



PROJETO DE INTELIGÊNCIA ARTIFICIAL APLICADA

RELATÓRIO TÉCNICO

L3C001-06/01

LABORATÓRIO DE CONEXIONISMO

E CIÊNCIAS COGNITIVAS

por

Analúcia Schiaffino Morales De Franceschi
Jorge Muniz Barreto
Mauro Roisenberg

Junho, 2001

Resumo

Este projeto estuda as aplicações de Inteligência Artificial Distribuída. Presentemente enfatiza-se a área de redes de computadores, sistema essencialmente distribuído. O principal objetivo é apresentar uma solução viável para o problema da automação de gerência de redes. A gerência de redes é composta por agentes e gerentes passivos. Ou seja, não existe autonomia nenhuma. A tomada de decisões normalmente é realizada por um administrador humano. 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. Os problemas estão sendo então, investigados conforme suas características estáticas ou dinâmicas. Além disso, também estão sendo considerados dois tipos de comportamentos da gerência de redes, reativo ou pró-ativo. O presente trabalho defende então, duas maneiras para solucionar os problemas de gerência de redes. Se o problema possuir caráter estático deverá ser empregada uma solução que satisfaça estas características, normalmente são utilizadas heurísticas para a solução do problema. As duas ferramentas mais conhecidas para solucionar este tipo de problema são as regras de produção (paradigma simbólico da IA) ou redes neurais diretas (paradigma conexionista da IA). Se o problema for de caráter dinâmico poderá ser “bem” solucionado apenas por ferramentas dinâmicas. Conhece-se até o presente momento, três formas de associar dinamismo a uma rede neural artificial: aplicando uma seqüência de retardos à rede, utilizando redes de retroação (com ciclos) ou através do uso de neurônios dinâmicos. Propõe-se então, o uso de exemplos para alcançar às soluções destes tipos de problemas. Quando os valores de entrada e saída são conhecidos, utiliza-se uma rede neural recorrente para estimar as trocas de estado do sistema através do treinamento da rede. Através desta metodologia pretende-se desenvolver soluções adequadas que respeitem as qualidades de cada área funcional da gerência de redes. Estas soluções tomarão a forma de agentes autônomos que auxiliarão o processo de automação da área de gerência de redes de computadores.

Conteúdo

1. Introdução	4
2. Metodologia	7
2.1 Agentes na Solução de Problemas	9
3. Agentes Autônomos	10
3.1 Agentes Autônomos Estáticos	11
3.2 Agentes Autônomos Dinâmicos	12
3.3 Exemplo.....	14
4. Sumário e Trabalhos Futuros.....	17
5. Referências	18

1. Introdução

Existem várias abordagens na área de Inteligência Artificial (IA), nas quais variam a manipulação do conhecimento, no sentido de como adquiri-lo, armazená-lo e empregá-lo. Classificando a IA quanto ao método de solução de problemas tem-se a IA Simbólica (IAS), a IA Conexionista (IAC), a IA Evolucionária (IAE) e a IA Híbrida (IAH). Quanto a localização espacial tem-se ainda a IA Monolítica (IAM) e a IA Distribuída (IAD). A IAS possui como ferramenta básica para manipular o conhecimento a lógica, com suas regras de inferência inspiradas nos silogismos enunciados há mais de 2.000 anos por Aristóteles. A IAC usando RNAs, aplica-se a problemas mal definidos, mas que são conhecidos através de exemplos. Entre os campos de aplicações de técnicas conexionistas, das quais destacam-se as redes neurais artificiais, estão: reconhecimento de padrões; controle de processos industriais; robótica; e também, como opção às técnicas de raciocínio baseado em casos para a resolução de problemas. A IAE os mecanismos utilizados são os mesmos encontrados na evolução biológica. É um exemplo de solução de problema bem definido de sobrevivência de uma espécie em ambiente variável. Pode ser encarada como um método de otimização com restrições variáveis e muitas vezes desconhecidas. A IAH reúne vantagens de mais de um tipo de método de abordagem para a resolução de problemas. A IAM envolve sistemas simples sem modularidade, como é o caso de sistemas especialistas. O funcionamento da IAD depende de um determinado conjunto de partes (ou módulos) para resolver de modo cooperativo um determinado problema. Sua modularidade para encontrar soluções de problemas está diretamente ligada ao conceito de agentes.

O conceito de agentes em IA surgiu no projeto de compreensão da linguagem HEARSAY-II quando surgiu o primeiro modelo (chamado Quadro Negro) para o problema de coordenação de ações entre várias partes do programa de IA. No entanto, um agente inteligente deve ter como característica a capacidade de assumir a função de outro agente que falhou [1].

