

MODELOS COMPUTACIONAIS BASEADOS NA BIOLOGIA: ALGORITMOS GENÉTICOS E REDES NEURAIS ARTIFICIAIS

RESUMO

Neste artigo são apresentados dois modelos computacionais inspirados na biologia, a saber, Algoritmos Genéticos e Redes Neurais Artificiais. Os Algoritmos Genéticos imitam o processo de evolução biológica da Seleção Natural na busca de indivíduos mais aptos em uma determinada população, de modo que eles podem ser utilizados com bastante eficiência em problemas de otimização. As Redes Neurais Artificiais, por sua vez, baseiam-se no processo de aprendizagem dos neurônios, altamente paralelo, sendo muito útil na modelagem e previsão de sistemas não-lineares complexos. É feito, ainda, um exemplo utilizando cada um dos modelos descritos.

P. F. Alencar

*Estudante do Curso de
Informática da Universida-
de de Fortaleza-UNIFOR*

A. V. R. Ribeiro

*Estudante do Curso de
Informática da Universida-
de de Fortaleza - UNIFOR
Bolsista do PIBIC/CNPq*

M. A. S. Freitas

*Prof. e Coord. do Grupo
de Pesquisas em Recur-
sos Hídricos, Meio Ambi-
ente e Computação
Aplicada da Universidade
de Fortaleza - UNIFOR*

ABSTRACT

It is presented two computational models based on biological mechanisms, which are: Genetic Algorithms and Artificial Neural Networks. Genetic Algorithms mimic biological evolution by means of natural selection in the search for better individuals of a population, therefore Genetic Algorithms are efficient optimizers. Artificial Neural Networks, otherwise, are based on the process of the human neurons learning, i.e. massively parallel, and they are good approaches to modelling and forecasting complex non-linear systems. This paper also describes examples of applications of both systems.

1. INTRODUÇÃO

Durante os últimos anos tem havido um interesse crescente na resolução de problemas utilizando sistemas/programas baseados na biologia, notadamente

nos princípios da evolução natural e no funcionamento dos neurônios no cérebro. Os programas que se norteiam nos fundamentos da Teoria da Evolução Natural são ge-

nericamente denominados de Programas Evolutivos (PE). Tais sistemas, muito usados em problemas de otimização, empregam uma população de possíveis soluções e por meio de diversos processos de seleção, com base na aptidão dos indivíduos, e de "operações genéticas" (acasalamento, recombinação ou crossover, mutação, etc.) buscam atingir a solução ótima de um determinado problema. As primeiras Estratégias Evolutivas (EE) remontam aos estudos de Ingo Rechenberg e Hans-Paul Schwefel, na década de 60, no Instituto de Mecânica dos Fluidos na Universidade Técnica de Berlim, Alemanha, na busca de formas ótimas de objetos em experimentos de túneis de vento (RECHENBERG, 1973; SCHWEFEL, 1981). Uma outra grande contribuição neste campo deve-se a J. H. HOLLAND (1975) com o livro *Adaptation in Natural and Artificial Systems*.

Um Programa Evolutivo, conforme MICHALEVICZ (1996) é um algoritmo probabilístico que utiliza uma população de indivíduos, $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ para a iteração t . Cada indivíduo representa uma solução potencial do problema em questão. Cada solução x_1^t é avaliada de modo a se obter uma medida de sua 'aptidão'. Uma nova população (iteração $t+1$) é formada pela seleção dos indivíduos mais aptos (etapa de seleção). Alguns indivíduos da nova população sofrem transformações (etapa de alteração) através de operações genéticas (mutações, crossover, etc.) para originar novas soluções. Após um determinado número de gerações o programa converge em direção a 'solução ótima'.

Já as Redes Neurais Artificiais (RNA) são sistemas baseados no atual entendimento do funcionamento dos sistemas nervosos biológicos, embora negligenciando muito dos detalhes biológicos. As RNA são sistemas massivamente paralelos que têm a capacidade de encontrar relações em sistemas não-lineares complexos.

2. ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AG) são exemplos de Programas Evolutivos que imitam a evolução biológica na procura por melhores indivíduos dentro de uma população variável. Em geral, eles executam procuras através de um conjunto de alternativas com o objetivo de en-

contrar a melhor alternativa respeitando os critérios requeridos. Estes critérios são expressos em termos de uma função-objetivo que é referida como *fitness function*. Os Algoritmos Genéticos são eficientes otimizadores. Eles são formados pela criação de uma **população de indivíduos** representados por **cromossomos**. Os cromossomos são representados normalmente por strings de comprimento fixo, análogas aos cromossomos do nosso próprio DNA. A maioria das aplicações de Algoritmos Genéticos utilizam o código binário ou o código grey para formar seus cromossomos.

