

# Usando uma rede GMDH como uma ferramenta para seleção de variáveis de entrada de uma rede neural

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Aluno: David John Freitas Fraga**  
**Orientador: Prof. Mêuser Jorge Silva Valença**

**David John Freitas Fraga**

**Usando uma rede GMDH como uma  
ferramenta para seleção de variáveis  
de entrada de uma rede neural**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, novembro de 2009.**

*A Steven Pinker, um visionário.*

# Agradecimentos

Agradeço a Deus pelas oportunidades que Ele sempre me dá.

Agradeço a minha família, especialmente meu pai, porque se ele não tivesse entrado em acordo comigo eu não estaria terminando minha graduação agora.

Agradeço aos amigos que me suportaram quando estive em épocas de grande pressão na minha vida pessoal e acadêmica.

Agradeço ao professor Mêuser, que me deu a oportunidade de ser seu orientando.

E agradeço a todos que torcem pelo meu sucesso.

# Resumo

As Redes Neurais estão cada vez mais sendo utilizadas nos sistemas complexos e os tornando mais sofisticados e eficientes. Áreas distintas como a área médica, de segurança, de administração, de economia, de redes de computadores usam essa técnica de Inteligência Artificial. Para melhorar sua confiabilidade, o projetista às vezes tem que realizar uma série de procedimentos, visando eliminar os ruídos e os elementos irrelevantes para serem processados pela rede neural. A seleção de entradas relevantes tende a ser uma tarefa mais manual, comum e trabalhosa, e com isso, mais passível de erros. A ferramenta desenvolvida nesse projeto utiliza uma técnica de aprendizagem indutiva para modelagem de sistemas complexos chamada Redes GMDH. Com essa ferramenta é possível selecionar as variáveis mais relevantes a serem usadas numa rede neural. Esse passo ainda é inicial, visto que a metodologia usada não forneceu claramente toda a vantagem de usar essas redes, mas é suficiente para perceber que é uma técnica promissora e bem consolidada.

# Abstract

Neural Networks are increasingly being used in complex systems and making them more sophisticated and efficient. Different areas such as medical, security, administration, economics, computer networks use this Artificial Intelligence technique. To improve its reliability, the designer sometimes has to perform a series of procedures designed to eliminate noise and irrelevant information to be processed by the neural network. The selection of relevant inputs tends to be a manual task more common and laborious, and therefore more prone to errors. The tool developed in this project uses an inductive learning technique for modeling complex systems called GMDH networks. With this tool you can select the most relevant variables to be used in a neural network. This step is still the beginning, because the methodology did not manage to show any clear advantage of using these networks, but it is enough to realize that is a promising and well established.

# Sumário

## Conteúdo

<b>Resumo</b>	<b>5</b>
<b>Abstract</b>	<b>6</b>
<b>Sumário</b>	<b>7</b>
<b>Índice de Figuras</b>	<b>9</b>
<b>Índice de Tabelas</b>	<b>11</b>
<b>Tabela de Símbolos e Siglas</b>	<b>12</b>
<b>Capítulo 1 - Introdução</b>	<b>13</b>
1.1    Objetivos	13
1.2    Descrição do problema	14
1.3    Próximos Capítulos	15
<b>Capítulo 2 - A Redes Neurais – Conceitos básicos</b>	<b>17</b>
<b>Capítulo 3 - A cibernética e as redes GMDH</b>	<b>21</b>
3.1    Fundamentação Teórica	22
3.2    As redes GMDH	23
3.3    Algoritmos GMDH	25
3.4    Um algoritmo GMDH modificado	27
<b>Capítulo 4 - Experimento</b>	<b>30</b>
4.1    Funcionamento do algoritmo	33
<b>Capitulo 5 - Resultados</b>	<b>36</b>
<b>Capítulo 6 Conclusão e Trabalhos Futuros</b>	<b>46</b>
6.1    Dificuldades encontradas	46
6.2    Conclusões sobre o experimento	46

6.3	Idéias para trabalhos futuros	47
6.4	Considerações finais	47
	<b>Bibliografia</b>	<b>49</b>



# Índice de Figuras

Figura 1. Neurônio biológico.....	17
Figura 2. Perceptron segundo Roseblatt.....	18
Figura 3. Rede MLP.....	19
Figura 4. Sistema cibernético nos seres humanos. Os hormônios coordenando a homeostase glicídica.....	21
Figura 5. Sistema cibernético não controlado.....	22
Figura 6. Rede GMDH com duas camadas ocultas.....	24
Figura 7. Equação de regressão.....	27
Figura 8. Esquema básico da propagação de variáveis.....	28
Figura 9. Exemplo de organização de um arquivo de entrada de dados para a rede GMDH. Acima à esquerda e abaixo, os dados não estão prontos para serem usados. Acima à direita, os dados estão prontos para conversão.....	33
Figura 10. Rede GMDH treinada (bolas pretas = elementos eliminados).....	35
Figura 11. Comparação entre as combinações de entradas selecionadas pela rede GMDH padrão.....	37
Figura 12. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 3, 4 e 5 pela rede GMDH padrão.....	37
Figura 13. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1 e 2 pela rede GMDH padrão.....	38
Figura 14. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2, 3, 4 e 5 pela rede GMDH padrão.....	38
Figura 15. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 7 pela rede GMDH padrão.....	39
Figura 16. Comparação entre todas as combinações de entradas selecionadas pela rede GMDH modificada.....	40

Figura 17. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 3 pela rede GMDH modificada.....	40
Figura 18. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2, 4, 5, 6 e 7 pela rede GMDH modificada.....	40
Figura 19. Comparação entre todas as combinações de entradas selecionadas pela GMDH padrão .....	41
Figura 20. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 5 pela rede GMDH padrão.....	41
Figura 21. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 7 pela rede GMDH padrão.....	42
Figura 22. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1 e 2 pela rede GMDH padrão.....	42
Figura 23. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2, 3, 4, 5 e 6 pela rede GMDH padrão.....	43
Figura 24. Comparação entre todas as combinações de entradas selecionadas pela GMDH modificada .....	43

# Índice de Tabelas

Tabela 1. Combinações realizadas com os dados da represa de Sobradinho com 8 entradas.....	38
Tabela 2. Combinações realizadas com os dados da represa de Três Marias com 8 entradas.....	38
Tabela 3. Combinações realizadas com os dados da represa de Sobradinho com 10 entradas.....	45
Tabela 4. Combinações realizadas com os dados da represa de Três Marias com 10 entradas.....	46
Tabela 5. Combinações realizadas com os dados da represa de Sobradinho com 10 entradas sendo 2 ruidosas (entrada 2 e 6).....	46
Tabela 6. Combinações realizadas com os dados da represa de Três Marias com 10 entradas sendo 2 ruidosas (entrada 3 e 9).....	46

# Tabela de Símbolos e Siglas

I.A. – Inteligência Artificial

GMDH – Group Method of Data Handling

SOM – Self-Organizing Map

MLP – Multi-Layer Perceptron

CCS – Combined Control Systems

COMBI – Combinatorial algorithm

MIA – Multilayered Interaction

OSA – Objective System Analysis

ONS – Operador Nacional do Sistema Elétrico

CVS – tipo de arquivo texto com os dados separados por vírgulas

BMSE – Batch Mean Square Error

# Capítulo 1 - Introdução

A aplicação de técnicas de aprendizagem estatística em Inteligência Artificial (IA) foi uma área de pesquisa no início da década de 70, mas se separou da corrente principal da IA à medida que esse último campo se concentrou em métodos simbólicos [1].

As redes neurais se constituem em uma técnica de IA cuja utilização prática está se tornando cada vez mais presente no nosso dia-a-dia. Elas apresentam uma abordagem alternativa aos métodos estatísticos tradicionais em solução de problemas de previsão de séries temporais. Aplicações realizadas com redes neurais artificiais têm apresentado desempenho satisfatório em diversas áreas de pesquisa como: classificação, reconhecimento, aproximação de funções, processamento de séries temporais [2].

Apesar de cada vez mais tecnologias e técnicas surgirem, as redes neurais se tornaram bastante popular por fornecer resultados satisfatórios aos problemas que dificilmente seriam resolvidos utilizando computação convencional, além do que muitas pesquisas são produzidas por ano mostrando situações onde a rede neural pode ser usada.

## 1.1 Objetivos

Em geral, antes dos dados serem processados pela rede neural, é necessária a realização de um pré-processamento. A seleção de variáveis de entrada automatizada é o objetivo geral e é o ponto chave para a ferramenta em desenvolvimento nesse projeto. Lidar com esses dados já é possível através das redes Group Method of Data Handling (GMDH) [3.2].

Nesse Trabalho de Conclusão de Curso, mostraremos a grande utilidade que a rede GMDH pode ter em situações que sejam necessários um processamento

complexo dos dados de entrada, reduzindo os erros por fator humano ou por uso de técnicas menos eficientes e facilitando o trabalho dos projetistas. Para isso será desenvolvida uma ferramenta para selecionar as variáveis de entrada mais relevantes usando uma rede GMDH com mudanças no seu algoritmo original. A idéia é q essa mudança pode melhorar o processo como um todo (ver algoritmo 3.3).

## 1.2 Descrição do problema

Em alguns casos, para melhorar o desempenho da rede neural, um ótimo pré-processamento dos dados de entrada é necessário. Em processamento de imagens, por exemplo, esse pré-processamento é dividido em etapas, a fim de diminuir os ruídos que serão processados pela rede neural. Em algumas situações, pode-se usar técnicas de *clustering*<sup>1</sup> para separar elementos de classes distintas que estejam interferindo em outras classes ou que pertença a certa classe, mas não sejam classificadas corretamente. Em classificação de proteínas, por exemplo, existe um grande número de informações que podem ser filtradas. Nesse caso, um conhecimento básico da sua sequência de aminoácidos nos dá muitas informações, como suas funções biológicas ou sua estrutura tridimensional. O problema é que a grande massa de dados exigiria um processamento com complexidade computacional de ordem  $O(n^2)$  [3].

Para esses tipos de problemas existem varias técnicas, dentre elas a Análise discriminante (Haykin, 1998), que realiza combinações lineares das variáveis do modelo preditivo que fornece a melhor discriminação entre as amostras, Algoritmo C4.5 (Quinlan, 1993), que gera um classificador na forma de uma árvore de decisão [4], Mapa Auto-Organizável (SOM em inglês - Kohonen, 1982), que são redes

---

<sup>1</sup> *Cluster é um sistema que compreende dois ou mais computadores ou sistemas (denominados nodos) na qual trabalham em conjunto para executar aplicações e são usados quando os conteúdos são críticos ou quando os serviços têm que estar disponíveis e/ou processados o quanto mais rápido possível.*

neurais de duas camadas e suas saídas são interconectados por uma relação de vizinhança, baseados no neurônio vencedor de um processo competitivo [5].

A rede GMDH, por sua vez, realiza combinações entre todas as variáveis de entrada, elimina aquelas que não satisfaçam certo critério, cria um novo conjunto de variáveis que serão recombinações até atingirem um critério de parada. Com isso é descoberto quais das variáveis são relevantes para o problema em questão.

