

DESENVOLVIMENTO DE AGENTES AUTÔNOMOS EM GERÊNCIA DE REDES DE COMPUTADORES

A.S.M. DE FRANCESCHI¹, J.M. BARRETO², M. ROISENBERG²

Universidade Federal de Santa Catarina
Programa de Pós-graduação em Engenharia Elétrica
Grupo de Pesquisas em Engenharia Biomédica¹
Departamento de Informática e Estatística²
Caixa Postal 476 Trindade - 88.040-900 Florianópolis SC

RESUMO

O presente trabalho apresenta uma metodologia para aplicação de redes neurais artificiais na solução de problemas de gerência de redes de computadores. As redes neurais são implementadas através de agentes de software autônomos estáticos ou dinâmicos. Agentes estáticos podem ser construídos utilizando-se heurísticas obtidas de um especialista ou administrador de redes. Agentes dinâmicos podem ser construídos a partir de exemplos obtidos da própria rede. O tipo de comportamento de gerência, reativo ou pró-ativo, também é considerado. O comportamento reativo é o mais comum, após o problema ocorrer busca-se a solução. É o caso de sistemas de diagnóstico de falhas (ou "troubleshooting"). O comportamento pró-ativo, ao contrário, é uma espécie de controle preventivo da rede. É possível utilizar as cinco áreas funcionais propostas pelo modelo de referência OSI para classificar a necessidade de agentes autônomos estáticos ou dinâmicos.

ABSTRACT

This work presents a methodology to develop autonomous agents for network management. There are two kinds of agents to develop: static or dynamic agents. The first one can be implemented, using heuristics obtained from an expert or the network administrator, through production rules or feed forward neural networks. Using the network examples we can construct dynamic agents. The neural network may be trained to solve a problem using some examples. Furthermore, the behavior of the management must be considered, the network management may be reactive or proactive. Normally, we have the reactive behavior when the problem occurs and after we will search for a solution. We may see in diagnostic or troubleshooting systems for Fault Management. On the contrary, the proactive behavior is a preventive control of the network.

We divided the network management in the five functional areas proposed by OSI Model Reference. Thus, each area has a different intelligent solution. The Fault and Accounting management can be solved by heuristics, because they are distributed problems with static features. However, the Performance, Configuration and Security management have dynamic characteristics, they have some properties which

change conforming some network attributes, as users, network elements, topologies, utilization rates, etc.

1. INTRODUÇÃO

Avançam o uso de novas técnicas para resolução de problemas complexos, tais como agentes autônomos, redes neurais artificiais e computação evolucionária. No entanto, existem poucos trabalhos voltados ao estudo da computabilidade e da complexidade dos problemas solucionáveis por estas técnicas. A computabilidade das redes neurais foi tratada inicialmente por McCulloch e Pitts (1943) usando lógica [1]. Eles provaram a equivalência entre uma rede neural com dispositivos de entrada a uma Máquina de Turing. Arbib [2] propôs uma demonstração intuitiva desta equivalência em 1964. Entretanto, as duas abordagens se diferenciam em relação à complexidade, porque a quantidade de recursos empregados na solução do mesmo problema são diferentes.

No sentido de criar uma teoria da complexidade para redes neurais artificiais, a primeira contribuição foi feita por Minsky e Papert em 1969. Eles provaram que uma rede neural direta precisa ter uma camada intermediária entre a camada de entrada e a de saída para solucionar problemas não-linearmente separáveis [3].

Mais recentemente, Barreto (1996) provou que é mais adequado utilizar redes recorrentes do que redes diretas para resolver problemas dinâmicos [4]. Entretanto, é muito comum encontrar, na literatura científica, o uso de redes neurais diretas com neurônios estáticos para solucionar problemas dinâmicos. A aplicação desta abordagem estática causa dois inconvenientes: o primeiro é que uma rede neural direta não consegue cobrir todos os diferentes estados de um sistema dinâmico. O segundo é que para cobrir todos os estados seria necessário uma rede direta muito grande, talvez impossível de convergir a uma solução.

