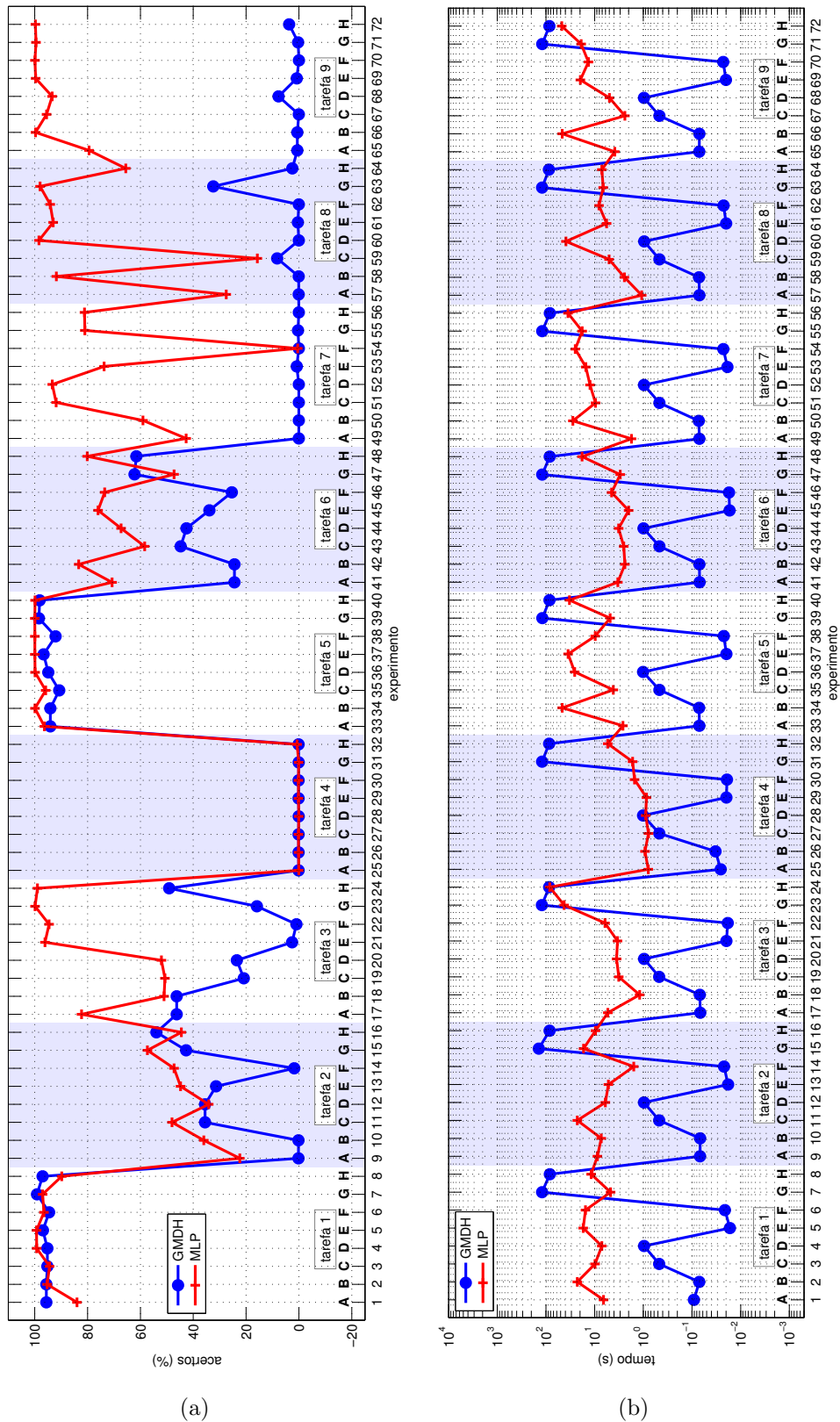


Figura 4.5: Resultados dos experimentos com o *Sistema 2*. (a) Percentuais de acerto. (b) Tempos de treinamento (eixo dos tempos em escala logarítmica). Os eixo x de ambos os gráficos mostram o número do experimento, como visto na Tab. A.3. Cada tarefa contém oito experimentos, com configurações de rede variando de A a H. Os fundos com cores alternadas destacam as tarefas (ver Tab. 4.2).

hante ao dos experimentos anteriores: as redes GMDH foram treinadas bem mais rapidamente do que as redes MLP, a não ser as redes C-GMDH, que tomaram tempos bem maiores e que, em alguns casos, até mesmo ultrapassaram os tempos de treinamento de suas redes MLP associadas. Embora os tempos dos treinamentos das C-GMDH dos experimentos 31 e 32 tenham sido bem menores do que das outras C-GMDH, seus percentuais de acertos mostram que elas foram malsucedidas.

4.10.4 Resultados para o Sistema 4

A Tabela A.5 apresenta os resultados dos experimentos com o Sistema 4. A Fig. 4.7(a) apresenta o gráfico com os percentuais de acertos de cada experimento e a Fig. 4.7(b) retrata os tempos de treinamento de cada rede. Seguindo o padrão das figuras anteriores, cada grupo de 8 experimentos corresponde a uma das tarefas definidas na Tabela 4.2. As redes neurais foram configuradas segundo as características definidas na Tabela 4.3, cujos identificadores aparecem sobre o eixo x (letras de A a H).

Os percentuais de acertos para o Sistema 4 (ver Fig. 4.7(a)) mostram que ambos os tipos de rede, GMDH e MLP, apenas conseguiram realizar as tarefas 1 e 5, que usam entradas senoidais e senoidais estocásticas, respectivamente, e que se destinam à predição de saídas futuras com uma amostra de antecedência. Apenas as redes MLP dos experimentos 21, 22 e 25, da tarefa 3 – predição de uma amostra com entradas estocásticas – conseguiram percentuais de acertos superiores a 90%.

As redes GMDH dos experimentos 67, 68, 71 e 72, conseguiram modelar com sucesso o sistema, utilizando identificação paralela. As redes MLP alcançaram acertos superiores a 90% em todos os experimentos de modelagem.

No que tange os tempos de treinamento, manteve-se no Sistema 4 o mesmo padrão obtido nos sistemas anteriores, ou seja, os treinamentos das redes MLP foram mais demorados do que os das redes GMDH, a não ser em alguns casos, geralmente no treinamento das redes C-GMDH. Note-se, contudo, que as redes GMDH dos experimentos 67 e 68 (O-GMDH com encaminhamento de nós) realizaram com sucesso a modelagem dos sistemas e seus tempos de treinamento foram menores do que os de suas contrapartes em MLP.

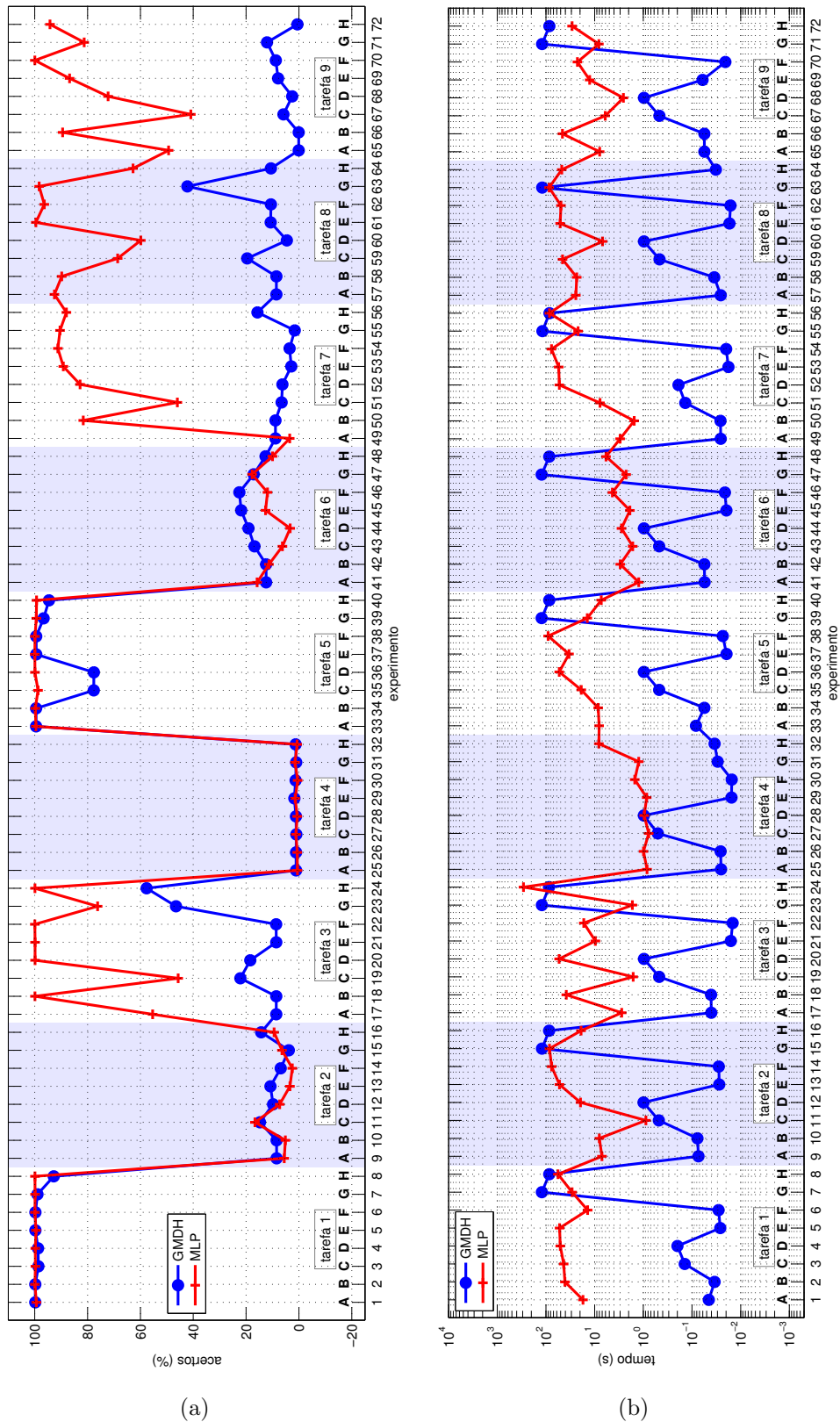


Figura 4.6: Resultados dos experimentos com o *Sistema 3*. (a) Percentuais de acerto. (b) Tempos de treinamento (eixo dos tempos em escala logarítmica). Os eixo x de ambos os gráficos mostram o número do experimento, como visto na Tab. A.4. Cada tarefa contém oito experimentos, com configurações de rede variando de A a H. Os fundos com cores alternadas destacam as tarefas (ver Tab. 4.2).

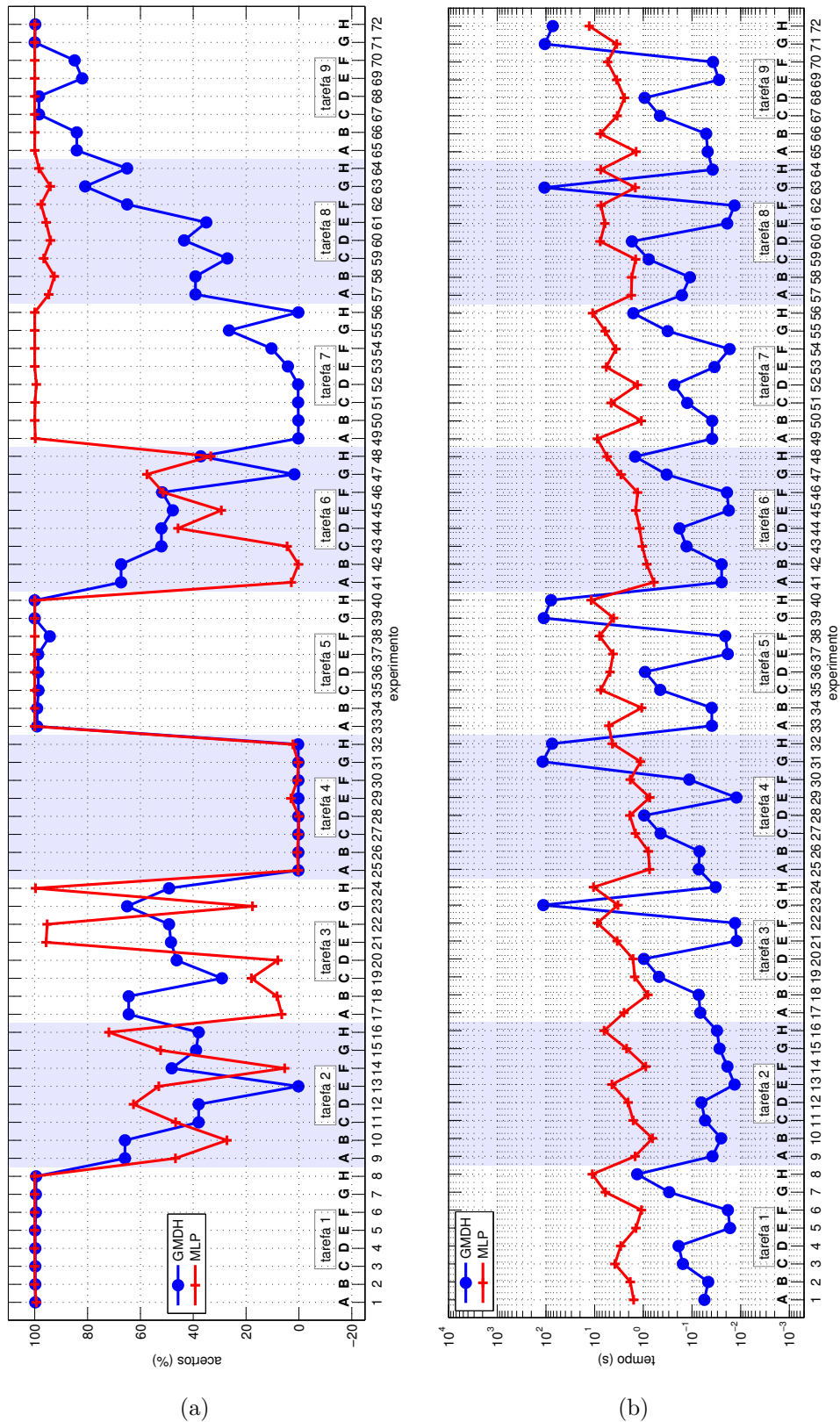


Figura 4.7: Resultados dos experimentos com o *Sistema 4*. (a) Percentuais de acerto. (b) Tempos de treinamento (eixo dos tempos em escala logarítmica). Os eixo x de ambos os gráficos mostram o número do experimento, como visto na Tab. A.5. Cada tarefa contém oito experimentos, com configurações de rede variando de A a H. Os fundos com cores alternadas destacam as tarefas (ver Tab. 4.2).

4.11 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou comparações entre diversas implementações de redes GMDH e MLP para as tarefas de predição e de modelagem de sistemas dinâmicos não lineares. As análises demonstraram que, na maioria dos casos, as redes GMDH podem ser treinadas mais rapidamente que as redes MLP e que seus desempenhos gerais foram compatíveis em relação às tarefas de predição. Por outro lado, as redes MLP apresentaram-se superiores para as tarefas de modelagem. Apesar desses resultados, as redes GMDH mostraram um interessante potencial para a aplicação proposta.

Este trabalho introduziu uma técnica de treinamento para a predição de saídas futuras dos sistemas, utilizando amostras e alvos de treinamento defasados no tempo entre si, de forma que uma amostra atual possa ser usada para estimar o comportamento do sistema em um tempo futuro. A predição de 1 amostra pode ser útil principalmente em um processo considerado lento, onde a estimativa da próxima saída possa ajudar no ajuste das suas entradas. A predição de entradas futuras mais distantes pode ser aplicada em sistemas de resposta mais rápida. A escolha da distância do valor predito em relação ao valor atual depende das características e das necessidades dos sistemas.

Ainda no que diz respeito à predição de saídas, muitas vezes os sistemas têm seus parâmetros modificados durante o tempo. As redes MLP podem sofrer ajustes pontuais nessas situações, enquanto que as redes GMDH precisariam ser retreinadas. Entretanto, o tempo de treinamento das O-GMDH, sendo consideravelmente menor do que os das MLPs, pode ainda ser uma vantagem nesses casos. Adicionalmente, existem relatos de utilização de algoritmos como o *backpropagation* para o ajuste dos coeficientes das descrições parciais. Essa capacidade ainda deve ser acrescentada no *GMDH Box*.

Ficou demonstrado, contudo, que é possível utilizar-se GMDH para a predição, que, como visto nos resultados apresentados, pode ser implementado mais rapidamente por causa dos menores tempos de treinamento requeridos em muitos casos. Apesar de que os experimentos com predição de 20 amostras utilizando GMDH somente tenham sido bem sucedidos para o Sistema 1, é possível supor que os resultados sejam melhores caso a predição seja feita para uma saída futura menos distante.

Outra contribuição deste projeto foi a aplicação de redes GMDH em identificação *paralela* de sistemas não lineares. A identificação paralela pode ser aplicada, por exemplo, em substituição de modelos físicos complexos, que demorariam um tempo ex-

cessivo para serem processados. Outra aplicação é a substituição de sensores de difícil manutenção ou de custo elevado. Sendo possível modelar a relação entre as entradas e as saídas desses dispositivos, seria possível aplicarem-se modelos baseados em GMDH no lugar dos sensores reais. Devido ao menor custo computacional das redes GMDH em comparação com outros modelos de RNA, especialmente na fase de execução (redes GMDH utilizam apenas polinômios, que podem ser implementados por somas e multiplicações), podem-se utilizar sistemas de processamento embarcados para executá-las.

Os resultados experimentais demonstraram que as tarefas de predição de uma amostra podem ser realizadas com redes O-GMDH, que necessitam de pouco tempo para processar e que apresentam estruturas de conexão bem simples e pequena quantidade de neurônios. Em alguns casos, a utilização de entradas *senoidais estocásticas* foi benéfico para o treinamento, mas as entradas originais produziram resultados excelentes. Por outro lado, o uso de entradas *senoidais estocásticas* auxiliou sempre positivamente os resultados tanto das redes MLP como das GMDH, embora sua aplicabilidade, como mencionado anteriormente, apenas foi bem sucedida no Sistema 1.

Percebeu-se, ainda, que a utilização de entradas meramente *estocásticas* não favoreceram a predição dos sistemas por ambos os modelos de RNAs, apresentando, contudo, as redes MLP, mais aptidão para o processo, apesar dos seus altos tempos de processamento. Em contrapartida, a modelagem (identificação *paralela*) utilizando entradas estocásticas mostrou bons percentuais de acertos e, mais ainda, para o Sistema 2, percebe-se que esse tipo de entrada foi benéfica na modelagem.

As redes GMDH tiveram dificuldade nas identificações *paralelas* mas, mesmo assim, produziram alguns resultados muito bons nos sistemas 1b e 4. Destaque-se o fato de que as configurações de GMDH bem sucedidas nesse caso utilizaram sempre encaminhamento de nós (configurações C, D, G e H).

Com base nos dados colhidos nos experimentos apresentados neste capítulo, pode-se inferir que a aplicação de redes GMDH para a identificação de sistemas não lineares pode ser realizada, especialmente quando se aplica a identificação *série-paralela*. Os resultados também demonstraram que é possível fazer a modelagem dos sistemas, usando a abordagem *paralela* e redes C-GMDH.

É plausível afirmar-se que vale a pena continuar a desenvolver os algoritmos de treinamento com o intuito de aplicar esse modelo a problemas reais de engenharia, con-

siderando-se que (a) as redes O-GMDH podem ser mais simples que as redes MLP em termos dos recursos computacionais consumidos; (b) que existem relatos de outros trabalhos utilizando O-GMDH para a aproximação de sistemas não lineares complexos e; (c) que as redes C-GMDH foram capazes de realizar a modelagem dos sistemas em alguns dos experimentos realizados.

5 PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS USANDO GMDH

5.1 CONSIDERAÇÕES INICIAIS

Proteínas são longas sequências de resíduos de aminoácidos que, em condições fisiológicas, adotam uma estrutura **tridimensional** particular [66]. Essa estrutura tridimensional, também denominada **estrutura nativa da proteína**, permite que a proteína realize sua função na célula [67]: atividade enzimática, armazenamento e transporte, imunológica (na forma de anticorpos), dentre outras.

A predição da estrutura terciária de proteínas é um dos problemas mais importantes e ainda não resolvidos da Bioinformática Estrutural [68]. A estrutura de uma molécula de proteína pode ser determinada através de métodos experimentais como a cristalografia por Raio X e a ressonância magnética nuclear (NMR – *Nuclear Magnetic Resonance*). Entretanto, ambos os métodos são custosos, demorados e apresentam certas limitações [69].

Ao longo dos últimos anos, esforços têm sido feitos com o intuito de se encontrarem estratégias computacionais para prever a estrutura tridimensional de proteínas. Os métodos desenvolvidos podem ser divididos em quatro classes [68]:

1. Métodos de Primeiros Princípios sem banco de dados [70];
2. Métodos de Primeiros Princípios Primário com banco de dados [71, 72];
3. Métodos de Reconhecimento de Enovelamento (*Fold Recognition* - **FR**) [73, 74, 75, 76] e;
4. Métodos de Modelagem Comparativa (*Comparative Modeling* - **CM**) [77, 78].