## 1.3 Próximos Capítulos

Para o resto desse documento serão apresentados os capítulos como segue:

- **Capítulo 2: As Redes Neurais – Conceitos Básicos**

É um capítulo necessário para o entendimento da evolução e funcionamento das redes neurais, suas necessidades, suas vantagens e seus problemas.

- **Capítulo 3: A cibernética e as redes GMDH**

Explica o surgimento e avanço da cibernética, bem como seus conceitos básicos e como as redes GMDH começaram a fazer parte das soluções computacionais. Além disso, são mostradas as vantagens e desvantagens de se usar as redes GMDH, assim como uma descrição geral do seu algoritmo.

- **Capítulo 4: Experimento**

Mostra a metodologia usada para o desenvolvimento da ferramenta de seleção de variáveis, incluindo os requisitos necessários e as outras ferramentas usadas no processo. É também explicado detalhadamente como o algoritmo desse Trabalho de Conclusão de Curso funciona e como ele se comporta, suas limitações e suas vantagens.

- **Capítulo 5: Resultados**

Exibe os resultados obtidos com essa ferramenta. O leitor também compreenderá melhor o que essa ferramenta oferece como resposta e como interpretá-la e verá como ela se comporta em configurações distintas.

- **Capítulo 6: Conclusões**

Dá uma descrição resumida dos resultados obtidos baseado nos objetivos propostos. Exibe as dificuldades encontradas, conclusões obtidas e expectativas para trabalhos futuros.



# Capítulo 2 - A Redes Neurais –

## Conceitos básicos

McCulloch e Pitts (1943) criaram um modelo matemático simples do neurônio em que ele é ativado quando uma combinação linear de suas entradas, que no caso do neurônio biológico são os seus estímulos nervosos, excede algum limiar (lei do Tudo ou Nada). A figura abaixo mostra um neurônio biológico.

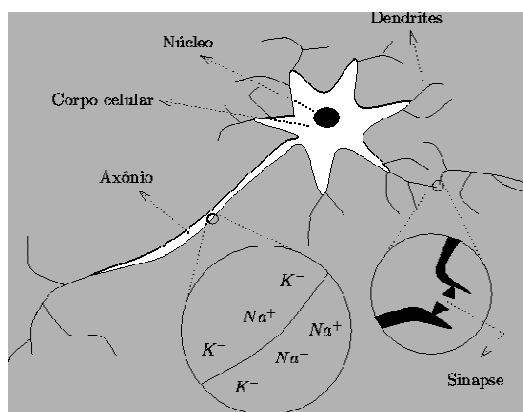


Figura 1. Neurônio biológico

Baseado nessa proposição considere  $x_1, x_2, x_3, \dots, x_n$ , como sendo as variáveis de entrada,  $x_j$  ( $j = 1, \dots, n$ ) do neurônio de saída  $i$ . A entrada líquida  $net_i$  é dada pela seguinte regra de propagação:

$$net_i = \sum_{j=1}^m w_{ij} \cdot x_j - \theta$$

onde  $W_{ij}$  são os pesos sinápticos e  $\theta$  é o limiar. A função de ativação  $f(net_i)$  neste caso é a função degrau, isto é:

$$f(net_i) = \begin{cases} 1, \forall net_i \geq 0 \\ 0, \forall net_i < 0 \end{cases}$$

Pesquisadores em IA e estatística ficaram interessados nas propriedades mais abstratas de redes neurais, como sua habilidade para executar computação distribuída, de tolerar entradas ruidosas e de aprender.

Frank Rosenblatt (1957) criou o perceptron moderno e provou o teorema de convergência de perceptrons (1960) [1]. A similaridade com o cérebro humano está na capacidade de aprender. Isso é feito modificando os pesos sinápticos em função dos exemplos de entrada que são repetidamente apresentados. A partir disso, novas funções de ativação puderam ser usadas, como a sigmóide logística

$$\left( y = f(\text{net}_i) = \frac{1}{1 + e^{-n}} \right) \text{ e a tangente hiperbólica } \left( y = f(\text{net}_i) = \frac{e^{\text{net}_i} - e^{-\text{net}_i}}{e^{\text{net}_i} + e^{-\text{net}_i}} \right) \text{ e assim, as}$$

redes neurais puderam aprender por exemplos. Abaixo mostramos o modelo de um perceptron.

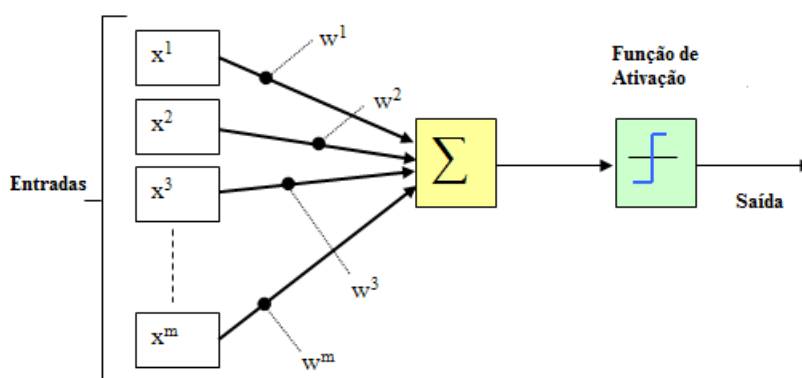


Figura 2. Perceptron segundo Roseblatt

Os valores  $x^1, x^2, x^3$  e  $x^m$  são os valores de entrada,  $w^1, w^2, w^3$  e  $w^m$  são os pesos respectivos a cada entrada,  $\Sigma$  representa a soma de todos os valores obtidos através do produto da entrada pelo seu respectivo peso sináptico e a função de ativação tratará essa soma, fornecendo um valor de saída que pode representar, por exemplo, uma classe de dados simbólicos.

Com o advento do algoritmo *backpropagation*, uma grande atenção tem-se dado às redes neurais, especialmente às redes MLP (*multilayer perceptron*), em função da sua capacidade de poder aproximar uma função arbitrária não-linear por superposição de funções sigmóides. Uma rede MLP com uma única camada

escondida é capaz de aproximar qualquer função contínua definida sobre um conjunto compacto limitado [2].

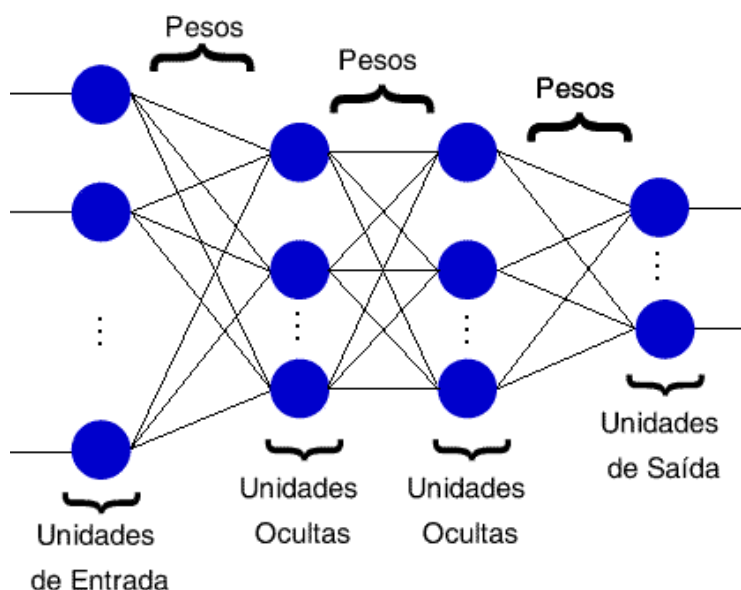


Figura 3. Rede MLP

Uma rede neural básica é composta por entradas, pesos sinápticos, uma função erro, a taxa de aprendizagem, critérios de parada (prevenindo um superajuste da rede) e funções de saída. Treinar uma rede neural consiste em ajustar os pesos através de uma regra de aprendizagem até que esta forneça respostas satisfatórias ao problema analisado. Para isso, apresentam-se exemplos com os valores de entradas e suas respectivas saídas (para aprendizagem do tipo supervisionado). Esse processo se repete para todo conjunto de exemplos, onde completará um ciclo (ou época). Caso o erro não esteja correspondendo a um critério, o processo é repetido e começa-se um novo ciclo.

A validação cruzada é o critério mais comumente usado pelos projetistas para a interrupção do treinamento de uma rede neural evitando seu superajustamento, o que faria a rede perder sua generalização.

Para que uma rede neural forneça respostas eficazes, deve-se saber que:

- É importante a escolha do conjunto inicial de pesos, pois valores iniciais diferentes podem alterar a solução encontrada.

- A forma de apresentação dos exemplos a cada ciclo também é importante. Recomenda-se apresentar os exemplos de forma aleatória.
- A taxa de aprendizagem fica entre 0 e 1.
- Devemos definir um critério de parada de treinamento satisfatório.

Para modelar uma rede neural do tipo MLP, por exemplo, temos que realizar uma divisão dos dados, sendo parte para treinamento, parte para o critério de parada de treinamento, e parte para testes, além de normalizar os dados de entrada, selecionar essas variáveis que serão usadas no modelo, determinar a arquitetura ótima da rede, treinar e verificar seu desempenho [6].

A utilização de dados diferentes para o processo de treinamento, validação e testes deve ser usado devido ao fato de que a rede neural tem que se comportar de forma similar com diferentes dados, o que comprova sua capacidade de generalização.

# Capítulo 3 - A cibernética e as redes

## GMDH

A cibernética foi originalmente concebida por Nobert Wiener como a ciência do controle e comunicação em animais e máquinas. Ele, junto com outros cientistas, observou que os seres vivos podiam modificar, ou melhor, adaptar seu comportamento em resposta a mudanças no ambiente e que essa mudança era resultado de um mecanismo regulador tentando minimizar o "erro" - a diferença entre o estado atual e o estado do objetivo. Na figura abaixo é mostrado um exemplo de cibernética: a homeostase. Sistemas complexos, como o corpo humano, precisam de homeostase para manter a estabilidade e sobreviver. Mais do que apenas sobreviver, esses sistemas devem ter a capacidade de se adaptar ao seu ambiente externo [7].

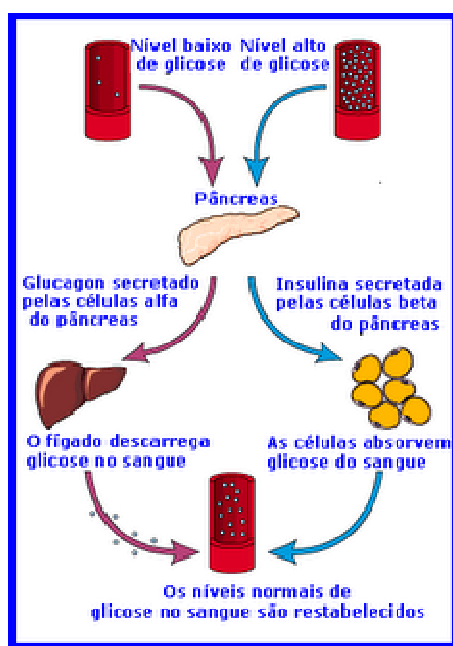


Figura 4. Sistema cibernético nos seres humanos. Os hormônios coordenando a homeostase glicídica [8]

### 3.1 Fundamentação Teórica

Os fundamentos teóricos de um sistema cibernético baseiam-se no conceito de “caixa preta” (que caracteriza um estudo da relação entre entrada e saída de um sistema), uma abordagem que utiliza o conceito de limiar lógico e connexionismo, uma abordagem indutiva que utiliza o conceito de mecanismo indutivo para manter o controle composto do sistema, uma abordagem probabilística que utiliza funções multiplicativas da teoria hierárquica de decisões estatísticas e uma abordagem lógica matemática de Gödel (teorema da incompletude) que utiliza o princípio de “complemento externo” como um critério de seleção [9]. Na próxima figura é exibido, de uma forma geral, como um sistema cibernético não-controlado funciona. Existe uma função que representa a relação entre a entrada e a saída. Essa relação é a função de transferência  $f_t$ .