Nos últimos anos, o paradigma de agentes tem aumentado significativamente sua popularidade. Segundo Hayes [2] deve-se a características como flexibilidade, modularidade e aplicabilidade na solução de uma variedade de problemas. O autor atribui um aumento parcial aos desenvolvimentos tecnológicos na área de computação distribuída e robótica. Estas áreas necessitam de agentes para modelar soluções adequadas, principalmente para problemas distribuídos. Paralelamente tem-se um crescimento de paradigmas de programação orientada-a-objetos, em parte devido à importância deste tipo de programação na estruturação de sistemas baseados em agentes.

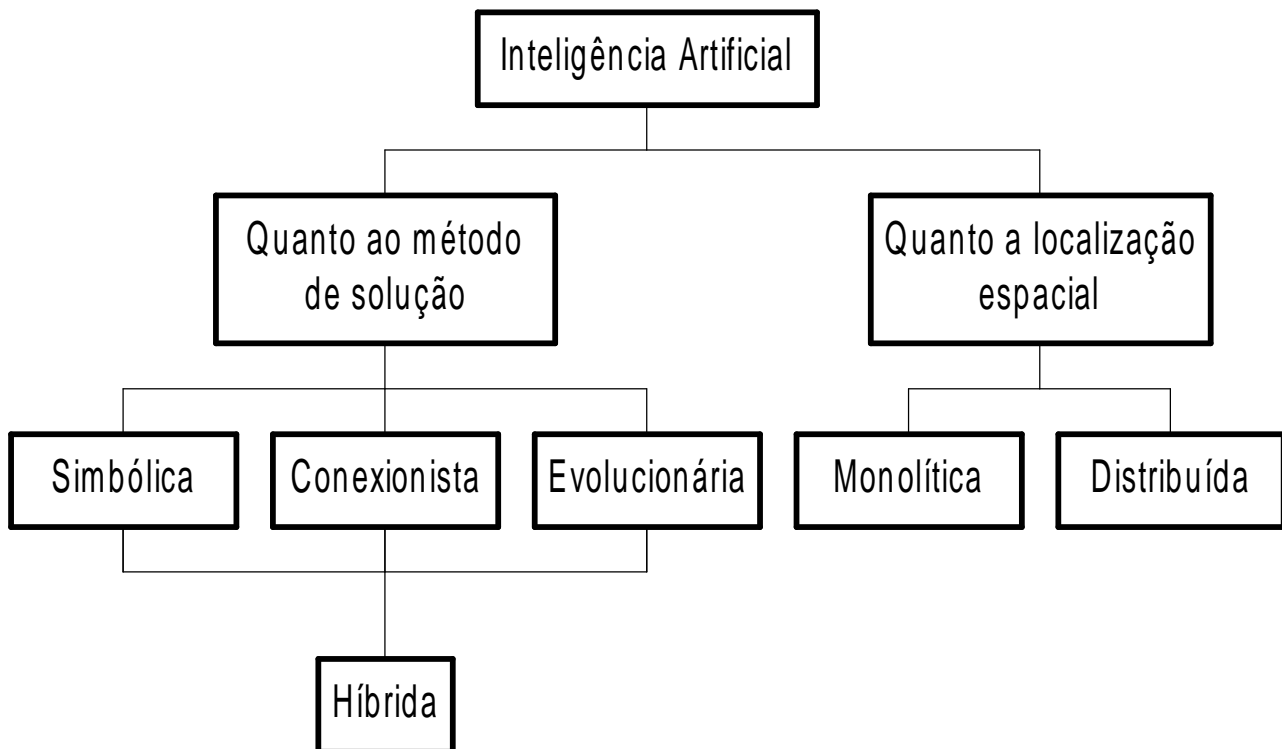


Figura 1 – Várias abordagens de IA.

No ambiente de gerência de redes existem três conceitos importantes: gerentes, agentes e objetos gerenciados. “...Um gerente pode obter informações atualizadas sobre os objetos gerenciados e controlá-los. Para isso, transmite operações de gerenciamento aos agentes. Um agente executa operações de gerenciamento sobre objetos gerenciados. Pode, ainda, transmitir ao gerente as notificações emitidas pelos objetos gerenciados...”[3]. Estes conceitos foram apresentados inicialmente nos protocolos CMIP (*Common Management Information Protocol*), do Modelo de Referência OSI (*Open System Interconnection*), e SMNP (*Simple Network Management Protocol*), da Internet. No entanto, estes gerentes e agentes não possuem nenhuma autonomia, ou seja, não participam do processo de tomada de decisões. A tomada de decisões é normalmente realizada por um especialista conhecido como administrador da rede. Segundo Lesser [4], a autonomia de um

agente está relacionada a capacidade do agente em tomar suas próprias decisões sobre as atividades que devem ser realizadas para alcançar seu objetivo. A autonomia pode ser limitada por políticas embutidas no próprio agente por seu projetista, ou como resultado de interações cooperativas entre outros agentes.