Os métodos do primeiro grupo buscam realizar a predição de novas estruturas apenas através da simulação computacional de propriedades físico-químicas específicas que acontecem com as proteínas na natureza durante o enovelamento. O objetivo é

encontrar, para cada proteína, o mínimo global dos valores de sua energia livre, que corresponde a sua estrutura nativa ou conformação funcional [70].

Na modelagem comparativa, a sequência alvo é alinhada com a sequência de um modelo evolucionário relacionado do qual se conhece a estrutura [77]. Para obterem-se modelos conhecidos, utiliza-se o Banco de Dados de Proteínas (*Protein Data Bank* - **PDB**) [79]. Encontrando-se uma homologia, a modelagem continua através da cópia de informações estruturais do modelo.

Métodos de *primeiros princípios* com informações de banco de dados, por sua vez, frequentemente comparam pequenos fragmentos de resíduos de aminoácidos e os combinam para encontrar a estrutura tridimensional com o menor energia potencial. Nos métodos de alinhamento, o objetivo principal é ajustar uma sequência proteica corretamente em um modelo estrutural. Métodos de reconhecimento de enovelamento são motivados pela noção de que a estrutura é mais preservada evolucionariamente do que a sequência; isto é, proteínas com sequências aparentemente sem nenhuma similaridade podem apresentar enovelamentos similares [80, 81].

Apesar de ser possível obterem-se predições de alta qualidade com a modelagem comparativa por homologia e por reconhecimento de enovelamento, este processo apresenta uma limitação: a incapacidade para reconhecer novas estruturas. Isso se explica pelo fato de que a metodologia somente pode prever estruturas de sequências proteicas que são similares ou quase idênticas a outras sequências de estruturas cadastradas no **PDB**. Métodos de *primeiros princípios* sem banco de dados, por sua vez, podem prever novas estruturas mas têm que lidar com o problema da complexidade e da alta dimensionalidade do espaço de busca conformacional [82, 69]. Na última edição do **CASP** (*Critical Assessment of Techniques for Protein Structure Prediction* - Avaliação Crítica de Técnicas de Previsão de Estruturas de Proteínas) não foram observados muitos progressos nos métodos de *princípio primário* sem banco de dados [83]. O maior avanço obtido no **CASP9** foram apresentados por métodos que utilizam informações de banco de dados [84, 83].

Esforços para a predição de estruturas de proteínas têm sido realizados em muitas áreas da ciência da computação. Durante a maior parte da década passada, redes neurais artificiais têm produzido predições precisas de estruturas secundárias de proteínas. Redes neurais são capazes de mapear estruturas através de suas sequências de aminoácidos. Em trabalho recente [85], uma rede do tipo MLP foi aplicada para que se obtivessem

estruturas tridimensionais aproximadas de proteínas.

Neste capítulo, apresenta-se a predição de estruturas de proteínas com um método baseado no treinamento de redes GMDH para o mapeamento de estruturas de fragmentos de sequências proteicas extraídas do PDB e posterior utilização dessas redes para a predizer a estrutura de sequências similares de resíduos. Diversas instâncias de redes baseadas em O-GMDH e C-GMDH foram aplicadas para realizar os experimentos. Os resultados dos experimentos com GMDH foram comparados com outros resultados obtidos utilizando-se redes MLP. Estes experimentos foram publicados em periódico internacional sob o título *A GMDH Polynomial Neural Network-based Method to Predict Approximate Three-dimensional Structures of Polypeptides* [52].

5.2 ESTRUTURAS E REPRESENTAÇÃO DE PROTEÍNAS

Uma proteína, ou polipeptídeo, é uma cadeia linear ordenada de resíduos de aminoácidos conectados por *ligações peptídicas*. Os aminoácidos são os blocos construtores das proteínas. Como pode ser visto através da Fig. 5.1(a), cada aminoácido possui uma molécula de carbono central, também chamada de *carbono alfa*, à qual estão ligados quatro grupos: (1) um hidrogênio, (2) um grupo amina (H_3N^+), (3) um grupo carboxílico (COO^-) e (4) uma cadeia lateral, também conhecida como *grupo R* [86]. As Figuras 5.1(b) e 5.1(c) apresentam as estruturas de dois aminoácidos, a glicina e a alanina, respectivamente. A diferença entre eles se encontra exatamente na região do grupo R, que essencialmente define as propriedades físico-químicas de cada proteína [87].

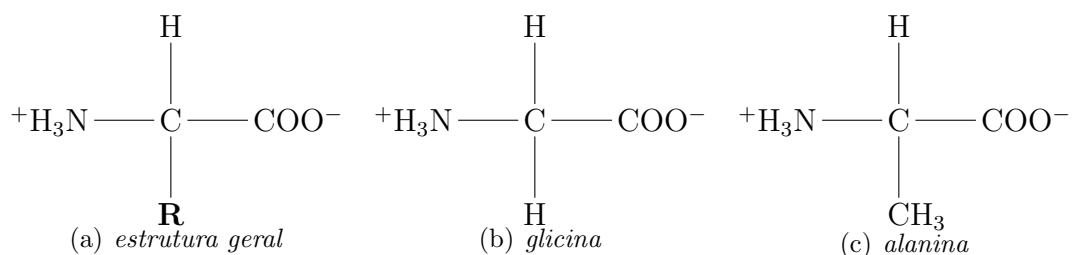


Figura 5.1: Estruturas dos aminoácidos: (a) *estrutura geral*, com o grupo R apresentado abaixo, o grupo amina à esquerda, o hidrogênio acima e o grupo carboxílico à direita; (b) estrutura do aminoácido *glicina*; (c) estrutura do aminoácido *alanina*.

Existem vinte tipos de aminoácidos comumente encontrados na natureza. Essas moléculas diferem entre si através de suas cadeias laterais. Suas estruturas são apresentadas no

apêndice B. A conformação tridimensional de uma proteína e sua função são determinadas por complexas interações entre as cadeias laterais [87].

Ao se combinarem dois aminoácidos, forma-se uma *ligação peptídica* e uma molécula de água é eliminada, como apresentado na Fig. 5.2, onde a ligação peptídica é apresentada em negrito. A junção de três aminoácidos forma um *tripeptídeo* e a junção de vários forma um *polipeptídeo*. Como há perda de uma molécula de água na combinação dos aminoácidos, a cadeia peptídica é formada pelo que restou das moléculas, ou seja de *resíduos de aminoácidos* [87, 88]. No início da cadeia, aparece o grupo amina do primeiro resíduo e no final da cadeia, encontra-se o grupo carboxílico do último resíduo. Esses terminais são conhecidos, respectivamente, por *Região N-Terminal* e *C-Terminal*.

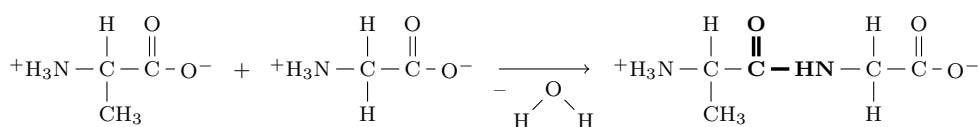


Figura 5.2: Formação de um dipeptídeo, com a eliminação de uma molécula de água. Uma ligação peptídica é estabelecida entre os dois aminoácidos, destacada em negrito.

A Fig. 5.3 mostra que um peptídeo possui três ângulos de torção principais chamados *phi* (ϕ), *psi* (ψ) e *ômega* (ω). As ligações entre N e o carbono alfa (C_α) e entre o carbono alfa e o carbono da ligação peptídica são livres para rotacionar. As rotações, descritas pelos ângulos ϕ e ψ são as características mais importantes na determinação da conformação adotada pela cadeia principal. O conjunto de ângulos de torção de um polipeptídeo determina sua conformação ou enovelamento. Os textos de Tramontano [89] e de Lesk [90] fornecem uma boa visão geral dos métodos de predição de estruturas de proteínas.

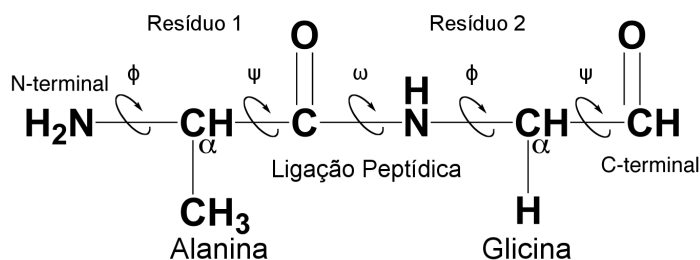


Figura 5.3: Representação esquemática de um modelo de peptídeo. A ligação peptídica é formada entre os aminoácidos Alanina e Glicina. N representa Nitrogênio, C and C_α são átomos de carbono. Baseada em [52]. O ângulo *ômega* (ω) ocorre na ligação peptídica. Os ângulos *phi* (ϕ) e *psi* (ψ) de cada resíduo são livres para rotacionarem.

As estruturas proteicas podem ser estudadas em quatro níveis:

1. **Estruturas Primárias:** descrevem a sequência linear de resíduos de aminoácidos simplesmente por sua ordem linear (N-Terminal \rightarrow C-Terminal)
2. **Estruturas Secundárias:** definidas pela presença de ligações de hidrogênio entre a sequência principal e os grupos de carboxila na cadeia polipeptídica. As duas principais estruturas secundárias são as α -hélices [91] e as folhas- β [92]. O terceiro grupo, laços, possui estruturas que se ligam aos dois outros grupos. Também chamadas de irregulares, as estruturas desse terceiro grupo são formadas em regiões onde o polipeptídeo modifica suas direções, isto é, após uma estrutura secundária regular em uma α -hélice ou uma folha- β
3. **Estruturas Terciárias:** agrupamentos de α -hélices e folhas- β determinam as estruturas em nível terciário.
4. **Estruturas Quaternárias:** compostas por grupos de subunidades de estruturas terciárias.

5.3 O MÉTODO

O método utilizado neste trabalho é composto de dois estágios principais: (a) a aquisição de padrões estruturais e; (b) a predição das estruturas tridimensionais aproximadas usando redes GMDH.

O *primeiro estágio* é realizado em cinco passos [85, 93, 94]: (1) fragmentação da sequência alvo de aminoácidos; (2) busca por modelos de proteínas; (3) cálculo da estrutura secundária e das informações dos ângulos de torção; (4) divisão dos ângulos de torção em grupos e; (5) construção dos padrões estruturais.

O *segundo estágio* compreende mais dois passos: (6) construção e treinamento das redes GMDH e; (7) predição da estrutura tridimensional aproximada da sequência proteica alvo. Mais detalhes sobre esse processo podem ser obtidos no trabalho de Dorn e Norberto de Souza [85, 93]. Uma breve descrição de cada passo é apresentada a seguir:

1. **Fragmentação da sequência alvo de aminoácidos:** a sequência alvo é dividida em curtos fragmentos com cinco resíduos. Um conjunto S formado de todos os possíveis fragmentos contíguos é criado, como mostra a Eq. 5.1, onde s_i e s_p são, respectivamente, o primeiro e o último fragmentos.

$$S = [s_i, s_{i+1}, \dots, s_p] \quad (5.1)$$

A Fig. 5.4 ilustra o processo. Um fragmento s_i se inicia no i -ésimo resíduo e termina no j -ésimo resíduo. Cada fragmento consiste em um vetor de ângulos de torção consecutivos (Eq. 5.2) que descreve as rotações internas do fragmento.

$$W = [\omega_{i-1}, \phi_i, \psi_i, \dots, \omega_{j-1}, \phi_j, \psi_j] \quad (5.2)$$

Um fragmento possui um número ímpar de resíduos (apenas 5) tendo em conta que somente os ângulos do resíduo central são considerados nos passos seguintes. O resíduo central é influenciado por seus vizinhos através dos padrões de disposição das ligações de hidrogênio e apresenta valores específicos para os ângulos de torção. Esses valores podem ser dispostos num diagrama de Ramachandran, que é uma forma de visualizar os ângulos diedrais ¹ ϕ sobre ψ dos resíduos de aminoácido em uma estrutura de proteínas.

G	S	E	K	M	S	T	A	I	Sequência Alvo
G	S	E	K	M					Fragmento 1: GSEKM
	S	E	K	M	S				Fragmento 2: SEKMS
		E	K	M	S	T			Fragmento 3: EK MST
			K	M	S	T	A		Fragmento 4: KMSTA
				M	S	T	A	I	Fragmento 4: MSTAI

Figura 5.4: Representação esquemática da fragmentação de uma sequência proteica. Baseada em [52].

- Busca dos modelos de proteínas:** modelos para cada fragmento s_i são obtidos do banco de proteínas PDB [79], através do alinhamento de sequências usando o algoritmo **BLASTp** (*Basic Local Alignment Search Tool* - Ferramenta de Busca de Alinhamento Local Básico) [96]. Os seguintes parâmetros foram utilizados no algoritmo: *Base de Dados:* PDB; *“Cut off” esperado:* 200000; *Matriz de substituição:* PAM70 [97]; *Tamanho da lista de acertos:* 100.
- Obtenção de Informações sobre a estrutura secundária e os ângulos:** o *software* PROMOTIF [98] foi utilizado para calcular a estrutura secundária dos ângulos de torção de cada resíduo nos templates obtidos do PDB. O PROMOTIF

¹**Diedro:** Diz-se do ângulo formado pelo encontro de dois planos [95].

implementa um algoritmo baseado no método **DSSP** (*Define Secondary Structure of Proteins* - Definir Estrutura Secundária de Proteínas) [99] e adota onze estados conformacionais que podem ser assumidos pelos resíduos de aminoácidos de uma proteína, a saber: 'H'; 'h' ; 'E' ; 'e' ; 'G' ; 'g' ; 'T' ; 't' ; 'S' ; 'B' ; 'b' (ver Tabela 5.1). Para uma descrição completa de cada estado conformacional e do algoritmo DSSP, verificar Kabsh e Sander [99]. O PROMOTIF também é utilizado para calcular os ângulos de torção (ϕ , ψ and ω) de cada fragmento usado como modelo. Por exemplo, para o fragmento **SEKMT** da proteína com identificador PDB número 1L2Y, a estrutura secundária e os ângulos de torção calculados são os seguintes:

Estrutura secundária: HHHHH

Ângulos de torção:

S: [67,5, 172,8, -179.9]

E: [-59,6, -19,4, 179.9]

K: [-66,4, -61,7, 179.8]

M: [-64,1, -17,9, 180.0]

T: [-78,3, -33,7, 179.9]

4. **Agrupamento de ângulos de torção:** todos os ângulos dos modelos são calculados e agrupados através do método **EM** (*Expectation Maximization* – Maximização de Expectativa) [100]. Isso permite a identificação das regiões onde os ângulos estão concentrados quando apresentados no diagrama de Ramachandran [101].
5. **Construção dos padrões estruturais:** As informações sobre a estrutura secundária (Tabela 5.1), a sequência de aminoácidos (Tabela 5.2) e os ângulos de torção (ϕ and ψ) do resíduo central de cada fragmento modelo são utilizados para construir os padrões de treinamento, que tem a forma definida pela Eq. 5.3, onde SS_i representa a estrutura secundária do i -ésimo resíduo do fragmento; a_i representa o i -ésimo resíduo e t representa um dos ângulos de torção ϕ ou ψ .

$$[SS_1, SS_2, SS_3, SS_4, SS_5 + a_1, a_2, a_3, a_4, a_5] : t \quad (5.3)$$

A informação sobre a estrutura secundária concatenada às identificações dos resíduos formam a amostra a ser apresentada para a rede durante o seu treinamento. O ângulo (ϕ ou ψ) configura a saída desejada da rede (alvo) para aquela

amostra. Dois conjuntos de treinamentos são construídos: (a) um conjunto contendo como alvos os ângulos ϕ e; (b) um conjunto contendo como alvos os ângulos ψ . Todos os fragmentos modelo obtidos do PDB são utilizados. A Fig. 5.5 ilustra a construção do padrão estrutural. A Tabela 5.1 mostra o esquema para converter as informações da estrutura secundária em valores numéricos. A Tab. 5.2 apresenta o esquema adotado para converter os códigos de uma letra que identificam os aminoácidos em valores numéricos.

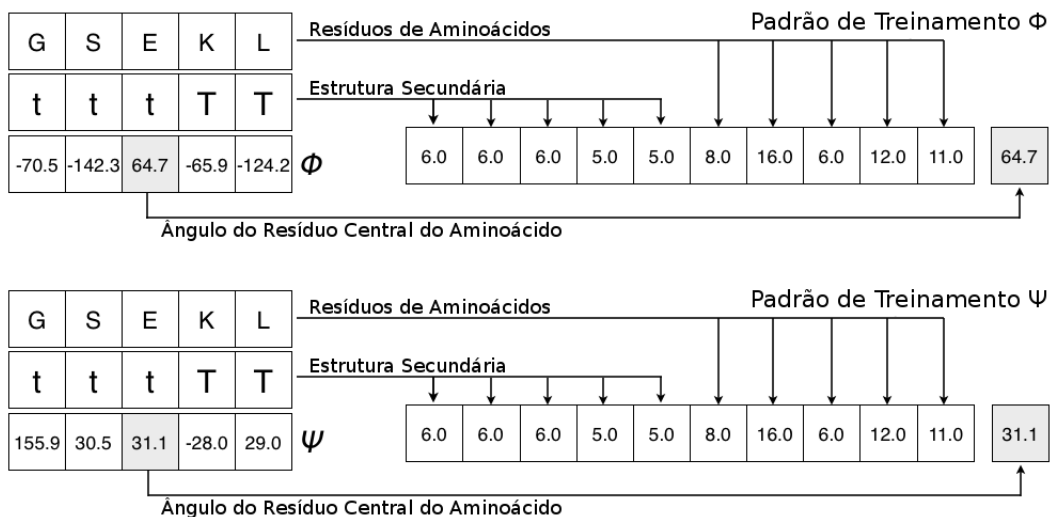


Figura 5.5: Construindo as amostras de treinamento. A estrutura secundária, os identificadores dos resíduos e os ângulos de torção são usados. Dois conjuntos são criados, um para os ângulos ϕ e outro para os ângulos ψ . Baseada em [52].

- 6. Construção e treinamento das redes GMDH:** Para cada fragmento alvo s_i obtido no passo 1, duas redes GMDH são construídas. Uma é utilizada para prever os ângulos ϕ e a outra para prever os ângulos ψ do resíduo central do fragmento (Fig. 5.6).
- 7. Predição das estruturas estereoquímicas aproximadas da sequência alvo:** As amostras para predição são montadas com as informações obtidas a partir dos fragmentos desconhecidos e submetidas às duas redes treinadas. As saídas dessas redes representam os ângulos de torção preditos ϕ e ψ apresentados pelo resíduo central de cada fragmento. Para construir a amostra desconhecida, a estrutura secundária da sequência alvo é calculada usando um servidor disponível *on-line* chamado **SCRATCH** [102]. A sequência de caracteres que representa a estrutura secundária da sequência alvo K forma um fragmento. As informações sobre resíduos de aminoácidos de K são combinadas com a estrutura secundária correspondente, montando o padrão a ser submetido às redes. A Fig. 5.6 ilustra o processo.

Tabela 5.1: Estados conformacionais. Os caracteres indicam a sequência das unidades estruturais secundárias como fornecidas pelo PROMOTIF. Um asterisco (*) indica um resíduo extra que representa um aminoácido pertencente a uma região em que a estrutura principal onde proteína começa a ter modificado o seu estado conformacional. A terceira coluna apresenta o código de conversão para se construir as amostras conformacionais.

Estado	Estrutura secundária	Código de conversão
H	α -hélice	1,0
h	α -hélice*	2,0
E	folha- β	3,0
e	folha- β *	4,0
T	volta- β	5,0
t	volta- β *	6,0
G	hélice-3 ¹⁰	7,0
g	hélice-3 ^{10*}	8,0
S	curva	9,0
B	curva	10,0
b	curva*	11,0

O esquema de fragmentação com cinco resíduos causa uma perda das informações sobre o primeiro e o último aminoácido da sequência alvo. O par de ângulos de torção para cada um desses resíduos é definido arbitrariamente em 180°. Após a predição de todos os ângulos de torção da sequência, a estrutura estereoquímica é construída utilizando-se as rotinas do pacote de modelagem molecular **TINKER** [103, 104] e o modelo de campos de força de ácidos nucleicos **CHARMM27** [105].

5.4 RESULTADOS EXPERIMENTAIS

O método descrito na seção 5.3 foi aplicado para predizer as estruturas de sete sequências de proteínas obtidas do PDB [79]. Os códigos identificadores dessas sequências são: 1A11 [106] (Fig. 5.7-A), 1ZDD [107] (Fig. 5.7-B), 1ACW [108] (Fig. 5.7-C), 1ENH [109] (Fig. 5.7-D), 1K43 [110] (Fig. 5.7-E), 1L2Y [111] (Fig. 5.7-F) e 1PJE [112] (Fig. 5.7-G). A Tabela 5.3 apresenta as sete sequências usadas nos estudos de casos.