Figura 5. Sistema cibernético não controlado [10]

Com a descoberta do funcionamento e comportamento dos sistemas cibernéticos, houve então um grande crescimento dessa ciência, resultando no desenvolvimento de outras ciências aliadas. Mas no final dos anos 60, seu desenvolvimento ficou cada vez mais interno. A modelagem matemática foi o novo foco necessário para a solução de grande variedade de problemas da sociedade, incluindo o desenvolvimento de computadores. Mesmo assim, a cibernética se tornou a ciência da modelagem de comunicação e controle de sistemas complexos.

No início, com o trabalho inicial de Wiener, ninguém considerava a Teoria do Controle Automático como contraditória à Teoria da Informação, apesar da diferença das duas abordagens. Um método combinado é mais eficiente. Através desse método, sistemas complexos são modelados com a ajuda de métodos experimentais de auto-organização [11].

Com o objetivo de representar a abordagem indutiva destes sistemas cibernéticos, vários métodos têm sido desenvolvidos. Esses métodos de aprendizagem indutiva são também chamados GMDH, *Self-Organization*, *Sorting Out* ou Métodos Heurísticos.

## 3.2 As redes GMDH

*Group Method of Data Handling* (GMDH), também chamado de método de aprendizagem indutiva, é a realização da abordagem indutiva para modelagem matemática de sistemas complexos. Este método foi desenvolvido desde 1968 no Institute of Cybenetics em Kiev, Ucrânia, pelo grupo de sistemas de controle combinado (CCS), principalmente pelo professor e autor de vários livros Alexey G. Ivakhnenko. Contribuições para este campo vieram de muitas áreas de pesquisas de diferentes disciplinas [12].

A.G. Ivakhnenko é um ciberneticista. Com suas pesquisas, percebeu que existe um problema na maioria dos modelos matemáticos que é o fato desses modelos requererem que o modelador saiba coisas sobre o sistema que é praticamente impossível de encontrar. Era o momento dos pesquisadores desenvolverem modelos que apenas lidem com os dados em si [11]. A figura 6 mostra a rede GMDH proposta pelo professor A.G. Ivakhnenko, onde  $y$  está em função de uma combinação das entradas  $x_1, x_2, x_3, x_4$ . Notamos também que alguns elementos (bolas pretas) são eliminados e as bolas brancas são selecionadas para combinar os elementos da nova camada. (ver tópico 3.4)

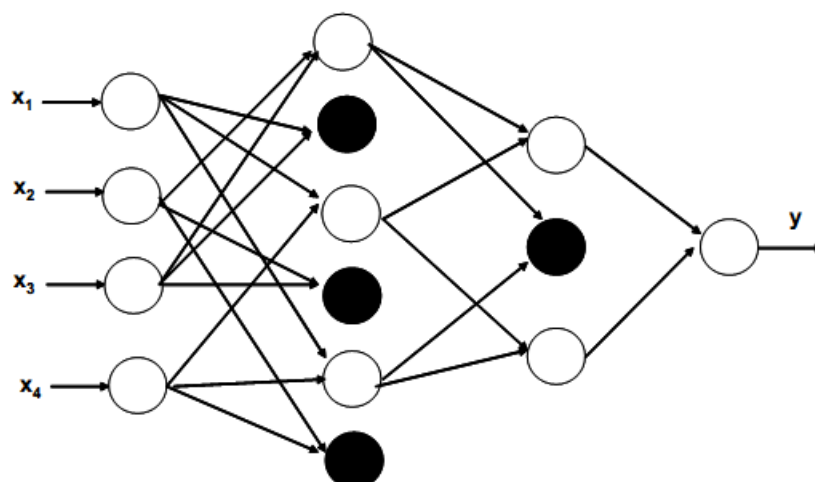


Figura 6. Rede GMDH com duas camadas ocultas

As redes GMDH são ainda pouco conhecidas e pouco utilizadas no Brasil, basta ver os raros trabalhos publicados por autores brasileiros utilizando estes modelos. Portanto, em função de sua simplicidade de implementação estas redes podem ser utilizadas como uma poderosa ferramenta para auxiliar na seleção das variáveis de entrada para outras modelagens. Tradicionalmente redes neurais artificiais, redes com aprendizado estatístico e especialmente algoritmos do grupo GMDH (Ivakhnenko (1971); Muller e Lemke (1995); Valença (1999)), são campos de pesquisa distintos. Entretanto, existe uma razoável semelhança se forem considerados seus objetivos e metodologias. Existem vários trabalhos publicados em inglês e russo sobre o algoritmo GMDH, entretanto, são poucos os livros publicados sobre a teoria do algoritmo GMDH e suas aplicações [2].

As redes GMDH são usadas em campos como mineração de dados, predição, modelagem de sistemas complexos, otimização e reconhecimento de padrões e possuem as seguintes vantagens e desvantagens:

a) Vantagens:

- Podem ser vistas como um identificador universal de estruturas através de equações de regressão;
- Realiza uma aproximação direta;



- Utiliza técnicas estatísticas simples como o método dos mínimos quadrados ou da máxima verossimilhança;
- A arquitetura da rede (quantidade de camadas intermediárias e de elementos) é construída durante o processo de treinamento;
- A quantidade de neurônios em cada camada intermediária é estimada com base em um critério de seleção (função objetiva) de forma automática;
- O treinamento é rápido e os parâmetros desnecessários são descartados automaticamente.

b) Desvantagens:

- Não são aproximadores universais de funções;
- Faz-se necessário testar vários modelos para o problema;
- Ao utilizar técnicas estatísticas simples, o modelo tem que trabalhar com funções de ativação que sejam lineares em relação aos coeficientes;
- Quando o número de variáveis aumenta, pode ocorrer uma explosão combinatorial quanto ao número de modelos a serem analisados.

### 3.3 Algoritmos GMDH

Os algoritmos GMDH são caracterizados por um procedimento indutivo que realiza um arranjo de modelos polinomiais complexos e seleciona a melhor solução através de um critério externo. Um modelo GMDH tem como função base,

$$Y(x_1, \dots, x_n) = a_0 + \sum_{i=1}^m a_i f_i$$

onde  $f_i$  são funções elementares dependentes de diferentes conjuntos de entrada e  $a_i$  são os coeficientes.

Para encontrar a melhor solução, o algoritmo considera vários subconjuntos de componentes da função base chamados *modelos parciais*. A partir daí ele aumenta gradualmente o número de modelos parciais e encontra uma estrutura com complexidade ótima indicada pelo mínimo valor de um critério externo. Esse processo é chamado de auto-organização de modelos [13].

A função base mais popular é o polinômio de Kolmogorov-Gabor

$$Y(x_1, \dots, x_n) = a_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=i}^n a_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=i}^n \sum_{k=j}^n a_{ijk} x_i x_j x_k + \dots$$

Esse polinômio também é chamado de polinômio de Ivakhnenko e representa a equação resposta da rede GMDH. Essa equação é obtida substituindo as variáveis da equação de saída pelas equações de entrada, até que chegue aos dados iniciais de entrada (ver 0).

Dentre os vários modelos de algoritmos GMDH, apresentamos aqui os mais conhecidos [12]:

- O algoritmo *Combinatorial* (COMBI). Baseia-se no arranjo gradual completo ou reduzido de modelos complicados e avaliação dos mesmos pelo critério externo em uma parte separada da amostra dos dados;
- O algoritmo *Multilayered Iteraction* (MIA) usa em cada camada do processo de organização a mesma descrição parcial (regra de iteração). Deve ser usado quando for necessário para lidar com um grande número de variáveis;
- O algoritmo *Objective System Analysis* (OSA). A principal característica dele, é que ele não examine as equações simples, mas os sistemas de equações algébricas ou de equações diferenciais, obtidos por modelos implícitos (sem função objetiva). Uma vantagem do algoritmo é que a informação contida na amostra de dados utilizada é melhor;
- O algoritmo *Two-level* (ARIMAD) para a modelagem de longa duração cíclica (como ações ou condições meteorológicas) processos.

O algoritmo que será usado nessa ferramenta será descrito logo a seguir.

### 3.4 Um algoritmo GMDH modificado

Através das variáveis de entrada do sistema, são criadas equações de regressão do tipo

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_i x_j$$

Onde  $x_i$  e  $x_j$  é um par de entradas e  $y$  é a saída. A ilustração da equação é exibida abaixo.

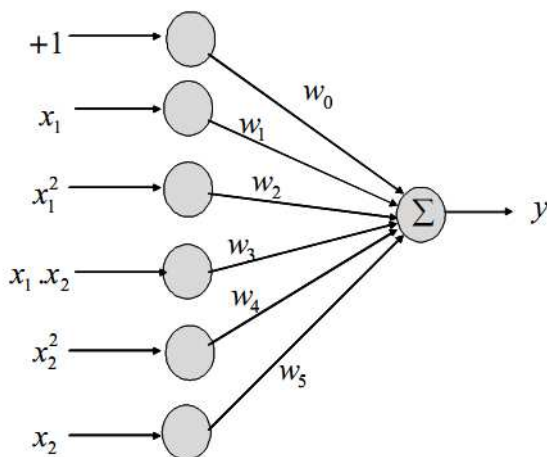


Figura 7. Equação de regressão

Combinando todos os pares possíveis de entradas, teremos  $\frac{m(m-1)}{2}$  novas variáveis de ordem mais alta que substituirá as originais  $x_1, x_2, \dots, x_m$ , objetivando uma nova equação de saída.

Após encontrar essas equações de regressão, será decidido quais delas serão eliminadas e quais delas prosseguirão. As equações quadráticas salvas serão usadas para gerar novas observações independentes. Dessas novas variáveis independentes, uma nova recombinação será feita assim como antes, duas por vez.

Isso nos dá uma nova coleção de  $\frac{m(m-1)}{2}$  equações de regressão que buscam obter o valor de  $y$  das novas variáveis, que por sua vez são estimadas de  $y$  das variáveis mais antigas. O processo continua até que as equações de regressão tenham uma resposta pior do que as anteriores. Isso acontece quando o modelo começa a se tornar superespecializado. Esse é o critério de parada, normalmente definido pelo projetista. Depois de parado o processo, são escolhidos os melhores polinômios quadráticos daquela geração, que foram estimados de outras duas variáveis, que são polinômios quadráticos de mais duas variáveis, ..., que são polinômios quadráticos das variáveis originais [11]. A figura abaixo mostra uma rede GMDH com duas gerações.