Outra preocupação no desenvolvimento de agentes que utilizam técnicas de IA diz respeito a complexidade de redes neurais. Entenda-se por complexidade a quantidade de recursos necessários para solucionar determinados tipos de problemas. Neste sentido, a primeira grande contribuição foi feita por Minsky e Papert. 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 [5]. Mais recentemente, Barreto provou que é mais adequado utilizar redes recorrentes do que redes diretas para resolver problemas dinâmicos [6]. A aplicação de uma abordagem estática na solução de problemas dinâmicos causa dois inconvenientes, o primeiro porque não permite cobrir todos os diferentes estados de um sistema dinâmico. E o segundo, porque seria necessário uma rede neural muito extensa para cobrir todos os estados possíveis, e desta forma seria muito difícil convergir a uma solução.

No entanto, é possível encontrar na literatura científica vários trabalhos com abordagens de agentes inteligentes em ambientes de redes de computadores em que não existe nenhuma preocupação com características estáticas ou dinâmicas dos problemas abordados. Exemplos de trabalhos voltados a criação de agentes para auxiliar na busca por informações na Internet são encontrados em [7][8]. Outros agentes que possuem aplicação na área de gerência de redes propriamente dita podem ser encontrados em [9][10][11]. A grande maioria destes agentes foram desenvolvidos utilizando regras de produção que são implementadas com o auxílio de uma biblioteca da IBM conhecida como *agglerts*. A metodologia apresentada neste trabalho considera características estáticas e dinâmicas para auxiliar na escolha correta para desenvolver agentes voltados à área de gerência de redes de computadores.

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: gerência de falhas, gerência de desempenho, gerência de configuração, gerência de contabilização e gerência de segurança. 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.

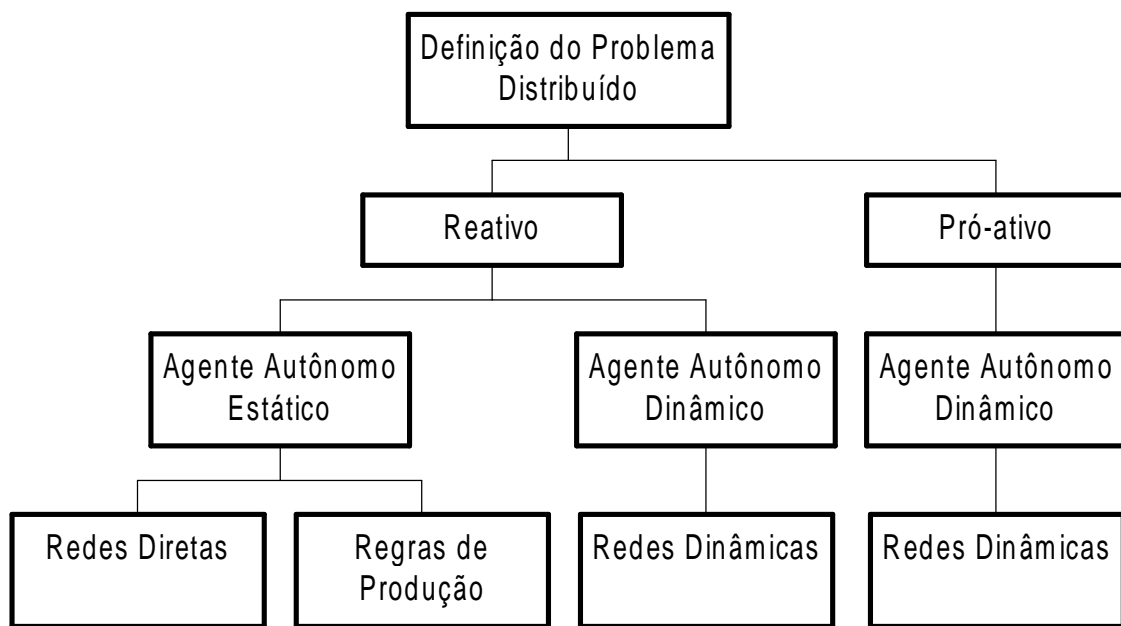


Figura 2 - Metodologia para solução de problemas distribuídos de gerência de redes.

Definição do Problema [1]: Um problema pode ser modelado como 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. 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.

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				
Desempenho				
Configuração				
Contabilização				
Segurança				

Com base na metodologia apresentada anteriormente, a Tabela 1 apresenta uma interpretação para as cinco áreas funcionais de gerência de redes definidas pelo modelo de referência OSI [13]. A Gerência de Falhas pode assumir os dois tipos de comportamento. Reativo no caso de falhas impossíveis de se prevenir, como é o caso das falhas provocadas pela ação do ambiente ou provocadas pela má qualidade de peças e equipamentos utilizados. Estes comportamentos poderão ter problemas de caráter estático ou dinâmico.