Este trabalho propõe uma metodologia híbrida para o desenvolvimento de agentes autônomos considerando como característica principal a análise do problema que poderá ser estático ou dinâmico. Esta abordagem serve como alternativa ao gerenciamento de redes centralizado e proprietário, porque os agentes deverão possuir autonomia e ser independente de qualquer plataforma de gerência.

2. METODOLOGIA

Analisando a gerência de redes como um todo tem se um problema extremamente complexo. No entanto, é possível subdividi-la em cinco áreas funcionais utilizando o modelo funcional proposto pela OSI. Considera-se também dois tipos de comportamentos da gerência de redes: reativo ou pró-ativo. Os problemas foram então, investigados conforme suas características estáticas ou dinâmicas. Na literatura encontram-se disponíveis diversos mecanismos que podem ser auxiliar no desenvolvimento de agentes autônomos, como é o caso do modelo do quadro-negro [5]. Porém estes mecanismos são normalmente aplicados para realizar tarefas como a comunicação, a coordenação e a cooperação entre os agentes na sua fase de implantação. A presente metodologia aborda como extrair o conhecimento necessário para a construção de cada agente de modo que uma solução satisfatória para um problema distribuído.

Definição do Problema [6]: Um problema é o objeto matemático $P = \{D, R, q\}$, consistindo de dois conjuntos não vazios, D os dados e R os resultados possíveis e de uma relação binária $q \subset D \times R$, a condição que caracteriza uma solução satisfatória, associando a cada elemento do conjunto de dados a solução única desejada. Segundo Barreto [6], o problema pode ser representado matematicamente por uma função. Resolver o problema será então encontrar um modo de implementar esta função ou de aproximá-la com o conhecimento que se dispõe.



Fig.1 - Metodologia Híbrida para solução de problemas distribuídos.

Após a definição do problema é necessário definir o comportamento de gerência que será adotado: reativo quando as ações de gerência são realizadas após o aparecimento de algum problema ou, pró-ativo no caso de se adotar um gerenciamento com ações preventivas. Outra questão, é determinar se a solução do problema deve ter características estáticas ou dinâmicas. Estáticas quando não existe o conceito de estado, e dinâmicas quando forem identificados trocas de estados na solução do problema.

Tabela 1 - Caracterização das áreas funcionais de gerência de redes.

Área Funcional	Comportamento da Gerência		Características	
	Reativa	Pró-ativa	Estáticas	Dinâmicas
Falhas	X	X	X	X
Desempenho		X		X
Configuração	X			X
Contabilização	X		X	
Segurança	X	X		X

A Tabela 1 apresenta uma caracterização das cinco áreas funcionais da gerência de redes definidas pelo modelo de referência OSI [7]. Na Gerência de Falhas podem ser adotados os dois comportamentos. Reativo no caso de falhas impossíveis de se prevenir, como é o caso da ação do ambiente (sobre cabos e placas de redes) e a qualidade de peças e equipamentos utilizados. Estes comportamentos poderão ter características estáticas e dinâmicas. A Gerência de Desempenho deve ser pró-ativa para impedir um baixo desempenho ou a degradação do serviço fornecido aos seus usuários. A Gerência de Configuração normalmente é reativa e possui características dinâmicas, ela envolve o crescimento do número de equipamentos e de usuários o que altera o estado da rede. A Gerência de Contabilização é estática e não exige uma solução dinâmica. E finalmente, a Gerência de Segurança pode ser tanto reativa quanto pró-ativa e normalmente, possui características dinâmicas. A partir desta análise é possível definir se o problema exige agentes autônomos estáticos ou dinâmicos.

3. AGENTES AUTÔNOMOS ESTÁTICOS

Para desenvolver agentes autônomos estáticos (AAE) busca-se normalmente heurísticas para a solução do problema. Ou seja, uma forma declarativa de resolver uma função, dando as propriedades que devem ser satisfeitas para solucionar o problema, neste caso, é necessário auxílio de um especialista. O especialista que neste caso poderá ser o administrador de redes, deverá indicar quais os sintomas, para diagnosticar o problema e sugerir as possíveis soluções. Neste caso, tem-se o conjunto de dados, o conjunto das possíveis soluções e a relação entre eles de forma bem definida. Estas informações podem ser implementadas de duas maneiras: Redes Neurais Diretas (paradigma conexionista) ou através de Regras de Produção (paradigma simbólico) ilustradas na Figura 2.