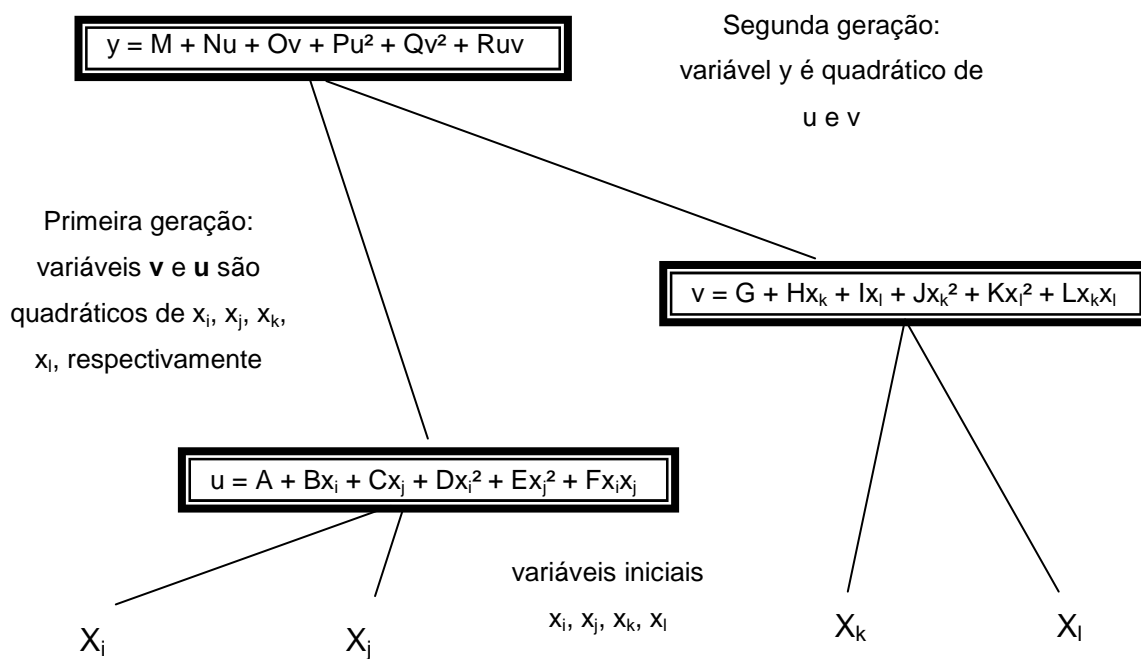


Figura 8. Esquema básico da propagação de variáveis

Para esse Trabalho de Conclusão de curso, faremos uma modificação na equação de regressão. Os valores de  $x_i$  e  $x_j$  que tendem a aumentar seu valor ao quadrado a cada nova camada criada, podendo levar a uma equação com um grau bastante alto, além do que como os dados de entrada ficarão entre 0 e 1, seus valores podem ficar muito pequenos com poucas camadas. Por isso, as variáveis originais serão substituídas pela função sigmóide logística como mostrada a seguir,

$$f(x_n) = \left( \frac{1}{1 + e^{-n}} \right)$$

Ficando no seguinte formato:

$$y = A + B \left( \frac{1}{1 + e^{-x_i}} \right) + C \left( \frac{1}{1 + e^{-x_j}} \right) + D \left( \frac{1}{1 + e^{-x_i}} \right)^2 + E \left( \frac{1}{1 + e^{-x_j}} \right)^2 + F \left( \frac{1}{1 + e^{-x_i}} \right) \left( \frac{1}{1 + e^{-x_j}} \right)$$

## Capítulo 4 - Experimento

Para experimentar o desempenho do algoritmo será utilizada a linguagem mais usada atualmente: JAVA. Além disso, Java é usada há um tempo considerável a ponto de ser uma linguagem conhecida bem a fundo pela maioria dos projetistas atuais. Como ferramenta de edição e compilação, usaremos o Netbeans. Quem usa outras ferramentas como Eclipse não terá grandes dificuldades em caso de adaptação.

Os dados que serão usados na rede neural para atuar como um sistema de previsão foram obtidos pelo ONS (Operador Nacional do Sistema Elétrico), onde contêm as vazões naturais mensais desde janeiro de 1931 a dezembro de 2003 da represa de Três Marias e a de Sobradinho, ambas fazendo parte da Bacia do rio São Francisco.

Para obter a rede GMDH, serão usados dados com 8 entradas e 1 saída, com 10 entradas e 1 saída e com 8 entradas + 2 entradas ruidosas (dados não pertencentes aos originais) e 1 saída. A resposta da rede definirá as variáveis mais importantes. Após isso, serão organizadas apenas as entradas escolhidas e então usadas como entradas para uma rede neural.

Visando medir a veracidade da ferramenta, foram definidos alguns critérios:

- a) Usar uma rede MLP para experimentar os dados de entrada selecionados. Para evitar interferência entre os dados de entrada da rede GMDH que serão executadas na ferramenta NetBeans e os dados de entrada da rede neural, a rede neural foi executada na ferramenta Eclipse. O código-fonte foi fornecido em anexo ao livro Meuser – fundamentos das redes neurais [6]. Ela será adaptada para processar um arquivo CSV e exportar dados como resposta em um arquivo CSV, que será adicionado a uma tabela comparativa.
- b) As variáveis threshold e alpha serão combinadas entre si a fim de analisar os resultados da rede.

- threshold – variável que representa a razão do erro do elemento em análise pelo erro mínimo de todos os elementos. Seu valor tem que ser maior ou igual a 1, não devendo ser superior a 1.5, pois isso poderá fazer com que a rede selecione variáveis não tão relevantes ou que a rede selecione todas as variáveis. Nessa ferramenta, usaremos os valores 1.01 e 1.1.
  - alpha – variável que representa a capacidade de aprendizado da rede. Um valor baixo fará com que a rede precise de mais ciclos para definir certos dados específicos, ou então a rede será menos rígida em relação aos critérios de seleção. O valor alto provocará o efeito inverso. Deve-se usar esse valor maior que 0 e menor ou igual a 1. Nessa ferramenta, usaremos os valores 0.1 e 0.9.
- c) Os dados de saída da rede GMDH devem ser usados para testes em uma rede MLP com os seguintes dados fixos:
- Ciclos: 1000;
  - Número de entradas: 49;
  - Valor da semente: 1315;
  - Número de camadas escondidas: 2;
  - Alpha: 0.8;
  - Beta: 0.6;
  - Os pesos serão iniciados aleatoriamente;

Para analisar os resultados, uma tabela será criada com um comparativo entre uma simulação de redes MLP com as 8 variáveis, e outras simulações com redes MLP com o número de entradas correspondente à resposta dada pela rede GMDH.

Todos esses dados devem estar em formato CSV (texto separado por vírgulas ou ponto-e-vírgula). Isso pode ser conseguido através do Microsoft Excel, em que o projetista pode organizar seus dados na tabela e salvá-los na forma CSV automaticamente.

Deve-se tomar cuidado em que estilo vai ficar o arquivo. Cada conjunto de entradas e saídas serão separados por uma linha enquanto o valor de cada entrada e saída ficará separado por vírgulas ou outro caractere delimitador. Caso o usuário use outro caractere delimitador, este deve informar à ferramenta qual caractere está usando. Essa estrutura generaliza o funcionamento da rede GMDH. Assim, ela pode tratar dados de previsão e de classificação de forma similar. Outro ponto importante é que caso as primeiras linhas do arquivo não possuam dados úteis, o usuário deve informar à ferramenta a partir de que linha os dados devem ser processados. Além disso, deve-se ter o cuidado de não deixar linhas incompletas ou linhas em branco (ou apenas preenchidas com o caractere separador) ou de mudar os estilos dos números caso necessário. A figura 9 ilustra melhor essa situação. Ao lado direito, os dados estão no estilo *número*, mas quando convertido para CSV, as vírgulas dos números se confundem com o caractere separador padrão (vírgula). Já ao lado direito, os dados estão sem formatação específica. Essa é a forma correta. A figura abaixo das duas mostra um arquivo salvo em CSV e ainda assim impróprio para uso na ferramenta. É necessária a retirada de espaços e linhas em branco ou incompletas no final do arquivo.

	A	B	C	D	E
1	SÃO FRANCISCO				4.676
2	SOBRADINHO			4.676	4.805
3	<b>Série revista</b>		4.676	4.805	6.630
4	<b>m3/s</b>	4.676	4.805	6.630	7.333
5	4.676	4.805	6.630	7.333	4.467
6	4.805	6.630	7.333	4.467	2.397
7	6.630	7.333	4.467	2.397	1.870
8	7.333	4.467	2.397	1.870	1.619
9	4.467	2.397	1.870	1.619	1.448
10	2.397	1.870	1.619	1.448	1.376
11	1.870	1.619	1.448	1.376	1.935
12	1.619	1.448	1.376	1.935	2.207
13	1.448	1.376	1.935	2.207	4.036
14	1.376	1.935	2.207	4.036	4.531
15	1.935	2.207	4.036	4.531	3.214
16	2.207	4.036	4.531	3.214	1.976
17	4.036	4.531	3.214	1.976	1.425
18	4.531	3.214	1.976	1.425	1.292
19	3.214	1.976	1.425	1.292	1.176
20	1.976	1.425	1.292	1.176	995
21	1.425	1.292	1.176	995	905

	A	B	C	D	E
1	SÃO FRANCISCO				
2	SOBRADINHO				
3	<b>Série revista</b>				
4	<b>m3/s</b>				
5	4676.354839	4805.36	6629.84	7333.47	4467.35
6	4805.357143	6629.84	7333.47	4467.35	2397.4
7	6629.83871	7333.47	4467.35	2397.4	1869.81
8	7333.466667	4467.35	2397.4	1869.81	1618.71
9	4467.354839	2397.4	1869.81	1618.71	1448.23
10	2397.4	1869.81	1618.71	1448.23	1375.81
11	1869.806452	1618.71	1448.23	1375.81	1935.37
12	1618.709677	1448.23	1375.81	1935.37	2207.06
13	1448.233333	1375.81	1935.37	2207.06	4036.16
14	1375.806452	1935.37	2207.06	4036.16	4531.28
15	1935.366667	2207.06	4036.16	4531.28	3213.81
16	2207.064516	4036.16	4531.28	3213.81	1976.07
17	4036.16129	4531.28	3213.81	1976.07	1424.87
18	4531.275862	3213.81	1976.07	1424.87	1291.77
19	3213.806452	1976.07	1424.87	1291.77	1175.52
20	1976.066667	1424.87	1291.77	1175.52	995.129



866	665,936,1871,4118,4387
867	936,1871,4118,4387,2737
868	1871,4118,4387,2737,2604
869	4118,4387,2737,2604,1374
870	4387,2737,2604,1374,983
871	2737,2604,1374,983,890
872	2604,1374,983,890,747
873	1374,983,890,747,680
874	983,890,747,680,506
875	890,747,680,506,843
876	747,680,506,843,1611
877	680,506,843,1611,
878	506,843,1611,,
879	843,1611,,,
880	1611,,,,
881	/////
882	/////
883	/////
884	/////
885	/////
886	/////
887	/////
888	/////

**<< Eliminar as partes incompletas ou vazias**

Figura 9. Exemplo de organização de um arquivo de entrada de dados para a rede GMDH. Acima à esquerda e abaixo, os dados não estão prontos para serem usados. Acima à direita, os dados estão prontos para conversão.