A Gerência de Desempenho deve ser pró-ativa em todos os seus aspectos. É uma forma preventiva de garantir a qualidade do serviço oferecido aos usuários. A Gerência de Configuração normalmente é reativa com características dinâmicas. O dinamismo é identificado pelo crescimento do número de equipamentos e de usuários que altera o estado da rede. A Gerência de Contabilização é estática e reativa. A Gerência de Segurança pode ser tanto reativa quanto pró-ativa. Contém características dinâmicas, uma delas é o controle de acesso que é diretamente ligado ao número de usuários. Com esta interpretação é possível definir se o problema a ser resolvido necessita de agentes autônomos estáticos ou dinâmicos.

2.1 Agentes na Solução de Problemas

Nos dias de hoje utiliza-se muito a palavra agente para descrever módulos de software que desempenham uma determinada função. Propõe-se então uma classificação para este paradigma de programação. Não é possível considerar a função de um robô (que é uma agente autônomo de hardware) e um programa coletor de dados de gerência (que também é chamado de agente) no mesmo nível de classificação.

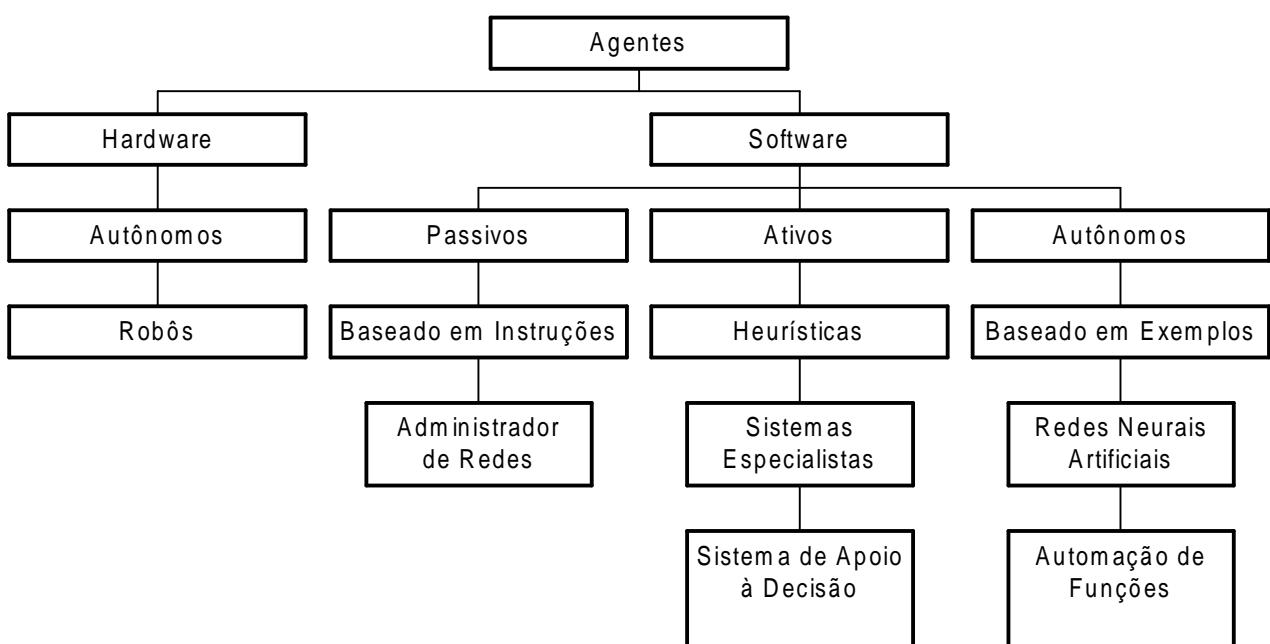


Figura 3 – Classificação do paradigma de agentes.

3. Agentes Autônomos

“Agentes são sistemas computacionais que operam em ambientes dinâmicos e imprevisíveis. Eles interpretam dados obtidos pelos sensores que refletem eventos ocorridos no ambiente e executam comandos em motores que produzem efeitos no ambiente. O grau de autonomia de um agente está relacionado à capacidade de decidir por si só como relacionar os dados dos sensores com os comandos dos motores em seus esforços para atingir objetivos, satisfazer motivações, etc.” [14].

De uma forma geral um agente autônomo (AA) é composto basicamente por: sensores, atuadores e por uma entidade cognitiva responsável pela autonomia do agente. No caso de um agente de software os sensores são as entradas, e os atuadores as saídas. Por autonomia entende-se que é a capacidade de governar-se por si próprio.

