

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



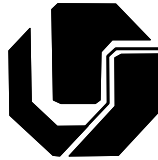
**Sistema Tutor Monitorado Por Rede
Neural Artificial**

Mauricio Corrêa de Almeida

UBERLÂNDIA - MG

Julho 2006

MAURICIO CORRÊA DE ALMEIDA



**Sistema Tutor Monitorado Por Rede Neural
Artificial**

Dissertação apresentada por Mauricio C. de Almeida à
Universidade Federal de Uberlândia para obtenção de título de
Mestre em Ciências pela Faculdade de Engenharia Elétrica.

Orientador: Prof. Ph.D. Keiji Yamanaka.

UBERLÂNDIA - MG

Julho 2006

FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UFU / Setor de Catalogação e Classificação

A447s Almeida, Mauricio Corrêa de.
Sistema tutor monitorado por rede neural artificial / Mauricio Corrêa
de Almeida. - Uberlândia, 2006.

97f. : il.

Orientador: Keiji Yamanaka.

Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.

Inclui bibliografia.

1. Inteligência artificial - Teses. I. Yamanaka, Keiji. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.3:007.52

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Sistema Tutor Monitorado Por Rede Neural Artificial

Dissertação apresentada por Mauricio C. de Almeida à Universidade Federal de Uberlândia
para obtenção de título de Mestre em Ciências pela Faculdade de Engenharia
Elétrica.

Banca Examinadora:

Prof. Ph.D. Keiji Yamanaka - Orientador
Prof. Dr. João Inácio da Silva Filho - UNISANTA
Profa. Dra. Elise Barbosa Mendes - FACHED/UFU
Prof. Dr. Luciano Vieira Lima - FEELT/UFU

Sistema Tutor Monitorado Por Rede Neural Artificial

Mauricio Corrêa de Almeida

Texto da dissertação apresentada à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Prof. Keiji Yamanaka, Ph.D.
Orientador

Prof. João Batista Vieira Junior, Dr.
Coordenador do Curso de Pós Graduação

Agradecimentos

Primeiramente agradeço a Deus por ter me iluminado durante a execução desta pesquisa.

Agradeço a Escola Técnica Estadual “Jorge Street”, sua Direção e aos meus alunos por terem me proporcionado a possibilidade de aplicar o sistema Tutor durante as aulas.

Minha gratidão aos professores da Universidade Federal de Uberlândia em especial ao meu orientador prof. Ph.D. Keiji Yamanaka pelo suporte, motivação e por me guiar no desenvolvimento da pesquisa.

Sou muito grato a Universidade Santa Cecília que firmou o Convênio com a UFU e proporcionou mais este grande passo na minha carreira pessoal e profissional.

Dedico este trabalho à minha esposa Tânia que com muita paciência entendeu mais este passo em minha vida e privou-se de muitas horas de lazer para estar ao meu lado.

Ao meu filho Thiago que também entendeu o significado desta pesquisa para o meu crescimento pessoal.

Aos meus Pais por sempre terem me apoiado nos momentos difíceis.

Resumo

Os sistemas de ensino a distância tem recebido cada vez mais apoio para pesquisa e desenvolvimento por proporcionarem vantagens tais como: reduzir os problemas causados pela distância entre o aluno e escola e flexibilizar os horários de estudo. Apesar de muitos sistemas de instrução assistida por computador serem bem sucedidos ainda encontramos oportunidades de melhoria no processo de monitoração.

O principal objetivo desta pesquisa é propor um sistema tutor inteligente baseado em rede neural artificial para reduzir o índice de alunos com baixo desempenho em linguagem de programação. O sistema monitora a navegação do aprendiz durante a etapa de estudo sugerindo sempre o melhor caminho a ser percorrido visando o melhor resultado. A rede neural foi treinada para monitorar uma determinada sequência de estudo e adaptativamente ajusta esta sequência a medida que novos caminhos de sucesso forem encontrados.

Este sistema foi aplicado a um grupo de aprendizes do ensino técnico do nível médio na disciplina de linguagem de programação permitindo uma melhora de 10% no índice de aprovação.

Abstract

Distance learning systems have been receiving support to research because they can provide flexibility for the student. There are many systems already developed but with great opportunity to improved the way of tracking the student.

The main purpose of this research is to propose an intelligent tutor based on artificial neural networks to reduce the number of students with low performance in computer language learning. The system monitor the student during his study stage and suggest the best path to be followed seeking for the best performance. The neural network will be trained to monitor a predetermined sequence of study but adaptively can adjust this sequence to the performance of the student. The system was applied to a group of students in a technical college and the learning performance increased in 10 % compared to another group without the assistance of the intelligent tutor.

Conteúdo

Capítulo 1. Introdução.....	1
1.1. Justificativa.....	1
1.2. Motivação	2
1.3. Objetivo Geral	2
1.4. Metodologia.....	3
 Capítulo 2. Estrutura da Dissertação.....	 6
 Capítulo 3. Perfil do Sistema Tutor.....	 7
3.1. Escopo e limitações do sistema monitorado por rede neural.....	16
 Capítulo 4. Fundamentos Teóricos de IA	 19
4.1. Trabalhando com aprendizado em IA	19
4.2. Modelos Conexionistas	22
4.3. Rede Neural de Retropropagação	24
4.4. Rede Multicamada.....	25
4.5. Ajuste dos pesos	25
4.6. Algoritmo de treinamento.....	27
 Capítulo 5. Aplicação da Rede Neural	 32
5.1. Atuação do sistema para montagem das entradas da RNA	32
5.2. Definição das entradas da RNA	33
5.3. Padrões de treinamento.....	34
5.4. Arquitetura da RNA para o sistema tutor	35
5.5. Software de treinamento da RNA.....	36
5.6. Software para o teste de resultados do treinamento da RNA	39
5.7. Etapas de treinamento da RNA:	40

Capítulo 6. Software do Sistema Tutor.....	42
6.1. Recursos utilizados no desenvolvimento do STI.....	42
6.2. Tecnologia aplicada no desenvolvimento do STI.....	48
6.3. Sistema de navegação do STI.....	53
6.4. Sistema de avaliação do STI.....	62
6.5. Sistema de gerenciamento do STI.....	64
6.6. Critérios para auto-aprendizado do STI	71
 Capítulo 7. Conclusão	 72
 ANEXO A1 – Padrões de treinamento.....	 76
 Referências Bibliográficas	 80
 Revisão Bibliográfica	 81
 Publicação	 83

Lista de Figuras

1 – Diagrama proposto para STI.	12
2 - Fluxo de rastreamento durante navegação do aprendiz.	16
3 -Ação da Rede Neural Artificial durante rastreamento.	18
4 – Diagrama da rede do tipo retropropagação.	28
5 - Hiperlink de navegação de estudos.	32
6 - Fluxograma da montagem das entradas da RNA.	32
7 – Entradas ativadas para o tópico 2.	33
8 – Tópico de estudo recuperado do banco de dados.	33
9 – Valores de entrada na rede neural.	33
10 – Arquitetura final da Rede Neural Artificial.	35
11 - Tela do software de treinamento da rede neural.	36
12 – Estrutura da tabela delta.	37
13 – Estrutura da tabela inout.	37
14 – Tela do programa para teste da RNA.	39
16 – Gráfico do Erro Total para 12 neurônios na camada	41
17 – Ambiente integrado de desenvolvimento JBuilder.	46
18 – Visualização dos recursos no ambiente integrado.	47
19 – Diagrama modelo adotado na aplicação STI	51
20 – Arquitetura básica JDBC.	52
21 – Diagrama de estados do STI para navegação.	55
22 – Interface de entrada do STI.	56
23 – Interface de carga de tópicos.	57
24 – Interface para os tópicos de estudo.	58

25 – Menu de seleção para alterar o status da monitoração do STI.....	59
26 – Mensagem de alerta para aprendiz monitorado pelo STI.	60
27 – Interface para consulta de desempenho.	61
28 - Página de avaliação do STI.....	62
29 - Página de administração do STI	64
30 - Página de administração do STI	65
31 – Interface para consulta aos padrões de treinamento da RNA.	65
32 – Lista de aprendizes não monitorados.	66
33 – Lista de caminho não monitorado por aprendiz.....	66
34 – Interface para validação de um novo padrão de treinamento.	67
35 – Chamada do aplicativo de treinamento da RNA através do STI.	67
36 – Tela para autorização de execução.....	68
37 – Programa para execução de treinamento da RNA.	68
38 – Lista de desempenho por grupo de aprendizes.	69
39 – Diagrama de estados do sistema de gerenciamento do STI.....	70

Lista de Tabelas

Tabela 1: Quadro resumo da metodologia da pesquisa.	5
Tabela 2 - Definição dos padrões de treinamento.....	34
Tabela 3 – Quadro resumo do critério de decisão	45
Tabela 4 – Tabela de resultados obtidos na pesquisa.....	74
A.1 - Padrões de treinamento capítulo 1, 3, 4, 5.....	76
A.1 - Padrões de treinamento capítulo 1, 3, 4, 5(cont.).	77
A.2 - Padrões de treinamento capítulo 2.....	78
A.2 - Padrões de treinamento capítulo 2(cont.).	79

Lista de Siglas e Abreviaturas

API	<i>Application Program Interface</i>
CD	<i>Compact Disc</i>
CGI	<i>Common Gateway Interface</i>
DBMS	<i>Database Management System</i>
EAD	<i>Ensino a Distância</i>
HTML	<i>Hypertext Markup Language</i>
IA	<i>Inteligência Artificial</i>
IAC	<i>Instrução Assistida por Computador</i>
JDK	<i>Java Development Kit</i>
JSP	<i>Java Server Page</i>
MVC	<i>Model View Controller</i>
RAM	<i>Random Access Memory</i>
RNA	<i>Rede Neural Artificial</i>
ODBC	<i>Open Database Connectivity</i>
STI	<i>Sistema Tutor Inteligente</i>
URL	<i>Universal Resource Locator</i>

Capítulo 1. Introdução

1.1. Justificativa

Os sistemas de ensino a distância tem recebido cada vez mais apoio para pesquisa e desenvolvimento por proporcionarem vantagens tais como: reduzir os problemas causados pela distância entre o aluno e escola e flexibilizar os horários de estudo.

No Brasil até 2005 já foram cadastrados no MEC pelo menos 30 cursos a distância em graduação e o dobro de pós-graduação. Já existem mais de 190 países que adotam esse tipo de prática e instituições como a Open University¹ do Reino Unido a qual tem hoje cadastrado 180.000 alunos em ensino a distância dentro e fora de seu país [1].