## 4.1 Funcionamento do algoritmo

O algoritmo da rede GMDH realizará os passos a seguir [14]:

- Pre-processar dados: os dados de entrada serão normalizados entre 0 e 1. Utilizando a normalização linear:

$$y = 0.7 \left( \frac{x - x_{\min}}{x_{\max} - x_{\min}} \right) + 0.15$$

Os valores ficarão ente 0,85 e 0,15 de forma a evitar saturação, pois nos valores extremos a derivada tende para zero.  $x_{\max}$  e  $x_{\min}$  são os valores máximos e mínimos das entradas a serem normalizadas.

- Separar o conjunto de dados que serão usados no treinamento da rede (50%) GMDH e o conjunto de dados que servirão para o processo de seleção (50%).

- c. Criar uma camada de  $m(m-1)/2$  elementos, onde  $m$  é o número de entradas iniciais. Cada par de entradas produzira um novo elemento.
- d. Inicializar os pesos dos elementos provisoriamente com valor 0 (zero).
- e. Usar o conjunto de dados de treinamento para treinar todas as entradas na camada criada. A explicação do procedimento de treinamento é dada por:
  - Obter os valores de entrada e a saída desejada.
  - Calcular os erros de cada elemento (diferença entre o valor de entrada e o de saída)

- Calcular os pesos utilizando a seguinte equação

$$W_m = W_m + \alpha \frac{X_m}{|X_m|^2} (y_d^m - W_m^T X_m)$$

A aplicação dessa equação faz com que  $\mathbf{W}$  seja modificado a fim de minimizar a diferença entre a saída desejada e a saída obtida.

Um lote de treinamento termina quando o conjunto inteiro de treinamento já foi apresentado à rede.

- O erro quadrático das entradas será somado por todo o lote, obtendo assim o BMSE (Batch Mean Square Error) daquela entrada. Esse valor será armazenado.
  - Se o BMSE para esta entrada for menor que o anterior, realizar um novo lote de treinamento. Para outros casos, parar o treinamento. O treinamento da camada atual cessa quando todas as entradas já estiverem treinadas.
- f. Entrar com o conjunto de dados usados para seleção e obter os BMSEs de todos os elementos da camada recém treinada e obter as proporções de cada BMSE em relação ao menor BMSE. Eliminar, no mínimo, metade dos elementos baseado no critério de erro. Os elementos que possuírem o valor

maior de erro baseado num valor limiar (*threshold*) são eliminados. Os "sobreviventes" serão armazenados.

- g. Se o menor BMSE na camada atual é maior que o BMSE da camada anterior ou a camada atual possui apenas um elemento (assumindo que a rede já tenha duas camadas ou mais), parar o processo de treinamento. Se o motivo da parada for devido a aumento do valor mínimo de BMSE, usar a camada anterior como a camada de saída da rede treinada. Se o treinamento é parado porque existe apenas um elemento na camada atual e o BMSE é menor que o BMSE da camada anterior, usar a camada atual como saída da rede treinada. Para outros casos, usar os "sobreviventes" para criar uma nova camada e retornar ao passo (d);
- h. Testar o desempenho da rede treinada com novos dados (se possível), ou então com dados combinados do conjunto de dados usados para treinamento e seleção.

Se os dados para avaliação da rede treinada consistem de dados de treinamento e de seleção e apenas os dados de treinamento foram usados para determinar os pesos da rede, a habilidade de generalização dessa rede pode ser testada. Caso novos dados sejam usados para avaliar a desempenho, uma maior faixa de generalização pode ser testada. Abaixo é ilustrada a estrutura da rede GMDH ótima após os procedimentos citados acima.

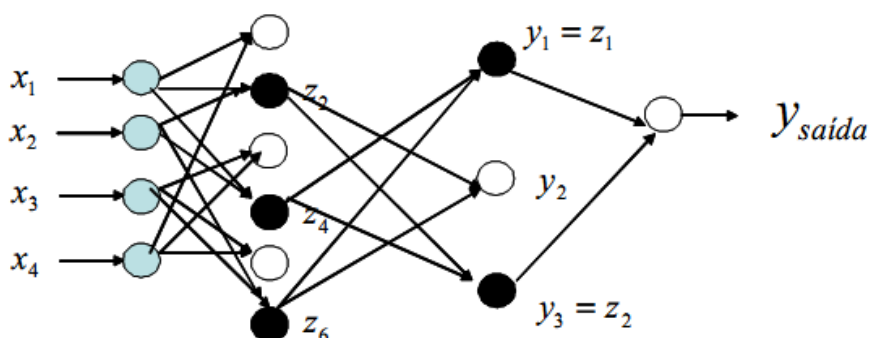


Figura 10. Rede GMDH treinada (bolas pretas = elementos eliminados)

## Capítulo 5 - Resultados

A partir de agora, os dados obtidos serão analisados. Como explicado no capítulo anterior, os erros obtidos no ciclo 1000 da rede MLP serão armazenados numa tabela. Nessa tabela constarão os erros de varias redes MLP com diferentes entradas.

As figuras abaixo mostram os gráficos comparativos entre as entradas e entre uma rede com todas as entradas com cada rede sugerida pela rede GMDH. Além disso, serão exibidos os resultados para uma rede GMDH padrão e uma rede GMDH modificada. Quanto mais similar com a análise de todas as entradas for o gráfico, melhor é sua escolha.

Para facilitar, as tabelas abaixo mostram as combinações realizadas nas duas represas com 8 entradas.

Tabela 1. Combinações realizadas com os dados da represa de Sobradinho com 8 entradas

Sobradinho					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,3,4,5	0.9	1.01	0,1,2,3
0.1	1.01	0,1,2	0.1	1.01	0,1,2,7
0.1	1.1	0,1,2,7	0.1	1.1	0,1,2,4,5,6,7
0.9	1.1	0,1,2,3,4,5	0.9	1.1	todas

Tabela 2. Combinações realizadas com os dados da represa de Três Marias com 8 entradas

Três Marias					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,1,2,5	0.9	1.01	0,1,2
0.1	1.01	0,1,2	0.1	1.01	0,1,2,7
0.1	1.1	0,1,2,7	0.1	1.1	todas
0.9	1.1	0,1,2,3,4,5,6	0.9	1.1	todas

Através dessas tabelas, podemos perceber os comportamentos das variáveis alpha e threshold e comprovar as definições feitas no capítulo anterior.

Agora exibiremos os gráficos da represa de Sobradinho com 8 entradas.

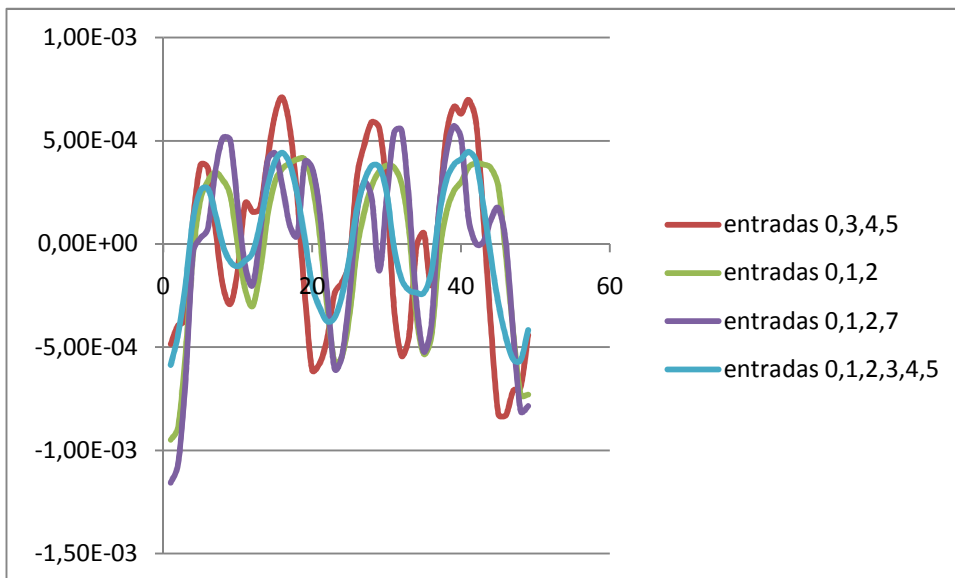


Figura 11. Comparação entre as combinações de entradas selecionadas pela rede GMDH padrão

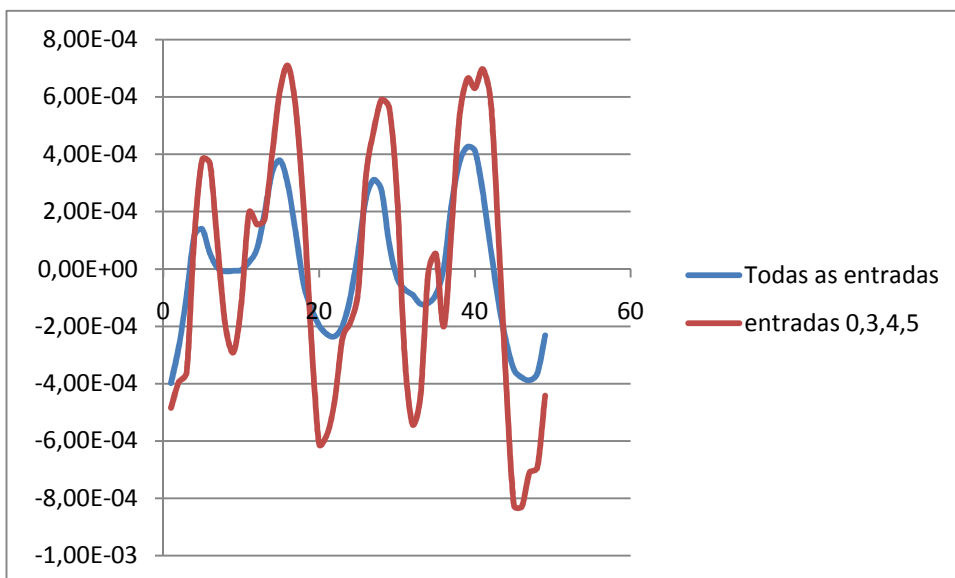


Figura 12. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 3, 4 e 5 pela rede GMDH padrão