A Figura 4 ilustra os componentes básicos de um AA inserido em seu ambiente. Existem outros conceitos para composição de AAs que envolvem desde arquiteturas genéricas, até comportamentos e aprendizado de AAs. Além da autonomia, outro conceito importante para trabalhar-se com AAs é o comportamento. O comportamento é a resposta fornecida pelo AA aos estímulos recebidos do ambiente. Os estímulos são recebidos pelos sensores, mapeados pela entidade cognitiva e refletidos no ambiente através dos atuadores. Este comportamento está diretamente ligado ao grau de autonomia do agente [15].

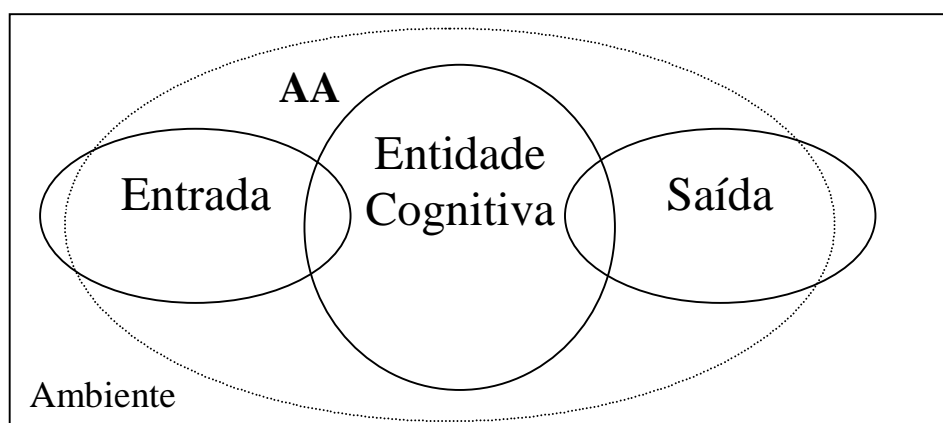


Figura 4 - Agente autônomo em seu ambiente.

3.1 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. Normalmente, extrai-se o conhecimento com o 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. Assim, 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 9.

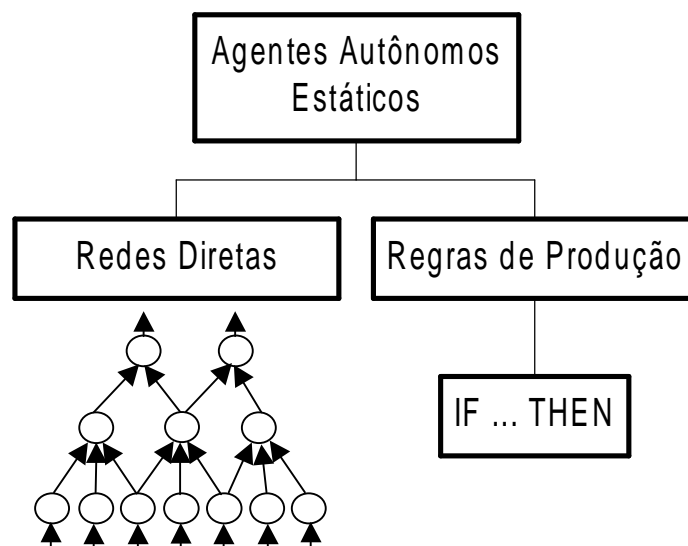


Figura 5 – Desenvolvimento de AAE.

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. Estas regras foram utilizadas em um trabalho anterior que implementava um sistema especialista para diagnóstico de falhas [16][17].

SE Δ pacotes de entrada > baseline	(a)
E pacotes de saída < baseline	(b)
E erros de saída > baseline	(c)
ENTÃO “pacotes errados”	(X)

SE Δ pacotes de entrada $>$ baseline	(a)
E pacotes de saída $<$ baseline	(b)
E erros de saída $>$ baseline	(c)
E fila de pacotes de saída $>$ baseline	(d)
ENTÃO "rajada de pacotes errados"	(Y)

Existem vários métodos para converter regras de produção em redes neurais diretas. Em [18], os autores apresentam uma taxonomia de arquiteturas híbridas e como converter regras em RNAs diretas. É necessário que as regras iniciais estejam no formato de Clausulas de Horn, porque deseja-se apenas uma saída, ou seja, uma conclusão. Segue abaixo a uma rede conexionista baseada nas regras acima apresentadas. A arquitetura da rede é caracterizada por alternar camadas conjuntivas (E) e disjuntivas (OU), porque a RNA direta equivalente às regras é construída utilizando grafos do tipo E/OU.