Várias estimativas diferentes de cada estrutura da Tab. 5.3 foram realizadas utilizando-se o procedimento descrito na seção 5.2. As estruturas estimadas foram avaliadas através da similaridade com as estruturas originais obtidas do PDB. As medidas de qualidade das estimativas foram feitas em termos do desvio do valor quadrático médio

Tabela 5.2: Os vinte resíduos de aminoácidos. A terceira coluna lista o código de conversão adotado para construir as amostras para treinamento da rede.

Abreviação	Nome	Valor convertido
A	Alanina	1,0
R	Arginina	2,0
N	Asparagina	3,0
D	Aspartato	4,0
C	Cisteína	5,0
E	Glutamato	6,0
Q	Glutamina	7,0
G	Glicina	8,0
H	Histidina	9,0
I	Isoleucina	10,0
L	Leucina	11,0
K	Lisina	12,0
M	Metionina	13,0
F	Fenilalanina	14,0
P	Prolina	15,0
S	Serina	16,0
T	Treonina	17,0
W	Triptofano	18,0
Y	Tirosina	19,0
V	Valina	20,0

Tabela 5.3: Sequências alvo

PDB ID	Sequência Primária	Resíduos
1A11	GSEKMSTAISVLLAQAVFLLLSQR	25
1ZDD	FNMQCRRRFYEALHPNLNEEQRNAKIKSIRDCC	34
1ACW	VSCEDCPEHCSTQKAQAKCDNDKCVEPI	29
1ENH	RPRTAFSSEQLARLKREFNENRYLTERRRQQLSSELGLNEAQIKIWFQNKRAKI	54
1K43	RGKWTYNGITYEGR	14
1L2Y	NLYIQWLKGGPSSGRPPPS	20
1PJE	MQPIQIAIVALVVAIIIAIVVWSIVIIIEGRGGKKKK	36

(RMSD - *Root Mean Square Deviation*) entre as posições dos átomos nas estruturas experimentais e estimadas, calculado segundo a Eq. 5.4, onde $\mathbf{r}_{A,i}$ and $\mathbf{r}_{B,i}$ são vetores representando as posições do i -ésimo átomo nas duas estruturas, \mathbf{A} e \mathbf{B} , respectiva-

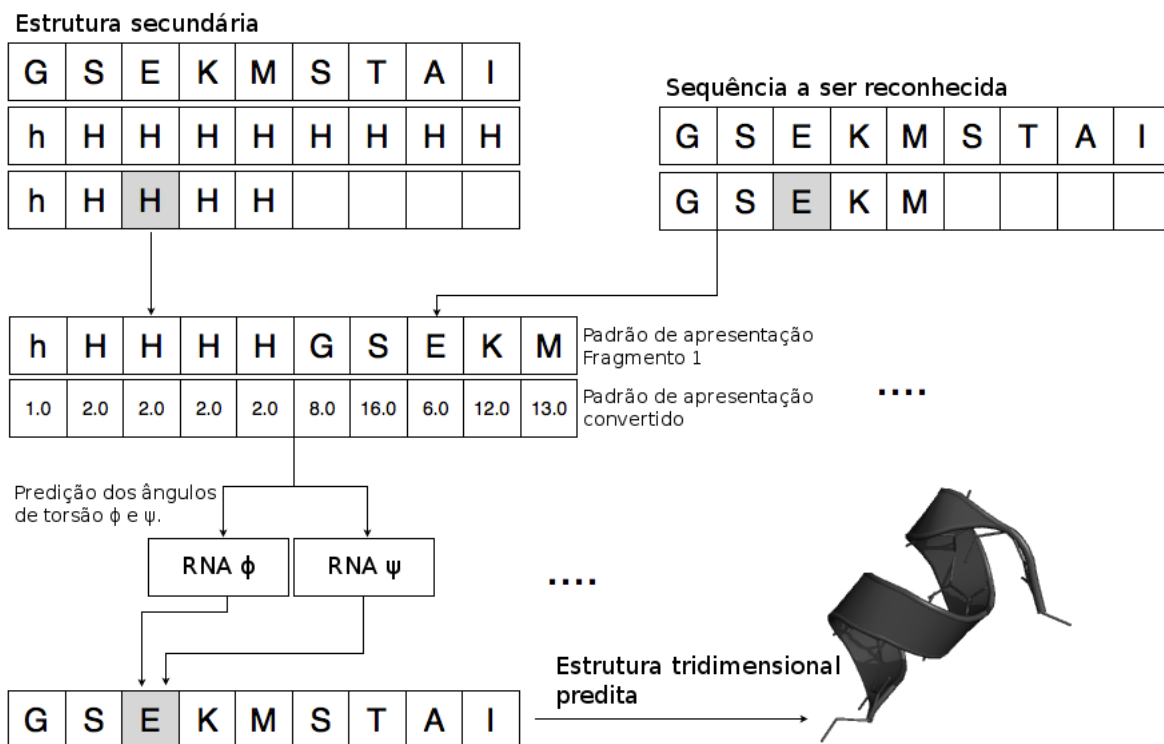


Figura 5.6: Predição dos ângulos de torção ϕ e ψ do aminoácido central de cada fragmento da sequência alvo k . Os resíduos do fragmento são concatenados com as informações da estrutura secundária para construir o padrão a ser apresentado para as redes GMDH. Baseada em [52].

mente, sobrepostas de maneira ótima.

$$\text{RMSD}(\mathbf{A}, \mathbf{B}) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{r}_{A,i} - \mathbf{r}_{B,i}\|^2} \quad (5.4)$$

O RMSD calculado entre as duas estruturas é obtido através do *software* **PROFIT** [114]. A Tab. 5.4 apresenta os valores obtidos de cada proteína alvo utilizando redes MLP, C-GMDH e O-GMDH.

Para cada uma das sete sequências de aminoácidos, três versões do método foram implementadas: uma usando o modelo O-GMDH, outra usando o C-GMDH com polinômios de graus crescentes (ver seção 3.4) e uma versão de referência utilizando redes neurais do tipo MLP, totalizando vinte e um experimentos.

Os experimentos se compuseram de vinte diferentes ensaios nos quais as amostras foram montadas com fragmentos modelares extraídos do PDB e usados para treinar as RNAs. Em cada ensaio, as amostras foram embaralhadas e, em seguida, divididas em

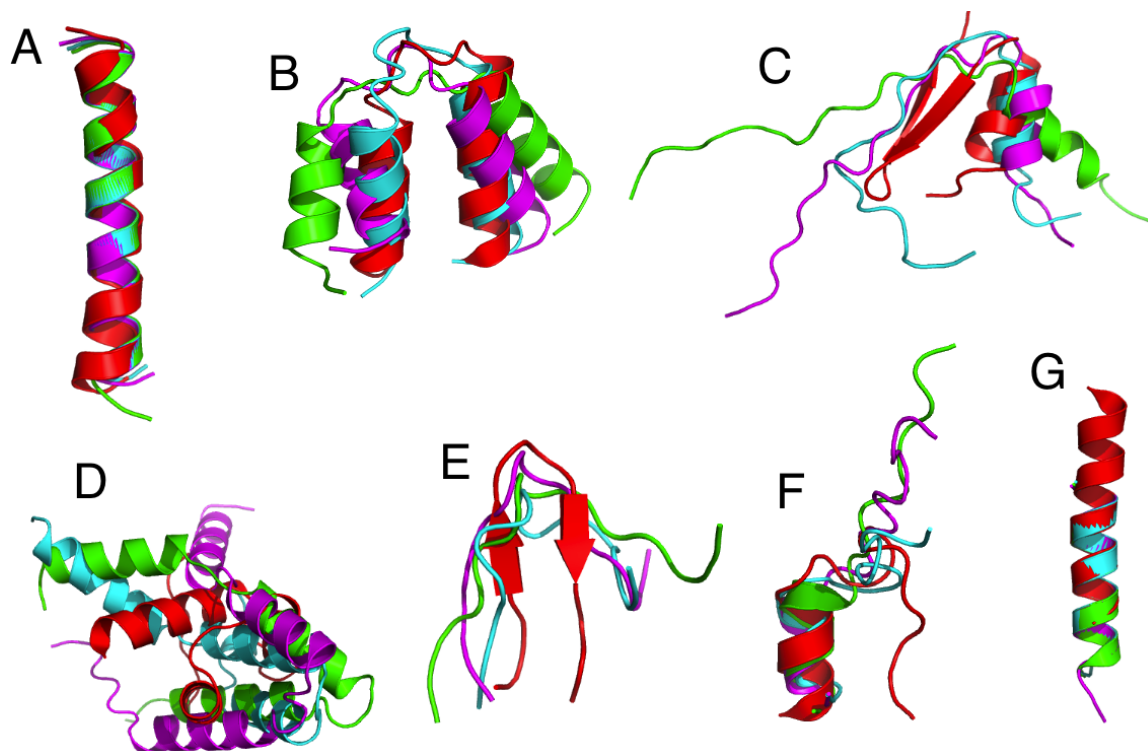


Figura 5.7: Representação em fita das estruturas obtidas pelo método experimental (em vermelho) e as estruturas preditas: MLP (em ciano), O-GMDH (em magenta) e C-GMDH (verde). Os carbonos-alfa (C_{α}) das estruturas experimentais e preditas estão alinhados. (A) Estruturas experimental e preditas da proteína com identificador PDB = 1A11; (B) Estruturas com identificador PDB = 1ZDD; (C) Identificador PDB = 1ACW; (D) Identificador PDB = 1ENH; (E) Identificador PDB = 1K43; (F) Identificador PDB = 1L2Y; (G) Identificador PDB = 1PJE. As cadeias paralelas de aminoácidos não são apresentadas para melhor clareza. Os gráficos foram gerados pela ferramenta **PYMOL** [113]. Fonte: [52].

dois grupos: (a) treinamento e (b) seleção, de acordo com os requisitos do treinamento de redes GMDH. Devido ao embaralhamento, cada ensaio gerou um valor de RMSD diferente. Nos ensaios com redes MLP, a separação entre os dados (treinamento, validação e teste) foi deixada a cargo da *toolbox* de redes neurais do MATLAB [115]. Os ensaios baseados em MLP também produziram diferentes resultados de RMSD. Para analisar a variabilidade das soluções, foram utilizados os valores mínimos, as médias e os desvios padrões das distribuições de RMSDs de cada experimento. Os resultados podem ser observados na Tab. 5.4 assim nas figuras 5.8, 5.9 e 5.10.

A Fig. 5.7 mostra a representação em fitas das estruturas experimentais e preditas, produzidas pelo aplicativo **PYMOL** [113]. A inspeção visual mostra que as hélices individuais e as estruturas secundárias estão bem formadas em todos os estudos de caso.

RMSDs Mínimos

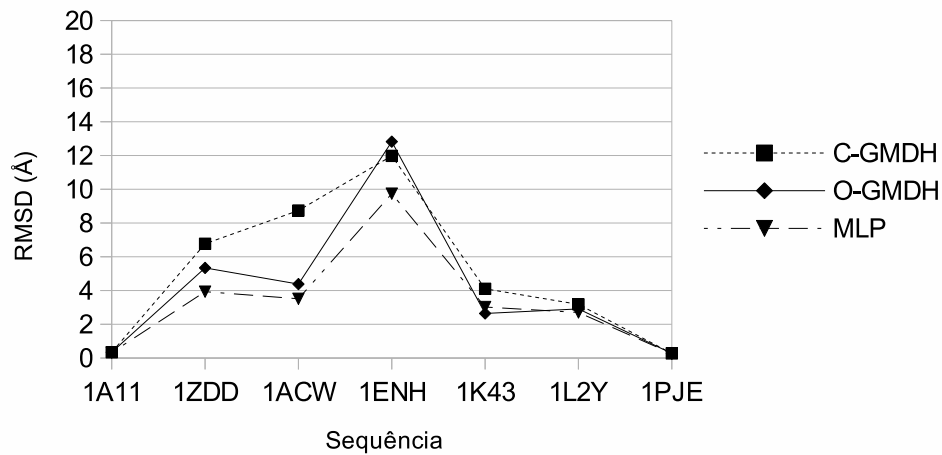


Figura 5.8: Valores mínimos dos RMSDs de cada experimento. Baseada em [52].

RMSDs Médios

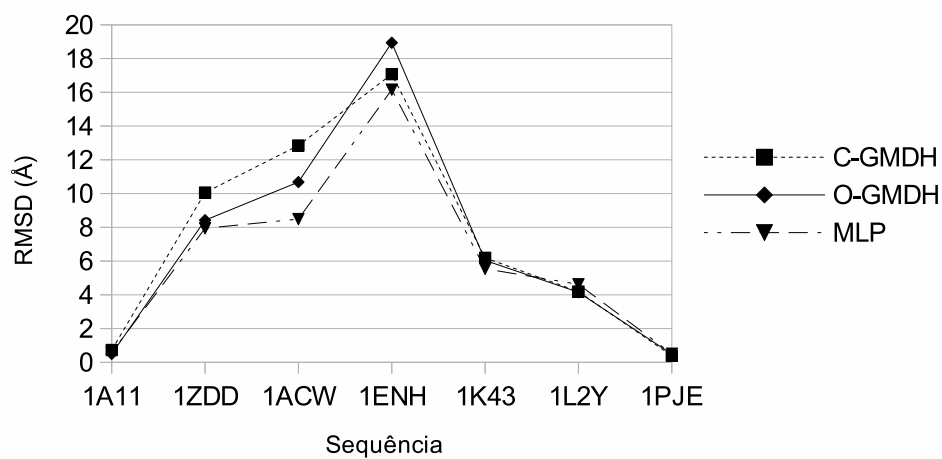


Figura 5.9: Valores médios dos RMSDs de cada experimento. Baseada em [52].

Tabela 5.4: RMSD dos carbonos alfa das estruturas preditas em relação às estruturas experimentais

Identificador	RNA	Resíduos (i-j)	RMSD Mínimo (Å)	RMSD Médio (Å)	σ (Å)
1A11	C-GMDH	3-23	0,34	0,73	0,45
1A11	O-GMDH	3-23	0,34	0,50	0,18
1A11	MLP	3-23	0,32	0,58	0,36
1ZDD	C-GMDH	3-32	6,77	10,06	1,48
1ZDD	O-GMDH	3-32	5,34	8,42	1,71
1ZDD	MLP	3-32	3,93	7,94	1,89
1ACW	C-GMDH	3-27	8,74	12,85	1,82
1ACW	O-GMDH	3-27	4,38	10,68	3,07
1ACW	MLP	3-27	3,51	8,49	3,27
1ENH	C-GMDH	3-52	11,98	17,08	2,90
1ENH	O-GMDH	3-52	12,82	18,24	2,52
1ENH	MLP	3-52	9,74	16,16	2,97
1K43	C-GMDH	3-12	4,10	6,19	0,89
1K43	O-GMDH	3-12	2,64	6,04	1,29
1K43	MLP	3-12	3,02	5,54	1,15
1L2Y	C-GMDH	3-17	3,18	4,19	0,89
1L2Y	O-GMDH	3-17	2,91	4,15	0,83
1L2Y	MLP	3-17	2,70	4,62	1,12
1PJE	C-GMDH	3-16	0,27	0,39	0,12
1PJE	O-GMDH	3-16	0,27	0,48	0,17
1PJE	MLP	3-16	0,26	0,51	0,24

Os valores de RMSD resultantes das diferentes abordagens são similares (ver Tab. 5.4). É possível observar que as versões baseadas em MLP do método apresentado produzem resultados ligeiramente melhores e, ainda, que os experimentos com O-GMDH são mais precisos que os baseados em C-GMDH. Os gráficos dos mínimos e das médias (Figuras 5.8 e 5.9) caracterizam os resultados e mostram que, entre os experimentos com GMDH, aqueles baseados em O-GMDH realizaram melhor as tarefas. A Fig. 5.10 mostra que os RMSDs obtidos com O-GMDH e MLP produziram curvas de desvios padrão similares, sugerindo que as soluções são compatíveis em termos de dispersão. As implementações com C-GMDH forneceram desvios padrão ligeiramente menores para a maioria dos casos e um desvio significativamente menor para a sequência com identificador 1ACW apesar de que seus valores mínimos de RMSD tenham sido maiores do que os produzidos pelas outras RNAs na maioria dos casos.

Nos estudos para a sequências 1A11, 1L2Y e 1PJE, os três modelos de rede neural

Desvios Padrão dos Valores de RMSD

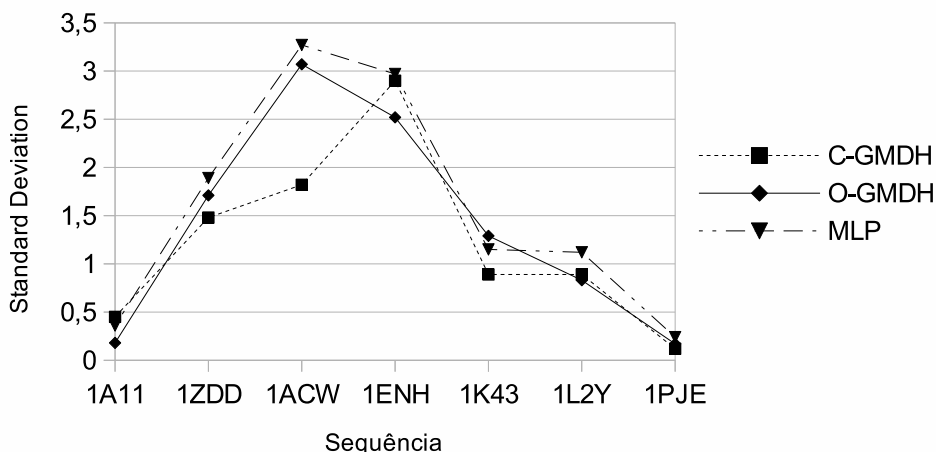


Figura 5.10: Desvios padrão entre os RMSDs dos ensaios de cada experimento. Baseada em [52].

apresentaram RMSDs similares, em cada sequência: para 1A11, foram próximos de 0,3 Å, para 1L2Y, 3,0 Å e para 1PJE, próximos de 0,27 Å. Os estudos realizados com a sequência 1ENH apresentaram os maiores valores de RMSD, a saber: C-GMDH, 11,98 Å; O-GMDH, 12,82 Å; MLP, 9,74 Å. Este resultado era de certa forma esperado já que a proteína possui um padrão de enovelamento mais complexo quando comparado com as utilizadas nos outros experimentos.

Análises das disposições estruturais secundárias das estruturas preditas (Fig. 5.7 A-G – verdes, cianos e magentas) em relação às estruturas experimentais (Fig. 5.7 A-G – vermelhos) foram realizadas utilizando o PROMOTIF. A Tab. 5.5 mostra o número (%) de resíduos de aminoácidos que ocorrem em um dos estados conformacionais (estados conformacionais regulares: folhas- β , α -hélices, hélices-3¹⁰; estados conformacionais irregulares: regiões de voltas ou de laços). As formações das estruturas secundárias preditas são muito próximas das apresentadas pelas estruturas experimentais. As menores diferenças ocorrem nas regiões onde as estruturas têm modificados seu estados conformacionais (α -hélices ou folhas- β - para regiões de voltas ou laços).

As estruturas preditas para a proteína 1K43 (Fig. 5.7-E — verde, ciano e magenta) apresentam as maiores diferenças nos valores dos estados conformacionais quando comparadas com a estrutura experimental. Isso ocorre porque a disposição da região em forma de *volta* do peptídeo não permite que os filamentos-*beta* se arranjam de forma que os grupos $N-H$ na estrutura principal de uma sequência estabeleçam ligações de hidrogênio com os grupos $C=O$ na estrutura principal de uma sequência adjacente.

Tabela 5.5: Análises das disposições das estruturas secundárias das conformações previstas. P denota uma estrutura prevista. E denota uma estrutura experimental.

PDB	Alg.	Folha	α -helix	3^{10}	Outras
1A11-E	-	0,0	92,0	0,0	8,0
1A11-P	C-GMDH	0,0	80,0	0,0	20,0
1A11-P	O-GMDH	0,0	84,0	0,0	16,0
1A11-P	MLP	0,0	84,0	0,0	16,0
1ZDD-E	-	0,0	73,5	0,0	26,5
1ZDD-P	C-GMDH	0,0	64,7	0,0	35,3
1ZDD-P	O-GMDH	0,0	70,6	0,0	29,4
1ZDD-P	MLP	0,0	64,7	0,0	35,3
1ACW-E	-	34,5	24,1	0,0	41,4
1ACW-P	C-GMDH	0,0	20,7	0,0	79,3
1ACW-P	O-GMDH	0,0	20,7	0,0	79,3
1ACW-P	MLP	0,0	17,2	0,0	82,8
1ENH-E	-	0,0	70,4	0,0	29,6
1ENH-P	C-GMDH	0,0	68,5	0,0	31,5
1ENH-P	O-GMDH	0,0	66,7	0,0	33,3
1ENH-P	MLP	0,0	66,7	7,4	25,9
1K43-E	-	42,9	0,0	0,0	57,1
1K43-P	C-GMDH	0,0	0,0	100,0	0,0
1K43-P	O-GMDH	0,0	0,0	0,0	100,0
1K43-P	MLP	0,0	0,0	0,0	100,0
1L2Y-E	-	0,0	35,0	20,0	45,0
1L2Y-P	C-GMDH	0,0	36,8	0,0	63,2
1L2Y-P	O-GMDH	0,0	21,1	0,0	78,9
1L2Y-P	MLP	0,0	36,8	15,8	47,0
1PJE-E	-	0,0	84,2	0,0	15,8
1PJE-P	C-GMDH	0,0	83,3	0,0	16,7
1PJE-P	O-GMDH	0,0	77,8	0,0	22,2
1PJE-P	MLP	0,0	83,3	0,0	16,7

Entretanto, essa pequena distorção pode ser corrigida através de métodos mecânicos [116, 117] que podem simular e otimizar as interações entre todos os átomos, o que permite a formação de ligações de hidrogênio que estabilizarão aquelas regiões.