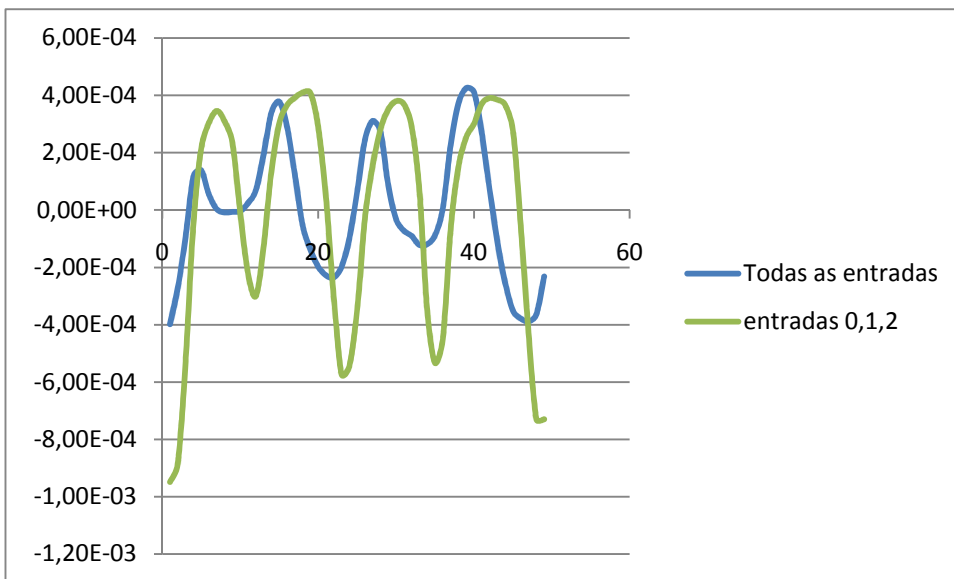


Figura 13. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1 e 2 pela rede GMDH padrão

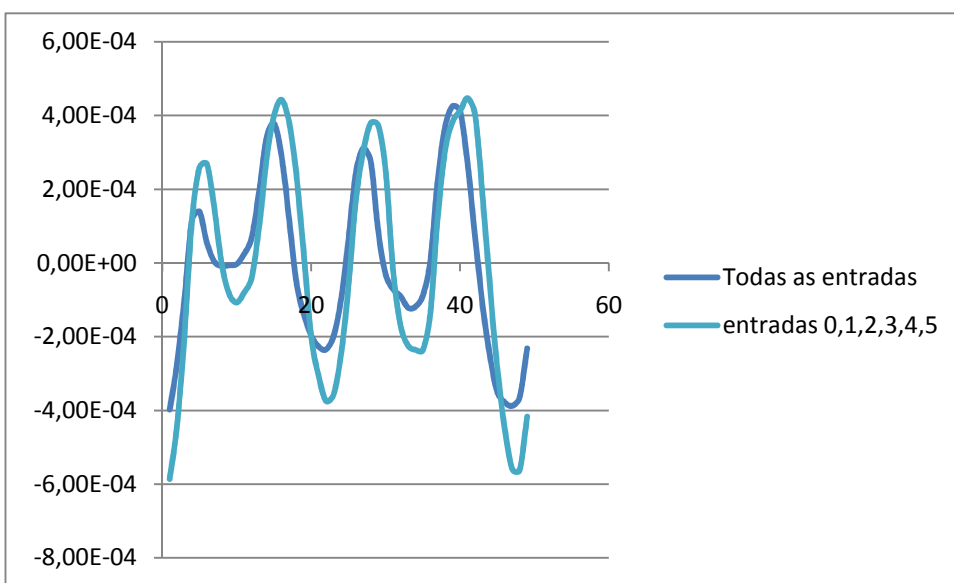


Figura 14. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2, 3, 4 e 5 pela rede GMDH padrão

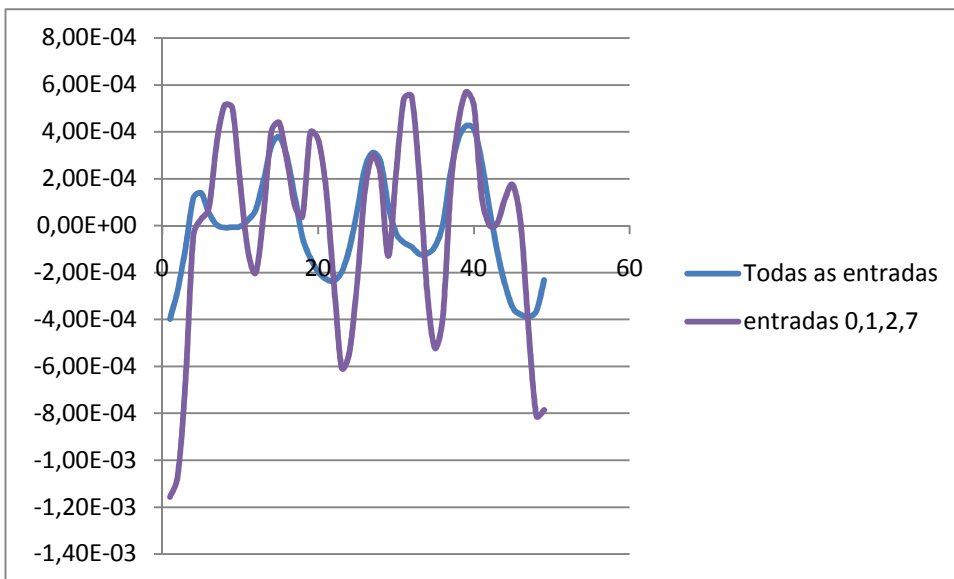


Figura 15. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 7 pela rede GMDH padrão

Percebe-se pela tabela que as variáveis mais antigas são as mais relevantes quando se trata de previsão.

A figura 13 e a figura 14 nos mostram um fato interessante: em caso de a quantidade de entradas for pequena (figura 13) ou se for grande (figura 14) em relação ao máximo valor possível, o gráfico produzido é similar, indicando uma tendência a uma convergência igual.

Abaixo, são mostrados os gráficos para a rede GMDH modificada

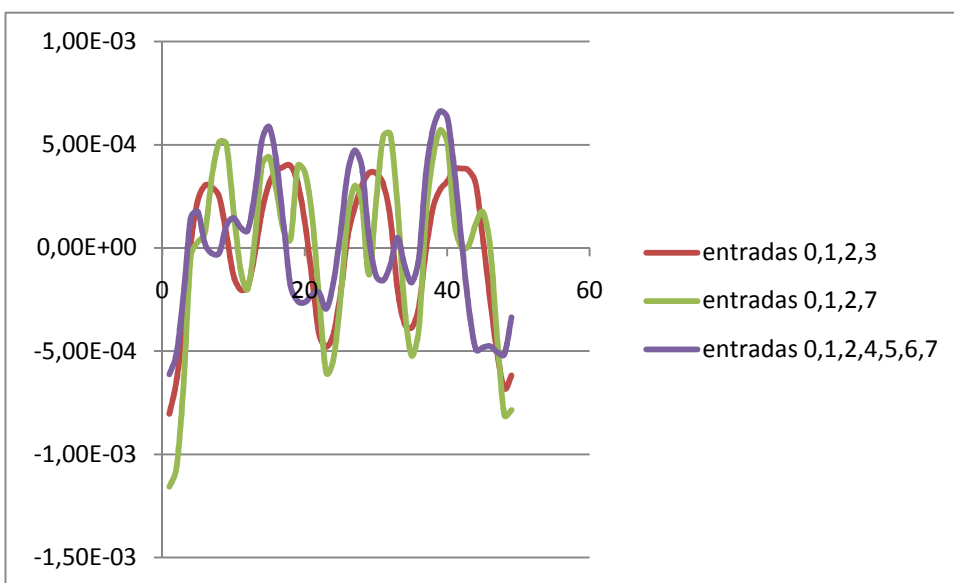


Figura 16. Comparação entre todas as combinações de entradas seleccionadas pela rede GMDH modificada

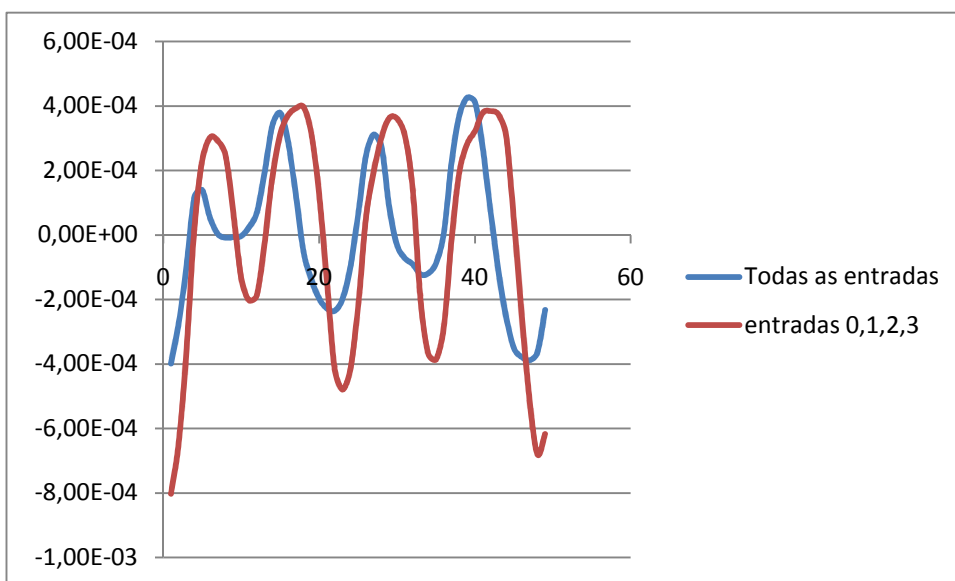


Figura 17. Comparação entre a rede com todas as entradas e a rede com as entradas seleccionadas 0, 1, 2 e 3 pela rede GMDH modificada

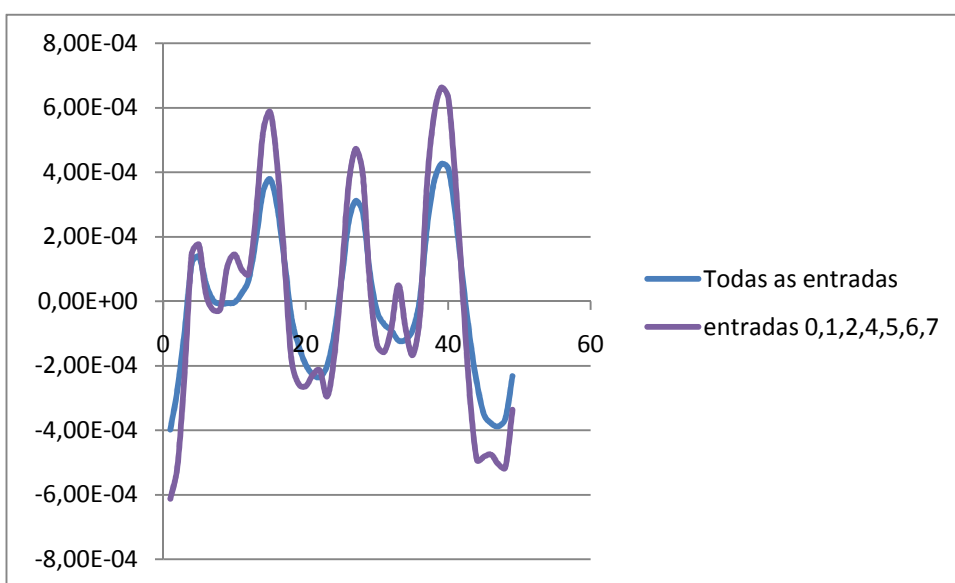


Figura 18. Comparação entre a rede com todas as entradas e a rede com as entradas seleccionadas 0, 1, 2, 4, 5, 6 e 7 pela rede GMDH modificada

Dos gráficos, os que demonstram melhores comportamentos são das entradas seleccionadas 0,1,2,4,5,6,7 e 0,1,2,3 do algoritmo GMDH modificado e 0,1,2 e 0,1,2,3,4,5 do algoritmo GMDH padrão. Na tabela, isso representa as seguintes



respectivas combinações:  $\alpha = 0.1$ ,  $\text{threshold} = 1.1$ ;  $\alpha = 0.9$ ,  $\text{threshold} = 1.01$ ;  
 $\alpha = 0.1$ ,  $\text{threshold} = 1.01$ ;  $\alpha = 0.9$ ,  $\text{threshold} = 1.1$ .