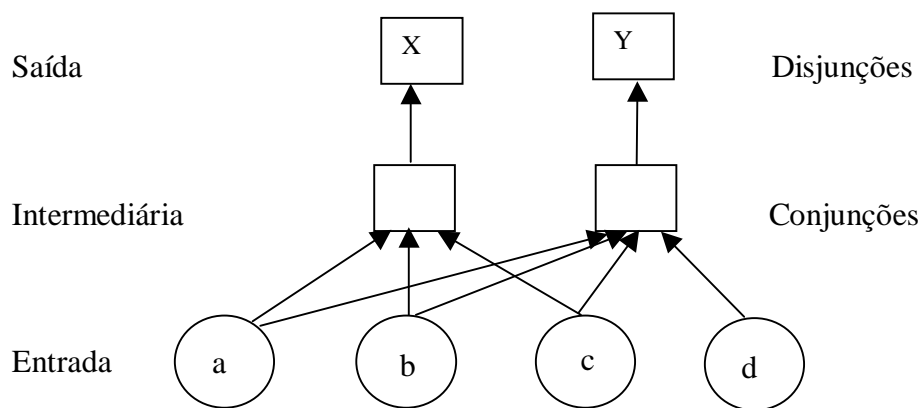


Figura 6 - RNA baseada em regras.

As Redes Neurais Diretas são estáticas, 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 da retropropagação ("backpropagation" em inglês) [1].

3.2 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 conhecida 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, deve-se:

- Inserir linhas de atrasos após a camada de entrada em redes neurais diretas;
- Utilizar 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 [19].

As Redes Neurais Recorrentes ou Recursivas são redes com ciclos e neurônios dinâmicos, esta dinâmica é provocada pelo retardo entre a saída do neurônio e sua entrada [1]. 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.

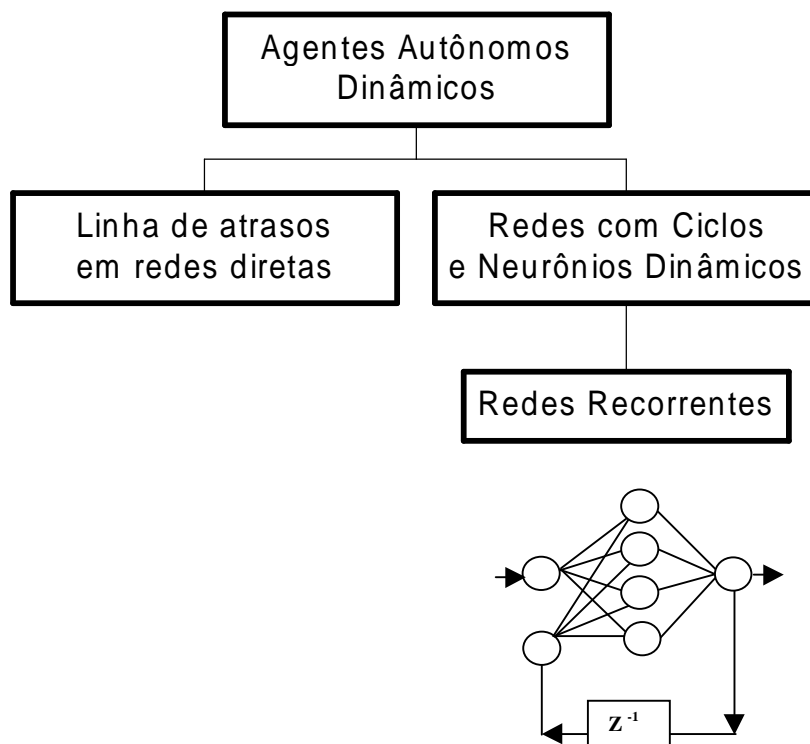


Figura 7– Desenvolvimento de AAD.

Este tipo de solução é adotada quando o problema é mal definido, mas é conhecido através de exemplos. É o mesmo caso de técnicas de raciocínio baseado em casos, só que neste caso existe o problema de encontrar os pesos das diferenças entre os casos. Por exemplo, tempo de resposta longo em uma rede de computadores pode significar vários tipos de falhas. Fazendo uma analogia a sistemas médicos, vários tipos de doenças tem como sintoma febre alta. Como diferenciar o peso de cada sintoma para cada doença.

3.3 Exemplo

Supondo um agente interpretador de eventos que poderia ser ativado por sinais de *polling* e alarme de eventos.

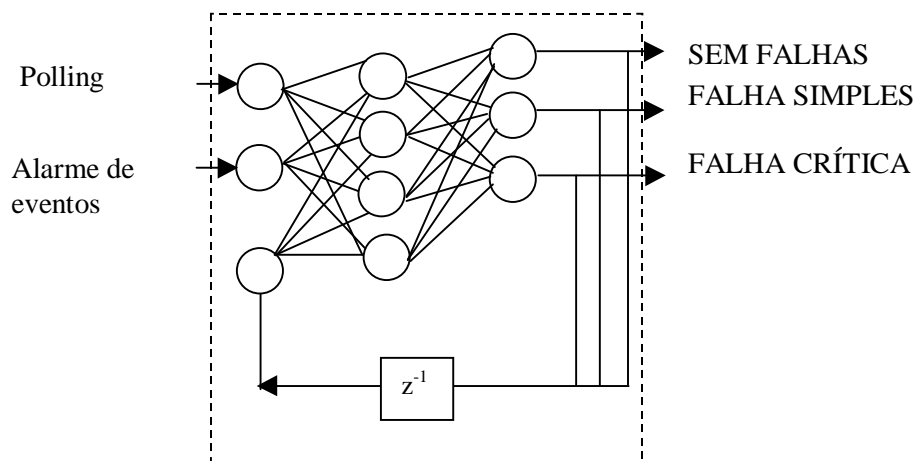


Figura 8 - Rede recorrente que implementa o agente interpretador de eventos.