O *software* PROCHECK [118] foi utilizado para obter as informações sobre as posições

dos resíduos de aminoácidos na distribuição de Ramachandran. Esses resultados estão sumarizados na Tabela 5.6. A distribuição de Ramachandran estabelece regiões de posicionamento das moléculas nas proteínas, baseadas nos ângulos ϕ e ψ . Moléculas nas regiões denominadas *mais favoráveis* são aquelas em que seus ângulos não as sujeitam a choques com outras moléculas. Além desta, os resíduos podem se encontrar nas regiões *adicionalmente permitidas*, *geralmente permitidas* e *não permitidas*. A qualidade das predições das estruturas terciárias pode ser avaliada também por este critério.

Nas estruturas preditas com GMDH, os resíduos de todos os experimentos que foram encontrados nas regiões *favoráveis* ou *adicionalmente permitidas* totalizam 95,6 %, para os métodos baseados em O-GMDH, e 94,5 % para aqueles que utilizaram C-GMDH. Por outro lado, os experimentos baseados em MLP apresentaram resultados onde 92,4 % dos resíduos de todas as proteínas foram encontrados nas regiões *mais favoráveis* e *adicionalmente permitidas*. Os métodos experimentais encontraram 99,4 % dos resíduos nessas duas regiões.

Uma grande quantidade de resíduos em uma região favorável significa que as estruturas têm um pequeno número de maus contatos. As estruturas obtidas com C-GMDH e O-GMDH são similares às obtidas pelos métodos experimentais. Os resultados obtidos com o PROMOTIF e com o PROCHECK demonstram que os elementos helicoidais nas estruturas preditas são bem formados e têm estruturas que se assemelham às experimentais.

5.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou o uso de redes GMDH para a predição de estruturas de polipeptídeos até o terceiro nível. O método se inicia com a fragmentação das proteínas em pequenas sequências com cinco resíduos de aminoácidos. A seguir, dados conformacionais de sequências proteicas similares aos fragmentos da proteína analisada são extraídos da Base de Dados de Proteínas – PDB. As estruturas secundárias e os ângulos de torção são calculados utilizando-se o *software* PROMOTIF e as similaridades entre as sequências são determinadas por alinhamentos de sequências utilizando-se a ferramenta BLASTp.

Os padrões estruturais são montados com as informações obtidas e utilizados para treinar redes GMDH com o intuito de identificar os ângulos de torção ϕ e ψ de cada

Tabela 5.6: Valores numéricos do diagrama de Ramachandran para as conformações experimentais e preditas. P denota uma estrutura predita. E denota uma estrutura experimental.

PDB	Algoritmo	Mais Fav.	Adic. Perm.	Geralm. Perm.	Não Perm.
1A11-E	-	91,3	4,3	4,3	0,0
1A11-P	C-GMDH	87,0	13,0	0,0	0,0
1A11-P	O-GMDH	93,5	6,5	0,0	0,0
1A11-P	MLP	91,3	8,7	0,0	0,0
1ZDD-E	-	87,1	12,9	0,0	0,0
1ZDD-P	C-GMDH	71,0	19,4	6,5	3,2
1ZDD-P	O-GMDH	80,6	12,9	6,5	0,0
1ZDD-P	MLP	74,2	16,1	9,7	0,0
1ACW-E	-	93,5	6,5	0,0	0,0
1ACW-P	C-GMDH	56,0	36,0	8,0	0,0
1ACW-P	O-GMDH	68,0	28,0	4,0	0,0
1ACW-P	MLP	69,0	32,0	4,0	0,0
1ENH-E	-	100,0	0,0	0,0	0,0
1ENH-P	C-GMDH	78,0	12,0	8,0	2,0
1ENH-P	O-GMDH	76,0	24,0	0,0	0,0
1ENH-P	MLP	90,0	10,0	0,0	0,0
1K43-E	-	93,5	6,5	0,0	0,0
1K43-P	C-GMDH	88,9	0,0	11,1	0,0
1K43-P	O-GMDH	55,6	33,3	11,1	0,0
1K43-P	MLP	33,3	22,2	33,3	11,1
1L2Y-E	-	90,9	9,1	0,0	0,0
1L2Y-P	C-GMDH	63,6	36,4	0,0	0,0
1L2Y-P	O-GMDH	45,5	45,5	9,1	0,0
1L2Y-P	MLP	63,6	36,4	0,0	0,0
1PJE-E	-	100,0	0,0	0,0	0,0
1PJE-P	C-GMDH	93,3	6,7	0,0	0,0
1PJE-P	O-GMDH	93,3	6,7	0,0	0,0
1PJE-P	MLP	93,3	6,7	0,0	0,0

fragmento. Esses ângulos são utilizados para construir toda a estrutura da proteína com o auxílio das rotinas do pacote de modelagem molecular TINKER.

Os experimentos revelaram que o método utilizando GMDH pode fornecer predições corretas e que resultam em estruturas secundárias e terciárias com formas e disposições

próximas às obtidas pelos métodos experimentais. A conformação completa das estruturas secundárias não pôde ser alcançada em algumas das predições, entretanto, elas se apresentaram compatíveis com os valores experimentais. Essas imprecisões podem ser perfeitamente ajustadas ou corrigidas por protocolos de refinamento bem projetados através de métodos de mecânica molecular, particularmente através de simulações MD [119, 117, 120, 121]. As estruturas aproximadas obtidas com o método proposto podem reduzir o tempo total demandado por métodos *ab initio*, que normalmente se iniciam a partir da conformação totalmente estendida do polipeptídeo.

A contribuição global deste método tem três aspectos:

1. O uso do algoritmo GMDH para desenvolver uma nova estratégia para um problema biológico relevante: a predição de estruturas estereoquímicas de proteínas;
2. A comparação do modelo GMDH de redes neurais com o já estabelecido modelo Perceptron de camadas múltiplas – MLP;
3. A predição de estruturas de proteínas de diversas classes (SCOP).

Devido ao exposto, considera-se que esta investigação abre o caminho para a aplicação de redes GMDH no âmbito das aplicações de biologia computacional e bioinformática.

No que se refere ao estudo das redes GMDH em si, percebeu-se que a variação C-GMDH do método produziu resultados menos significativos do que o algoritmo original para o problema estudado. Em teoria, o C-GMDH permite uma maior interação entre os nós de diferentes camadas, o que aumentaria a busca no espaço de soluções. Entretanto, à época da realização dos experimentos, esse método não se encontrava plenamente implementado na ferramenta *GMDH Box*. Em particular, a expressão utilizada nos neurônios C-GMDH nesses experimentos, apesar de ser quadrática, ou seja, de representar uma inovação em relação ao C-GMDH proposto por Ivakhnenko (ver seções 2.4 e 3.4), era formada por polinômios incompletos.

Por exemplo, um neurônio com três entradas conteria, no modelo C-GMDH de Ivakhnenko, um polinômio com 8 termos, como visto na Eq. 2.19. Com a introdução das inovações já implementadas na ferramenta, um neurônio C-GMDH com três entradas poderia apresentar um polinômio do terceiro grau com treze termos, no formato da Eq. 3.4.

Nos experimentos deste capítulo, as expressões C-GMDH eram polinômios com sete termos como mostra a Eq. 5.5.

$$y = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2 + a_4x_2^2 + a_5x_3 + a_6x_3^2 \quad (5.5)$$

É possível que os resultados um tanto piores obtidos com o C-GMDH tenham sido causados pelos polinômios mais limitados que compunham as descrições parciais, em contraste com os aplicados nos neurônios O-GMDH.

Utilizar neurônios completos pode proporcionar uma busca mais apurada sobre o espaço de soluções. Alguns experimentos simplificados, feitos após os estudos apresentados neste capítulo, baseados em novas versões do *GMDH Box*, sugerem que redes C-GMDH com polinômios completos (mesmo os de primeira ordem) apresentam melhores resultados do que redes de descrições parciais formadas por expressões quadráticas incompletas, com uma pequena contrapartida relacionada a um ligeiro aumento da rede após o treinamento. Somente a realização de novos experimentos poderá determinar se isso é uma tendência geral ou se pode ser verdade para alguns casos e não ser verdade para outros, ou mesmo se não há qualquer contribuição efetiva.

6 IMPLEMENTAÇÃO DE GMDH EM *HARDWARE* USANDO FPGAs

6.1 CONSIDERAÇÕES INICIAIS

A implementação de redes neurais artificiais em computadores de uso geral foi levada a um nível avançado em termos de desempenho e precisão. Entretanto, redes neurais, especialmente no que se refere ao seu treinamento, não são facilmente aplicadas em *sistemas embarcados*, pois estes têm restrições importantes de memória e consumo de energia, sendo regidos por relógios a baixa frequência (geralmente menores que 500 MHz), apresentando desta maneira limitações no seu poder computacional. Contudo, a adequação para esse tipo de plataforma é muito importante, pois ela representa mais de 90% dos sistemas computacionais existentes no mundo [122]. Este capítulo mostra os resultados iniciais do desenvolvimento de uma arquitetura GMDH em *hardware* reconfigurável, tanto nas fases de treinamento como de execução. A arquitetura foi desenvolvida e testada em uma plataforma de co-projeto de *hardware/software*.

Para realizar o teste da plataforma, utilizou-se uma versão simplificada do problema de reconhecimento de estruturas de proteínas apresentado no capítulo 5. Um conjunto de experimentos foi realizado em um PC utilizando um FPGA como um coprocessador acessado através de *sockets* TCP/IP. Embora apenas um único elemento processador GMDH tenha sido realizado em um FPGA, a metodologia empregada demonstrou que é possível implementar redes GMDH em *hardware* para serem utilizadas em sistemas embarcados. Adicionalmente, esta aplicação demonstrou que um dos passos mais importantes do algoritmo, o cálculo dos coeficientes das descrições parciais, pode ser efetivamente implementado em *hardware*, o que foi alcançado utilizando-se um projeto de um sistema solucionador de equações lineares, desenvolvido anteriormente. Os experimentos demonstraram que a implementação é eficaz para encontrar soluções de boa qualidade para o problema em avaliação e seus resultados foram publicados em uma conferência internacional sob o título *Hardware Implementation of GMDH-Type Artificial Neural Networks and its use to Predict Approximate Three-dimensional Structures of Proteins* [53].

Diversas implementações de RNAs têm sido propostas para lidar com problemas como:

(a) identificação de sistemas dinâmicos [123], (b) compressão de imagens digitais [124, 125], (c) solução de problemas de otimização [126], dentre muitos outros. Em geral, essas abordagens demonstram a adequabilidade do uso de FPGAs para a solução de questões relacionadas a sistemas inteligentes, caracterizados por sua alta complexidade e que, portanto, requerem um grande esforço computacional para que se produzam resultados práticos aproveitáveis. Desta forma, a utilização de *hardware* dedicado para a computação de sistemas evolutivos e neurais vem se tornando uma questão chave para os projetistas [127].

Um tema dominante para qualquer implementação de RNAs em FPGAs é o compromisso entre consumo de recursos, precisão e desempenho. Uma característica problemática para o desenvolvimento de redes neurais em *hardware* é a arquitetura da função de ativação [128], que é responsável por restringir ou modular as saídas dos neurônios em diversos modelos de RNAs. A função de ativação pode tomar diversas formas, como, por exemplo, a de uma tangente hiperbólica ($\varphi(v) = \tanh(v)$). Existem diferentes alternativas para o projeto dessas funções, como (a) LUTs (*Look-Up Tables* - Tabelas de Consulta), que podem consumir uma grande quantidade de blocos de memória ou; (b) aproximação linear por partes, isto é, a aproximação da saída das funções não lineares através de uma sequência finita de saídas lineares.

Outra matéria importante é o nível de paralelismo empregado. A implementação de muitos elementos processadores (neurônios) requer uma grande quantidade de recursos de *hardware*. Além disso, à medida que o número de neurônios cresce, mais interconexões entre eles são necessárias, o que também limita o tamanho dos projetos. Apesar de que o paralelismo intrínseco às redes neurais possa sugerir a sua implementação em *hardware* (o que permitiria explorar esse paralelismo), na realidade o que está implícita é a ideia de que as implementações em *hardware* serão mais rápidas do que as implementações em *software*. O foco no paralelismo se justifica apenas quando o alto desempenho é atingido com um custo razoável [129]. Como custo, entretanto, é possível considerar fatores outros além do valor monetário. Para sistemas embarcados, por exemplo, especialmente aqueles que funcionam à bateria, um “custo” a ser avaliado é a potência dissipada e, mais ainda, a energia consumida para realizar-se uma tarefa. Mesmo considerando-se que FPGAs apresentam um alto consumo estático de energia, isto é, energia utilizada apenas para mantê-los ligados, soluções baseadas em *hardware* especializado podem dissipar menos energia quando realizam cálculos intensos como os presentes no processamento de RNAs. A questão da dissipação de potência em implementações de RNAs em *hardware* foi avaliado em trabalho anterior, publicado em

conferência internacional com o título *Performance, accuracy, power consumption and resource utilization analysis for hardware/software realized Artificial Neural Networks* [59].

A implementação computacional de redes neurais artificiais do tipo GMDH é relativamente simples porque seus neurônios dispensam o uso de funções de ativação. Como visto no capítulo 2, as descrições parciais são formadas por polinômios, o que é vantajoso para a implementação em *hardware*, já que exigem apenas operações matemáticas de implementação relativamente fácil: adições, subtrações, multiplicações e potências com expoentes inteiros positivos (que podem ser calculados por séries de multiplicações).

Para o trabalho apresentado neste capítulo, foram desenvolvidas algumas bibliotecas que permitem que o *GMDH Box* utilize um neurônio O-GMDH implementado em um FPGA, juntamente com um Módulo de Solução de Sistemas Lineares (**LSSM** - *Linear Systems Solving Module*) [130], que usa a eliminação gaussiana para realizar o cálculo dos coeficientes das descrições parciais. Tanto o neurônio GMDH como o módulo LSSM foram escritos em VHDL e testados em FPGA.

Os experimentos foram realizados com o neurônio O-GMDH implementado no dispositivo, acessado via protocolo TCP/IP, enquanto o módulo LSSM foi executado na ferramenta de simulação ModelSim [131] da empresa Mentor Graphics. A comunicação com o ModelSim foi abstraída através de modelagem ao nível de transação (*transaction level modelling*). A comunicação via TCP/IP foi realizada utilizando-se o processador MicroBlaze [132] da Xilinx, com periféricos a ele conectados através de um barramento PLB (*Processor Local Bus*) [133].

6.2 MÓDULOS IMPLEMENTADOS EM *HARDWARE*

As duas operações mais importantes das redes GMDH foram implementadas em FPGA: (a) o processamento dos neurônios, usado na execução normal das redes, e (b) o cálculo dos coeficientes das descrições parciais, necessário para o treinamento das redes.

Um diagrama do neurônio O-GMDH em *hardware* é apresentado na Fig. 6.1. No lado esquerdo do diagrama estão representadas duas unidades FIFO (*First In First Out*). Elas são responsáveis pela recepção dos pesos e das entradas do neurônio. Os controladores das FIFOs, conectados a suas entradas, são responsáveis por ler e paralelizar os

dados recebidos de acordo com os requisitos do elemento processador e por armazená-los nos locais apropriados dentro do banco de registradores. Para os dados de entrada, o controlador responsável lê dois valores da FIFO de entrada e os armazena.

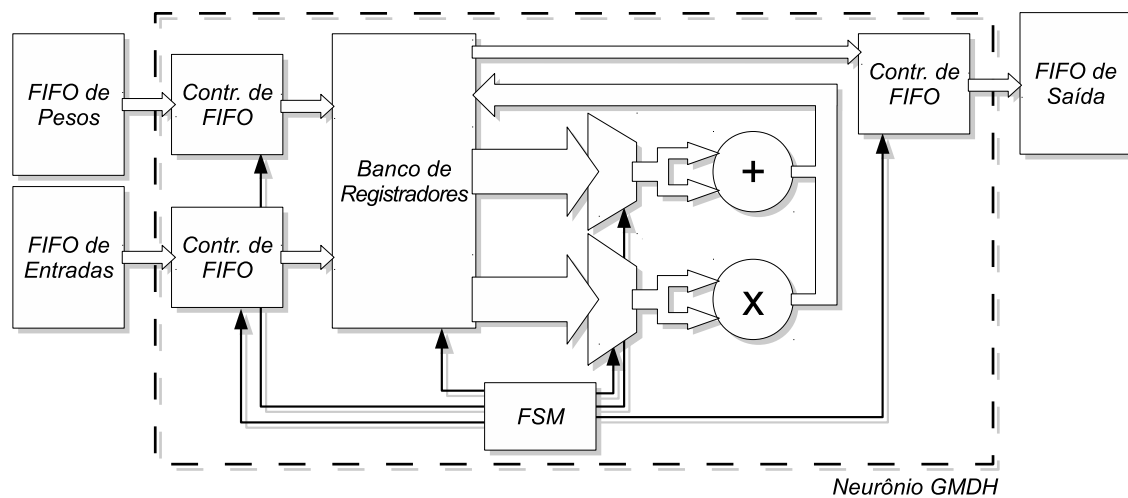


Figura 6.1: Diagrama do neurônio GMDH em *hardware*. Baseada em [53].

Uma *máquina de estados finitos* (FSM - *Finite State Machine*), apresentada na parte inferior da Fig. 6.1, é responsável por controlar o processo de leitura bem como todos os outros passos da operação do neurônio. Assim que os primeiros termos são disponibilizados no banco de registradores, a FSM comuta os multiplexadores conectados ao somador e ao multiplicador, que são unidades aritméticas de ponto flutuante com *pipeline*, construídas com a ferramenta Xilinx CORE Generator (versão 13.4) [134]. As FIFOs também foram implementadas com o CORE Generator.

Todas as unidades no projeto funcionam em paralelo, permitindo, assim, que as operações matemáticas iniciem-se antes que todos os dados estejam presentes no banco de registradores. Levando-se em consideração a latência das unidades de ponto flutuante, a máquina de estados as alimenta a cada ciclo de *clock* com os dados que já estiverem disponíveis. As saídas dos operadores aritméticos são armazenadas novamente no banco de registradores e, quando os cálculos são finalizados, a saída do neurônio é disponibilizada para outro controlador de FIFO (Fig. 6.1, à direita) que, por sua vez, o envia para a FIFO de saída.

A máquina de estados foi implementada de tal forma que os pesos e as entradas do neurônio podem ser fornecidos em ciclos independentes. Assim que um grupo de seis coeficientes e um grupo de duas entradas estiverem disponíveis, o neurônio inicia seus cálculos. Se nenhum outro coeficiente for enviado à FIFO de pesos, o neurônio continuará a ler e processar as entradas, duas a duas. A FIFO de entradas possui 256

posições, permitindo o processamento de 128 saídas do neurônio em uma rodada de comunicação com o FPGA.

6.2.1 Co-verificação de *software-hardware*

Os projetos implementados neste trabalho foram testados usando um fluxo de projeto que emprega uma comunicação TCP/IP via *sockets* entre o PC e o FPGA onde o neurônio e o módulo LSSM estão programados. Enquanto o neurônio GMDH foi programado no FPGA, e acessado ali durante os experimentos, o módulo LSSM foi simulado pela ferramenta ModelSim, com a qual a comunicação foi abstraída através de métodos de *transaction level modeling*. Esta abstração foi alcançada através de algumas funções em código M que lidam com os detalhes da comunicação com os projetos, isolando, assim, os algoritmos de treinamento e execução de redes GMDH da complexidade envolvida nessas operações.

Para realizar a comunicação TCP/IP, um processador *soft-core* MicroBlaze foi utilizado. O neurônio GMDH e o módulo LSSM foram projetados para funcionar como periféricos do processador, conectados ao barramento PLB, como visto na Fig. 6.2. Isso foi realizado através de invólucros (*wrappers*) em VHDL, que encapsulam a lógica dos módulos como periféricos do MicroBlaze. O processo de criação dos periféricos é auxiliado pelo ambiente integrado da ferramenta XPS - *Xilinx Platform Studio* [135]. O projeto apresentado na Fig. 6.2, empregando o neurônio GMDH, foi testado em um *kit* de desenvolvimento XUPV5-LX110T que contém um dispositivo Xilinx Virtex 5 vlx110T [136]. Apesar de o módulo LSSM já ter sido testado no FPGA, seu encapsulamento como um periférico do MicroBlaze ainda não tinha sido desenvolvido à época dos experimentos apresentados, o que determinou sua simulação no ModelSim.