2.1 Analogia entre os Algoritmos Genéticos e a Genética

O vocabulário dos Algoritmos Genéticos provém da Genética Natural. Assim fala-se, por exemplo, de indivíduos, cromossomos, genes, genótipo, fenótipo, etc. A tabela 1 a seguir apresenta alguns desses paralelos:

| Algoritmos Genéticos | Genética Natural |
|------------------------|------------------|
| Strings | Cromossomos |
| Bits Caracteres | Gens |
| Função-objetivo | Fitness Function |
| Posição do string | Locus |
| Estrutura | Genótipo |
| Conjunto de Parâmetros | Fenótipo |
| Não-linearidade | Epistase |

Tabela 1: Algoritmos Genéticos versus Genética Natural

2.2 Vantagens no Emprego de Algoritmos Genéticos na Otimização

Em problemas de otimização de pequeno porte existem diversos métodos clássicos disponíveis (GILL et al., 1978). Em resolução de problemas complexos e de grande porte os Algoritmos Genéticos têm se mostrado entretanto muito úteis e ágeis. A seguir são listadas algumas das vantagens da utilização de AG em relação aos métodos tradicionais de otimização:

- Os AGs partem de um conjunto de possíveis soluções (população) e não de um único ponto;
- Os AGs utilizam diretamente a função-objetivo (fitness function) e não as derivadas 1a. ou 2a.;
- São algoritmos randômicos e utilizam paralelismo implícito;
- São algoritmos robustos (simples e eficientes: *exploration vs. exploitation*)

2.3 Operações Genéticas

As operações genéticas são importantes para que a população se diversifique e preserve as características de adaptação adquiridas pelas gerações anteriores. As mais usuais são as descritas a seguir:

- **População:** um AG começa com a seleção randômica de uma população de n indivíduos dos 2^n possíveis candidatos.
- **Seleção:** uma nova população de cromossomos é selecionada da população inicial através da seleção de cromossomos com mais alto valor da fitness function, ou seja, indivíduos mais capazes.
- **Mutação:** Dado um cromossomo qualquer, um de seus bits é escolhido, com determinada probabilidade, para ser trocado simulando a mutação.

Exemplo: Antes: 1011010 Depois: 1011110

- **Crossover (Recombinação Genética):** É feita uma troca de cromossomos, entre os pares selecionados, a partir de uma certa posição na cadeia de bits.

Exemplo: Antes: 1011|011 Depois: 1011|110
 1101|110 1101|011

2.4 Um Algoritmo Genético

De modo geral um Algoritmo Genético pode ser explicitado como a seguir. O fluxograma equivalente pode ser visto na figura 1:

1. Selecionar uma população inicial p^k .
2. Determinar para cada cromossomo x o valor da fitness function $f(x)$.
3. Gerar uma nova população por algum procedimento de seleção natural.
4. Se os valores já satisfazem o critério então parar, senão passo 5.

5. Produzir uma população de novos cromossomos $p^{(k+1)}$ utilizando uma das operações descritas.
6. Substituir a população p^k pela população $p^{(k+1)}$ produzida em 4, aumentar uma unidade do valor de k e ir para passo 2.

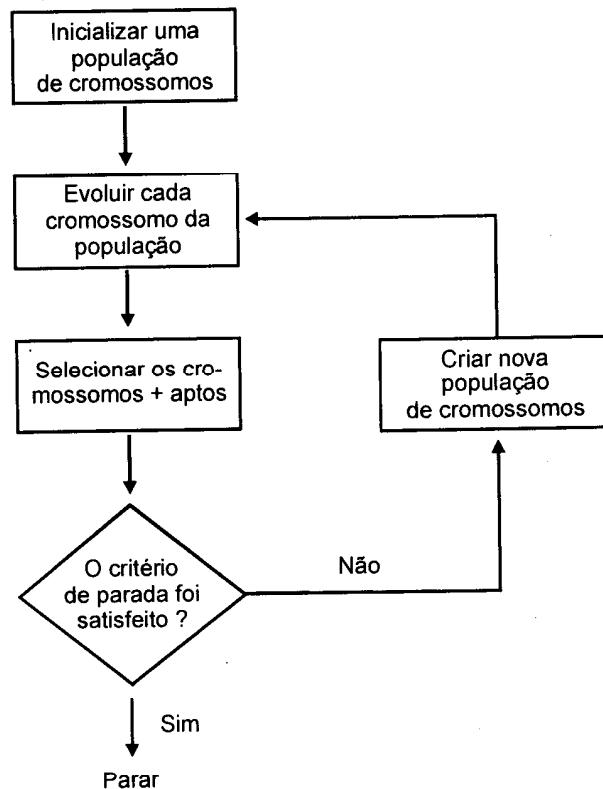


Figura 1: Fluxograma de um Algoritmo Genético (KLIR & YUAN, 1995)