Em primeiro lugar, para construir tal agente é necessário verificar as entradas e as saídas. Como entrada tem-se a sequência de *polling* e a sequência de alarme de eventos. Supondo os valores da tabela abaixo como ativação dos sensores para classificar se existe ou não falha na rede. Se pelo menos um sinal 1 aparecer na sequência é porque ocorreu um evento. Se nas duas entradas aparecer sinal 1 é porque existe uma evento grave.

Tabela 2 - Sinais de ativação do agente.

P	E	SF	FS	FC
0	0	1	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1

Após a definição das entradas e saídas utiliza-se um algoritmo para treinamento da rede. Neste trabalho está sendo aplicado o algoritmo proposto por Roisenberg [14] para treinamento de redes recorrentes. Considera-se o retardo da rede como uma terceira entrada e em seguida treina-se com o algoritmo de retropropagação.

É necessário esclarecer que deve existir uma hierarquia de agentes cada um com suas funções específicas. As informações de entrada deste agente podem ser recebidas de outros agentes considerados coletores de informações. Semelhante aos agentes de gerência de redes que não possuem autonomia, apenas coletam informações da base de dados.

Em [20], o autor apresenta agentes autônomos para gerência de redes utilizando regras de produção e uma pequena biblioteca para programação de uma rede neural direta. Sugere ainda, os seguintes agentes para gerência de redes:

- Agentes facilitadores de comunicação, responsáveis pela comunicação entre os agentes desenvolvidos;
- Agentes para análise de dados, como exemplo, o autor propõe um agente para analisar tráfego de pacotes IP em concentradores, verificando os pontos de acesso com maior volume de informações;
- Agentes para consolidar dados, agentes para agrupar os dados de um ou mais coletores de dados com o objetivo de estipular estatísticas sobre a rede;
- Agentes para prever dados futuros, com a característica das RNAs serem utilizadas em predição de séries temporais, o autor sugere um agente para previsão de falhas;

- Agentes para classificação de valores, poderia ser criado um agente para classificar o fluxo de informações em alto, médio ou baixo.

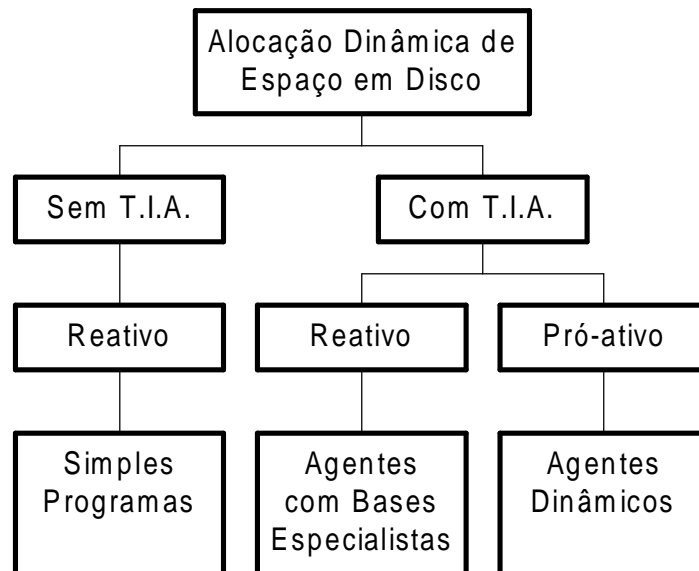


Figura 9 - Exemplo.

4. Sumário e Trabalhos Futuros

O presente trabalho apresentou uma metodologia para desenvolvimento de agentes de software autônomos para gerência de redes. Dois tipos de agentes foram propostos, dependendo do tipo de solução que o problema exige. Se o problema possui características estáticas existem heurísticas para solucioná-lo. As heurísticas podem ser extraídas do conhecimento de um especialista, neste caso, o administrador da rede. Caso contrário, utiliza-se exemplos da rede em questão para determinar padrões de entrada e saída. Estes padrões serão fornecidos a uma rede neural recorrente encarregada de estimar as trocas de estado para alcançar a solução do problema.