Os avanços tecnológicos nas áreas de rede, internet e linguagens, tais como Java, têm propiciado a abertura de caminho para desenvolvimento de aplicações na área de ensino a distância que nativamente necessitam de acesso por meios diversos e disseminação em larga escala.

O grande desafio tem sido transformar estas tecnologias em soluções ou até mesmo em um produto, considerando que um sistema de treinamento não pode ser simplesmente o transporte dos métodos tradicionais de ensino presencial da sala de aula para um meio eletrônico [2].

Apesar de muitos sistemas de instrução assistida por computador serem bem sucedidos nas empresas e nas instituições de ensino, ainda é possível encontrar falhas no processo de avaliação ou monitoração. Isto se deve ao fato de que nem todos os sistemas desenvolvidos levam em consideração o nível ou grau de conhecimento do aprendiz.

Muitos sistemas foram projetados, essencialmente, para tabular e mostrar o resultado de um determinado treinamento [3]. O fato de não possuírem enfoque na monitoração do aprendiz para identificação de suas falhas e problemas durante o aprendizado tem levado muitos aprendizes e instrutores à frustração. Muitos sistemas são questionados se os mesmos

¹ Open University é a única Universidade do Reino Unido dedicada a ensino a distância.
< <http://www.open.ac.uk/about/ou/> >

conduzem o aprendiz ao entendimento daquilo que o instrutor desejava. Isso normalmente provoca perda de tempo e recursos fazendo com que o aprendiz refaça todas as etapas do treinamento sem nenhum tipo de realimentação daquilo que não foi absorvido durante a primeira tentativa de aprendizado.

Conforme estudo apresentado por Silva, D. R.; Vieira, Marina T. P (2001) que traz a relação de 18 ambientes para realização de Educação a Distância (EAD) e suas ferramentas de avaliação e acompanhamento, foram identificados quatro mecanismos para suporte à realização do acompanhamento do aprendizado: rastreamento das ações do aluno, redirecionamento por avaliação, registros de mensagens de listas mensagens e chat ² [4].

1.2. Motivação

Ao analisar o tipo de rastreamento normalmente adotado por sistemas de EAD nota-se que estes não destacam a realimentação para o aprendiz feita por meio de um agente tutor artificial capaz de atuar em casos de erro ou desvio do objetivo. Cabe ao professor, então, analisar esses dados e tomar as atitudes que julgar necessário. Fica assim evidenciado que existe liberdade para pesquisa no sentido de rastreamento do aprendiz por um agente tutor artificial que estaria guiando o mesmo durante todo o aprendizado, assim como o instrutor o faz estando presente no momento da instrução.

Esta pesquisa tem destaque para aplicação de um agente tutor de inteligência artificial baseado nos conceitos de redes neurais artificiais (RNA) e sistemas tutores inteligentes (STI).

1.3. Objetivo Geral

O objetivo desta pesquisa é reduzir o índice de alunos com baixo rendimento no período de seis meses de avaliação, através da aplicação do treinamento por sistema assistido por computador. Sendo assim esta pesquisa se propõe a:

- a. Aplicar um Sistema Tutor capaz de guiar e monitorar o aprendiz durante o treinamento para obtenção do melhor caminho de estudo a ser percorrido e melhor rendimento possível no resultado da avaliação.
- b. Aplicar técnicas de Redes Neurais Artificiais para representar o domínio do instrutor dentro do sistema de rastreamento.
- c. Fornecer uma plataforma de navegação, através de hipertexto, capaz de monitorar os caminhos de estudo do aprendiz.

² Chat é um modo de troca de mensagens de maneira síncrona com uma ou mais pessoas, através de um programa de computador. < <http://dictionary.reference.com/search?q=chat>>

- d. Apresentar como produto um software desenvolvido em linguagem Java capaz de implementar a monitoração.

1.4. Metodologia

Pode-se definir pesquisa como o procedimento racional e sistemático que tem por objetivo proporcionar respostas aos problemas que são propostos. É requerida quando não se dispõe das informações suficientes para responder ao problema ou quando estas informações não estão organizadas de forma adequada (GIL, 1996 apud FONSECA, 2002) [5].

Este tópico tem por objetivo descrever os procedimentos e métodos científicos disponíveis para o desenvolvimento de uma pesquisa, bem como identificar o que melhor se enquadra às necessidades deste trabalho e permitir a reprodução deste trabalho por outros pesquisadores.

Abordagem da pesquisa

A escolha da abordagem é fundamental para nortear a pesquisa de forma válida e confiável e, para isto, deve estar atrelada aos objetivos centrais do estudo e à natureza do problema em questão.

Segundo Bryman (1989 apud FONSECA, 2002), duas são as abordagens para a pesquisa organizacional: abordagem quantitativa e abordagem qualitativa. A distinção entre a pesquisa quantitativa e a qualitativa não está na existência ou não de quantificação; a principal diferença da abordagem qualitativa em relação à quantitativa é a ênfase na perspectiva sobre o indivíduo estudado. A pesquisa quantitativa é impulsionada por considerações prévias da teoria e literatura, enquanto a pesquisa qualitativa evita a idéia de que o pesquisador é a fonte do que é relevante em relação ao objeto da pesquisa [5].

A abordagem qualitativa ressalta os seguintes pontos:

- Maior proximidade do pesquisador às circunstâncias na qual está envolvido (abordagem perceptiva).
- Menor rigidez na estruturação da pesquisa, proporcionando maior flexibilidade para redirecionamento (caráter exploratório).
- Seqüência dos eventos ao longo da pesquisa, diminuindo as dificuldades em reproduzir as etapas da pesquisa.
- Utilização de mais de uma fonte de dados dentro do universo da pesquisa.

A abordagem quantitativa salienta:

- Restrições iniciais sobre o alcance da pesquisa.

- A evidência de relações causais na formulação da hipótese.
- A operacionalização dos conceitos apresentados nas hipóteses.
- A inadequação de variáveis difíceis de quantificar.
- A replicação dos fenômenos estudados, de um estudo para outro.
- As conclusões que servem a generalizações, ultrapassando os limites restritos da pesquisa.

Os tipos de pergunta da pesquisa podem ser categorizados pela série: “quem”, “o que”, “onde”, “como” e “por que”. Segundo Yin (1994 apud FONSECA, 2002), os problemas de pesquisa do tipo “como” e “por que” estão relacionados ao caráter exploratório e mais próximo da abordagem qualitativa, enquanto questões do tipo “o que”, “quem” e “onde” implicam em tratamento quantitativo visando discriminar a incidência do fenômeno estudado [5].

Considerando os objetivos a serem atingidos a abordagem Quantitativa apresenta-se como a mais recomendada para esta pesquisa.

Métodos de pesquisa

Com base nos procedimentos técnicos utilizados para a coleta de dados, os métodos de pesquisas são: pesquisa bibliográfica, pesquisa documental, pesquisa experimental, pesquisa ex-post-facto, levantamento, estudo de caso, pesquisa-ação e pesquisa participante [5].

Estudo de caso

Estudo de caso pode ser caracterizado de acordo como um estudo de uma entidade bem definida como um programa, uma instituição, um sistema educativo, uma pessoa, ou uma unidade social. Visa conhecer em profundidade o seu “como” e os seus “porquês”, evidenciando a sua unidade e identidade própria. É uma investigação que se assume como particularística, isto é, que se debruça deliberadamente sobre uma situação específica que se supõe ser única em muitos aspectos, procurando descobrir o que há nela de mais essencial e característico. O fato de somente um objeto ser selecionado é possível obter a seu respeito uma grande quantidade de informações.

O pesquisador não pretende intervir sobre o objeto, mas revelá-lo tal como ele o percebe. O estudo de caso pode decorrer de acordo com uma perspectiva interpretativa, que procura compreender como é o mundo do ponto de vista dos participantes, ou uma perspectiva pragmática, que visa simplesmente apresentar um panorama global, tanto quanto possível completo e coerente do objeto de estudo do ponto de vista do investigador.

Resumo da metodologia de pesquisa

Analisando-se os conceitos e as considerações anteriormente descritas sobre a metodologia de pesquisa, mostra-se como mais adequado, que este trabalho deve receber uma abordagem quantitativa e estudo de caso é o método mais apropriado aos objetivos desta pesquisa.

Tabela 1: Quadro resumo da metodologia da pesquisa.

Metodologia	
Abordagem	Quantitativa
Método	Estudo de Caso

Este sistema será aplicado a uma turma de 40 alunos do primeiro ano do 2º Grau do ensino de nível técnico em escola pública na disciplina Introdução a Linguagem C++. O tempo estimado de treinamento por semestre é de 50 horas/aula, após este período serão obtidas as conclusões.

O critério para aprovação dos alunos nesta disciplina é formado por 30% exercícios em sala de aula + 20% avaliação do sistema tutor + 50 % prova prática individual.

O modelo de conceito aplicado é determinado por “I” para desempenho insuficiente, “B” para desempenho bom e “MB” para desempenho muito bom.

Para obtenção dos ganhos com a utilização STI o rendimento dos alunos deste semestre será comparado com a média de alunos que ficavam em processo de recuperação nas turmas anteriores onde não foi aplicado o sistema tutor e o número total de alunos aprovados.

Capítulo 2. Estrutura da Dissertação

No Capítulo 1 são apresentadas as justificativas que levaram ao desenvolvimento desta pesquisa, o objetivo a ser atingido, a metodologia de pesquisa utilizada neste trabalho e a proposta para obtenção dos resultados.

No Capítulo 2 é mostrada a estrutura desta dissertação.

No Capítulo 3 é mostrado um cenário dos sistemas tutores e as características principais destes sistemas.

No Capítulo 4 é feita toda a fundamentação do STI como um sistema tutor inteligente baseado nos conceitos de IA. É apresentado também o modelo de RNA chamado de retropropagação aplicado a esta pesquisa e seus fundamentos teóricos.

No Capítulo 5 está descrito a arquitetura, padrões de entrada, etapas de treinamento e o software desenvolvido para aplicação de RNA nesta pesquisa.

No Capítulo 6 estão as funcionalidades e a descrição do software do sistema tutor desenvolvido e aplicado nesta pesquisa assim como as ferramentas e as justificativas para que estas tenham sido utilizadas.

No Capítulo 7 são mostrados os resultados obtidos após aplicação do STI desta pesquisa e as conclusões deste trabalho.

Capítulo 3. Perfil do Sistema Tutor

Os Sistemas Tutores Inteligentes (STI) tem uma história interessante, originada com o movimento da Inteligência Artificial (IA) no final da década de 50 e início da década de 60, segundo Urban (1993), no início, pensavam os pesquisadores que para obter um sistema que raciocinasse como o ser humano era necessário somente ter computadores com maior capacidade de processamento [2].