No que se refere ao *GMDH Box*, o treinamento e a execução das redes operam como se não estivessem realizando uma co-verificação *hardware/software*, mas quando o método de execução do neurônio é chamado no MATLAB, os dados são transferidos para o FPGA através da conexão *socket* e os resultados são recuperados, quando prontos, pelo mesmo canal. Por outro lado, quando o método de treinamento requer o cálculo dos pesos dos neurônios, as funções em código M chamam, de forma transparente, o módulo LSSM em simulação no ModelSim.

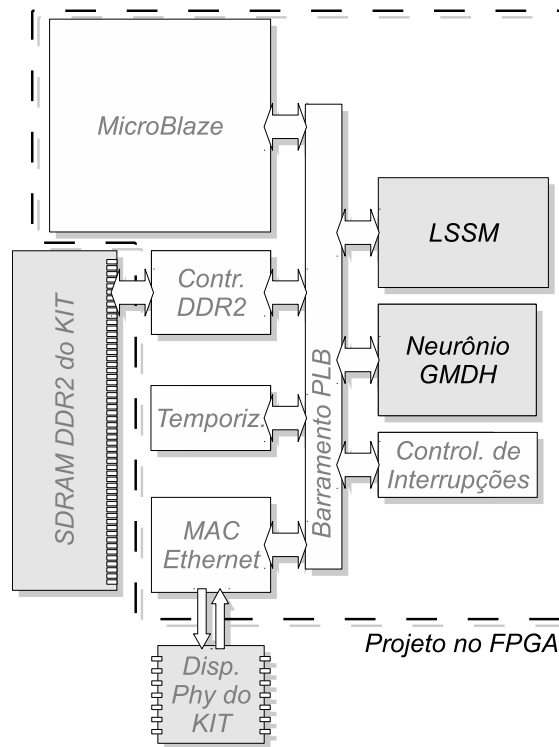


Figura 6.2: Diagrama mostrando o neurônio GMDH e o módulo LSSM como periféricos de um processador MicroBlaze, conectados em um barramento PLB. Baseada em [53].

6.2.2 O Módulo de Solução de Sistemas Lineares (LSSM)

O módulo LSSM foi desenvolvido para resolver *sistemas de equações lineares em geral*, utilizando a eliminação gaussiana. Para atender aos requisitos apresentados neste trabalho, ele foi adaptado para calcular os coeficientes dos neurônios GMDH, isto é, para resolver um sistemas com seis equações e seis incógnitas. Outros métodos que realizem a solução de sistemas de equações podem ser utilizados, caso existam descrições ou mesmo IP *cores* disponíveis para esse fim, entretanto, a eliminação gaussiana é apropriada para o método GMDH por causa da grande variedade de problemas que podem ser apresentados às redes neurais artificiais.

A Fig. 6.3 mostra que o LSSM é composto de: (a) um bloco de RAM, onde os elementos da matriz são armazenados; (b) a Unidade de Eliminação Gaussiana (GEU - *Gaussian Elimination Unit*) que iterativamente lê e escreve dados dos blocos de RAM até que todas as incógnitas tenham sido encontradas; (c) um conjunto de controladores de FIFO, responsáveis pela conexão com as FIFOs das seis linhas da matriz usada na eliminação gaussiana e com as FIFOs de saída; (d) uma máquina de estados finitos (FSM), responsável pela coordenação da transferência de dados entre os outros blocos.

Os coeficientes de um neurônio são calculados através da Eq. 2.17. O LSSM realiza

esse cálculo tomando como entrada a matriz formada pela Eq. 6.1, onde $[\mathbf{M}, \mathbf{p}]$ é uma matriz com dimensões 6×7 enviada para o módulo.

$$[\mathbf{M}, \mathbf{p}] = [(\mathbf{X}^T \mathbf{X}), \mathbf{X}^T \mathbf{b}] \quad (6.1)$$

Internamente, uma *matriz identidade* com dimensões 6×6 é incorporada aos valores armazenados (ver Eq. 6.2). Depois da eliminação gaussiana, realizada através de operações simples de multiplicação e deslocamento de dados, a memória interna do módulo armazena os valores apresentados na Eq. 6.3, onde \mathbf{I} e \mathbf{M}^{-1} são matrizes com dimensões 6×6 e \mathbf{a} é um vetor com seis elementos contendo os coeficientes que resolvem o sistema de equações lineares.

$$[\mathbf{M}, \mathbf{I}, \mathbf{p}] \quad (6.2)$$

$$[\mathbf{I}, \mathbf{M}^{-1}, \mathbf{a}] \quad (6.3)$$

Quando o LSSM termina a eliminação gaussiana, sua máquina de estados começa a copiar a última coluna da matriz armazenada na memória interna para a FIFO de saída. Mais detalhes sobre o módulo, por exemplo, sobre as otimizações realizadas na implementação em *hardware*, podem ser encontrados em Arias-Garcia et. al [130].

6.3 RESULTADOS EXPERIMENTAIS

Para testar a implementação dos módulos em *hardware*, utilizou-se a metodologia de predição de proteínas aplicada no capítulo 5 para reconhecer a estrutura estereoquímica da proteína com identificador PDB = 1WQC [137]. A toxina do escorpião *Opisthacanthus madagascariensis* (PDB ID: 1WQC) é um polipeptídeo composto de 26 resíduos de aminoácidos (DPCYEVLQHQHGNVKECEEACKHPVE) conhecido por ser disposto em duas α -hélices conectadas por uma volta, um formato estrutural conhecido como um “grampo α -helicoidal”. Dados estruturais dessa sequência alvo foram obtidos do PDB e uma rede O-GMDH foi construída para cada fragmento com modelos extraídos da base de dados e utilizados para montar as amostras de treinamento.

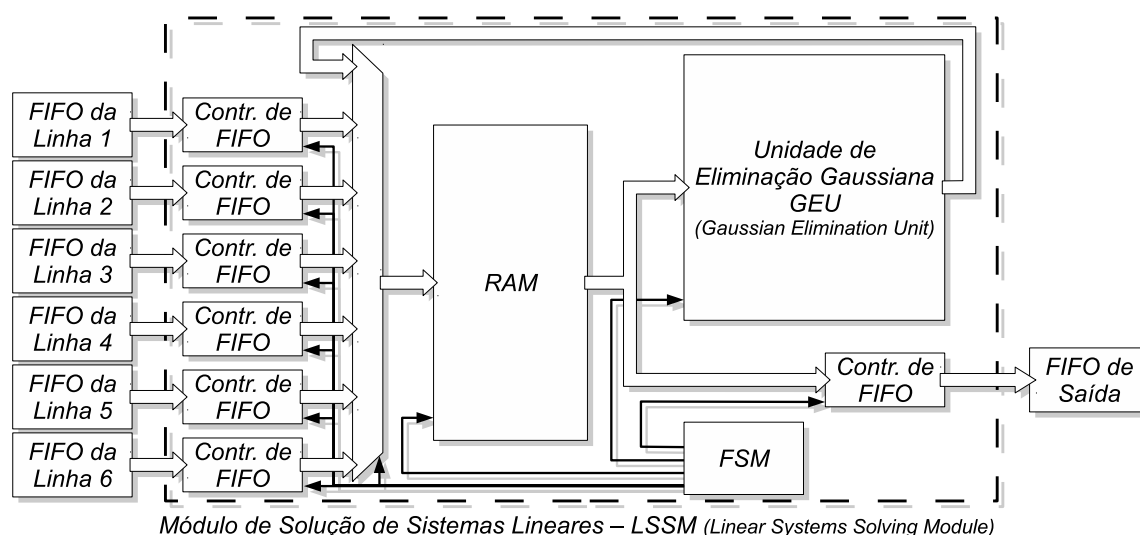


Figura 6.3: LSSM – Módulo de solução de sistemas lineares utilizado para calcular os pesos dos neurônios. Baseada em [53].

Em cada experimento, foram realizados dez diferentes ensaios, nos quais as amostras foram embaralhadas e divididas nos grupos de treinamento e seleção, como necessário para o treinamento de redes GMDH. Para fins de comparação, os experimentos foram realizados tanto com a versão em *hardware* como com a versão em *software* da rede. Para as análises estruturais e bioquímicas, foram considerados os menores valores de RMSD (Eq. 5.4) entre os dez ensaios de cada experimento. Para maiores detalhes sobre o cálculo do RMSD, verificar a seção 5.4.

A coluna 2 da Tab. 6.1 apresenta os valores obtidos para a sequência alvo utilizando as implementações em *hardware* e em *software*. A Fig. 6.4 mostra a representação em fita das estruturas como descritas experimentalmente (A – cinza), bem como as estruturas previstas em *software* (B – magenta) e em *hardware* (C – magenta). Por inspeção visual, percebe-se que as hélices individuais das estruturas secundárias se apresentam bem formadas, tanto nos resultados obtidos em *software* com em *hardware*.

Tabela 6.1: RMSD entre os carbonos alfa das estruturas previstas e experimental e análise da estrutura secundária das conformações previstas. A letra **E** denota a estrutura experimental, **S** denota a estrutura prevista por *software* e **H** denota a estrutura prevista por *hardware*.

PDB	RMSD (Å)	Folha	α -hélice	3^{10}	Outras
1WQC-E	0,0	0,0%	65,4 %	0,0%	34,6 %
1WQC-S	2,8	0,0%	61,5 %	0,0%	38,5 %
1WQC-H	2,6	0,0%	65,4 %	0,0%	34,6 %

Para analisar a disposição das estruturas secundárias das configurações previstas (ver

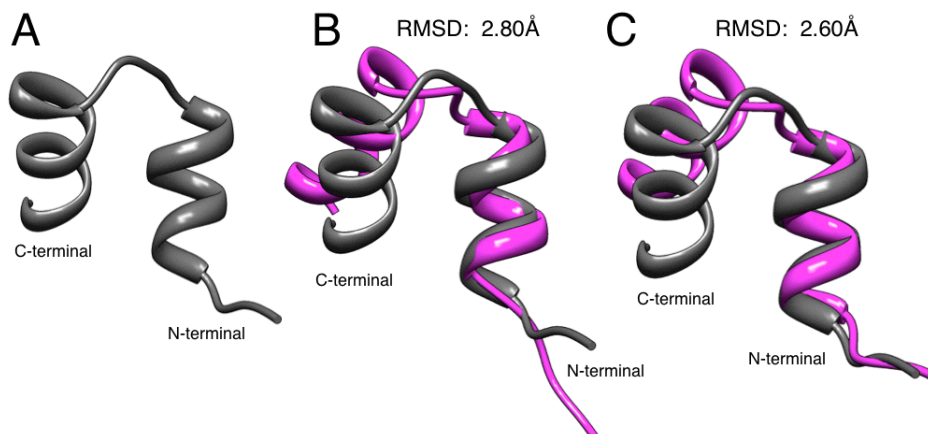


Figura 6.4: Representação em fita das estruturas experimental (cinza) e preditas (magenta) da proteína com identificador PDB 1WQC. Os carbonos-alfa (C_{α}) das estruturas sobrepostas estão corretamente alinhados. (A) Estrutura estereoquímica obtida experimentalmente; (B) Estruturas experimental e predita em *software*; (C) Estruturas experimental e predita em *hardware*. Cadeias laterais de aminoácidos não apresentadas para melhor clareza. Imagem gerada com a ferramenta CHIMERA [138]. Fonte: [53].

Fig. 6.4 B e C – magenta), foi utilizada a ferramenta PROMOTIF [98]. A Tab. 6.1 mostra o número (%) de resíduos que ocorrem em um dos quatro estados conformacionais (Estado regular: folha- β , α -hélice, hélice-3¹⁰ e outros: espirais e laços). Percebe-se pela Fig. 6.4 que as estruturas secundárias preditas por ambos os modelos se aproximam muito das estruturas obtidas pelos métodos experimentais.

Para se obterem as informações sobre a distribuição dos resíduos no diagrama de Ramachandran, utilizou-se a ferramenta PROCHECK. Os resultados estão ilustrados na Fig. 6.5 e sumarizados na Tab. 6.2. As predições em *software* e em *hardware* apresentaram mais de 90% de resíduos na região mais favorável. Uma alta quantidade de aminoácidos nessa região denota que a estrutura possui um número pequeno de maus contatos.

O diagrama de Ramachandran destaca as combinações dos ângulos ϕ e ψ que produzem bons posicionamentos das moléculas, em outras palavras, que evitam choques entre as moléculas. O eixo x representa os ângulos ϕ e o eixo y os ângulos ψ . As combinações de ângulos ϕ e ψ que são mais favoráveis são identificadas no diagrama por regiões em vermelho. As regiões em amarelo destacam as combinações de ângulos adicionalmente favoráveis. Em amarelo-claro, destacam-se as regiões geralmente permitidas e em branco, as não permitidas.

Os pontos negros no diagrama de Ramachandran representam os ângulos dos resíduos

das sequências estudadas. Na Fig. 6.5, o diagrama sob a letra **A** mostra os ângulos da sequência estudada, obtidos experimentalmente. O diagrama **B** apresenta os resultados com a predição via *software* e o diagrama **C** mostra os resultados obtidos com a co-verificação *software/hardware*.

Tabela 6.2: Valores de Ramachandran obtidos a partir das conformações experimental e preditas. E denota a estrutura experimental. S denota a estrutura predita por *software*. H denota a estrutura predita em *hardware*.

PDB	Mais favoráveis	Adicionalmente permitidas	Geralmente permitidas	Não Permitidas
1WQC-E	90,5 %	0,0%	9,5%	0,0%
1WQC-S	90,5 %	4,8%	4,8%	0,0%
1WQC-H	95,2 %	0,0%	4,8%	0,0%

As menores diferenças entre o número de resíduos da estrutura experimental e das estruturas preditas ocorre nas regiões onde a estrutura inteira modifica seu estado conformacional (regiões de α -hélices ou folhas- β para regiões de voltas ou laços). Essa pequena distorção, contudo, pode ser corrigida por métodos mecânicos que podem simular e otimizar as interações entre todos os átomos, permitindo a formação de ligações de hidrogênio para estabilizar essas regiões [116].

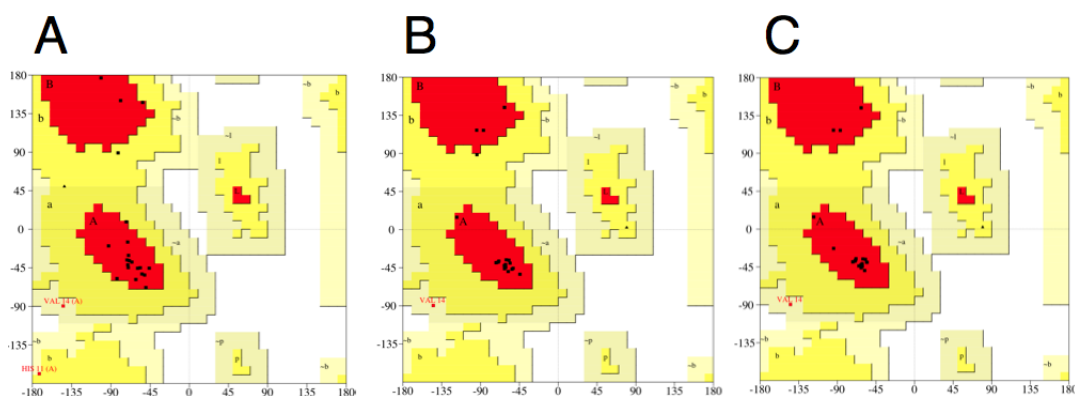


Figura 6.5: Diagramas de Ramachandran para as estruturas experimentais e preditas. (A) Diagrama para os valores experimentais. (B) Diagrama para as estruturas preditas por *software*. (C) Diagramas para estruturas preditas em *hardware*. O eixo x representa os ângulos ϕ e o eixo y os ângulos ψ . As cores representam as regiões mais favoráveis (vermelhas), adicionalmente favoráveis (amarelas); geralmente favoráveis (amarelo-claras) e; não permitidas (brancas). Diagramas gerados pela ferramenta PROCHECK [118]. Fonte: [53].

6.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo demonstrou a aplicação de módulos em *hardware* para o cálculo das principais tarefas relacionadas ao algoritmo GMDH: (a) o processamento das expressões polinomiais que definem os neurônios e; (b) os cálculos dos coeficientes que compõem os polinômios, durante o processo de treinamento.

Dois módulos foram desenvolvidos na linguagem VHDL: (a) o neurônio GMDH e (b) o módulo de solução de sistemas lineares – LSSM, que foram sintetizados e testados em um dispositivo FPGA Xilinx Virtex 5 vlx110T.

Uma estratégia de execução foi desenvolvida para avaliar a funcionalidade dos projetos em VHDL. Nesta estratégia, foi desenvolvido um conjunto de funções que abstraem a comunicação do sistema *GMDH Box* com os módulos de *hardware*. Essas funções foram utilizadas para realizar a comunicação com o FPGA via TCP/IP, onde o neurônio foi implementado, e com a ferramenta ModelSim, onde o *design* do LSSM foi simulado.

O neurônio foi implementado como um periférico conectado a um barramento PLB do qual um *soft-processor* MicroBlaze era o mestre, responsável por estabelecer a comunicação via *socket* e por transferir os dados para o neurônio e obter suas saídas. O módulo LSSM foi testado apenas por simulação, pois sua implementação como um periférico de um barramento PLB ainda não estava terminada à época dos experimentos.

Para examinar a adequabilidade do projeto, uma aplicação de exemplo foi desenvolvida. Os módulos foram utilizados como auxiliares na execução e treinamento de redes O-GMDH criadas para realizar aproximações de estruturas tridimensionais de sequências de proteínas. Os resultados demonstraram que o método baseado em GMDH, utilizando co-verificação de *software/hardware* é efetivo para lidar com o problema proposto.

Apesar do fato de que os computadores de propósito geral possuem capacidade suficiente para processarem todas as fases dos algoritmos GMDH, os desempenhos obtidos pelos módulos em *hardware* em uma plataforma com frequências bem menores (como acontece na maioria dos sistemas embarcados) podem vir a ser uma característica chave no campo das aplicações embarcadas. Avaliações preliminares sugerem que um único

neurônio O-GMDH, que toma apenas 3% dos recursos oferecidos pelo FPGA empregado neste trabalho (usando a descrição atual), pode atingir a marca de 89 KFLOPS/MHz.

Os próximos passos no desenvolvimento do projeto em *hardware* serão a finalização do periférico contendo o módulo LSSM e a avaliação da infraestrutura desenvolvida em aplicações embarcadas. Caminhos possíveis são a utilização de mais neurônios GMDH como periféricos, a serem executados em paralelo, ou mesmo a implementação de mais partes do algoritmo GMDH no nível dos microcontroladores ou mesmo diretamente em *hardware*. Com os resultados obtidos, ficou provado que o emprego do algoritmo GMDH para plataformas embarcadas é possível e funcional.

7 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

7.1 CONCLUSÕES

Muitos sistemas apresentam grande complexidade para que sejam modelados analiticamente. Em outros casos, não é possível conhecer todas as variáveis que influenciam o seu funcionamento. Independentemente desses fatores, a criação de modelos para a análise dos sistemas sob diferentes circunstâncias ou para prever suas respostas futuras é uma necessidade crescente.

O desenvolvimento da filtragem adaptativa está muito ligado ao desenvolvimento das redes neurais artificiais [11]. Widrow e Hoff desenvolveram o método dos mínimos quadrados médios (LMS) juntamente com seu modelo de neurônio, o *Adaline*. As pesquisas na área de aproximação de sistemas seguem diversos caminhos, como os métodos de regressão e interpolação e diversos outros métodos de filtragem. Segundo Gabor [6], a aplicação de um filtro como um preditor foi realizada primeiramente por Kolmogorov e Wiener, mas foi Zadeh [139] quem primeiro percebeu que um filtro universal também pode ser usado para reconhecimento. Gabor argumenta que o termo “filtro” pode ser relacionado a qualquer operação quantitativa sobre informações que devem ser processadas *online*. Dessa forma, o método do acadêmico A. G. Ivakhnenko, fundamentado nos trabalhos de Gabor e Kolmogorov, também pode ser visto como uma forma de filtragem.

Existem muitos relatos na literatura sobre a aplicação de redes GMDH para o modelamento e a predição de sistemas e para o reconhecimento de padrões, entretanto, há uma carência no estudo da convergência e do consumo de recursos computacionais relacionados a diversos aspectos do algoritmo, como os critérios de parada do treinamento e de seleção dos neurônios, a forma das expressões utilizadas nas descrições parciais ou a conectividade entre os neurônios. Também não existem estudos aprofundados sobre problemas relacionados aos ajustes dos coeficientes, como a solução dos sistemas de equações.