A implementação dos agentes está sendo desenvolvida em 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 ou de sistema operacional e processador [21]

Como trabalhos futuros pretende-se desenvolver classes de objetos para linguagem JAVA para auxiliar no desenvolvimento de agentes autônomos estáticos e dinâmicos para gerência de redes de computadores.

5. Referências

- [1] BARRETO, J.M. **Inteligência Artificial No limiar do Século XXI Abordagem Híbrida Simbólica, Conexionista e Evolucionária**. 2^a edição revista e aumentada. Florianópolis, 2000.
- [2] HAYES, C.C. “Agents in A Nutshell – A very Brief Introduction”. **IEEE Transactions on Knowledge and Data Engineering**, vol. 11, no. 1, Jan./Fev., 1999. Pp.127-132.
- [3] BRISA. **Gerenciamento de Redes – Uma abordagem de sistemas abertos**. Makron Books do Brasil ed. : São Paulo, 1993.
- [4] LESSER, V.R. “Cooperative Multiagent Systems: A Personal View of the State of the Art”. **IEEE Transactions on Knowledge and Data Engineering**, vol. 11, no. 1, Jan./Fev., 1999. Pp.133-142.
- [5] MINSKY, M.L., PAPERT, S. A. **Perceptrons: an introduction to computational geometry**, MIT Press, 1988
- [6] 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.
- [7] BOGONIKOLOS, N.; FRAGOUDIS, D.; LIKOTHANASSIS, S. “ARCHIMEDES: An Intelligent Agent for Adaptative – Personalized Navigation within a WEB Server”. In: **Proceedings of the 32nd Hawaii International Conference on System Sciences**, IEEE Press, 1999.
- [8] NISSEN, M.E. “An Intelligent Agent for Web-based Process Redesign”. In: **Proceedings of the 32nd Hawaii International Conference on System Sciences**, IEEE Press, 1999.
- [9] COVACI, S.; ZHANG T.; BUSSE, I. “Java-based Intelligent Mobile Agents for Open System Management”. In: **Proceedings of the 9th International Conference on Tools with Artificial Intelligence**, ICTAI’97, IEEE Press, 1997.

- [10] KURFESS, F.J.; SHAH, D.P. “Monitoring Distributed Process with Intelligent Agents”. In: **Proceedings of IEEE Conference and Workshop on Engineering of Computer-Based Systems**, ECBS’98, IEEE Press, 1998.
- [11] PINHEIRO, R.; POYLISHER, A.; CALDWELL, H. “Mobile Agents Aggregation of Network Management Data”. In: **Proceedings of First International Symposium on Agent Systems and Applications – Third International Symposium on Mobile Agents**, 1999.
- [12] RICH,E., KNIGHT, K. **Inteligência Artificial**. Tradução Maria Claudia S. R. Ratto. 2a edição, McGraw-Hill : São Paulo, 1993.
- [13] MOTOROLA CODEX. **The Basics Book of OSI and Network Management**. Motorola University Press, Addison-Wesley, 1993.
- [14] ROISENBERG, M. “Emergência da Inteligência em Agentes Autônomos através de Modelos Inspirados na Natureza”. **Tese de Doutorado** (PPGEEL, UFSC), Florianópolis, 1998.
- [15] DE FRANCESCHI, A.S.M. “Desenvolvimento de Agentes Autônomos para Gerência de Redes de Computadores”. **Exame de Qualificação do Curso de Doutorado** (PPGEEL, UFSC), Florianópolis, 2000.
- [16] DE FRANCESCHI, A.S.M., BARRETO, J.M. “Distributed Problem Solving Based on Recurrent Neural Networks Applied to Computer Network Management” In: **ICT’99 International Conference on Telecommunications**, IEEE Press, Cheju, Korea, 1999.
- [17] 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. In: **Proceedings of International Workshop on Application of Neural Networks in Telecommunications 3**. Lawrence Erlbaum Associates Publishers : Mahwah, New Jersey, 1997. pp.116-123.
- [18] DE AZEVEDO, F.M.; BRASIL, L.M., OLIVEIRA, R.C.L. de. **Redes Neurais com Aplicações em Controle e Sistemas Especialistas**. Florianópolis, SC : Visual Books Editora, 2000.
- [19] ROISENBERG, M., BARRETO, J.M., DE AZEVEDO, F.M. “Uma Proposta de Modelização para Agentes Autônomos Baseada na Teoria de Sistemas”. In: **Anais do 3º Simpósio Brasileiro de Automação Inteligente**, Vitória, ES, 1997. pp.500-507. AZEVEDO, F.M. de,

- [20] KOCH, F. “Agentes Autônomos para Gerenciamento de Redes de Computadores”. **Dissertação de Mestrado** (CPGCC, UFSC), Florianópolis, 1997.
- [21] LEMAY, L., PERKINS, C.L. **Teachyourself JAVA in 21 days**. SamsNet Publishing, 1996.