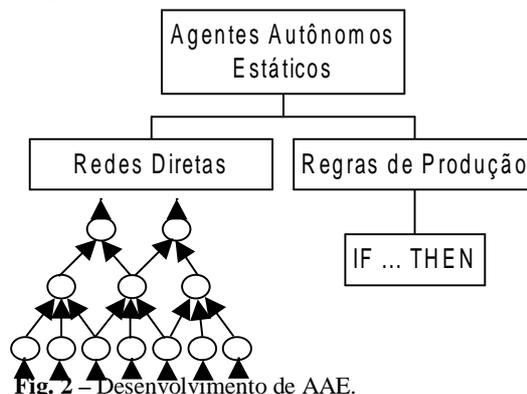


Fig. 2 - Desenvolvimento de AAE.

Um bom exemplo de sistema que pode ser implementado com agentes autônomos estáticos é o de diagnóstico de falhas, ou até mesmo uma base de dados com fichas de falhas da rede e suas possíveis soluções (“troubleshooting”). A seguir segue um exemplo de regras de produção que identificam falhas em pacotes recebidos pela rede [8][9].

```

IF Δ pacotes de entrada > baseline
  AND pacotes de saída < baseline
  AND erros de saída > baseline
  THEN “pacotes errados”

IF Δ pacotes de entrada > baseline
  AND pacotes de saída < baseline
  AND erros de saída > baseline
  AND fila de pacotes de saída > baseline
  THEN “rajada de pacotes errados”
    
```

As Redes Neurais Diretas (Figura 2) são estáticas e não possuem ciclos e são representadas por camadas. Normalmente, possuem camadas de entrada, intermediária e de saída. Além disso são as mais populares, principalmente por existirem métodos de aprendizado fáceis de usar, o mais conhecido é o “backpropagation” [6].

4. AGENTES AUTÔNOMOS DINÂMICOS

Agentes autônomos dinâmicos (AAD) podem ser desenvolvidos com o auxílio de exemplos. Através de exemplos tem-se uma representação indireta do problema, ou seja, apenas a definição do problema é conhecido para um subconjunto de dados possíveis. Problemas dinâmicos, só podem ser “bem resolvidos” se forem utilizadas redes neurais artificiais com características dinâmicas.

Para incluir dinamismo a uma solução conexionista, tem-se:

- Inserindo linhas de atrasos após a camada de entrada em redes neurais diretas;
- Utilizando redes com ciclos e neurônios dinâmicos: redes recorrentes ou redes de Hopfield;

Deseja-se então, conhecer os elementos do conjunto de respostas admissíveis para todos os elementos do conjunto de dados, mesmo aqueles que não estão incluídos na definição da função. Os exemplos, portanto, são utilizados para treinar uma rede neural com algoritmo supervisionado e obter os valores estimados da solução para os outros valores, utilizando a propriedade de generalização. Neste trabalho utiliza-se redes recorrentes [10].

As Redes Neurais Recursivas são redes com ciclos e neurônios dinâmicos, esta dinâmica é provocada pelo retardo entre a saída do neurônio e sua entrada [6]. Através desta topologia de rede neural é possível estimar o conjunto de estados a partir de um padrão de entrada e saída. Estes padrões são fornecidos por uma coleta de exemplos sobre o problema de gerência a ser solucionado.

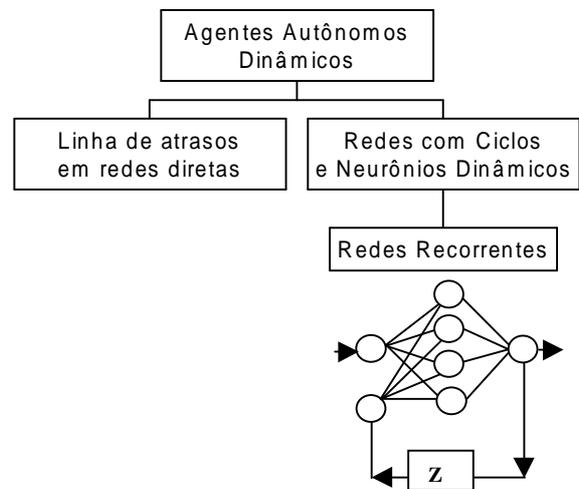


Fig. 3 – Desenvolvimento de AAD.