Este trabalho intentou preencher uma lacuna, fornecendo dados experimentais e conclusões que colaboram para o desenvolvimento da técnica, ainda sendo descoberta pela comunidade, além de introduzir inovações no método que facilitam o seu uso na prática. Um dos resultados desse esforço já foi apresentado ao público: uma versão da ferramenta (*GMDH Box*) desenvolvida para a realização dos estudos; um sistema flexível e expansível, que permite a aplicação de GMDH de uma forma geral[48]. Não existe nenhuma outra iniciativa divulgada que forneça a flexibilidade e a aplicabilidade apresentada pelo sistema, que foi disponibilizado com uma licença de *software* livre [49], o que permite que seja imediatamente utilizado por qualquer outro pesquisador (ou mesmo para fins comerciais), sem restrições.

O sistema desenvolvido foi utilizado para a avaliação do método em diversas aplicações. Um estudo sobre as capacidades das redes foi realizado utilizando modelos teóricos de plantas não lineares. Uma aplicação prática inédita para as redes GMDH foi feita no campo do reconhecimento de estruturas estereoquímicas de proteínas.

Seguindo o caminho rumo à implementação do modelo em sistemas embarcados, este trabalho também produziu a primeira implementação de um neurônio GMDH em FPGA. A avaliação do método em *hardware* foi feita através da técnica de co-verificação de *software-hardware*, onde o neurônio foi utilizado como um periférico em um sistema contando com um barramento PLB, tendo um microprocessador MicroBlaze como mestre. Ademais, incluiu-se, no projeto, um módulo em *hardware* de solução de sistemas de equações lineares, com isso completando-se o núcleo da implementação de GMDH em *hardware*.

Este trabalho não apenas forneceu um mecanismo inédito para a avaliação das redes do tipo GMDH (a ferramenta *GMDH Box*), mas introduziu novas parametrizações nos seus processos e, ainda, as utilizou em aplicações inovadoras, tais como as listadas abaixo.

1. Ferramenta de treinamento flexível:

A ferramenta *GMDH Box*, apresentada neste trabalho, permite que se façam diferentes configurações que afetam o treinamento das redes GMDH, além de ter introduzido novos critérios de avaliação do desempenho e de interrupção do treinamento. No que se refere aos critérios de avaliação, foi acrescentado o critério do *limiar dinâmico* do MSE gerado pelos neurônios e permitiu-se a utilização em

conjunto do *limiar dinâmico* com o critério do *número máximo de neurônios na camada*. No que tange a interrupção do treinamento, a ferramenta acrescentou os critérios: (a) *desempenho mínimo*, que determina o término do processo quando o menor MSE de uma camada alcança um valor mínimo predeterminado e; (b) *número máximo de camadas*. Além disso, o sistema possibilita a utilização de diferentes sinais como entradas da rede. Em significativa parcela dos artigos estudados, utilizou-se GMDH para identificar sistemas SISO (*Single Input, Single Output*). O sistema *GMDH Box* permite a identificação de sistemas MISO (*Multiple Inputs, Single Output*).

A ferramenta foi construída de forma modular, o que facilita a substituição de partes dos processos por novas versões, como por exemplo, a função de treinamento dos neurônios ou das camadas. Uma vez desenvolvido um módulo substituto, pode-se especificar o nome da nova função durante a criação da rede, fazendo com que o treinamento e a execução se comportem de maneira diferente.

Outro ponto de interesse foi a solução de diversas questões de desempenho e estabilidade. Foram desenvolvidos, por exemplo, métodos para calcular as quantidades de combinações e para a criação dessas combinações melhorados em relação aos fornecidos pelos ambientes MATLAB e Octave. Nos cálculos dos coeficientes, utilizou-se o método SVD (*Single Value Decomposition*), que diminui os efeitos gerados por sistemas mal condicionados. O SVD se encontra implementado em ambas as plataformas, mas sua utilização para a solução de sistemas de equações não é automática e requer alguma programação para que seja aplicada.

Pelo que se encontrou na literatura, não existe uma outra ferramenta que consolida uma gama de possibilidades de treinamento como o faz o *GMDH Box*.

2. Utilização de GMDH para identificação de sistemas não lineares:

A partir do artigo de Narendra e Parthasarathy, que realizaram estudos sobre a utilização de redes neurais na identificação de sistemas, este trabalho apresentou a utilização de redes GMDH para a modelagem *paralela* de plantas não lineares, além de mostrar resultados na área de identificação *série-paralela*.

3. Aplicação em problemas de bioinformática:

Redes GMDH foram utilizadas num método que realiza a predição das estruturas estereoquímicas de proteínas. O método é baseado no reconhecimento de padrões de fragmentos das sequências proteicas em estudo. Os padrões de treinamento são criados com fragmentos de sequências já conhecidas, obtidas do Banco de Dados de Proteínas (PDB - *Proteins Database*). Este estudo forneceu a primeira

avaliação do método GMDH neste campo de pesquisa. Fazendo-se uma comparação com redes MLP, verificou-se que GMDH consegue realizar a tarefa de maneira compatível, no que se refere às distâncias das posições dos resíduos nas estruturas preditas em relação às estruturas identificadas por métodos experimentais. Quanto à disposição dos resíduos nas regiões de Ramachandran (*mais favoráveis, adicionalmente permitidas, geralmente permitidas e não permitidas*), os resultados com GMDH se mostraram mais próximos dos resultados experimentais do que os obtidos com MLPs.

4. Aplicação em problemas objetivando sistemas embarcados:

O treinamento de redes neurais em sistemas embarcados é uma tarefa que tende a ser de difícil aplicação devido ao custo computacional envolvido nos cálculos das funções de ativação e nos métodos iterativos geralmente utilizados por diversos modelos de RNAs. GMDH, ao contrário, utiliza descrições polinomiais para implementar seus neurônios, o que permite sua realização por meio de operações simples, como multiplicações e somas, tarefas para as quais os processadores para aplicações embarcadas comumente estão preparados. Entretanto, o cálculo dos coeficientes ainda pode ser uma tarefa difícil, pois os métodos para solução de sistemas de equações lineares também consomem consideráveis recursos computacionais.

Com o intuito de avaliar a possibilidade da aplicação de GMDH nessas plataformas, buscou-se uma abordagem baseada em sistemas reconfiguráveis, resultando no desenvolvimento de uma plataforma de co-verificação de *software/hardware* que vem a ser a primeira incursão do método de Ivakhnenko em *hardware*, mais especificamente, em FPGAs (*Field Programmable Gate Arrays*). No fluxo de projeto desenvolvido, utilizou-se, ainda, com sucesso, um módulo de *hardware* para solução de sistemas de equações lineares (LSSM - *Linear Systems Solver Module*), através do qual os coeficientes dos neurônios foram calculados. Com o suporte da característica modular do *GMDH Box*, foi possível desenvolver funções para comunicação com o FPGA onde o projeto do neurônio GMDH estava em funcionamento, bem como com a ferramenta de simulação ModelSim, onde o projeto físico do LSSM se encontrava em simulação.

O projeto de co-verificação foi aplicado em um problema exemplo: a predição da estrutura tridimensional do polipeptídeo que forma a toxina do escorpião *Opisthacanthus madagascariensis*. Essa aplicação demonstrou que é possível implementar partes importantes do algoritmo GMDH em plataformas reconfiguráveis, abrindo o caminho para mais incursões nesse sentido.

7.2 TRABALHOS FUTUROS

Após muitos dos experimentos apresentados neste relatório, foram realizadas diferentes melhorias e novas funcionalidades foram incluídas no aplicativo *GMDH Box*. Essas melhorias foram aproveitadas nos problemas de identificação de sistemas não lineares, que havia sido a primeira investigação do GMDH utilizando o *GMDH Box* em suas versões estáveis iniciais. Esse problema foi reavaliado com uma versão mais atual da ferramenta, entretanto, os testes com reconhecimento de proteínas e com *hardware* reconfigurável ainda podem ser revistos.

Apesar da evolução atual do sistema, ainda há muito campo para a investigação do método, por exemplo:

1. Utilizar algoritmos que realizem a adaptação das redes após o treinamento, permitindo sua evolução sem necessidade de criar-se novas redes quando os parâmetros dos sistemas se modifiquem. Para tanto, pode-se seguir as seguintes abordagens:
 - Utilização de algoritmos bio-inspirados para a otimização dos modelos. Durante a implementação do *GMDH Box*, foram desenvolvidas funções de treinamento baseadas no algoritmo PSO (*Particle Swarm Optimization* – Otimização por Enxame de Partículas). Experimentos preliminares mostraram que é possível alcançar resultados com esse algoritmo. Uma vantagem do PSO, por exemplo, é a possibilidade de utilizar-se funções custo independentes do erro produzido entre as saídas dos neurônios e os alvos da rede.
 - Adaptar algoritmos de treinamento de rede MLP para o treinamento GMDH, como o *backpropagation*. Neste sentido, já foi iniciada a implementação do treinamento baseado no algoritmo RPROP [140].
2. Traduzir partes do algoritmo de treinamento para a linguagem C ou C++ e, eventualmente, implementar outras partes em linguagens de descrição de *hardware*. Essas implementações podem levar ao desenvolvimento do método inteiramente adaptado a plataformas embarcadas.
3. Utilizar redes GMDH para o processamento de sinais de sensores (de infravermelho ou ultrassom) para realizar o desvio de obstáculos em robótica móvel. Este é um problema que já foi estudado preliminarmente durante o desenvolvimento deste projeto e que pode ser continuado, principalmente se puder ser abordado com versões embarcadas do método.

Outro ponto ainda a ser estudado é o da real capacidade de aproximação das redes GMDH. Os neurônios GMDH não apresentam funções discriminantes em seus elementos e redes GMDH são formadas por diferentes números de camadas, portanto, não podem ser enquadrados nas conclusões de Cybenko, que demonstraram que redes neurais artificiais com funções de ativação sigmoide são aproximadores universais [141] (ver seção 2.5). Análises formais das propriedades das redes polinomiais devem ser consideradas como importantes trabalhos futuros.

APÊNDICES

A RESULTADOS EXPERIMENTAIS DAS MODELAGENS DE SISTEMAS DINÂMICOS

Resultados dos experimentos realizados com a identificação de sistemas dinâmicos, apresentados no Capítulo 4.

Tabela A.1: Resultados para o *Sistema 1*, variação **a**.

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
1	1	A	99,99%	0,59	3	99,98%	252,37	5
2	1	B	99,99%	0,04	3	99,98%	304,84	10
3	1	C	99,99%	0,13	3	99,95%	263,34	7
4	1	D	99,99%	0,10	3	99,96%	351,30	15
5	1	E	99,99%	0,03	4	99,99%	277,88	20
6	1	F	99,99%	0,03	4	100,00%	256,33	40
7	1	G	99,99%	0,59	7	99,91%	221,57	30
8	1	H	99,99%	4,69	8	100,00%	299,17	60
9	2	A	92,24%	0,05	3	91,28%	218,00	5
10	2	B	92,24%	0,04	3	88,24%	276,76	10
11	2	C	87,50%	1,10	14	88,49%	199,01	7
12	2	D	87,41%	3,82	13	73,41%	340,39	15
13	2	E	93,00%	0,02	4	87,62%	371,67	20
14	2	F	92,36%	0,02	4	85,28%	575,11	40
15	2	G	86,50%	453,20	25	89,71%	642,02	30
16	2	H	88,72%	311,84	14	89,75%	1552,40	60
17	3	A	67,44%	0,06	6	67,63%	9,26	5
18	3	B	67,44%	0,06	6	64,27%	34,41	10
19	3	C	65,69%	1,21	13	68,71%	16,82	7
20	3	D	65,69%	2,88	13	95,82%	331,81	15
21	3	E	48,37%	0,02	4	95,47%	70,11	20
22	3	F	48,11%	0,03	4	99,98%	620,14	40
23	3	G	7,80%	452,66	24	99,52%	162,52	30
24	3	H	5,56%	313,34	16	79,98%	2324,04	60
25	4	A	9,39%	0,07	6	20,31%	6,19	5
26	4	B	9,39%	0,06	6	14,95%	4,90	10
27	4	C	3,13%	1,36	16	1,43%	5,15	7
28	4	D	3,91%	2,74	15	1,19%	8,67	15
29	4	E	1,77%	0,02	4	0,87%	35,96	20
30	4	F	1,79%	0,02	4	1,14%	10,88	40
31	4	G	4,14%	489,85	27	1,23%	17,59	30

continua...

Tabela A.1 - continuação

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
32	4	H	1,07%	336,81	16	1,11%	42,19	60
33	5	A	99,99%	0,03	3	99,97%	164,59	5
34	5	B	99,99%	0,06	3	100,00%	273,82	10
35	5	C	99,99%	0,12	3	99,65%	57,71	7
36	5	D	99,99%	0,08	3	100,00%	364,11	15
37	5	E	99,99%	0,02	4	100,00%	417,52	20
38	5	F	99,99%	0,03	4	100,00%	611,48	40
39	5	G	99,99%	0,71	8	100,00%	640,91	30
40	5	H	99,99%	4,78	8	100,00%	1886,59	60
41	6	A	92,45%	0,04	3	90,23%	26,89	5
42	6	B	92,45%	0,04	3	96,84%	166,81	10
43	6	C	87,42%	1,31	15	87,77%	39,81	7
44	6	D	88,83%	3,30	15	96,43%	362,58	15
45	6	E	92,82%	0,02	4	96,95%	381,44	20
46	6	F	92,84%	0,02	4	96,05%	245,67	40
47	6	G	90,14%	477,11	28	96,33%	360,51	30
48	6	H	89,51%	333,19	16	98,02%	470,54	60
49	7	A	5,69%	0,04	3	4,91%	242,05	5
50	7	B	5,69%	0,04	3	12,67%	314,00	10
51	7	C	8,86%	0,09	3	5,88%	261,55	7
52	7	D	8,86%	0,08	3	5,23%	360,28	15
53	7	E	1,35%	0,02	4	1,43%	391,27	20
54	7	F	3,73%	0,02	4	41,61%	594,68	40
55	7	G	2,70%	0,58	7	57,76%	327,37	30
56	7	H	10,40%	3,93	8	75,16%	571,09	60
57	8	A	13,24%	0,06	6	70,47%	242,87	5
58	8	B	13,24%	0,04	6	92,42%	286,37	10
59	8	C	1,82%	1,17	13	35,91%	256,93	7
60	8	D	1,82%	2,63	13	10,90%	351,01	15
61	8	E	0,16%	0,02	4	59,35%	379,28	20
62	8	F	0,22%	0,03	4	63,08%	590,82	40
63	8	G	0,02%	449,86	23	91,68%	617,00	30
64	8	H	0,36%	300,34	17	28,07%	1481,35	60
65	9	A	0,76%	0,03	3	9,13%	13,39	5
66	9	B	0,76%	0,03	3	99,33%	285,46	10
67	9	C	3,06%	0,10	3	5,85%	256,47	7
68	9	D	3,06%	0,08	3	84,10%	350,28	15
69	9	E	0,92%	0,02	4	99,32%	380,12	20
70	9	F	0,75%	0,02	4	99,98%	270,70	40
71	9	G	7,60%	0,58	8	99,92%	597,33	30
72	9	H	16,61%	3,87	8	100,00%	709,44	60

Tabela A.2: Resultados para o *Sistema 1*, variação **b**.

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
1	1	A	99,99%	0,68	3	99,98%	280,84	5
2	1	B	99,99%	0,08	3	99,94%	70,04	10
3	1	C	99,99%	0,15	3	99,96%	300,43	7
4	1	D	99,99%	0,13	3	99,94%	412,59	15
5	1	E	99,99%	0,04	4	100,00%	341,12	20
6	1	F	99,99%	0,03	4	100,00%	174,80	40
7	1	G	99,99%	0,88	8	99,95%	338,24	30
8	1	H	100,00%	5,93	7	100,00%	33,16	60
9	2	A	99,60%	0,07	6	99,83%	282,08	5
10	2	B	99,60%	0,09	6	99,81%	281,64	10
11	2	C	98,86%	0,27	6	96,45%	252,60	7
12	2	D	98,76%	0,43	7	98,64%	342,56	15
13	2	E	98,64%	0,02	4	99,39%	373,78	20
14	2	F	99,72%	0,02	4	99,82%	566,12	40
15	2	G	93,13%	14,54	15	99,55%	24,50	30
16	2	H	40,63%	286,53	13	79,55%	230,00	60
17	3	A	87,00%	0,04	6	99,77%	240,26	5
18	3	B	87,00%	0,04	6	100,00%	277,55	10
19	3	C	65,41%	1,07	10	39,51%	10,34	7
20	3	D	65,41%	2,67	10	100,00%	343,82	15
21	3	E	46,56%	0,02	4	100,00%	248,78	20
22	3	F	46,60%	0,02	4	76,16%	391,35	40
23	3	G	100,00%	451,01	20	97,26%	181,74	30
24	3	H	100,00%	301,87	13	100,00%	2275,87	60
25	4	A	2,02%	0,05	6	1,62%	12,94	5
26	4	B	2,02%	0,05	6	1,78%	6,00	10
27	4	C	3,52%	1,17	14	1,68%	4,13	7
28	4	D	2,60%	2,63	16	1,40%	5,32	15
29	4	E	2,08%	0,02	4	1,94%	5,97	20
30	4	F	2,06%	0,02	4	1,80%	11,67	40
31	4	G	2,88%	447,33	25	1,10%	11,23	30
32	4	H	2,83%	304,77	16	1,68%	45,15	60
33	5	A	99,97%	0,03	3	99,86%	169,14	5
34	5	B	99,97%	0,03	3	100,00%	122,29	10
35	5	C	99,99%	0,09	3	99,94%	252,21	7
36	5	D	99,99%	0,08	3	99,95%	342,89	15
37	5	E	99,99%	0,02	4	100,00%	373,66	20
38	5	F	99,99%	0,02	4	100,00%	580,13	40
39	5	G	99,89%	0,58	8	100,00%	238,36	30
40	5	H	100,00%	3,91	7	100,00%	931,12	60

continua...

Tabela A.2 - continuação

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
41	6	A	99,10%	0,05	6	99,77%	47,17	5
42	6	B	99,10%	0,04	6	99,79%	107,16	10
43	6	C	99,28%	1,15	13	99,90%	165,86	7
44	6	D	98,33%	2,64	14	99,46%	214,86	15
45	6	E	98,62%	0,02	4	99,71%	10,39	20
46	6	F	99,34%	0,02	4	99,86%	105,81	40
47	6	G	99,90%	445,25	24	99,60%	15,73	30
48	6	H	99,89%	295,10	14	99,83%	40,86	60
49	7	A	0,39%	0,03	3	1,06%	239,19	5
50	7	B	0,39%	0,04	3	10,25%	263,61	10
51	7	C	7,41%	0,09	3	13,28%	251,19	7
52	7	D	7,41%	0,09	3	98,87%	234,34	15
53	7	E	3,01%	0,02	4	0,00%	132,41	20
54	7	F	1,44%	0,02	4	99,98%	70,46	40
55	7	G	13,34%	0,60	8	99,94%	576,12	30
56	7	H	100,00%	3,67	7	99,92%	166,86	60
57	8	A	9,97%	0,05	6	99,91%	240,60	5
58	8	B	9,97%	0,04	6	100,00%	276,70	10
59	8	C	18,48%	1,15	14	76,73%	251,61	7
60	8	D	1,44%	2,62	16	71,99%	344,32	15
61	8	E	1,04%	0,02	4	62,89%	298,83	20
62	8	F	1,00%	0,02	4	99,89%	265,49	40
63	8	G	3,24%	445,76	23	55,16%	574,43	30
64	8	H	100,00%	298,85	15	63,52%	937,37	60
65	9	A	2,15%	0,03	3	45,12%	239,30	5
66	9	B	2,15%	0,03	3	99,92%	277,56	10
67	9	C	14,46%	0,08	3	99,84%	252,06	7
68	9	D	14,46%	0,09	3	100,00%	232,27	15
69	9	E	2,12%	0,02	4	99,96%	368,33	20
70	9	F	1,04%	0,02	4	100,00%	559,81	40
71	9	G	0,00%	0,56	8	100,00%	291,77	30
72	9	H	100,00%	3,56	7	100,00%	1068,19	60

Tabela A.3: Resultados para o *Sistema 2*

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
1	1	A	95,61%	0,09	13	84,01%	6,58	5
2	1	B	95,61%	0,07	13	95,33%	22,73	10
3	1	C	95,21%	0,47	15	94,41%	9,81	7
4	1	D	95,21%	0,96	15	99,29%	7,13	15
5	1	E	96,89%	0,02	6	99,33%	17,19	20
6	1	F	94,57%	0,02	7	96,53%	15,42	40
7	1	G	99,16%	120,10	23	97,05%	4,70	30
8	1	H	97,05%	83,45	17	89,80%	11,78	60
9	2	A	0,16%	0,07	17	22,39%	8,76	5
10	2	B	0,16%	0,07	17	36,02%	7,29	10
11	2	C	35,53%	0,47	15	48,06%	22,74	7
12	2	D	35,53%	0,97	15	34,14%	6,04	15
13	2	E	31,36%	0,02	7	44,81%	5,21	20
14	2	F	1,78%	0,02	7	47,25%	1,57	40
15	2	G	42,69%	140,05	34	57,37%	16,85	30
16	2	H	53,96%	84,10	15	44,50%	9,59	60
17	3	A	46,28%	0,07	16	82,29%	5,34	5
18	3	B	46,28%	0,07	16	51,05%	1,19	10
19	3	C	20,88%	0,47	18	50,71%	3,20	7
20	3	D	23,52%	0,97	15	52,05%	3,53	15
21	3	E	2,60%	0,02	7	96,13%	3,39	20
22	3	F	0,94%	0,02	6	94,57%	6,13	40
23	3	G	15,93%	121,34	28	99,89%	42,27	30
24	3	H	49,14%	86,15	14	98,97%	83,80	60
25	4	A	0,12%	0,03	3	0,04%	0,79	5
26	4	B	0,12%	0,03	3	0,12%	0,92	10
27	4	C	0,12%	0,47	11	0,18%	0,78	7
28	4	D	0,08%	1,01	14	0,04%	0,89	15
29	4	E	0,12%	0,02	7	0,16%	0,86	20
30	4	F	0,12%	0,02	6	0,16%	1,52	40
31	4	G	0,08%	120,06	22	0,08%	1,66	30
32	4	H	0,12%	85,83	16	0,68%	5,37	60
33	5	A	94,09%	0,07	14	96,37%	2,63	5
34	5	B	94,09%	0,07	14	99,93%	46,45	10
35	5	C	90,73%	0,47	15	95,89%	4,13	7
36	5	D	94,89%	1,01	17	99,89%	25,70	15
37	5	E	96,65%	0,02	7	99,97%	35,09	20
38	5	F	92,13%	0,02	7	99,97%	9,74	40
39	5	G	98,41%	118,78	31	99,97%	4,82	30
40	5	H	98,17%	84,51	17	99,89%	32,96	60

continua...