2.5 Os Conceitos Schemata e Building Blocks

Cada indivíduo pode ser considerado como representante de uma classe de indivíduos com características similares. Por exemplo: o indivíduo 1000101 pertence à classe de indivíduos com o primeiro bit=1 e o símbolo que o representa é 1*****. Cada classe e seu símbolo correspondente é chamada de **schema** (plural: **schemata**). Schematas com comprimento pequeno, baixa ordem e fitness function acima da média irão receber um aumento no número de tentativas nas gerações subsequentes. Tais schematas são denominados de **building blocks**.

3. REDES NEURAIS ARTIFICIAIS

Recentemente foi alcançado significativo progresso nos campos do Reconhecimento do Padrão e da Teoria de Sistemas com o uso de Redes Neurais Artificiais (RNA). As redes neurais são estruturas matemáticas bastante flexíveis capazes de identificar relações não-lineares e de descrever processos complexos. A denominação Redes Neurais Artificiais é dada aos modelos que tentam reproduzir a estrutura e o funcionamento dos neurônios no cérebro (KOSKO, 1992; FREITAS & BILLIB, 1996). Uma RNA compõe-se de um elevado número de elementos, denominados neurônios (*cells, units*) e um grande número de ligações, conhecidas como sinapses. A cada ligação (*links, connections*) é associado um peso, que está intrinsecamente relacionado à capacidade de aprendizado da rede (FREITAS, 1998). Fala-se, portanto, de um sistema "inteligente". Além disso, trata-se de um processo distribuído paralelo (*parallel distributed processing* - PDP).

A criação de uma RNA baseia-se nas seguintes tarefas (FREITAS, 1997):

- Determinação das características da rede: topologia; tipos de ligações; ordenamento das ligações e pesos.
- Determinação das características dos neurônios: funções de entrada, ativação ou transferência e de saída.
- Determinação da dinâmica da rede: geração dos valores iniciais dos pesos das ligações; processos de otimização e regras de aprendizagem.

3.1 Formas de Aprendizagem

Uma característica fundamental das RNA é a capacidade de aprendizado. Existem diversas formas teóricas de como uma Rede Neural Artificial pode aprender, quais sejam: 1) Acréscimo de novas ligações; 2) Retirada de ligações existentes; 3) Modificação dos valores dos pesos w_{ij} das ligações; 4) Modificação dos valores patamares dos neurônios; 5) Modificação das funções de ativação, propagação e saída; 6) Acréscimo de novos neurônios; 7) Retirada de neurônios. A forma mais largamente utilizada é a modificação dos valores dos pesos das ligações entre os neurônios.

3.2 Tipos de Redes Neurais Artificiais

Dependendo da dinâmica da rede, bem como dos neurônios e da topologia foi desenvolvida uma infinidade de Redes Neurais Artificiais, a saber: o modelo Perceptron (ROSENBLATT, 1962); o modelo Backpropagation (RUMMELHART et al., 1986); o modelo ADALINE (WIDROW & HOFF, 1960); a rede de KOHONEN (KOHONEN, 1984); o modelo HOPFIELD (HOPFIELD, 1982), o modelo ART (CARPENTER & GROSSBERG, 1987) etc. Doravante são explicados alguns aspectos dos modelos mais freqüentes na literatura.

- 1) Backpropagation (retroalimentação): é o método mais conhecido e utilizado. Uma função de aprendizagem simples é dado apenas com o parâmetro de aprendizagem η . Uma versão deste método emprega a taxa de aprendizagem, 'momentum' e 'flat spot elimination'. Uma terceira versão é a denominada batch-Backpropagation.
- 2) Radial-Basis Functions (RBF) e Hyper-Basis Functions (HBF): são redes feedforward com uma camada de neurônios escondida (hidden) com funções de ativação radiais simétricas. O centro dessas funções são pontos, os quais tentam aproximar a rede e ao mesmo tempo preencher os critérios de suavidade. Uma vantagem deste tipo de rede é que, existe uma solução matemática quando há uma igualdade entre o número de padrões de entrada e o número de neurônios escondidos. Para elevado número de dados de entrada faz-se necessário usar, ou uma escolha aleatória ou um algoritmo de agrupamento não-supervisionado.
- 3) Time-Delay-Netze (TDNN): são redes feedforward especiais nas quais neurônios recebem suas entradas de um campo receptivo de uma camada anterior, a qual representa a série de tempo de um único neurônio.
- 4) Jordan-Netze, Elman-Netze e Rede Hierárquica Elman generalizada: apropriadas para o prognóstico de série de tempo. Essas redes podem perceber o decurso de uma seqüência de padrões em neurônios internos contextuais e dessa forma melhor prognosticar séries de tempo do que redes feedforward com uma janela de entra-

da fixa. Elas podem, entretanto, ser ainda com relativa rapidez treinadas com variantes dos métodos de retroalimentação.