Nos anos 60 pesquisadores criaram inúmeros sistemas chamados de Instrução Assistida por Computador (IAC), estes programas geravam um conjunto de problemas que visavam melhorar as habilidades do aprendiz, primariamente em aritmética e vocabulário.

A quase totalidade dos esforços nesta época estava dedicada a lutar com os desafios das técnicas de programação deste tipo de sistema nos custosos computadores de grande porte. Estes sistemas não eram focados explicitamente no aprender das pessoas, portanto era assumido que uma vez a informação apresentada, o aprendiz absorveria.

De acordo com McArthur; Lewis; Bishay (1993) estes sistemas tentavam implementar os métodos tradicionais de ensino exercitando e estimulando a prática até que certa habilidade tivesse sido conquistada, estes essencialmente foram projetados para apresentar o problema, receber, gravar, tabular e mostrar o desempenho do aprendiz [3].

O destaque na utilização desde método é o fato do mesmo ter muito bem definido seus objetivos de aprendizado. Os pesquisadores deste período tentaram mostrar que este método podia melhorar significativamente a velocidade e a qualidade do aprendizado e de certo modo foram bem aceitos e muitos tiveram sucesso.

Porém opiniões contrárias a este sucesso começam a aparecer porque uma vez que as novas tecnologias permitiam automatizar os processos de ensino e aprendizado as mesmas poderiam também permitir criar novos métodos e agregar valor aos objetivos educacionais.

Como resultado constatou-se que trabalhar com novas tecnologias, mas com métodos tradicionais de ensino, aprendizado e avaliação faziam cada vez menos sentido. Esta situação

mostrou-se como uma enorme dificuldade para o desenvolvimento de aplicações efetivamente com tecnologia aplicada a educação. Os objetivos e métodos de aprendizado tradicionais já eram profundamente conhecidos e estavam definidos, mas novos métodos tais como aprendizado por investigação, colaboração ou visualização ainda não eram bem compreendidos [3].

Ao final dos anos 60 e início dos anos 70 os pesquisadores decidiram ir além da apresentação de problemas e tabulação de resultados e passaram a considerar o aprendiz um fator do sistema de aprendizado (SUPPES, 1967 apud McArthur; Lewis; Bishay, 1993) [3].

Em 1982, o estado da Arte em Instrução Auxiliada por Computador foi revisado e batizado como um sistema chamado Sistema Tutor Inteligente (STI). Ficava implícito neste tipo de sistema que o aprendiz iria aprender executando ou fazendo. Estes sistemas baseados em computador deviam monitorar a solução de problemas, treinar, ser um instrutor e ser um conselheiro (SLEEMAN; BROWN, 1982 apud Urban, 1996) [2].

O destaque é que estes seriam ainda plataforma de pesquisa para o refinamento das teorias de inteligência artificial, mas agora os pesquisadores pensavam em representar os conhecimentos do aprendiz. Aqui nós encontramos pela primeira vez o termo “modelo do estudante” para descrever uma representação abstrata do aprendiz dentro do programa do computador. Ficava caracterizado por similaridade que se é possível transferir habilidades básicas de conhecimento do professor para o computador é provável que seja possível delegar certa instrução as máquinas e assim aperfeiçoar os resultados.

O STI busca capturar o método de ensino e aprendizado exemplificado como tutor interativo individualizado. Para os pesquisadores de IA da época este parecia ser o método mais natural por várias razões, tal como a versão de instruir, exercitar e praticar era amplamente aceita pela comunidade educadora e pela nossa cultura. Por isso o sistema individual de aprendizado foi considerado uma das melhores alternativas para o STI.

O STI tem em sua filosofia central um alto controle do sistema tutor e das etapas compostas por tarefa e resposta, sendo que o estudante aprende trabalhando com uma série de questões que são colocados pelo sistema. O sistema trabalha como um especialista em tarefas, controlando a seleção das atividades ou problemas, enquanto o aprendiz é responsável por executar as mesmas e responder as questões. O processo baseia-se em regras determinadas para ser um crítico no aprendizado, e decide pelo aprendiz quando este pode avançar ou retornar nas tarefas executadas.

O centro de um STI é ser um sistema especialista. O sistema especialista tem embutido em si conhecimento suficiente sobre uma área em particular que possibilita prover respostas

para as questões nele postadas, corrigir os resultados em todas as etapas e não só no final, trabalhando em pequenas parcelas do aprendizado. Sendo assim o sistema especialista permite ao STI demonstrar ou modelar o caminho correto de solução de problema de maneira similar a um tutor humano que pode gerar diferentes caminhos para a resposta dentro do objetivo a ser alcançado (MCARTHUR et al., 1988, p.5) [3].

O STI pode agir tão bem no treinamento quanto um modelo de um sistema especialista quando se trata de solução de problema. O STI pode monitorar o aprendiz enquanto ele resolve o problema e determinar se cada passo está correto. Para os sistemas desenvolvidos anteriormente ao STI as questões podiam ser consideradas como uma unidade básica do todo, mas em um STI a unidade básica é o passo de progresso individual do raciocínio. O STI monitora cada passo do raciocínio do aprendiz durante a solução do problema, sempre que o estudante comete um erro o STI diagnostica o problema e na medida do possível atualiza o modelo de aprendizado para aquele aprendiz, então tenta remediar a situação com instruções detalhadas fazendo com que o sistema especialista atue no próximo passo. Este processo deve se repetir a cada passo enquanto o aprendizado evolui até a solução completa do problema.

Baseado no que foi exposto é possível construir um STI onde diferentes versões de tutorial, para um mesmo assunto, podem ser implementadas incluindo diferentes níveis de complexidade dentro de uma hierarquia, onde o papel do aprendiz inicia em um nível básico e partir daí o STI guiará o estudo para cada nível dependendo do desempenho obtido na etapa anterior.

Este tipo de STI foi experimentado em um sistema tutor de álgebra que possui as características principais do sistema especialista [3]. Neste sistema foi permitido experimentar duas componentes pedagógicas diferentes, uma delas permitia alto controle do estudante, decidindo quando necessitava de ajuda do sistema, quando pedir ao sistema especialista por esclarecimento, ou quando as questões deviam ser difíceis ou fáceis. A outra componente pedagógica experimentada estava no outro extremo, sendo que o aprendizado era totalmente controlado pelo sistema tutor, decidindo que questões fazer ao aprendiz e quando ajudar o estudante (MCARTHUR et al., 1989, p.5) [3].

A primeira competência de um STI pode ser exemplificada quando ressaltamos que o sistema permite mostrar ao instrutor, onde é necessário dar mais atenção e onde estão as maiores dificuldades do aprendiz, aumentando assim a sua motivação. A segunda refere-se à importância do sistema que deve ser função daquilo que tenta ensinar e não do programa representado por ele, assim sendo qualquer sistema que seja encarado somente como um

programa terá poucos resultados (OHLSSON, 1991 apud McArthur; Lewis; Bishay, 1993) [3].

Um resumo das principais características para o provável sucesso de um STI, estão descritas a seguir:

Tabela 2 - Características resumidas do Sistema Tutor

Sistema Tutor	Característica
Micro Tutor	<ul style="list-style-type: none"> - É o princípio fundamental do STI, - Fornece alto nível de realimentação para o aprendiz, - Extremamente detalhista quando mostra a solução do problema, - Realimentação detalhada permite alto grau de reconhecimento da falha do aprendiz, - Devido ao alto grau de detalhamento pode provocar dificuldade ao aprendiz em reconhecer a solução do problema e reconhecer seus erros.
Tutor controlando o aprendizado	<ul style="list-style-type: none"> - Tutor altamente competente para tomar a decisão de quando o aprendiz necessita ou não de maiores informações e experiências dentro do aprendizado, superando as vontades do próprio aprendiz. - Este sistema assume que conhece exatamente o que o aprendiz deseja estudar e conhece o seu estágio de conhecimento sobre o assunto.
Treinamento guiado por geração de impasse	<ul style="list-style-type: none"> - Estimula a ação do estudante criando dificuldade, impasses e solução para erros, - Organiza o aprendizado por tarefas com pequenas sessões de questionamentos, - O sistema deve reconhecer e superar o impasse para contribuir no aprendizado, - O sistema deve sempre prover realimentação imediata ao impasse.

Todas as variações do STI mostraram grande contribuição para o aprendizado dentro da sala de aula porque dão retorno imediato às dificuldades do aprendizado, geralmente

apontam para os objetivos e resultados desejados no currículo escolar tradicional e também se adaptam ao método tradicional de instrução e aprendizagem [3].

A maioria dos métodos de sala de aula combina lições, prática e exercícios, como consequência o instrutor não tem muitas dificuldades em adaptá-lo a sala de aula e ao currículo existente.

A limitação neste tipo de sistema é basicamente poder ensinar por apenas um método de ensino, ou seja, estes não tem domínio sobre diferentes métodos, enquanto um sistema especialista completo no futuro deve ser capaz de adotar vários métodos durante o aprendizado. O sistema não deve ficar limitado à prática e execução de exercício como modo de interação com o aprendiz e sim permitir durante a instrução que o aprendiz possa migrar para outros modos de aprendizado. O ideal seria um sistema capaz de adotar vários métodos pedagógicos no momento apropriado e permitir ao aprendiz diferentes estilos de aprendizado, contrastando com o método competente de ensino individualizado e variando de acordo com as necessidades do aprendiz entre o método de prática execução para métodos baseados no raciocínio.

Uma solução para este problema pode ser a expansão ou melhoria dos métodos de avaliação ou dos objetivos a serem atingidos, aperfeiçoando a qualidade do ensino e da disciplina dentro do que já existe nos sistemas atuais, ou mesmo expandindo as matérias que um STI pode ensinar.

O STI deve ser capaz, mesmo que sutilmente, de raciocinar com o aprendiz quando o mesmo estiver em um momento de impasse no aprendizado, variando as respostas para o problema até garantir o sucesso do aprendiz. Este tipo de sistema deve levar em conta vários aspectos para realimentar corretamente o aprendiz, entre eles estão: considerar o histórico de acertos e erros passados do aprendiz, incluir regras no sistema que permitam o planejamento da matéria de estudo, em tempo real, para cada aprendiz e não somente remediar o resultado sem individualização.

Wenger (1987 apud URBAN, 1996, p.7) analisou os objetivos implícitos e explícitos do projeto de STI e este o considerou como um projeto de comunicação do conhecimento focado na cognição e aprendizado somado aos aspectos da inteligência artificial. Wenger (1987 apud URBAN, 1996, p.7) propôs o que pode ser a base para uma disciplina que combina o trabalho de pesquisa da inteligência artificial, ciência cognitiva e educação.