Agora, as figuras abaixo exibirão os resultados da represa de Três Marias.

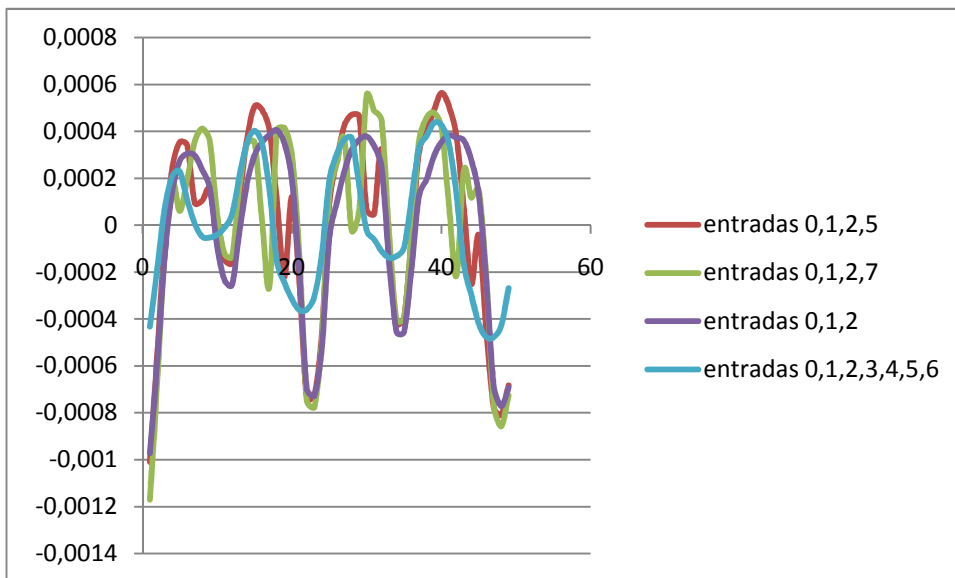


Figura 19. Comparação entre todas as combinações de entradas selecionadas pela GMDH padrão

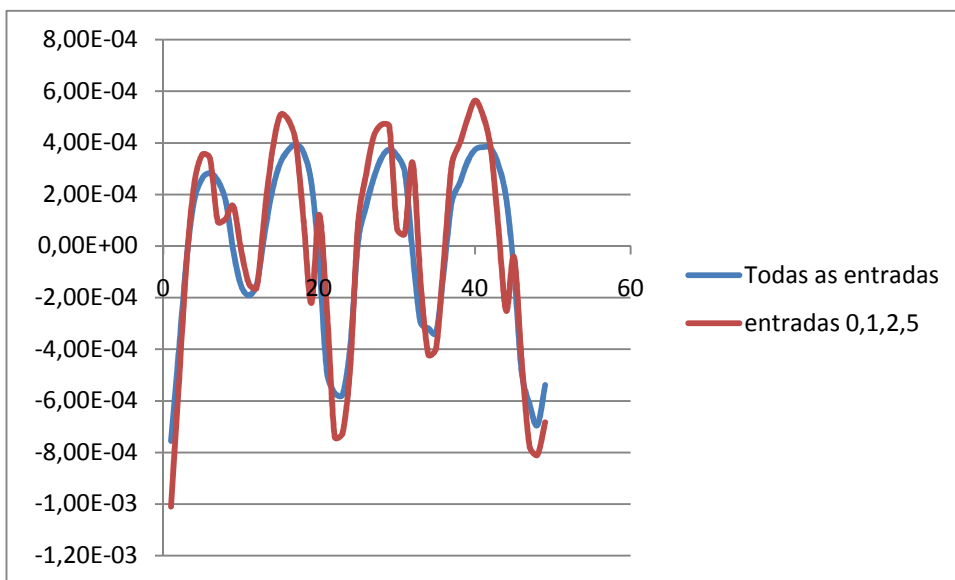


Figura 20. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 5 pela rede GMDH padrão

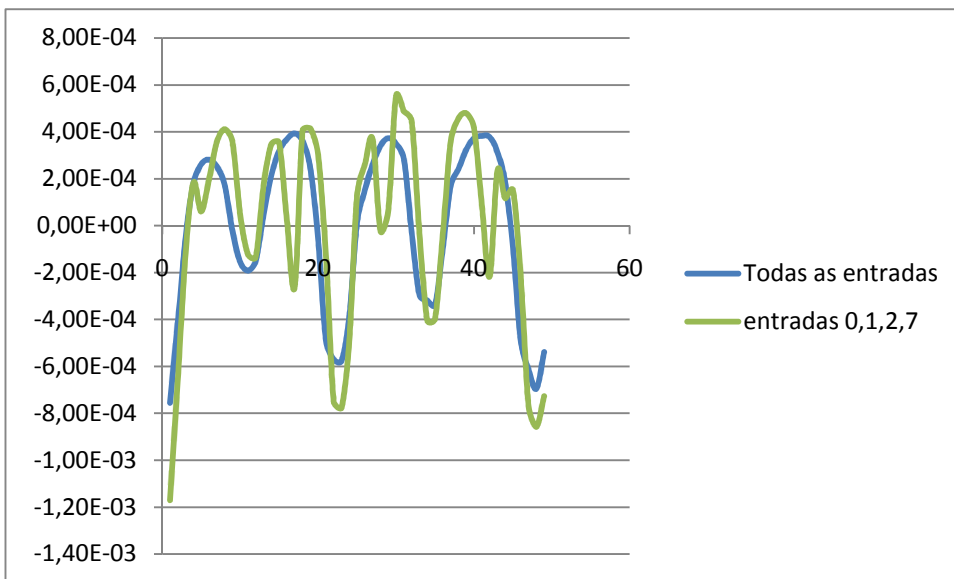


Figura 21. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2 e 7 pela rede GMDH padrão

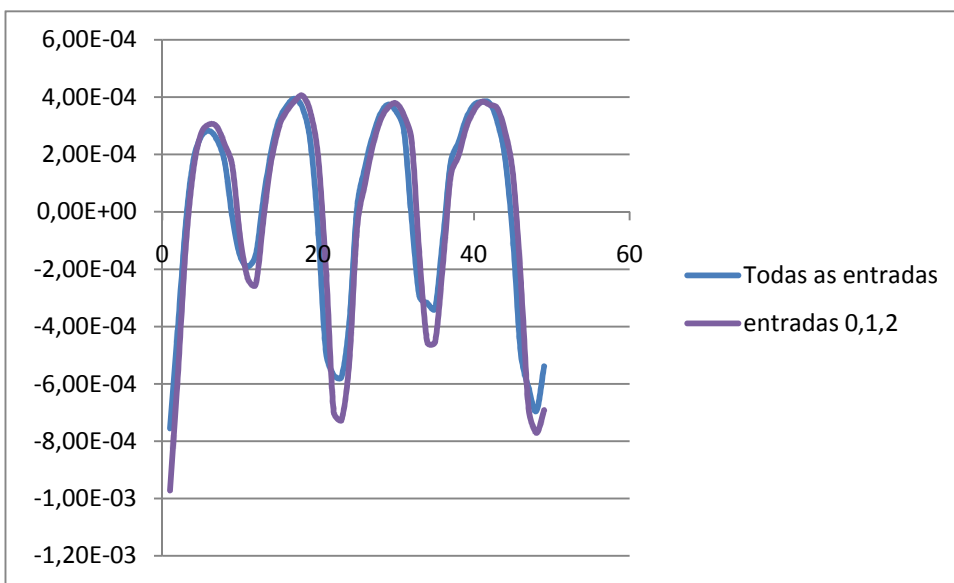


Figura 22. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1 e 2 pela rede GMDH padrão

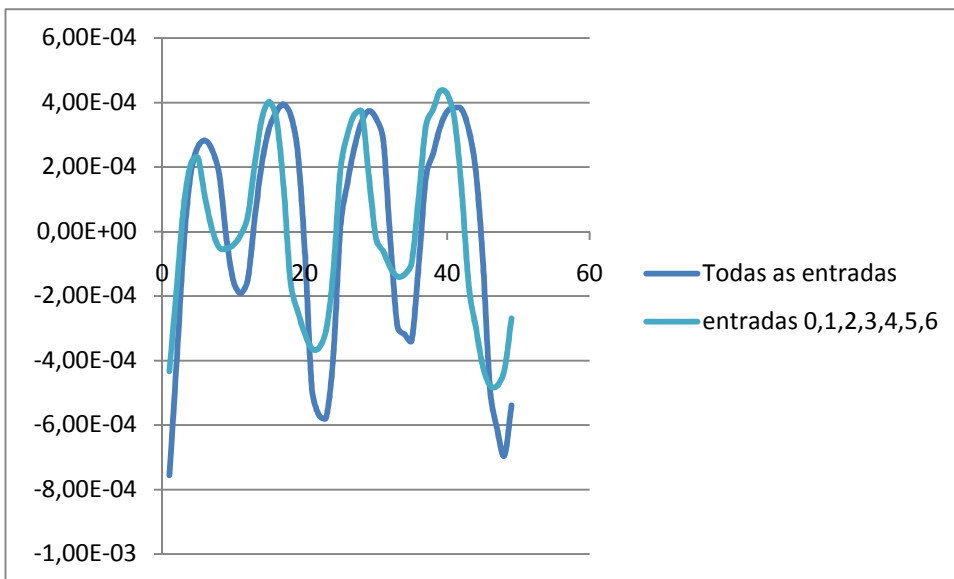


Figura 23. Comparação entre a rede com todas as entradas e a rede com as entradas selecionadas 0, 1, 2, 3, 4, 5 e 6 pela rede GMDH padrão

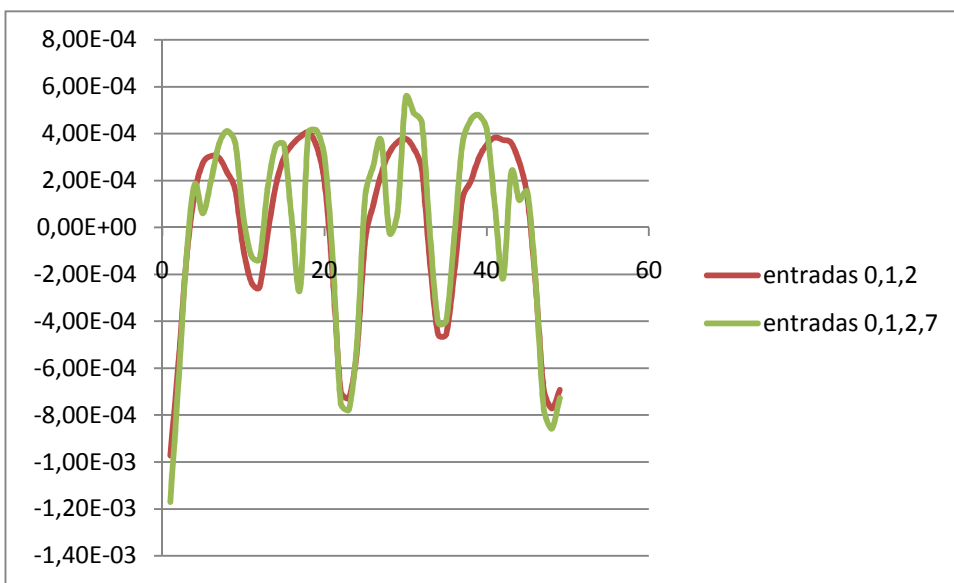


Figura 24. Comparação entre todas as combinações de entradas selecionadas pela GMDH modificada

Como no caso anterior, o conjunto de entradas 0, 1 e 2 e 0, 1, 2, 3, 4, 5 e 6 do algoritmo GMDH padrão e o 0,1,2 do algoritmo GMDH modificado foram os que deram melhores resultados, se comparado ao uso de todas as entradas.