Tabela A.3 - continuação

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
41	6	A	24,40%	0,07	15	70,73%	3,34	5
42	6	B	24,40%	0,07	15	83,35%	2,39	10
43	6	C	44,76%	0,47	10	58,37%	2,52	7
44	6	D	42,58%	0,99	12	67,31%	3,22	15
45	6	E	33,88%	0,02	6	76,04%	2,00	20
46	6	F	25,40%	0,02	6	73,55%	4,49	40
47	6	G	62,21%	119,02	30	47,28%	2,99	30
48	6	H	61,55%	83,72	17	80,12%	18,15	60
49	7	A	0,02%	0,07	13	42,70%	1,75	5
50	7	B	0,02%	0,07	13	59,04%	27,97	10
51	7	C	0,02%	0,47	15	91,89%	9,61	7
52	7	D	0,02%	0,97	15	93,37%	12,44	15
53	7	E	0,82%	0,02	6	73,73%	15,16	20
54	7	F	0,02%	0,02	7	0,49%	24,96	40
55	7	G	0,35%	119,44	23	81,05%	17,98	30
56	7	H	0,02%	83,41	17	81,17%	35,35	60
57	8	A	0,08%	0,07	14	27,50%	1,08	5
58	8	B	0,08%	0,07	14	91,81%	2,44	10
59	8	C	8,25%	0,47	15	15,74%	5,03	7
60	8	D	0,04%	0,96	19	98,41%	38,82	15
61	8	E	0,38%	0,02	7	93,01%	5,68	20
62	8	F	0,05%	0,02	7	94,17%	8,13	40
63	8	G	32,44%	120,25	31	97,97%	6,67	30
64	8	H	2,49%	86,83	16	65,57%	7,10	60
65	9	A	0,60%	0,07	13	79,45%	3,85	5
66	9	B	0,60%	0,07	13	99,73%	46,47	10
67	9	C	0,02%	0,47	14	95,54%	2,40	7
68	9	D	7,71%	0,97	15	93,41%	4,97	15
69	9	E	0,82%	0,02	7	99,69%	19,68	20
70	9	F	0,01%	0,02	7	99,93%	13,39	40
71	9	G	0,29%	119,45	29	99,53%	18,96	30
72	9	H	3,70%	84,90	15	99,77%	47,40	60

Tabela A.4: Resultados para o *Sistema 3*

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
1	1	A	99,79%	0,05	3	99,40%	17,33	5
2	1	B	99,79%	0,03	3	99,94%	40,79	10
3	1	C	98,62%	0,14	6	99,74%	43,14	7
4	1	D	98,75%	0,20	6	99,80%	50,89	15
5	1	E	99,76%	0,03	6	99,52%	52,38	20
6	1	F	99,76%	0,03	6	99,95%	13,91	40
7	1	G	98,96%	121,19	25	99,89%	29,07	30
8	1	H	92,81%	85,56	17	99,91%	56,42	60
9	2	A	8,45%	0,07	17	5,54%	7,09	5
10	2	B	8,45%	0,08	17	5,11%	8,03	10
11	2	C	14,74%	0,48	14	16,66%	0,88	7
12	2	D	9,80%	0,99	15	7,30%	19,52	15
13	2	E	10,76%	0,03	7	3,42%	52,66	20
14	2	F	6,88%	0,03	7	2,48%	76,53	40
15	2	G	3,84%	120,71	23	6,42%	84,82	30
16	2	H	14,29%	86,03	15	9,40%	18,97	60
17	3	A	8,56%	0,04	7	55,39%	2,78	5
18	3	B	8,56%	0,04	7	99,92%	38,16	10
19	3	C	22,17%	0,47	15	45,72%	1,62	7
20	3	D	18,40%	0,97	14	99,88%	53,58	15
21	3	E	8,56%	0,02	6	99,89%	9,68	20
22	3	F	8,62%	0,01	3	99,96%	16,72	40
23	3	G	46,55%	121,45	27	76,21%	1,68	30
24	3	H	57,66%	86,51	15	99,91%	290,01	60
25	4	A	1,09%	0,03	3	0,55%	0,84	5
26	4	B	1,09%	0,03	3	0,65%	0,99	10
27	4	C	0,97%	0,50	11	1,10%	0,77	7
28	4	D	1,13%	0,98	11	0,65%	0,96	15
29	4	E	1,70%	0,02	3	1,41%	0,85	20
30	4	F	1,26%	0,02	3	0,48%	1,50	40
31	4	G	1,02%	0,03	3	1,48%	1,24	30
32	4	H	1,26%	0,03	3	0,70%	8,17	60
33	5	A	99,49%	0,08	10	99,58%	8,08	5
34	5	B	99,49%	0,06	10	99,81%	8,43	10
35	5	C	77,69%	0,47	15	98,77%	18,77	7
36	5	D	77,63%	0,98	15	99,93%	53,24	15
37	5	E	99,47%	0,02	7	99,96%	33,62	20
38	5	F	99,45%	0,02	7	99,91%	90,05	40
39	5	G	96,64%	122,65	27	99,43%	14,15	30
40	5	H	94,63%	85,54	14	99,31%	7,33	60

continua...

Tabela A.4 - continuação

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
41	6	A	12,40%	0,06	10	15,83%	1,24	5
42	6	B	12,40%	0,06	10	11,32%	2,98	10
43	6	C	16,80%	0,47	14	6,26%	1,65	7
44	6	D	19,10%	0,97	16	3,30%	2,81	15
45	6	E	21,88%	0,02	7	12,63%	1,89	20
46	6	F	22,56%	0,02	7	11,90%	4,33	40
47	6	G	17,12%	122,67	28	17,71%	2,25	30
48	6	H	12,62%	85,64	15	10,00%	5,85	60
49	7	A	8,88%	0,03	3	3,48%	3,00	5
50	7	B	8,88%	0,03	3	81,64%	1,54	10
51	7	C	6,57%	0,14	6	46,02%	7,75	7
52	7	D	6,23%	0,19	6	82,85%	52,92	15
53	7	E	2,90%	0,02	6	89,20%	54,73	20
54	7	F	3,52%	0,02	6	91,28%	76,67	40
55	7	G	1,57%	117,87	25	90,43%	21,82	30
56	7	H	15,70%	84,57	17	88,02%	84,57	60
57	8	A	8,54%	0,03	3	92,56%	24,28	5
58	8	B	8,54%	0,03	3	89,79%	23,37	10
59	8	C	19,57%	0,47	15	68,63%	45,57	7
60	8	D	4,58%	0,97	15	59,87%	6,90	15
61	8	E	10,68%	0,02	6	99,60%	50,96	20
62	8	F	10,62%	0,02	3	96,40%	49,09	40
63	8	G	42,23%	119,33	28	98,33%	85,83	30
64	8	H	10,62%	0,03	3	62,80%	47,47	60
65	9	A	0,07%	0,06	10	49,33%	7,87	5
66	9	B	0,07%	0,06	10	89,46%	45,97	10
67	9	C	5,87%	0,47	13	40,94%	6,06	7
68	9	D	2,53%	0,97	15	72,24%	2,56	15
69	9	E	7,88%	0,06	7	86,80%	12,64	20
70	9	F	8,74%	0,02	7	99,98%	22,63	40
71	9	G	12,09%	119,97	21	81,29%	8,18	30
72	9	H	0,54%	84,97	14	94,25%	29,11	60

Tabela A.5: Resultados para o *Sistema 4*

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
1	1	A	99,80%	0,06	6	99,40%	1,58	5
2	1	B	99,80%	0,05	6	100,00%	1,85	10
3	1	C	99,80%	0,15	5	100,00%	3,84	7
4	1	D	99,80%	0,19	7	100,00%	2,91	15
5	1	E	99,90%	0,02	6	100,00%	1,42	20
6	1	F	99,60%	0,02	6	100,00%	1,09	40
7	1	G	99,60%	0,29	8	100,00%	5,98	30
8	1	H	99,50%	1,34	7	100,00%	11,15	60
9	2	A	65,83%	0,04	3	46,75%	1,47	5
10	2	B	65,83%	0,03	3	27,27%	0,65	10
11	2	C	37,96%	0,05	3	46,65%	1,62	7
12	2	D	37,96%	0,06	3	62,64%	2,06	15
13	2	E	0,20%	0,01	3	53,05%	4,40	20
14	2	F	48,15%	0,02	3	5,39%	0,89	40
15	2	G	38,96%	0,03	3	52,35%	2,21	30
16	2	H	37,96%	0,03	3	71,93%	6,39	60
17	3	A	64,44%	0,07	15	6,49%	2,50	5
18	3	B	64,44%	0,07	15	8,29%	0,80	10
19	3	C	29,17%	0,48	18	17,98%	1,51	7
20	3	D	46,25%	0,97	15	7,99%	1,62	15
21	3	E	48,45%	0,01	3	95,70%	3,45	20
22	3	F	49,15%	0,01	3	95,30%	8,76	40
23	3	G	65,03%	112,81	28	17,58%	3,32	30
24	3	H	49,15%	0,03	3	99,70%	10,52	60
25	4	A	0,20%	0,07	14	0,30%	0,75	5
26	4	B	0,20%	0,07	14	0,60%	0,79	10
27	4	C	0,20%	0,44	11	0,20%	1,45	7
28	4	D	0,20%	0,96	15	0,00%	1,89	15
29	4	E	0,20%	0,01	3	3,00%	0,74	20
30	4	F	0,20%	0,11	3	0,60%	1,84	40
31	4	G	0,20%	116,96	32	0,20%	1,14	30
32	4	H	0,30%	74,77	20	2,30%	4,31	60
33	5	A	99,10%	0,04	6	100,00%	5,09	5
34	5	B	99,10%	0,04	6	100,00%	1,08	10
35	5	C	98,60%	0,45	15	100,00%	7,47	7
36	5	D	98,70%	0,92	17	100,00%	4,83	15
37	5	E	98,70%	0,02	7	100,00%	4,19	20
38	5	F	94,31%	0,02	7	100,00%	7,92	40
39	5	G	100,00%	110,69	28	100,00%	4,06	30
40	5	H	100,00%	78,46	15	100,00%	11,80	60


continua...

Tabela A.5 - continuação

Exp.	Tarefa	Rede	GMDH			MLP		
			Acertos (%)	Tempo (s)	Neur.	Acertos (%)	Tempo (s)	Neur.
41	6	A	67,33%	0,02	3	2,90%	0,60	5
42	6	B	67,33%	0,02	3	0,20%	0,85	10
43	6	C	52,05%	0,13	5	4,50%	1,05	7
44	6	D	52,05%	0,18	5	45,75%	1,19	15
45	6	E	47,75%	0,02	3	29,37%	1,43	20
46	6	F	51,75%	0,02	3	51,35%	1,31	40
47	6	G	1,70%	0,33	6	57,54%	2,88	30
48	6	H	37,16%	1,48	7	33,37%	5,56	60
49	7	A	0,20%	0,04	6	99,80%	8,78	5
50	7	B	0,20%	0,04	6	100,00%	1,10	10
51	7	C	0,30%	0,13	5	99,90%	4,58	7
52	7	D	0,30%	0,23	7	99,40%	1,32	15
53	7	E	4,20%	0,03	6	100,00%	5,76	20
54	7	F	10,39%	0,02	6	100,00%	3,66	40
55	7	G	26,47%	0,32	8	100,00%	6,06	30
56	7	H	0,20%	1,61	7	100,00%	10,96	60
57	8	A	39,16%	0,16	15	94,71%	1,76	5
58	8	B	39,16%	0,11	15	92,61%	1,74	10
59	8	C	27,07%	0,78	12	96,60%	1,43	7
60	8	D	43,46%	1,71	18	94,01%	7,68	15
61	8	E	35,06%	0,02	7	95,70%	6,15	20
62	8	F	65,03%	0,01	3	97,60%	7,38	40
63	8	G	80,92%	107,15	23	94,11%	1,46	30
64	8	H	65,03%	0,04	3	98,40%	7,52	60
65	9	A	84,12%	0,05	6	100,00%	1,41	5
66	9	B	84,12%	0,05	6	100,00%	7,65	10
67	9	C	98,40%	0,46	15	100,00%	3,45	7
68	9	D	98,40%	0,94	15	100,00%	2,45	15
69	9	E	82,02%	0,03	7	100,00%	3,53	20
70	9	F	84,92%	0,04	7	100,00%	5,40	40
71	9	G	100,00%	105,19	26	100,00%	3,50	30
72	9	H	99,80%	72,52	16	100,00%	12,91	60

B OS VINTE AMINOÁCIDOS E SUAS ESTRUTURAS

Tabela B.1: Os vinte aminoácidos comumente encontrados nas proteínas e suas estruturas [86].

A – Alanina	R – Arginina	N – Asparagina	D – Aspartato
$^+H_3N-CH-COO^-$ $ $ CH_3	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH_2 $ $ CH_2 $ $ NH $ $ $C=NH_2^+$ $ $ NH_2	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ C $/ \quad \backslash$ $H_2N \quad O$	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ COO^-
C – Cisteína	E – Glutamato	Q – Glutamina	G – Glicina
$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ SH	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH_2 $ $ COO^-	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH_2 $ $ C $/ \quad \backslash$ $H_2N \quad O$	$^+H_3N-CH-COO^-$ $ $ H
H – Histidina	I – Isoleucina	L – Leucina	K – Lisina
$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ $C-NH$ $ \quad \backslash$ $C-N \quad C-H$ $ \quad $ $H \quad H^+$	$^+H_3N-CH-COO^-$ $ $ $H-C-CH_3$ $ $ CH_2 $ $ CH_3	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH $/ \quad \backslash$ $H_3C \quad CH_3$	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH_2 $ $ CH_2 $ $ CH_2 $ $ NH_3^+
M – Metionina	F – Fenilalanina	P – Prolina	S – Serina
$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ CH_2 $ $ S $ $ CH_3	$^+H_3N-CH-COO^-$ $ $ CH_2 $ $ 	$^+H_2N-CH-COO^-$ $ \quad $ $H_2C \quad CH_2$ $\quad \backslash$ $\quad \quad CH_2$	$^+H_3N-CH-COO^-$ $ $ CH_2OH

continua...

Tabela B.1 - continuação

T - Treonina	W - Triptofano	Y - Tirosina	V - Valina
$ \begin{array}{c} +\text{H}_3\text{N}-\text{CH}-\text{COO}^- \\ \\ \text{H}-\text{C}-\text{OH} \\ \\ \text{CH}_3 \end{array} $	$ \begin{array}{c} +\text{H}_3\text{N}-\text{CH}-\text{COO}^- \\ \\ \text{CH}_2 \\ \\ \text{C}=\text{CH} \\ \quad \\ \text{NH} \\ \\ \text{C}_6\text{H}_5 \end{array} $	$ \begin{array}{c} +\text{H}_3\text{N}-\text{CH}-\text{COO}^- \\ \\ \text{CH}_2 \\ \\ \text{C}_6\text{H}_4 \\ \\ \text{OH} \end{array} $	$ \begin{array}{c} +\text{H}_3\text{N}-\text{CH}-\text{COO}^- \\ \\ \text{CH} \\ / \quad \backslash \\ \text{H}_3\text{C} \quad \text{CH}_3 \end{array} $

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Kunter, M. L. *Astronomy. A Physical Perspective*, capítulo 5, página 91. Cambridge, Cambridge, 1. ed., agosto 2003.
- [2] Hogg, H. S. Out of Old Books (The Titius-Bode Law and the Discovery of Ceres). *Journal of the Royal Astronomical Society of Canada*, 42:241–246, outubro 1948.
- [3] Forbes, E. G. Gauss and the Discovery of Ceres. *Journal for the History of Astronomy*, 2:195–199, 1971.
- [4] Coffey, J. The First Asteroid Discovered. <http://www.universetoday.com/35925/the-first-asteroid-discovered/>, julho 2009.
- [5] Gilat, A. e Subramaniam, V. *Métodos Numéricos para Engenheiros e Cientistas*, capítulo 5 - Ajuste de Curvas e Interpolação, páginas 199–251. Bookman, Porto Alegre, 1. ed., 2008.
- [6] Gabor, D., Wilby, W. e Woodcock, R. A universal non-linear filter, predictor and simulator which optimizes itself by a learning process. *Proceedings of the IEE - Part B: Electronic and Communication Engineering*, 108(40):422–435, julho 1961.
- [7] Oliver, A. e Fatmi, H. Nonlinear adaptive filtering by the gabor-kolmogorov method. In *Control 1991. Control '91., International Conference on*, volume 2, páginas 1105–1110, março 1991.
- [8] Ivakhnenko, A. G. Heuristic Self-Organization in Problems of Engineering Cybernetics. *Automatica*, 6:207–219, 1970.
- [9] Haykin, S. *Neural Networks: A Comprehensive Foundation*, capítulo 1 – Introduction, páginas 23–71. Pearson, Singapore, 2. ed., 1999.
- [10] Widrow, B., Hoff, M. E., et al. Adaptive switching circuits. <http://isl-www.stanford.edu/~widrow/papers/c1960adaptiveswitching.pdf>, 1960.

- [11] Haykin, S. *Redes Neruais, princípios e prática*, capítulo 3 – Perceptrons de camada única, páginas 142–181. Bookman, São Paulo, 2001.
- [12] Ivakhnenko, A. G. Polynomial theory of complex systems. *Systems, Man and Cybernetics, IEEE Transactions on*, 1(4):364–378, outubro 1971.
- [13] Farlow, S. J. The GMDH Algorithm of Ivakhnenko. *The American Statistician*, 35(4):210–215, novembro 1981.
- [14] Ivakhnenko, A. G. e Ivakhnenko, G. A. The review of problemas solvable with algorithms of the Group Method of Data Handling (GMDH). *Pattern Recognition and Image Analysis*, 5(4):527–535, 1995.
- [15] Kondo, T. e Pandya, A. GMDH-type neural networks with a feedback loop and their application to the identification of large-spatial air pollution patterns. In *SICE 2000. Proceedings of the 39th SICE Annual Conference. International Session Papers*, páginas 19–24, 2000.
- [16] Sakaguchi, A., Yamamoto, T., Fujii, K. e Monden, Y. Evolutionary GMDH-based identification for nonlinear systems. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 6, páginas 5812–5817, outubro 2004.
- [17] Abdel-Aal, R., Elhadidy, M. e Shaahid, S. Modeling and forecasting the mean hourly wind speed time series using GMDH-based abductive networks. *Renewable Energy*, 34(7):1686–1699, 2009.
- [18] Ivakhnenko, A. G., Ivakhnenko, G. A. e Müller, J.-A. Self-organization of neural networks with active neurons. *Pattern Recognition and Image Analysis*, 4(2):185–195, 1994.
- [19] Mingzhu, Z., Changzheng, H. e Liatsis, P. A D-GMDH model for time series forecasting. *Expert Systems with Applications*, 39(5):5711–5716, 2012.
- [20] Hyndman, R. Time series data library. <http://datamarket.com/data/list/?q=provider:tsdl>, outubro 2012.
- [21] Lee, S. H., No, Y. G., Na, M. G., Ahn, K.-I. e Park, S.-Y. Diagnostics of loss of coolant accidents using SVC and GMDH models. *Nuclear Science, IEEE Transactions on*, 58(1):267–276, fevereiro 2011.
- [22] Madandoust, R., Ghavidel, R. e Nariman-Zadeh, N. Evolutionary design of generalized GMDH-type neural network for prediction of concrete compressive strength using upv. *Computational Materials Science*, 49(3):556–567, 2010.