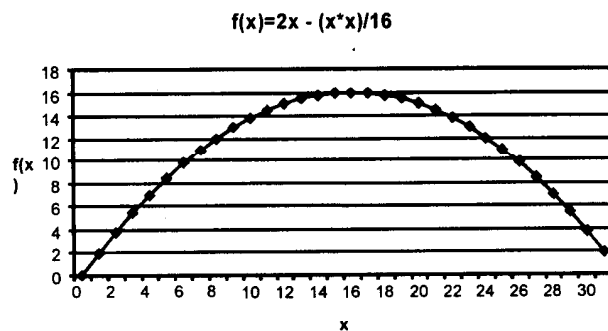
3.3 Campos de Aplicações das Redes Neurais Artificiais

As Redes Neurais Artificiais têm sido aplicadas nos mais diversos campos da ciência, a saber: prognóstico de estrutura secundária de proteínas; administração de projetos de investimentos; diagnóstico do sistema de controle de automóveis (piloto automático); identificação de alvos militares detectados pelo sonar submarino; previsão de ataques e epilepsia; previsão da metástase em câncer gástrico; previsão de crises econômicas; previsão da cotação das ações na bolsa de valores; previsão de falência de associações de poupança e empréstimo; reconhecimento de voz e caligrafia; reconhecimento de poços de petróleo (óleo e gás); otimização do manejo de sistemas de abastecimento d'água; diagnóstico de textura de materiais (pedras) / controle de qualidade, etc.

4. APLICAÇÕES

4.1 Aplicação de Algoritmos Genéticos

Para ilustrar o uso de Algoritmos Genéticos na determinação do máximo de função em um determinado intervalo é apresentada uma aplicação para a função $f(x) = 2x - x^2/16$ no intervalo $[0,31]$.



Dados iniciais:

- Tamanho da População=6
- Comprimento do cromossomo=5
- Número máximo de gerações=10
- Probabilidade de crossover= 0.25
- Probabilidade de mutação=0.10

1ª Geração

| String | X | Fitness |
|--------|-------|---------|
| 11010 | 26,00 | 9.7500 |
| 01011 | 11,00 | 14.4375 |
| 01010 | 10,00 | 13.7500 |
| 01011 | 11,00 | 14.4375 |
| 00101 | 5,00 | 8.4375 |
| 10110 | 22,00 | 13.7500 |
| Média | | 12.4271 |

2ª Geração

| String | X | Fitness |
|--------|-------|---------|
| 00010 | 2,00 | 3.75 |
| 10111 | 23,00 | 12.9375 |
| 00111 | 7,00 | 10.9375 |
| 01010 | 10,00 | 13.7500 |
| 01100 | 12,00 | 15.0000 |
| 10000 | 16,00 | 16.0000 |
| Média | | 12.0625 |

4.2 Aplicação de Redes Neurais Artificiais

A seguir é apresentada uma aplicação de RNA no aprendizado do problema do XOR (ou exclusivo). Para isso foi utilizada uma rede de apenas três camadas (2,2,1). O algoritmo empregado foi o Backpropagation, com função de ativação do tipo sigmodal, taxa de aprendizagem de 0.5 e termo de momento igual a 0.8. Para um erro de 2% durante o processo de treinamento o algoritmo converge após cerca de 510 iterações, conforme listagem dos resultados abaixo.

IO MATRIX

| | | |
|--------|--------|--------|
| 1.0000 | 1.0000 | 0.0000 |
| 0.0000 | 0.0000 | 0.0000 |
| 0.0000 | 1.0000 | 1.0000 |
| 1.0000 | 0.0000 | 1.0000 |
| 1.0000 | 1.0000 | 0.0000 |