Uma coisa importante a analisar é que em certas combinações no algoritmo GMDH modificado,  $\alpha = 0.1$ ,  $\text{threshold} = 1.1$  e  $\alpha = 0.9$ ,  $\text{treshold} = 1.01$ , todas

os quase todas as entradas foram selecionadas, indicando que todas elas são relevantes.

Além dessas experiências, foram feitos testes com as situações mostradas nas tabelas abaixo. A tabela 3 exibe os dados comparativos da represa de Sobradinho com 10 entradas e na tabela 4 os dados são da represa de Três Marias.

Tabela 3. Combinações realizadas com os dados da represa de Sobradinho com 10 entradas

Sobradinho					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,1,3,4	0.9	1.01	0,1,2,9
0.1	1.01	0,1,2	0.1	1.01	0,1,2,9
0.1	1.1	0,1,2,9	0.1	1.1	1,2,3,6,7,8,9
0.9	1.1	0,1,2,3,4,5,6,9	0.9	1.1	todas

Tabela 4. Combinações realizadas com os dados da represa de Três Marias com 10 entradas

Três Marias					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,1,2,3,4	0.9	1.01	0,1,2
0.1	1.01	0,1,2	0.1	1.01	0,1,2,9
0.1	1.1	0,1,2,9	0.1	1.1	0,1,2,6,7,8,9
0.9	1.1	todas	0.9	1.1	todas

O comportamento é similar ao do caso com 8 entradas. Nota-se que em alguns casos onde antes tinham como entradas selecionadas 0, 1, 2 e 7 o número 7 foi substituído pelo 9, indicando que o algoritmo considera nesse caso a última variável uma das mais relevantes e também que a combinação  $\alpha = 0.9$ ,  $\text{threshold} = 1.1$ , faz a rede selecionar todas ou quase toda as entradas. Essa combinação representa um alto grau de aprendizado com uma liberdade de 10% de erro em relação ao erro mínimo.

Nas próximas tabelas, serão exibidos as combinações dos dados das represas de Sobradinho e de Três Marias, ambas com 10 entradas sendo 2 delas

ruidosas. Esses dados são o mesmo valor para a entrada adicionada e esse valor ficou entre o menor e o maior valor das variáveis. No caso de Sobradinho, esse valor foi 5000. Em Três Marias, o valor foi de 1000.

Tabela 5. Combinações realizadas com os dados da represa de Sobradinho com 10 entradas sendo 2 ruidosas (entrada 2 e 6)

Sobradinho					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,1,2,9	0.9	1.01	0,1,2,9
0.1	1.01	0,1,2	0.1	1.01	0,1,2
0.1	1.1	0,1,2,3,7,9	0.1	1.1	1,2,3,6,7,8,9
0.9	1.1	0,1,2,7,9	0.9	1.1	todas

Tabela 6. Combinações realizadas com os dados da represa de Três Marias com 10 entradas sendo 2 ruidosas (entrada 3 e 9)

Três Marias					
GMDH padrão			GMDH modificado		
alpha	threshold	entradas	alpha	threshold	entradas
0.9	1.01	0,1,2,9	0.9	1.01	0,1,2,9
0.1	1.01	0,1,2	0.1	1.01	0,1,2,9
0.1	1.1	0,1,2,6,8,9	0.1	1.1	todas
0.9	1.1	0,1,2,5,8,9	0.9	1.1	todas

Com os dados ruidosos a rede não muda significativamente. Nos dados de Sobradinho, por exemplo, a entrada 2, que nesse caso é uma das entradas ruidosas, é selecionada em qualquer uma das combinações de alpha e threshold, e em qualquer algoritmo. Algo parecido acontece nos dados de Três Marias com a entrada 9.

Vemos também que em comparação entre os dados com 10 variáveis, os valores não mudam muito, principalmente nos dados processados pelo algoritmo GMDH modificado.

# Capítulo 6

## Conclusão e Trabalhos Futuros

Considerando todo o processo que levou ao desenvolvimento dessa ferramenta esse capítulo dará as conclusões, dificuldades encontradas e idéias para trabalhos futuros.

### 6.1 Dificuldades encontradas

A quantidade de variáveis que podem ser manipuladas é grande e isso acarretou numa dificuldade de uma configuração fixa de valores e quais dessas variáveis deveriam ser analisadas.

Medir o desempenho de uma rede neural também não é algo trivial. Dentre as possibilidades, o erro, que é a diferença entre o valor desejado e o valor obtido, nos diz o quanto a rede neural está próxima da resposta esperada e isso pode medir seu desempenho.

Não ficou claro exatamente o porquê de a rede GMDH dar respostas parecidas no caso de 10 variáveis originais e de 8 variáveis originais + 2 variáveis ruidosas. Os valores ruidosos parecem interferir, mas provavelmente com pouca importância na resposta da rede.

### 6.2 Conclusões sobre o experimento

Visto o experimento, seu comportamento, seus gráficos e tabelas, podemos concluir que:

- As redes GMDH podem realmente ser utilizadas como selecionadoras de variáveis de entrada para uma rede neural. Cada caso define seu grau de satisfação e exigência e através disso, podem-se configurar os valores da rede GMDH.
- Cada procedimento, algoritmo, deve ser adaptado à necessidade em questão, como os valores de alpha e threshold ou a função erro ou a função objetiva.

- Esperava-se que o algoritmo selecionasse a maioria das variáveis, pois em previsão podemos considerar que todas são relevantes, umas mais que outras.
- As variáveis de entrada mais importantes num conjunto de dados usados para previsão são as mais antigas, na ordem 0, 1, 2,..., m.
- O algoritmo proposto nessa ferramenta, o algoritmo GMDH com modificações, consegue uma estabilidade em relação às respostas. Além disso, é mais tolerante a mudanças de configurações das variáveis alpha e threshold que o algoritmo original.
- Os gráficos mostram nesse caso em estudo que quanto mais próximos da quantidade original de entradas, mais ela se comporta como o gráfico de uma rede sem seleção de variáveis, o que era esperado.
- As informações obtidas não foram suficientes para demonstrar a eficiência que as redes GMDH podem dar nas suas respostas.
- O algoritmo modificado é sensivelmente melhor que o algoritmo original, mas ainda abaixo das expectativas para essa ferramenta.

### **6.3 Idéias para trabalhos futuros**

Para efeito de testes futuros, recomendamos usar também dados classificatórios. Um estudo de caso pode mostrar uma forma de utilização dessas redes na prática. Além disso, outras variáveis podem ser levadas em consideração na rede neural, como o erro médio quadrático sobre os dados de validação. O número de ciclos pode ser variável e ser incluído na análise, mostrando se uma rede gerada pelo algoritmo GMDH pode ser treinada melhor e mais rapidamente.

### **6.4 Considerações finais**

Fica claro que essa é uma área de pesquisa que merece mais atenção. As redes GMDH já são utilizadas em variadas situações e está substituindo técnicas menos eficientes e menos eficazes. Apesar disso, ainda é uma rede pouco conhecida no Brasil e, portanto com menos pessoas especializadas. Esperamos que esse trabalho

de conclusão de curso, pelo ao menos tenha despertado interesse de novos pesquisadores que poderão encontrar mais utilidades para essa técnica.



# Bibliografia

- [1] RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 2ª Edição. ed. [S.I.]: Elsevier/Campus. Traduzida.
- [2] VALENÇA, M. **Aplicando Redes Neurais - um guia completo**. [S.I.]: Livro Rápido - Elógica, 2005. 23-28 p.
- [3] WHISTON. Protein classification artificial neural system. **Protein Sci**, n. 1, 1992.
- [4] THIAGO BAPTISTA RODRIGUES, J. L. R. M. E. C. M. Seleção de variáveis e classificação de padrões por redes neurais como auxílio ao diagnóstico de cardiopatia isquêmica, Rio de Janeiro, 2008.
- [5] SILVA, M. A. Mapas Auto-Organizáveis na Análise Exploratória de Dados Geoespaciais Multivariados, Recife, Março 2004.
- [6] VALENÇA, M. **Fundamentos das Redes Neurais**. [S.I.]: Livro Rápido - Elógica, 2007.
- [7] WIKIPEDIA. Homeostase. **Wikipedia**. Disponível em: <<http://pt.wikipedia.org/wiki/Homeostase>>..
- [8] SERGIO. Informativo digital Einstein. **Informativo digital Einstein: respostas**, Maio 2007. Disponível em: <<http://informativodigitaleinstein.blogspot.com/2007/05/respostas-com-comentarios-1.html>>.
- [9] VALENÇA, M. J.; VALENÇA, I. C. Uma Rede GMDH com algoritmo multiplicativo aditivo e neurônio ativo para previsão de vazões. **VIII Simpósio de Recursos Hídricos do Nordeste**, 2006.
- [10] SILVA, V. L. D. HOMEOSTASIA E REOSTASIA. **HOMEOSTASIA E REOSTASIA**, Recife. Disponível em: <<http://www.crono.icb.usp.br/homeostasiaereostasia.htm>>.
- [11] FARLOW, S. J. **Self-organizing methods in modeling - GMDH type algorithms**. [S.I.]: Dekker, v. 54, 1984.

[12] STUDIES, N. I. F. S. GMDH - History. **Group Method of Data Handling (GMDH) for data mining, forecast algorithm optimization, fuzzy models analysis, statistical learning networks and modeling software systems.** Disponível em: <[http://www.gmdh.net/GMDH\\_his.htm](http://www.gmdh.net/GMDH_his.htm)>.

[13] WIKIPEDIA. Group method of data handling. **Wikipedia.** Disponível em: <<http://en.wikipedia.org/wiki/GMDH>>.

[14] DUC T. PHAM, X. L. **Neural Networks for Identification, Prediction and Control.** [S.l.]: Springer.