“... considere o exemplo de um livro: eles certamente ultrapassaram o limite das pessoas na precisão e permanência em suas memórias e excederam sua credibilidade. Por esta razão, eles são inestimáveis para humanidade. Agora imagine livros ativos que podem

interagir com o leitor para comunicar conhecimento em um nível apropriado, seletivamente ressaltando a inter-conectividade e ramificações dos itens, recordando as informações relevantes, provando o entendimento, esclarecendo as áreas com dificuldades mais detalhadamente, saltando sobre o material aparentemente conhecido... sistemas de comunicação inteligente do conhecimento são realmente um sonho atrativo” [2].

Wenger (1987 apud URBAN, 1996) invocou para um movimento em direção a “forma orientada cognitivamente” em engenharia de software, no qual a cognição é central, ao invés de focar os modelos computacionais e pedagogia. Ele mostrou que até a metade dos anos 80 existiam dois pontos de vista opostos em relação ao STI: o ponto de vista tradicional onde o computador era um entregador de instrução e o sob o ponto de vista emergente, onde o computador atua como uma ferramenta exploratória do aprendizado [2].

Ele reivindicou que o STI fosse olhado com a visão de “comunicação de conhecimento”, então seria possível combinar os dois pontos de vista, aparentemente opostos.

Wenger (1987 apud URBAN, 1996) apresentou o seguinte modelo de STI que inclui um modelo chamado de comunicação com o aprendiz [2]:

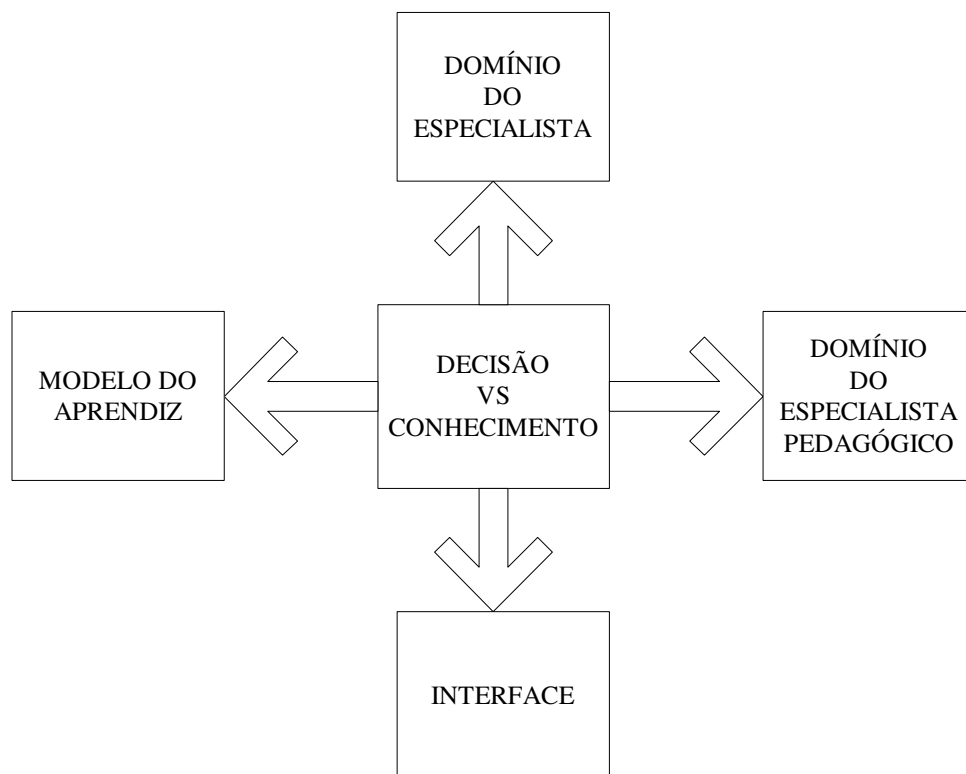


Figura 1 – Diagrama proposto para STI.

Este tipo de modelo é consistente com os trabalhos de STI que vinham sendo feitos até a época, Wenger (1987 apud URBAN, 1996) tentou abstrair os componentes por trás das definições mais tradicionais da engenharia de software [2].

Domínio do especialista: nos sistemas tradicionais de IAC (Instrução Assistida por Computador), o conhecimento do especialista está contido em blocos ou pedaços que podem ser chamados de molduras. O sistema apresenta blocos ao aprendiz, com a ordem de apresentação baseada na resposta do estudante no bloco anterior. Se o aprendiz responder corretamente um conjunto de questões, o sistema apresenta o próximo bloco na sequência. Se o aprendiz responder incorretamente, blocos alternativos são apresentados. Contudo, estes blocos são estáticos, eles não são capazes de “aplicar” conhecimento como um especialista humano seria. A característica chave que distingue o sistema de comunicação de conhecimento do STI padrão no domínio do especialista é representar o objeto problema não meramente como um conjunto de blocos estáticos, mas na verdade como um modelo dinâmico no domínio do conhecimento e com um conjunto de regras na qual o sistema pode “raciocinar”. Estes sistemas têm suas raízes em pesquisa em sistemas especialistas (tais como sistemas de diagnóstico médico ou sistemas eletrônicos de diagnóstico) que tem a habilidade de gerar múltiplos blocos de solução, além de uma simples solução idealista e específica.

Modelo do Aprendiz: o modelo do aprendiz permanece como a centro das pesquisas do STI (HOLT et al., 1994 apud URBAN, 1996)[2]. O que realmente distingue o STI do IAC é a capacidade de responder aos estilos de aprendizado individuais do estudante e fornecer instruções personalizadas para cada um. Apesar de alguns autores questionarem o objetivo do modelo do aprendiz, devido às limitações técnicas (MCCALLA, 1992 apud URBAN, 1996, p.8) ou pela fundamentação filosófica (SACK et al. 1994 apud URBAN, 1996, p.8), essa é uma área ainda em atividade de pesquisas [2].

Mitchel (1993 apud URBAN, 1996, p.8) argumenta que um modelo de STI deve representar o mundo, o aprendiz e a interação do aprendiz-instrutor (GREER; MCCALLA, 1994 apud URBAN, 1996) [2].

De acordo com Wenger (1987 apud URBAN, 1996), o modelo do aprendiz tem três tarefas. (1) Deve reunir dados do aprendiz e sobre o aprendiz. Estes dados podem ser explícitos, solicitando ao aprendiz para resolver problemas específicos, ou implícitos, seguindo o caminho de navegação do aprendiz ou comparando as mesmas respostas a de um aprendiz similar. (2) Esses dados devem ser usados para criar a representação do conhecimento do estudante em relação às divergências do conhecimento de um especialista.

O sistema usa então este modelo para prever qual a resposta do aprendiz na situação subsequente, compara com o previsto com a resposta atual do aprendiz e utiliza a informação para refinar o modelo. (3) O modelo do aprendiz deve coletar dados para a base executando algum tipo de diagnóstico, considerando o estado do conhecimento do aprendiz e em termos de seleção da melhor estratégia pedagógica para apresentar subsequente informação para o estudante. Um dos maiores desafios é saber considerar dados espúrios, ou seja o fato de que o aprendiz nem sempre tem uma resposta consistente e constante, particularmente quando seu conhecimento é frágil e ele não está certo sobre a resposta.

Domínio Pedagógico: o STI deve moldar o conhecimento atual do estudante e apoiar a transição para um novo estado de conhecimento. Wenger (1987 apud URBAN, 1996, p.8) exige que este requisito seja implícito no STI e que o sistema possa alternar entre diagnóstico e suporte didático [2]. Dentro do domínio pedagógico existe o diagnóstico o que significa que o STI deduz informação sobre o estado aprendiz em três níveis. Em um nível focando o comportamento, em outro tratando do nível de conhecimento e inferindo ações baseado na observação e finalmente no nível individual, cobrindo áreas como personalidade, motivação e auto avaliação. Wenger (1987 apud URBAN, 1996, p.9) observou que os sistemas tutores não têm levado muito em consideração o nível individual, sendo que é necessário considerar o aprendiz mais que um elemento passivo no aprendizado [2].

Reunir informações de diagnóstico apresenta um desafio para os desenvolvedores de STI. O único dado prontamente disponível é o resultado das tarefas individuais ou problemas apresentados e alguns dados provenientes da interface que podem indicar a quantidade de uso do teclado ou os movimentos do mouse. Existem dois modos de usar esta informação.

O primeiro é o modo de rastreamento que tenta mapear os dados externos coletados pelo sistema para uma representação do modelo interno dentro do espaço do problema. Este modo foi desenvolvido em sistemas IAC e tentou classificar o estado de conhecimento em um ou mais estados e os desenvolvedores de sistema tinham que antecipar o próximo quadro ao aprendiz.

O segundo é a interpretação reconstrutiva que é baseada no contexto de cumprimento de objetivos ou planos através da solução de problemas. Ao invés de classificar o dado dentro de um pequeno conjunto de modelos, a interpretação reconstrutiva inicia com um conjunto de planos que são decompostos em metas e submetas. Assim que os dados adicionais são coletados, o STI pode podar a lista para reduzir o espaço de busca e atribuir probabilidades para vários nós no espaço de busca. O STI ajusta estas probabilidades dinamicamente,

permitindo ao STI responder de maneira mais inteligente ao engano ou respostas ruidosas que o aprendiz freqüentemente se expõe quando está aprendendo um novo material.

Didática: A segunda faceta do especialista pedagógico foi identificada por Wenger (1987 apud URBAN, 1996, p.9) como sendo o suporte didático, o aspecto de suporte na instrução. Genericamente o STI esteve concentrado no modelo de manipulação do conteúdo, com pequena atenção dedicada a didática. Wenger (1987 apud URBAN, 1996, p.9) sugeriu que a didática pode ser organizada em torno de quatro princípios: (1) o plano de ação denominado como currículo mínimo usado para conduzir o aprendiz e prover o contexto para diagnóstico do sistema; (2) contexto estratégico que é o plano de ação a ser implementado com os objetivos do instrutor onde o aprendiz solicita por auxílio no aprendizado; (3) a base de decisão onde são atribuídas as regras ou diretrizes para determinar as limitações do sistema; (4) e o nível alvo para o modelo do estudante que é a seleção do nível limite onde instrutor passa atuar. O especialista humano deve entender, agir e responder apropriadamente a este nível caso o STI não seja capaz de dar respostas. Raramente os sistemas tutores desenvolvidos tinham determinado em qual nível ou momento o estudante necessita de orientação do instrutor para ajustar seu plano de ação [2].