- [23] Elattar, E., Goulermas, J. e Wu, Q. Generalized locally weighted GMDH for short term load forecasting. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(3):345–356, maio 2012.
- [24] Kondo, T., Kondo, C., Takao, S. e Ueno, J. Feedback GMDH-type neural network algorithm and its application to medical image analysis of cancer of the liver. *Artificial Life and Robotics*, 15:264–269, 2010. 10.1007/s10015-010-0805-8.
- [25] Chaudhuri, R. A combinatorial GMDH approach to identification, modelling and prediction of money market state variables. In *International Conference on Technology and Business Management*, páginas 208–221, março 2012.
- [26] Lin, J.-S. A novel design of wafer yield model for semiconductor using a GMDH polynomial and principal component analysis. *Expert Systems with Applications*, 39(8):6665–6671, 2012.
- [27] Chen, B., Wang, W. e Qin, Q. Stereo vision calibration based on GMDH neural network. *Appl. Opt.*, 51(7):841–845, março 2012.
- [28] Amanifard, N., Nariman-Zadeh, N., Borji, M., Khalkhali, A. e Habibdoust, A. Modelling and Pareto optimization of heat transfer and flow coefficients in microchannels using GMDH type neural networks and genetic algorithms. *Energy Conversion and Management*, 49(2):311–325, 2008.
- [29] Bueno, E. I. *Group Method of Data Handling (GMDH) e Redes Neurais na Monitoração e Detecção de Falhas em sensores de centrais nucleares*. PhD thesis, Instituto de Pesquisas Energéticas e Nucleares, Universidade de São Paulo, São Paulo, 2011.
- [30] Sakaguchi, A. e Yamamoto, T. A design of predictive PID control systems using GA and GMDH network. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 1, páginas 266–271, 2002.
- [31] Puig, V., Witczak, M., Nejjari, F., Quevedo, J. e Korbicz, J. A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test. *Engineering Applications of Artificial Intelligence*, 20(7):886–897, 2007.
- [32] Witczak, M., Korbicz, J., Mrugalski, M. e Patton, R. J. A GMDH neural network-based approach to robust fault diagnosis: Application to the DAMADICS benchmark problem. *Control Engineering Practice*, 14(6):671–683, 2006. A Benchmark Study of Fault Diagnosis for an Industrial Actuator.

- [33] Pham, D. T. e Liu, X. Modelling and prediction using GMDH networks of Adalines with nonlinear preprocessors. *International Journal of Systems Science*, 25:1743–1759, 1994.
- [34] GMDH Software. http://www.gmdh.net/GMDH_sof.htm, novembro 2012.
- [35] Zelezny, A. NeroNet GMDH for Java. <http://www.unet.univie.ac.at/~a9404026/andy/>, novembro 1999.
- [36] Peak Consulting. <http://www.peakconsulting.com/>, novembro 2012.
- [37] Microsoft Corporation. Microsoft Visual Basic for Applications. [http://msdn.microsoft.com/en-us/library/office/aa188202\(v=office.10\).aspx](http://msdn.microsoft.com/en-us/library/office/aa188202(v=office.10).aspx), novembro 2012.
- [38] Microsoft Corporation. Microsoft Excel. <http://office.microsoft.com/en-us/excel/>, novembro 2012.
- [39] Ward Systems Group. Neuroshell Trade. <http://www.neuroshell.com/learnmore.asp>, novembro 2012.
- [40] Tetko, I. V., Gasteiger, J., Todeschini, R., Mauri, A., Livingstone, D., Ertl, P., Palyulin, V. A., Radchenko, E. V., Zefirov, N. S., Makarenko, A. S., Tanchuk, V. Y. e Prokopenko, V. V. Virtual computational chemistry laboratory - design and description. *Journal of Computer-Aided Molecular Design*, 19(6):453–463, junho 2005.
- [41] Tetko, I. V., Gasteiger, J., Todeschini, R., Mauri, A., Livingstone, D., Ertl, P., Palyulin, V. A., Radchenko, E. V., Zefirov, N. S., Makarenko, A. S., Tanchuk, V. Y. e Prokopenko, V. V. Polynomial Neural Netowks. <http://www.vcclab.org/lab/pnn/>, novembro 2012.
- [42] Zaptron Systems. DataX Data Mining Software Suite. <http://www.zaptron.com/datax/dataxcatalog.htm>, novembro 2012.
- [43] Geos Research Group. GMDH Shell. <http://www.gmdhshell.com/>, novembro 2012.
- [44] National Institute for Strategic Studies – National Academy of Sciences of Ukraine. Group Method of Data Handling. <http://www.gmdh.net/>, novembro 2012.
- [45] Pham, D. T. e Liu, X. *Neural Networks for Identification, Prediction, and Control*. Springer-Verlag, Berlin, 1. ed., 1995.

- [46] MathWorks. MathWorks' website. <http://www.mathworks.com/>, janeiro 2011.
- [47] Free Software Foundation, Inc. The free software definition. <http://www.gnu.org/philosophy/free-sw.html.en>, julho 2012.
- [48] Braga, A. L. GMDH-T-ANN - GMDH Type Artificial Neural Network. <http://opengmdh.org/wiki/GMDH-T-ANN>, junho 2011.
- [49] Open Source Initiative. Common development and distribution license. <http://www.opensource.org/licenses/CDDL-1.0>, janeiro 2012.
- [50] Braga, A. L. Comparing artificial neural network implementations for prediction and modeling of dynamical systems. In *21st International Congress of Mechanical Engineering - COBEM 2011*, 2011.
- [51] Braga, A. L., Llanos, C. H. e dos S. Coelho, L. *ABCМ Symposium Series in Mechatronics*, volume 5, capítulo III.3 – Comparing Artificial Neural Network Implementations for Prediction and Modeling of Dynamical Systems, páginas 601–609. Associação Brasileira de Engenharia e Ciências Mecânicas, ABCM, Rio de Janeiro, oct 2012.
- [52] Dorn, M., Braga, A. L., Llanos, C. H. e dos S. Coelho, L. A GMDH polynomial neural network-based method to predict approximate three-dimensional structures of polypeptides. *Expert Systems with Applications*, (0):–, 2012.
- [53] Braga, A. L. S., Arias-Garcia, J., Llanos, C., Dorn, M., Foltran, A. e Coelho, L. S. Hardware implementation of GMDH-type artificial neural networks and its use to predict approximate three-dimensional structures of proteins. In *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2012 7th International Workshop on*, páginas 1–8, julho 2012.
- [54] Vissikirsky, V. A. e Stepashko, V. S. Growth dynamics of trees irrigated with wastewater: GMDH modeling, assessment, and control issues. *Instrumentation Science and Technology*, 2:229–249, 2005.
- [55] Sung-Kwun, O. e Pedrycz, W. The design of self-organizing polynomial neural networks. *Information Sciences*, 141(3-4):237–258, 2002.
- [56] Chua, L. e Ng, C.-Y. Frequency domain analysis of nonlinear systems: general theory. *Electronic Circuits and Systems, IEE Journal on*, 3(4):165–185, julho 1979.
- [57] Madala, H. R. e Ivakhnenko, A. G. *Inductive Learning Algorithms for Complex Systems Modeling*, capítulo 3. CRC Press, Boca Raton, Florida, 1994.

- [58] Mehra, R. K. Group method of data handling (GMDH): Review and experience. In *Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications, 1977 IEEE Conference on*, volume 16, páginas 29–34, dezembro 1977.
- [59] Braga, A., Llanos, C., Göhringer, D., Obie, J., Becker, J. e Hubner, M. Performance, accuracy, power consumption and resource utilization analysis for hardware / software realized artificial neural networks. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on*, páginas 1629–1636, setembro 2010.
- [60] Press, W. H., Teukolsky, S. A., Vetterling, W. T. e Flannery, B. P. *Métodos Numéricos Aplicados*, capítulo 2 - Solução de Equações Algébricas Lineares, páginas 58–129. Bookman, Porto Alegre, 1. ed., 2011.
- [61] Techniques Used for Linear Algebra. <http://www.gnu.org/software/octave/doc/interpreter/Techniques-Used-for-Linear-Algebra.html#Techniques-Used-for-Linear-Algebra>, novembro 2012.
- [62] Gilat, A. e Subramaniam, V. *Métodos Numéricos para Engenheiros e Cientistas*, capítulo 4 - Resolvendo um sistema de equações lineares, páginas 113–197. Bookman, Porto Alegre, 1. ed., 2008.
- [63] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. e Sorensen, D. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3. ed., 1999.
- [64] Moler, C. MATLAB Incorporates LAPACK. http://www.mathworks.com/company/newsletters/news_notes/clevescorner/winter2000.cleve.html, 2000.
- [65] Narendra, K. S. e Parthasarathy, K. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 1(1):4–27, março 1990.
- [66] Anfinsen, C., Haber, E., Sela, M. e White, F. H. J. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A.*, 47:1309, 1961.
- [67] Branden, C. e Tooze, J. *Introduction to protein structure*. Garland Publishing Inc., New York, 2. ed., 1998.

- [68] Floudas, C., Fung, H., McAllister, S., Moennigmann, M. e Rajgaria, R. Advances in protein structure prediction and de novo protein design: A review. *Chem. Eng. Sci.*, 61(3):966, 2006.
- [69] Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A. e Yannakakis, M. On the complexity of protein folding. *J. Comput. Biol.*, 5(3):423, 1998.
- [70] Osguthorpe, D. Ab initio protein folding. *Curr. Opin. Struct. Biol.*, 10(2):146, 2000.
- [71] Rohl, C., Strauss, C., Misura, K. e Baker, D. Protein structure prediction using rosetta. *Methods Enzymol.*, 383(2):66, 2004.
- [72] Srinivasan, R. e Rose, G. Linus - a hierarchic procedure to predict the fold of a protein. *Proteins: Struct., Funct., Bioinf.*, 22(2):81, 1995.
- [73] Bowie, J. U., Luthy, R. e Eisenberg, D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253(5016):164, 1991.
- [74] Jones, D., Taylor, W. e Thornton, J. A new approach to protein fold recognition. *Nature*, 358(6381):86, 1992.
- [75] Bryant, S. H. e Altschul, S. Statistics of sequence-structure threading. *Curr. Opin. Struct. Biol.*, 5(2):236, 1995.
- [76] Turcotte, M., Muggleton, S. e Sternberg, M. *Application of inductive logic programming to discover rules governing the three-dimensional topology of protein structure*. Springer, Madison, July 1998.
- [77] Martí-Renom, M., Stuart, A., Fiser, A., Sanchez, A., Mello, F. e Sali, A. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.*, 29(16):291, 2000.
- [78] Sánchez, R. e Sali, A. Advances in comparative protein-structure modeling. *Curr. Opin. Struct. Biol.*, 7(2):206, 1997.
- [79] Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bath, T., Weissig, H., Shindyalov, I. e Bourne, P. The protein data bank. *Nucleic Acids Res.*, 28(1):235–242, 2000.
- [80] Finkelstein, A. e Ptitsyn, O. Why do globular proteins fit the limited set of folding patterns? *Prog. Biophys. Mol. Biol.*, 50(3):171, 1987.
- [81] Levitt, M. e Chothia, C. Structural patterns in globular proteins. *Nature*, 261(5561):552, 1976.

- [82] Ngo, J., Marks, J. e Karplus, M. The protein folding problem and tertiary structure prediction. In Merz Jr, K. e Grand, S., editors, *Computational complexity, protein structure prediction and the Levinthal Paradox*, página 435. Birkhauser, Boston, 1997.
- [83] Xu, D., Zhang, J., Roy, A. e Zhang, A. Automated protein structure modeling in casp9 by i-tasser pipeline combined with quark-based ab initio folding and fg-md-based structure refinement. *Proteins: Struct., Funct., Bioinf.*, 79(10):147, 2011.
- [84] Cozzetto, D., Kryshchuk, A., Fidelis, K., Moult, J., Rost, B. e Tramontano, A. Evaluation of template-based models in casp8 with standard measures. *Proteins: Struct., Funct., Bioinf.*, 77(9):18, 2009.
- [85] Dorn, M. e Norberto de Souza, O. A3N: an artificial neural network n-gram-based method to approximate 3-d polypeptides structure prediction. *Expert Syst. Appl.*, 37(12):7497, 2010.
- [86] Bowen, R. Amino acids. Hypertexts for Biomedical Sciences – <http://www.vivo.colostate.edu/hbooks/molecules/aminoacids.html>, abril 2012.
- [87] Bowen, R. The structure of proteins. Hypertexts for Biomedical Sciences – <http://www.vivo.colostate.edu/hbooks/genetics/biotech/basics/prostruct.html>, abril 2012.
- [88] Clark, J. The structure of proteins. Chemguide – <http://www.chemguide.co.uk/organicprops/aminoacids/proteinstruct.html>, abril 2012.
- [89] Tramontano, A. *Protein structure prediction*. John Wiley and Sons, Inc., Weinheim, 1. ed., 2006.
- [90] Lesk, A. M. *Introduction to Bioinformatics*. Oxford University Press Inc., New York, 1. ed., 2002.
- [91] Pauling, L., Corey, R. e Branson, H. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. U.S.A.*, 37(4):205, 1951.
- [92] Pauling, L. e Corey, R. The pleated sheet, a new layer configuration of polypeptide chains. *Proc. Natl. Acad. Sci. U.S.A.*, 37(5):251, 1951.
- [93] Dorn, M. e Norberto de Souza, O. Mining the protein data bank with cref to predict approximate 3-d structures of polypeptides. *Int. J. Data Min. and Bioin.*, 4(3):281, 2010.

- [94] Dorn, M. e Norberto de Souza, O. Cref: A central-residue-fragment-based method for predicting approximate 3-d polypeptides structures. *Proceedings of the ACM Symposium on Applied Computing*, página 1261, 2008.
- [95] Dicionário Priberam da Língua Portuguesa. <http://www.priberam.pt/dlpo/>, agosto 2012.
- [96] Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W. e Lipman, D. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389, 1997.
- [97] Dayhoff, M., Schwartz, R. e Orcutt, B. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5(3):345–352, 1978.
- [98] Hutchinson, E. e Thornton, J. Promotif: A program to identify and analyze structural motifs in proteins. *Protein Sci.*, 5(2):212–220, 1996.
- [99] Kabsch, W. e Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.
- [100] Witten, I. H. e Frank, E. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Series in Data Management Systems, Oxford, UK, 2. ed., 2006.
- [101] Ramachandran, G. e Sasisekharan, V. Conformation of polypeptides and proteins. *Adv. Protein Chem.*, 23:283–438, 1968.
- [102] Scratch. Scratch protein predictor webpage, janeiro 2012.
- [103] Ponder, J. Jay Ponder Lab, Department of Biochemistry and Molecular Biophysics, Washington University School of Medicine, Saint Louis, 2010.
- [104] Ponder, J. e Richards, F. An efficient newton-like method for molecular mechanics energy minimization of large molecules. *J. Comput. Chem.*, 8(7):1016, 1987.
- [105] MacKerell, A. J., Banavali, N. e Foloppe, N. Development and current status of the charmm force field for nucleic acids. *Biopolymers*, 56(4):257, 2001.
- [106] Opella, S., Marassi, F., Gesell, J., Valente, A., Kim, Y., Oblatt-Montal, M. e Montal, M. Structures of the m2 channel-lining segments from nicotinic acetylcholine and nmda receptors by nmr spectroscopy. *Nat.Struct.Biol.*, 6:374–379, 1999.

- [107] Starovasnik, M., Braisted, A. e Wells, J. Structural mimicry of a native protein by a minimized binding domain. *Proc.Natl.Acad.Sci.USA*, 94:10080–10085, 1997.
- [108] Blanc, E., Fremont, V., Sizun, P., Meunier, S., Van Rietschoten, J., Thevand, A., Bernassau, J. e Darbon, H. Solution structure of p01, a natural scorpion peptide structurally analogous to scorpion toxins specific for apamin-sensitive potassium channel. *Proteins*, 24:359–369, 1996.
- [109] Clarke, N., Kissinger, C., Desjarlais, J., Gilliland, G. e Pabo, C. Structural studies of the engrailed homeodomain. *Protein Sci.*, 3:1779–1787, 1994.
- [110] Pastor, M., Lopez de la Paz, M., Lacroix, E., Serrano, L. e Perez-Paya, E. Combinatorial approaches: a new tool to search for highly structured beta-hairpin peptides. *Proc.Natl.Acad.Sci.USA*, 99:614–619, 2002.
- [111] Neidigh, J., Fesinmeyer, R. e Andersen, N. Designing a 20-residue protein. *Nat.Struct.Biol.*, 9:425–430, 2002.
- [112] Park, S., Mrse, A., Nevzorov, A., Mesleh, M., Oblatt-Montal, M., Montal, M. e Opella, S. Three-dimensional structure of the channel-forming trans-membrane domain of virus protein u (vpu) from hiv-1. *J.Mol.Biol.*, 333:409–424, 2003.
- [113] Pymol Website. Pymol website, janeiro 2012.
- [114] Martin, A. C. R. Profit web page, 2012.
- [115] MathWorks. MATLAB’s Neural Netowrk Toolbox. <http://www.mathworks.com/help/toolbox/nnet>, 2012.
- [116] Dorn, M., Breda, A. e Norberto de Souza, O. A hybrid method for the protein structure prediction problem. *Lect. Notes Bioinf.*, 5167:47, 2008.
- [117] Fan, H. e Mark, A. Refinement of homology-based protein structures by molecular dynamics simulation techniques. *Protein Sci.*, 13:211–220, 2004.
- [118] Laskowski, R., MacArthur, M., Moss, D. e Thornton, J. Procheck: a program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.*, 26(2):283–291, 1993.
- [119] van Gunsteren, W. e Berendsen, H. Computer simulation of molecular dynamics: methodology, applications, and perspectives in chemistry. *Angew. Chem., Int. Ed. Engl.*, 29(9):992, 1990.
- [120] Rapaport, D. C. *The art of molecular dynamics simulation*. Cambridge University Press, Cambridge, 2. ed., 2004.

- [121] Koza, J. R. *Molecular dynamics simulations: Elementary methods*. John Wiley and Sons, Inc., New York, 1. ed., 1992.
- [122] Estrin, D., Govidan, R. e Heidemann, J. Embedding the internet: introduction. *Communications of the ACM*, 43(5):38–41, maio 2000.
- [123] Atencia, M., Boumeridja, H., Joya, G., García-Lagos, F. e Sandoval, F. FPGA implementation of a systems identification module based upon hopfield networks. *Neurocomputing*, 70(16–18):2828–2835, 2007.
- [124] Kurdthongmee, W. A novel hardware-oriented kohonen SOM image compression algorithm and its FPGA implementation. *J. Syst. Archit.*, 54(10):983–994, outubro 2008.
- [125] Saif, S., Abbas, H. M., Nassar, S. M. e Wahdan, A. A. An FPGA implementation of a neural optimization of block truncation coding for image/video compression. *Microprocess. Microsyst.*, 31(8):477–486, dezembro 2007.
- [126] Grossi, G. e Pedersini, F. FPGA implementation of a stochastic neural network for monotonic pseudo-boolean optimization. *Neural Networks*, 21(6):872–879, 2008.
- [127] Kilts, S. *Advanced FPGA Design: Architecture, Implementation, and Optimization*. Wiley-IEEE Press, 1. ed., agosto 2007.
- [128] Saichand, V., Nirmala, D., Arumugam, S. e Mohankumar, N. FPGA realization of activation function for artificial neural networks. In *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, volume 3, páginas 159–164, novembro 2008.
- [129] Omondi, A. R. e RAJAPAKSE, J. C. *ASIC vs. FPGA neurocomputers*, capítulo 1, página 9. Springer, 2006.
- [130] Arias-Garcia, J., Llanos, C., Ayala-Rincon, M. e Jacobi, R. A fast and low cost architecture developed in FPGAs for solving systems of linear equations. In *Circuits and Systems (LASCAS), 2012 IEEE Third Latin American Symposium on*, páginas 1–4, março 2012.
- [131] Mentor Graphics. Modelsim - advanced simulation and debugging. <http://model.com/>, maio 2012.
- [132] Xilinx Inc. Microblaze soft processor core. <http://www.xilinx.com/tools/microblaze.htm>.

- [133] IBM Corporation. *32-bit Processor Local Bus Architecture Specifications Version 2.9*.
- [134] Xilinx Inc. Xilinx core generator system. <http://www.xilinx.com/tools/coregen.htm>, maio 2012.
- [135] Xilinx Inc. Xilinx platform studio. <http://www.xilinx.com/tools/xps.htm>, maio 2012.
- [136] Xilinx Inc. Xilinx University Program XUPV5-LX110T Development System. <http://www.xilinx.com/univ/xupv5-lx110t.htm>, abril 2012.
- [137] Chagot, B., Pimentel, C., Dai, L., Pil, J., Tytgat, J., Nakajima, T., Corzo, G., Darbon, H. e Ferrat, G. An unusual fold for potassium channel blockers: Nmr structure of three toxins from the scorpion *opisthacanthus madagascariensis*. *Biochem. J.*, 388:263–271, 2005.
- [138] Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C. e Ferrin, T. E. UCSF Chimera—a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, outubro 2004.
- [139] Zadeh, L. A. Optimum nonlinear filters. *Journal of Applied Physics*, 24(4):396–404, 1953.
- [140] Riedmiller, M. e Braun, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, páginas 586–591 vol.1, 1993.
- [141] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.