DESIRED MATRIX

0.0000
0.0000
1.0000
1.0000
0.0000

INPUT MATRIX

1.0000 1.0000
0.0000 0.0000
0.0000 1.0000
1.0000 0.0000
1.0000 1.0000

Event # 5 0.098458
Event # 25 0.091546
Event # 45 0.085091
Event # 65 0.079064
Event # 85 0.073436
Event # 105 0.068181
Event # 125 0.063275
Event # 145 0.059087
Event # 165 0.055537
Event # 185 0.052180
Event # 205 0.049006
Event # 225 0.046177
Event # 245 0.043981
Event # 265 0.041862
Event # 285 0.039820
Event # 305 0.037853
Event # 325 0.035962
Event # 345 0.034146
Event # 365 0.032487
Event # 385 0.031283
Event # 405 0.030090
Event # 425 0.028912
Event # 445 0.027753
Event # 465 0.026614
Event # 475 0.026054
Event # 485 0.025499
Event # 495 0.024951
Event # 505 0.024409

Network response:

inputvec: 1.00 1.00 response: 0.006
inputvec: 0.00 0.00 response: 0.006
inputvec: 0.00 1.00 response: 1.010
inputvec: 1.00 0.00 response: 1.003
inputvec: 1.00 1.00 response: 0.006

Final Weights

0.0000 0.0000 4.3802 -4.3088 0.0000 0.0000
0.0000 0.0000 -3.8796 3.7129 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 1.5366 0.0000
0.0000 0.0000 0.0000 0.0000 1.8051 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 -3.4103 -3.2185 -0.1126 0.0000

5. CONCLUSÕES

Com o aumento da capacidade computacional é cada vez maior o interesse por modelos inspirados na natureza. Neste artigo apresentou-se aplicação de dois desses modelos (Algoritmos Genéticos e Redes Neurais Artificiais) na determinação do máximo de uma função e no aprendizado do conectivo XOR (ou-exclusivo), respectivamente.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- CARPENTER, G. A. & S. GROSSBERG, 1987: **Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, Computer Vision, Graphics, and Image Processing**, vol. 37, 54-115.
- FREITAS, M. A. S., 1997: Regionale Dürreanalyse anhand statistischer Methoden und Neuro-Fuzzy-Systemen mit Anwendung für Nordost-Brasilien, **Tese de Doutorado**, Inst. für Wasserwirtschaft, Hydrologie und landwirtschaftl. Wasserbau, Univ. Hannover.
- FREITAS, M.A.S., 1998: **A Decision Support System for Drought Forecasting and Reservoirs Management in Northeast-Brazil**, SBMET (submitted to publication).
- FREITAS, M.A.S. & M.H.A. BILLIB, 1997: **Drought Prediction and Characteristic Analysis in Semi-arid Ceará - Northeast Brazil, Symposium, Sustainability of Water Resources Under Increasing Uncertainty**, IAHS Publ. N° 240, pág. 105-112, Rabat, Marrocos.
- GILL, P. E.; W. MURRAY & M. H. WRIGHT, 1978: **Practical Optimization, Academic Press**, London.
- HOLLAND, J. H., 1975: **Adaptation in Natural and Artificial Systems**, MIT Press.

- HOPFIELD, J. J., 1982: **Neural Networks and Physical Systems with Emergent Collective Computational Abilities**, Proc. of the National Academy of Science, USA, Biophysics, 79, 2554-2558.
- KLIR, G. J. & B. YUAN, 1995: **Fuzzy Sets and Fuzzy Logic: Theory and Application**, Prentice Hall, N.J.
- KOHONEN, T., 1984: **Self-Organization and Associative Memory**, Springer Verlag, New York.
- KOSKO, B., 1992: **Neural Networks and Fuzzy Systems**, Prentice Hall.
- MICHALEWICZ, Z., 1996: **Genetic Algorithms + Data Structures = Evolution Programs**, 3^o ed., Springer Verlag.
- RECHENBERG, I., 1973: **Evolutio-nsstrategie: Optimierung technischer System nach Prinzipien der biologischen Evolution**, Frommann-Holzboog Verlag, Stuttgart.
- ROSENBLATT, F., 1962: **Principles of Neurodynamics**, Spartan Books, New York.
- RUMMELHART, D. E.; G. E. HINTON & R. J. WILLIAMS, 1986: **Learning Representations by Back-propagations Errors**, Nature, 323(9), 533-536.
- SCHWEFEL, H.-P., 1981: **Numerical Optimization for Computer Models**, John Wiley, Chichester, UK.
- SERAPHIN, M., 1994: **Neuronale Netze und Fuzzy-Logik: Verknüpfung der Verfahren, Anwendungen, Vor- und Nachteile, Simulationsprogramm**, München, Franzis Verlag.
- WIDROW, B. & M. E. HOFF, 1960: **Adaptive Switching Circuits**, IRE WESCON Convention Record, New York, 96-104.
- ZELL, A., 1994: **Simulation Neuronaler Netze**, Addison Wesley.