Interface: A interface permite a comunicação entre o estudante e o STI. Aqui os guias com recomendações para criação de uma boa interface homem máquina e desenvolvimento de software devem ser aplicados, levando em conta as implicações dos aspectos pedagógicos na interface. O meio mais comum utilizado é a interface através de ligações de hipertexto, esta deve prover uma maneira em que o aprendiz seja capaz de ser guiado sem perder aspectos importantes da instrução.

Wenger (1987 apud URBAN, 1996, p.10) sugeriu que um sistema de comunicação de conhecimento requer uma interface que contenha um modelo de diálogo, que não provoque dúvidas nas respostas do estudante e na sua navegação pelo sistema. Uma vez que o aprendiz tende a responder as perguntas de maneira incorreta ou incompleta, principalmente quando se encontra em dificuldades, é importante prover na interface um suporte apropriado a estas repostas e que permitam o diagnóstico do sistema, auxiliando o avanço para etapas posteriores. Isto produz um ganho significativo para o STI pois ajuda a evitar a redundância e melhora a instrução.

A interface não deve simplesmente colocar o conhecimento disponível ao aprendiz mas deve permitir que durante toda a navegação seja possível a realimentação do aprendizado não devendo atuar somente como um sistema explanatório, caso típico de navegação pela internet, o que não contribui para os objetivos de um STI. O sistema exploratório coloca toda

a responsabilidade de acertar o caminho sob o aprendiz. Mesmo em sistemas explanatórios o STI deve tomar conta atuando como guia ou orientador evitando que o aprendiz perca o caminho a seguir ou aspectos importantes [2].

3.1. Escopo e limitações do sistema monitorado por rede neural

Neste sistema o agente tutor que faz a simulação do instrutor tem a função de monitorar e rastrear os passos de navegação do aprendiz durante todo o processo de aprendizado.

O tutor artificial tem a responsabilidade de sugerir um caminho alternativo, objetivando assim sempre o melhor desempenho. Este sistema avalia o desempenho baseado nos desejos do instrutor e no resultado da avaliação indicando consequentemente a próxima etapa a ser seguida no treinamento. Caso o aprendiz deseje seguir um caminho alternativo, ou seja, aquele caminho que inicialmente não havia sido proposto pelo instrutor, o tutor artificial alertará e permitirá o prosseguimento do estudo rastreando este novo caminho proposto.

Obtendo sucesso na avaliação final o tutor será novamente treinado através da rede neural artificial e adicionará este novo caminho de estudo como um caso de sucesso.

A partir deste momento qualquer novo aprendiz que desejar seguir um caminho alternativo do qual outro aprendiz já tenha tido sucesso, o tutor já re-treinado pela rede neural também levará em consideração e permitirá que este caminho seja percorrido.

Caso o caminho alternativo que foi percorrido leve o aprendiz a um baixo desempenho na avaliação final o tutor somente registrará este caminho rastreado como um caso de insucesso e a rede neural não será re-treinada e continuará alertando os próximos aprendizes para que não percorram este caminho.

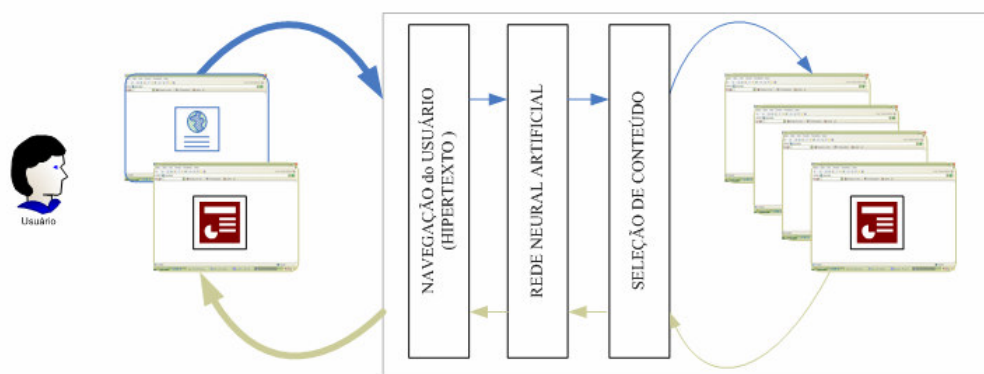


Figura 2 - Fluxo de rastreamento durante navegação do aprendiz.

O instrutor que inicialmente criou os caminhos de navegação para estudo, sempre terá domínio sobre a rede neural artificial para permitir ou não que um novo caminho possa ser considerado como sucesso para um re-treinamento da rede neural artificial. O sistema nesta pesquisa se propõe a guiar, monitorar e ser um crítico durante a instrução, apresentando sempre que ocorrer um desvio dos objetivos uma realimentação para o aprendiz. Para tanto o sistema terá 100% do tempo de rastreamento e redirecionamento.

Neste sistema tutor não serão adotados diferentes métodos ou modelos pedagógicos de ensino, adaptados ao sistema de aprendizado por computador, mas sim será mostrado que a rede neural artificial pode trazer mudanças, provocar ganhos, ser treinada com os conhecimentos do instrutor e trabalhar como um sistema especialista.

Esse projeto propõe apresentar a rede neural como um modo auxiliar na instrução, procurando obter os melhores resultados para o aprendiz. Esta pesquisa conta também com a presença do instrutor durante o aprendizado e a aplicação do sistema tutor.

Para confirmação dos resultados foi desenvolvido um treinamento com os requisitos mínimos necessários para que o aprendiz possa executá-lo e depois ser avaliado pelo próprio sistema.

O software desenvolvido para execução do treinamento nesta pesquisa executa em um servidor e o acesso ao treinamento é feito através dos computadores ligados em rede local onde o ambiente de navegação é por hipertexto e os testes finais de avaliação de múltipla escolha dentro do mesmo ambiente.

O sistema tutor faz o rastreamento e realimentação do aprendiz através da monitoração de cada “hiperlink”³ de texto ativado, que representa o tópico do capítulo que está sendo estudado. No caso da avaliação feita no final de cada capítulo o sistema considera as alternativas assinaladas como correta pelo aprendiz.

Uma vez tendo sido o sistema tutor treinado com os critérios determinados inicialmente pelo instrutor e o aprendiz realizado a avaliação, o resultado do desempenho pode ser medido e registrado para consulta posterior.

Naqueles casos onde o resultado tenha sido insatisfatório o próprio sistema tutor encaminha o aprendiz para repetição do capítulo e ao contrário se for satisfatório o aprendiz prossegue para o próximo capítulo.

³ Hiperlink é apresentação de informações escritas, organizada de tal maneira que o leitor tem liberdade de escolher vários caminhos, a partir de seqüências associativas possíveis entre blocos vinculados por remissões, sem estar preso a um encadeamento linear único.

Para execução do rastreamento das ações do aprendiz através da ativação do “hiperlink” de texto é utilizada a tecnologia Java⁴ e JSP⁵ (Java Server Pages) que permitem transmitir ao computador do cliente a página para navegação no decorrer do treinamento e de maneira transparente ao usuário executa do lado do servidor a ativação da rede neural artificial para monitoração.

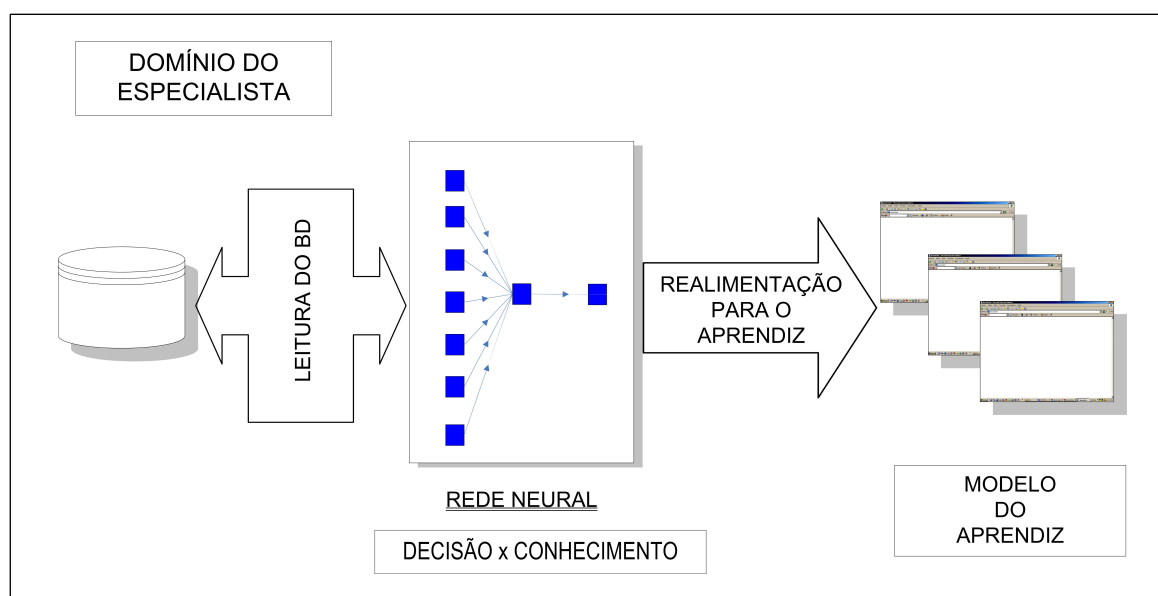


Figura 3 -Ação da Rede Neural Artificial durante rastreamento.

Todo “hiperlink” de texto ativado é armazenado pelo sistema no banco de dados. Ao mesmo tempo o sistema entrega os dados para a rede neural que avalia se o caminho pode ser percorrido.

Um formulário que apresenta questões de múltipla escolha usando tecnologia JSP permite ao aprendiz efetuar a avaliação ao final de cada capítulo. Os resultados são armazenados em um banco de dados e o sistema tutor recupera estes valores para verificar o desempenho do aprendiz, mostrando ao mesmo o resultado alcançado e as questões que obteve sucesso na resposta.

O sistema tutor só permite que um novo capítulo seja estudado se o aprendiz obteve o rendimento mínimo atribuído ao sistema pelo instrutor.

Está proposto que cada capítulo do treinamento seja composto em sua avaliação final por dez questões com alternativas de múltipla escolha.

⁴ Java é uma linguagem de programação desenvolvida pela Sun Microsystems para suportar larga escala de distribuição em particular para aplicações da Internet.< <http://dictionary.reference.com/search?q=Java>>.