5. SUMÁRIO E TRABALHOS FUTUROS

O presente trabalho apresentou uma metodologia para desenvolvimento de agentes autônomos para gerência de redes. Esta metodologia defende duas abordagens uma estática e outra dinâmica. A estática é baseada em heurísticas extraídas de um especialista. A vantagem da dinâmica é que dispensa a extração do conhecimento de um especialista, que é uma tarefa muito complicada. Muitas vezes, a solução de um problema é realizada com a intuição, e fica impossível indicar porque determinada ação foi realizada. Como desvantagem tem-se a não garantia da convergência de uma rede neural. Não existem métodos para garantir que uma rede neural chegue a uma solução. No entanto, podem ser fornecidos bons exemplos para que a probabilidade de que a rede neural aprenda um determinado padrão seja boa.

A implementação dos agentes será realizada em Linguagem JAVA. O ambiente desta linguagem é naturalmente distribuído e integra de forma natural os aspectos de segurança. A vantagem principal é a possibilidade de definir os agentes de forma independente, não proprietária a nenhuma plataforma de gerência de redes, muito menos de sistema operacional e processador. Um programa escrito em JAVA e processado em um Pentium, ou em um Power PC ou até mesmo numa SPARC terá um código interpretável por um navegador (“browser”, em inglês) sobre qualquer um desses equipamentos [11]. Além disso, como JAVA é composta por pacotes de classes de objetos, poderá ser criada como expansão da metodologia, uma classe para desenvolvimento de agentes autônomos para gerência de redes.

6. REFERÊNCIAS

- [1] McCulloch W.S., Pitts W.H. “A Logical Calculus of Ideas Immanent in Nervous Activity”, *Bull. of Mathematical Biophysics*, vol.5, pp.115-133, 1943.
- [2] Arbib M.A., *Brains, Machines and Mathematics*, McGraw-Hill, 1964.

- [3] Minsky M. L., Papert S. A. *Perceptrons: an introduction to computational geometry*, MIT Press, 1988
- [4] Barreto J. M. "Conexionismo e Resolução de Problemas". *Concurso de Professor Titular, Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, Florianópolis, SC, 1996.*
- [5] Rich E., Knight K. *Inteligência Artificial*. Tradução Maria Claudia S. R. Ratto. 2ª edição, McGraw-Hill : São Paulo, 1993.
- [6] Barreto J. M. *Inteligência Artificial No limiar do Século XXI Abordagem Híbrida Simbólica, Conexionista e Evolucionária*. 2ª edição revista e aumentada. Florianópolis, 1999.
- [7] Motorola Codex, *The Basics Book of OSI and Network Management*, Motorola University Press, Addison-Wesley, 1993.
- [8] De Franceschi A. S. M., Barreto J. M. "Distributed Problem Solving Based on Recurrent Neural Networks Applied to Computer Network Management". *Proceedings of ICT'99 International Conference on Telecommunications*, IEEE Press, Cheju, Korea, 1999.
- [9] De Franceschi A. S. M., da Rocha M. A., Weber H. L., Westphall C. B. "Employing Remote Monitoring and Artificial Intelligence Techniques to Develop the Proactive Network Management". *Proceedings of IEEE International Workshop on Application of Neural Networks in Telecommunications*, Melbourne, Australia, 1997. pp.116-123.
- [10] Roisenberg M., Barreto J. M., De Azevedo F. M. "Uma Proposta de Modelização para Agentes Autônomos Baseada na Teoria de Sistemas". *Anais do 3o Simpósio Brasileiro de Automação Inteligente*, Vitória, ES, 3 a 5 de setembro, 1997. pp.500-507.
- [11] Lemay L., Perkins C. L. *Teach yourself JAVA in 21 days*. SamsNet Publishing, 1996.