⁵ A tecnologia Java Server Page (JSP) provê uma maneira simplificada de criar conteúdo dinâmico para Internet.< <http://java.sun.com/products/jsp/>

Capítulo 4. Fundamentos Teóricos de IA

4.1. Trabalhando com aprendizado em IA

Segundo Rich e Knight (1993, p.513) a crítica mais forte á IA é que as máquinas só podem ser consideradas inteligentes quando forem capazes de aprender coisas novas e se adaptarem a novas situações, em vez de simplesmente seguir o que foi comandado.

Assim fica claro que uma importante característica das entidades inteligentes é a capacidade de adaptação a novos ambientes e de solução de novos problemas.

Ada Augusta, uma das primeiras filósofas da computação , escreveu :

“A máquina analítica não tem qualquer pretensão de originar nada. Ela pode fazer qualquer coisa desde que nós saibamos como mandá-la executar”. (LOVELACE, 1961 apud RICH; KNIGHT, 1993) [6].

Vários críticos de IA interpretaram este comentário como uma indicação de que os computadores não seriam capazes de aprender. Na verdade não é bem isto, porque nada impede que digamos a um computador como interpretar as informações recebidas de uma maneira que melhore gradualmente seu desempenho.

Ao invés de antecipar a pergunta se é possível ou não que os computadores aprendam, seria mais esclarecedor descrever exatamente a que atividades nos referimos quando falamos em aprender e que mecanismos podem ser usados para executar essa atividade.

A proposta para aprendizagem feita por Simom (1983 apud RICH; KNIGHT, 1993) denota:

“mudanças adaptáveis no sistema, no sentido de que permitem que o sistema, da próxima vez, faça a mesma tarefa ou tarefas tiradas do mesmo grupo com mais eficiência ou eficácia “[6].

Assim definida a aprendizagem esta passa a abranger uma ampla escala de condições onde é possível considerar que os computadores são capazes de aprender.

Em uma destas condições, está o refinamento de habilidades, modo mais próximo das pessoas, que podem refinar e melhorar suas habilidades através da prática. Em outra condição está a aquisição de conhecimento, sendo esta a fonte de força para os programas de IA.

A aquisição de conhecimento inclui atividades diferentes, entre elas está o armazenamento de informação ou aprendizagem por memorização, que pode ser considerada a mais básica entre outras. Muitos programas de IA são capazes de melhorar seu desempenho por meio de técnicas de aprendizagem por memorização.

Aprendemos também através do conselho de outras pessoas, o aconselhamento é semelhante a aprendizagem por memorização mas dependendo do tipo de conselho pode não ser possível utilizá-lo diretamente no programa. Os conselhos são úteis desde que seja possível interpretá-los, portanto deve haver alguma maneira do programa transformar o conselho ou instruções em um resultado final que possa ser operacionalizado.

Um outro modo de aprender utilizado pelas pessoas é através da própria experiência na solução dos problemas. Após a solução de um problema complexo, a estrutura e os métodos utilizados para resolução, são armazenados e então em outra ocasião é possível recorrer a esta solução. Isto torna as pessoas mais eficientes em sanar o mesmo tipo de problema.

Somos capazes também de generalizar a experiência e capazes de solucionar problemas relacionados.

Em contraste com o aconselhamento, a aprendizagem a partir de experiência na solução de problemas, em geral não envolve a coleta de novos conhecimentos anteriormente não disponíveis, ou seja, o programa é capaz de lembrar e generalizar sua experiência mas não acrescenta nenhuma dedução a seu conhecimento assim como faria um programa de aconselhamento que recebe estímulos do mundo externo.

Em termos práticos este processo de aprender pode significar a diferença entre a solução do problema de maneira rápida ou a não solução. Outro efeito que deve ser observado no processo de aprendizado é a redução de desempenho do programa ao tentar executar a tarefa de aprender e resolver o problema ao mesmo tempo.

Os programas que aprendem através da experiência na solução de problemas podem conseguir soluções qualitativamente melhores ao longo do tempo.

Uma outra forma de aprendizagem que envolve estímulos externos é a aprendizagem a partir de exemplos. Geralmente os seres humanos aprendem a classificar as coisas do mundo sem regras explícitas. A aprendizagem a partir de exemplos em geral envolve um instrutor

que nos auxilia na tarefa de classificação, aplicando uma correção quando algum erro é cometido.

Na verdade não há uma formulação precisa sobre aprendizagem o importante é manter em mente que a aprendizagem é em si mesma um processo de solução de problemas.

4.2. Modelos Conexionistas

Na busca de construir máquinas inteligentes, o modelo que surge naturalmente é o cérebro humano. Uma idéia óbvia para IA, então, é simular o funcionamento do cérebro humano diretamente em um computador. A idéia de criar uma máquina inteligente com neurônios artificiais iniciou com McCulloch e Pitts (1943), e outros pesquisadores que seguiram esta linha nas duas décadas seguintes como Ashly (1952), MinsKy (1954), Block e Rosenblatt (1962) [6]. Estas pesquisas foram virtualmente interrompidas na década de 70, quando as redes em estudo mostraram ineficiência em termos computacionais.

Prematuramente casos de sucesso foram obscurecidos pelo rápido progresso da computação digital e também aos exageros atribuídos a capacidade dos primeiros modelos de redes neurais, lançando dúvidas neste campo de pesquisa.

Recentemente ressurgiu o interesse nas redes neurais por razões relacionadas ao aumento de capacidade de processamento dos computadores, tornado possível a simulação de redes maiores, computadores com capacidade de processamento paralelo e o mais importante a descoberta de novas arquiteturas de redes neurais e poderosos algoritmos de aprendizagem.

Estas novas arquiteturas foram intituladas de conexionistas, porque receberam inspiração de fatos conhecidos sob o funcionamento do cérebro humano [6].

Redes neurais artificiais foram desenvolvidas como generalização de modelos matemáticos da cognição humana ou biologia neural [7].

As características principais são: (1) um grande número de elementos de processamento muito simples, parecido com o neurônio; (2) um grande número de conexões ponderadas entre os elementos sendo que os pesos das conexões codificam o conhecimento da rede; (3) controle distribuído e paralelo; e por último (4) ênfase na aprendizagem automática de representações internas.

Os pesquisadores conexionistas conjecturam que pensar sobre computação em termos da “metáfora do cérebro” em vez da “metáfora do computador digital” conduzirá a incursões sobre a natureza do comportamento inteligente.

Apesar das vantagens dos computadores se comparados ao ser humano em relação a capacidade de armazenamento, velocidade de processamento e capacidade para execução de cálculos aritméticos, os mesmos não são capazes de executar tarefas como andar, falar, etc.

Mesmo assim quando comparado o desempenho do computador com o desempenho do ser humano para certas atividades, o desempenho do ser humano será muito inferior. Isto indica que talvez a estrutura do cérebro humano não seja a mais apropriada para certas tarefas.

Mas os neurônios também oferecem suas limitações por serem mais lentos que os circuitos integrados dos computadores, estarem sujeitos a falhas e terem sua mortalidade natural dentro do cérebro. A grande vantagem é que o cérebro opera em processamento paralelo e isso torna o ser humano mais rápido e capaz de executar tarefas extremamente complexas como a interpretação de uma cena visual ou compreensão de uma frase em apenas décimo de segundo. De acordo com Rich e Knight (1993 apud FELDMAN e BALLARD ,1985, p.516) isto sugere buscar soluções em algoritmos paralelos que exijam não mais do que 100 etapas de processamento para encontrarmos a solução [6].

A idéia por trás das conexões é que podemos ter avanços significativos em IA quando abordamos os problemas do ponto de vista de computação do cérebro.

4.3. Rede Neural de Retropropagação

As duas razões principais para a interrupção das pesquisas nos anos 70 foram: a falha do perceptron de uma camada não ser capaz de resolver o problema da função XOR e a falta de métodos de treinamento para rede multicamada. Um método de propagar a informação do erro nas unidades de saída de volta para a camada intermediária foi descoberto na década de setenta por Werbos (1974), mas não teve muito reconhecimento [7].

O trabalho de David Parker (1985) foi reconhecido quando o grupo de processamento paralelo distribuído, liderado pelo psicólogo David Rumelhart (1986) da universidade da Califórnia em San Diego e James McClelland (1988) da Universidade Carnegie-Mello publicaram o assunto [7].

O sistema de representação do conhecimento empregado pelas redes neurais é um tanto obscuro porque as redes devem aprender suas próprias representações, sendo praticamente impossível programá-las manualmente.

Uma rede neural pode ser caracterizada pelo padrão de conexão entre os neurônios, chamada arquitetura, seu método de determinação do peso nas conexões, chamado de algoritmo de treinamento ou aprendizado e sua função de ativação. No caso da rede de retropropagação é necessário uma camada intermediária que conecta a camada de entrada e as saídas. Cada unidade da rede neural requer uma função de ativação para poder produzir valores entre -1 e 1 esta função deve ser contínua e diferenciável e normalmente é utilizado a função sigmóide (em forma de S).

No início do treinamento a rede tem um conjunto de pesos aleatórios atribuído a seus neurônios. A rede passa a ajustar estes pesos durante a etapa de treinamento, toda vez que recebe um par de entradas e saídas. Cada par requer dois estágios: uma passagem para frente e seu respectivo retorno para trás. A passagem para frente envolve apresentar um exemplo na entrada da rede e deixar as ativações dos neurônios transcorrerem até chegarem à camada de saída. Durante o retorno o resultado da saída da rede é comparado com a saída alvo e a estimativa de erro é calculada para a unidade de saída. Os pesos conectados às unidades de saída devem ser ajustados para redução do erro. Depois do erro da saída calculado, este valor é derivado para cálculo estimado do erro da camada intermediária e ajuste dos pesos desta camada.

Finalmente a estimativa do erro é propagado para a camada de entrada para correção dos pesos dos neurônios da camada de entrada [6]. Os detalhes de treinamento e o algoritmo utilizado na rede neural desta pesquisa estão descritos nos tópicos a seguir.

4.4. Rede Multicamada

Uma rede multicamada é uma rede com uma ou mais camadas (níveis) de nós (chamados de unidade oculta ou intermediária) entre as unidades de entrada e saída. Redes multicamadas podem resolver problemas complexos dos quais as redes com somente uma camada não podem resolver, por outro lado o treinamento pode ser mais complicado. Mesmo assim em alguns casos o treinamento pode não ser tão trabalhoso pelo fato da rede multicamada ser mais capaz de encontrar a solução do problema [6].

4.5. Ajuste dos pesos

Além da arquitetura o método de ajuste dos pesos conhecido como treinamento é outra característica fundamental para determinar o tipo de rede.

Para conveniência devemos considerar dois tipos distintos de treinamento, o supervisionado e o não-supervisionado. Mas é importante citar que existem redes que recebem pesos fixos e não passam por um processo iterativo de treinamento.

A maioria das tarefas para as quais as redes são treinadas estão no campo de mapeamento, agrupamento e otimização de restrições. Classificação e associação de padrões são consideradas formas genéricas de mapeamento de padrões de vetores de entrada para um específico vetor ou padrão de saída.

Existe uma ambigüidade na maneira como os métodos de treinamento são rotulados em supervisionados e não-supervisionados, visto que alguns autores denominam as rede de auto-treinamento supervisionado como uma terceira categoria. Contudo em geral existe uma proveitosa correspondência entre o tipo de treinamento e o tipo de problema que se deseja resolver. A seguir estão as características de uma rede supervisionada que foi aplicada nesta pesquisa.

O método supervisionado é talvez a maneira mais típica para a configuração de uma rede, pois o treinamento é realizado através da apresentação da uma seqüência de vetores de treinamento ou padrões, cada qual associado a seu respectivo vetor de saída alvo. Os pesos são ajustados de acordo com o algoritmo de aprendizado. As redes neurais mais simples foram projetadas para classificação de padrões, ou seja, classificar se um padrão de entrada pertence ou não a uma determinada categoria. Neste tipo de rede a saída é um elemento bivalente podendo ter seu valor de saída atribuído como -1 ou 1. O valor -1 corresponde aos vetores que não pertencem á categoria e inversamente os que têm valor 1 pertencem á

categoria. Para problemas complexos de classificação uma rede multicamada de retropropagação é apontada como a melhor solução [7].

O treinamento da rede de retropropagação também conhecido como regra delta generalizada, consiste em aplicar o método de gradiente descendente para minimizar o erro quadrático total das saídas computado pela rede. Neste caso o gradiente da função é o erro e as variáveis são os pesos. O gradiente desta função mostra a direção na qual a função aumenta mais rapidamente assim quando o mesmo for negativo mostra a direção na qual a função decresce mais rapidamente. Para esta técnica é aplicado também à derivada parcial do erro para cada peso, que mostra em que direção os pesos devem ser corrigidos para que haja diminuição do erro.

A derivada parcial tem valor inestimável neste algoritmo para busca do melhor desempenho.

A natureza geral do método de treinamento de retropropagação significa que uma rede deste tipo pode ser usada para resolver problemas em muitas áreas. Aplicações que utilizam tal rede podem ser encontradas em todos os campos que buscam a solução de problemas através do mapeamento de um conjunto de entradas que corresponde a um conjunto de saídas.

O foco é treinar a rede para conseguir um balanço entre a habilidade de dar a resposta correta para um determinado padrão de entrada usado no treinamento e a habilidade de responder com ponderação as entradas similares, mas não idênticas á aquelas empregadas durante o treinamento, a isto chamamos de generalização.

O treinamento da rede para retropropagação envolve três estágios: alimentar os padrões de entrada para que possam transcorrer para frente, executar o cálculo com a retropropagação do erro associado e o ajustar dos pesos. Este ciclo é repetido até que o Erro Total desejado tenha sido atingido. Uma rede neural pode necessitar de vários ciclos para que o erro seja minimizado, este ciclo é chamado de época.

Após o treinamento, a aplicação da rede só envolve o cálculo da fase de alimentar e transcorrer as entradas para frente para obtenção do valor na saída. Por isso fica bem caracterizado neste tipo de rede que o processo de treinamento é mais custoso do que o processo de aplicação onde a rede responde rapidamente. Existem variações no algoritmo de treinamento visando aumentar seu desempenho e diminuir o tempo despendido para treinamento da rede. Na maioria das redes de retropropagação, somente uma camada intermediária é suficiente para o treinamento da mesma [7].

4.6. Algoritmo de treinamento

No início do treinamento as entradas (X_i) são alimentadas e cada uma delas recebe um sinal e propaga este sinal para cada uma das unidades da camada oculta (Z_1, \dots, Z_p). Cada unidade oculta calcula sua ativação e envia o sinal (z_j) para a unidade de saída. Cada unidade de saída (Y_k) calcula sua ativação (y_k) para formar a resposta da rede para o padrão que foi alimentado na entrada.

Durante o treinamento cada unidade de saída compara sua ativação (y_k) com o valor alvo (t_k) para determinar o erro associado para o padrão naquela unidade. Baseado neste erro o fator δ_k ($k=1, \dots, m$) é calculado. O fator δ_k é utilizado para distribuir o erro na unidade de saída de volta para todas as unidades nas camadas anteriores, que são as camadas ocultas conectadas a y_k . Este valor também é utilizado mais tarde no algoritmo para atualizar os pesos entre as saídas e as camadas ocultas. De maneira similar, o fator δ_j ($j=1, \dots, p$) é calculado para cada unidade oculta Z_j . O erro não precisa ser propagado de volta para a camada de entrada, pois δ_j é utilizado para atualizar os pesos entre a camada oculta e a camada de entrada.

Após a determinação de todos os fatores δ os pesos para todas as camadas são ajustados simultaneamente. O ajuste dos pesos w_{jk} (da camada oculta Z_j para a saída Y_k) é baseado no fator δ_k e na ativação z_j da unidade oculta Z_j . O ajuste dos pesos v_{ij} (da unidade de entrada X_i para a unidade oculta Z_j) é baseado no fator δ_j e na ativação x_i da unidade de entrada.

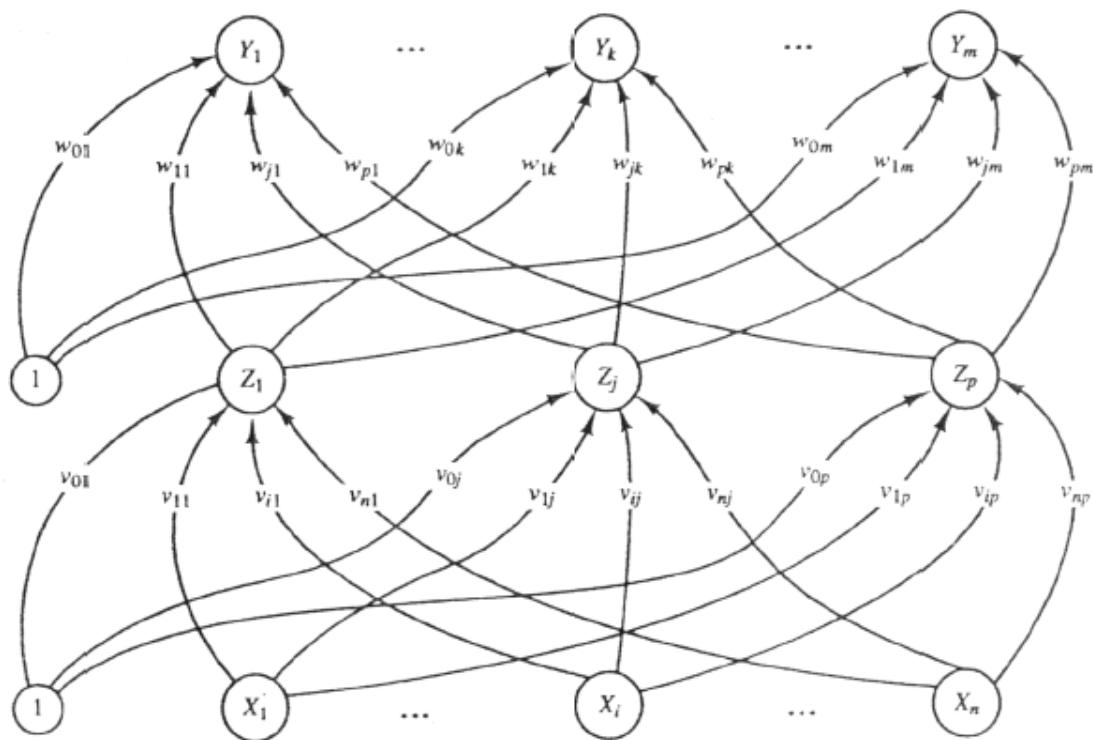


Figura 4 – Diagrama da rede do tipo retropropagação.

As saídas das unidades da rede estão quase sempre associadas a uma função de ativação que deve obrigatoriamente ser contínua, diferenciável devendo possuir uma derivada, e ser monotonicamente não decrescente, ou seja, ter uma sequência designada onde seus membros sucessivos aumentam ou diminuem gradativamente sem ter grandes oscilações sendo que o membro que aumenta deverá ser maior que seu predecessor.

Além disso, para aumentar a eficiência computacional é recomendado que a derivada da função de ativação seja fácil de ser calculada [7].

A derivada das funções de ativação mais comuns normalmente pode ser representada por uma função, espera-se que a função possa saturar no seu mínimo e máximo. As funções de ativação mais comuns são: a função sigmóide binária e a sigmóide bipolar.

O algoritmo de treinamento para rede de retropropagação recomendado por Fausset (1994, p.294) está descrito como:

Passo 0: inicializar os pesos para pequenos valores randômicos;

Passo 1: executar os passos de 2-9 enquanto erro total for maior que erro desejado;

Passo 2: executar os passos de 3-8 para cada par de treinamento;

Passo 3 (alimentação das entradas para saída) : receber o sinal de entrada x_i para cada unidade de entrada ($X_i, i=1,..., n$) e propagar este sinal para as camadas ocultas.

Passo 4: executar a somatória do peso para cada unidade oculta ($Z_j, j=1,..., p$) multiplicado pelo sinal da entrada. Somar o peso do bias da camada de entrada. Aplicar a função de ativação para o sinal de saída e propagar o sinal para as unidades de saída;

$$z_in_j = v_{oj} + \sum x_i \cdot v_{ij} \quad (i=1,..., n) \quad (4.1)$$

$$z_j = f(z_in_j) \quad (4.2)$$

Passo 5: executar a somatória do peso para cada unidade de saída ($Y_k, k=1,..., m$) multiplicada pelo sinal de entrada da camada intermediária. Somar o peso do bias da camada oculta. Aplicar a função de ativação e calcular o valor na saída ;

$$y_in_k = w_{ok} + \sum z_j \cdot w_{jk} \quad (j=1,..., p) \quad (4.3)$$

$$y_k = f(y_in_k) \quad (4.4)$$

Passo 6 (retropropagação do erro): calcular o erro δ_k para cada unidade de saída usando o padrão de entrada alvo e o valor da saída real, calcular a correção dos pesos da camada de saída Δw_{jk} , calcular a correção dos pesos do bias para camada de saída Δw_{ok} e enviar δ_k para as unidades ocultas;

$$\delta_k = (t_k - y_k) \cdot f'(y_in_k), \quad (4.5)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \text{ (usado para atualizar } w_{jk}), \quad (4.6)$$

$$\Delta w_{ok} = \alpha \delta_k \text{ (usado para atualizar } w_{ok}) \quad (4.7)$$

Passo 7: calcular a somatória do delta da entrada para cada unidade oculta δ_{in_j} (Z_j , $j=1, \dots, p$), multiplicado pelo peso da camada intermediária. Multiplicar o resultado pela derivada da sua função de ativação e calcular o erro δ_j . Calcular a correção dos pesos Δv_{ij} , calcular a correção para os pesos do bias Δv_{oj} e atualizar os pesos e os biases;

$$\delta_{in_j} = \sum \delta_k \cdot w_{jk} \quad (k=1, \dots, m), \quad (4.8)$$

$$\delta_j = \delta_{in_j} \cdot f'(z_{in_j}), \quad (4.9)$$

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \text{ (usado para corrigir } v_{ij}), \quad (4.10)$$

$$\Delta v_{oj} = \alpha \cdot \delta_j \text{ (usado para corrigir } v_{oj}). \quad (4.11)$$

Passo 8: atualizar os pesos e os biases para cada unidade de saída Y_k e atualizar os pesos e biases para cada unidade oculta Z_j .

$$w_{jk} \text{ (atual)} = w_{jk} \text{ (anterior)} + \Delta w_{jk} \text{ para } (j=0, \dots, p) \text{ e } (Y_k, k=1, \dots, m) \quad (4.12)$$

$$v_{ij} \text{ (atual)} = v_{ij} \text{ (anterior)} + \Delta v_{ij} \text{ para } (i=1, \dots, n) \text{ e } (Z_j, j=1, \dots, p) \quad (4.13)$$

Passo 9: Testar a condição de parada

Para utilização deste algoritmo as matrizes que armazenam os dados para o cálculo de delta das unidades de saída (passo 6, δ_k) e para o cálculo de delta das unidades ocultas (passo 7, δ_j) devem ser distintas para que a correção do erro seja distinta por camada.

Uma época é denominada um ciclo do algoritmo através de todo o conjunto de vetores de treinamento, tipicamente são necessárias várias épocas para que a rede de retropropagação seja treinada. O algoritmo mostrado atualiza os pesos sempre a cada padrão de treinamento apresentado.

Para evitar que uma pequena mudança na direção do gradiente provoque uma mudança brusca na correção do erro e não permita a convergência, a taxa de aprendizado (α) é adicionada ao algoritmo para limitar este passo de andamento da correção. O maior inconveniente em ter um passo de aprendizado muito pequeno é tornar o aprendizado também muito lento. Mesmo assim é sempre desejável utilizar uma taxa de aprendizado baixa para evitar que ocorra rompimento na direção do aprendizado caso um vetor de entrada incomum seja apresentado na entrada da rede. Sempre que os vetores de treinamento forem similares é recomendável aumentar a taxa de aprendizado.

O momentum também é uma alternativa para atualização dos pesos fazendo com que os mesmos mudem em uma direção dada pela combinação do gradiente atual e o gradiente anterior.

Para utilizar o momentum é necessário salvar o(s) peso(s) ou a atualização do(s) peso(s) dos padrões de treinamento da época anterior. O momentum permite a rede executar grandes atualizações de pesos mesmo com uma taxa de aprendizado pequena, também reduz a probabilidade que a rede encontre um mínimo local que não seja o mínimo global.

Quando o momentum é utilizado a rede não prossegue somente na direção do gradiente atual, mas sim na direção da combinação com o anterior. Esta é uma variação sobre o método do gradiente descendente onde as vantagens se destacam principalmente quando alguns dos vetores de treinamento são diferentes da maioria ou possivelmente estão incorretos. A convergência pode ficar mais rápida quando o termo momentum é adicionado.

A restrição para efetividade do momentum inclui o fato de que a taxa de aprendizado impõe um limite superior na porção que cada peso pode ser alterado e segundo o mesmo pode causar mudanças no peso em uma direção que aumenta o erro.

Capítulo 5. Aplicação da Rede Neural

Neste tópico da pesquisa é mostrada a solução encontrada para alimentar as entradas da RNA através de “hiperlink” de texto, sua arquitetura e as etapas para realização do treinamento da mesma.

5.1. Atuação do sistema para montagem das entradas da RNA

Cada “hiperlink” de um tópico de estudo quando ativado pelo mouse é rastreado sendo que os capítulos são formados por 10 tópicos.

Os “hiperlinks” para navegação do aprendiz através dos tópicos são apresentados pelo sistema conforme mostrado na figura 5.

Você está no capítulo1									
Índice do Capítulo 1									
Tópico 1	Tópico 2	Tópico 3	Tópico 4	Tópico 5	Tópico 6	Tópico 7	Tópico 8	Tópico 9	Tópico 10
Cada tópico deste capítulo deve ser seguido para que você obtenha sucesso.									

Figura 5 - Hiperlink de navegação de estudos.

O diagrama mostrado na figura 6 refere-se às ações adotadas pelo sistema toda vez que o aprendiz ativa um “hiperlink”.

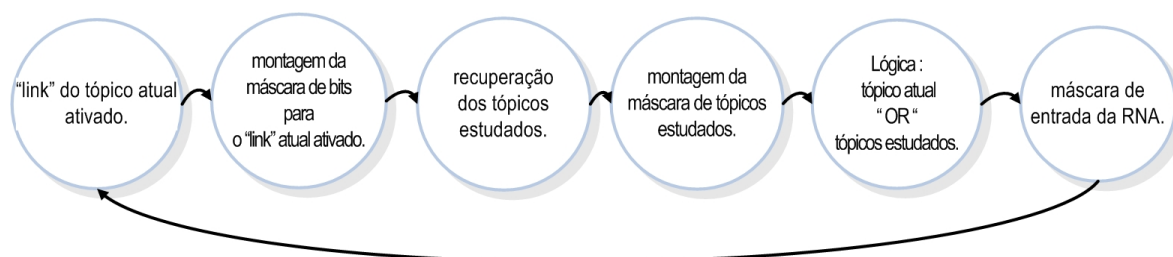


Figura 6 - Fluxograma da montagem das entradas da RNA.

5.2. Definição das entradas da RNA

Para demonstrar o mecanismo adotado neste sistema para a montagem das entradas da rede neural há um exemplo abaixo, onde o aprendiz estudou o tópico 1 e ativa o tópico 2 para prosseguir nos estudos. O sistema prepara as entradas para a rede neural baseado no “hiperlink” ativado pelo mouse, sendo que o “hiperlink” ativado carrega o valor (um) 1 e o “hiperlink” não ativado carrega o valor (menos um) -1. O sistema compara os 10 bits de entrada que representam a navegação do aprendiz naquele momento do estudo com os 10 bits armazenados no banco de dados que representam os tópicos já navegados e então disponibiliza nas entradas da RNA uma máscara de 10 bits.

A figura 7 exemplifica a ativação do tópico 2 pelo aprendiz.

Tópico 1	Tópico 2	Tópico 3	Tópico 4	Tópico 5	Tópico 6	Tópico 7	Tópico 8	Tópico 9	Tópico 10
-1	1	-1	-1	-1	-1	-1	-1	-1	-1

Figura 7 – Entradas ativadas para o tópico 2.

A figura 8 mostra a recuperação de dados do banco para um aprendiz que iniciou o treinamento e portanto havia ativado o tópico 1.

Tópico 1	Tópico 2	Tópico 3	Tópico 4	Tópico 5	Tópico 6	Tópico 7	Tópico 8	Tópico 9	Tópico 10
1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figura 8 – Tópico de estudo recuperado do banco de dados.

Após o sistema detectar qual “hiperlink” foi ativado e recuperar quais os tópicos navegados então é feita a operação “ou” lógico para montar a entrada para rede neural.

Tópico 1	Tópico 2	Tópico 3	Tópico 4	Tópico 5	Tópico 6	Tópico 7	Tópico 8	Tópico 9	Tópico 10
1	1	-1	-1	-1	-1	-1	-1	-1	-1

Figura 9 – Valores de entrada na rede neural.

Desse modo são montadas as entradas da rede sendo que para cada tópico de cada capítulo o sistema repete este modo de operação.

Assim fica demonstrado a solução encontrada para alimentar as entradas da rede neural o que permitiu definir então o número de neurônios na entrada, tipo de função de ativação para as entradas e saídas e a arquitetura da RNA.

5.3. Padrões de treinamento

Uma vez encontrado uma maneira de alimentar as entradas da rede neural foi possível definir os padrões de entrada.

Segundo Fausett (1993, p.298), a relação entre o número de padrões de entrada disponíveis, P , o número de pesos a serem treinados W e a precisão da classificação esperada, e , pode ser resumida na seguinte regra geral:

$$e=W/P \quad (6.1)$$

A questão é: sob quais circunstâncias podemos assegurar que a rede que está sendo treinada para classificar uma percentagem dos padrões de treinamento corretos irá também classificar corretamente padrões de teste escolhidos do mesmo espaço de amostras?

A resposta é: se existe um número suficiente de padrões de treinamento, a rede neural é capaz de generalizar para atingir o resultado desejado [7] .

Os padrões de treinamento escolhidos foram definidos baseado no desejo do instrutor visando os tópicos que devem ser estudados obrigatoriamente pelo aprendiz. Inicialmente foram feitas as seguintes definições:

Tabela 2 - Definição dos padrões de treinamento

Capítulo	Regra para estudo dos tópicos	Observação
1	exemplo: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	todos tópicos devem ser estudados
2	exemplo: 1, 3, 5, 7, 9	tópicos podem ser estudados alternadamente
3	exemplo: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	todos tópicos devem ser estudados
4	exemplo: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	todos tópicos devem ser estudados
5	exemplo: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	todos tópicos devem ser estudados

Uma vez determinadas as regras de estudo para cada capítulo então os padrões para treinamento da rede puderam ser selecionados. Para interpretar os padrões de treinamento deve-se olhar para os valores representados por “1” que significam que o tópico foi ativado ou estudado e “-1” o oposto. Quando se trata da saída da rede, o valor “1” significa que o aprendiz está liberado para estudo daquele tópico selecionado, caso contrário, quando a saída da rede é igual a “-1”, o tópico não foi liberado para estudo de acordo com as regras determinadas para a rede. Os padrões de treinamento encontram-se detalhados no anexo A.

5.4. Arquitetura da RNA para o sistema tutor

Uma vez determinados os padrões de treinamento da rede neural foi definida também sua arquitetura que sofreu modificações durante o treinamento. A configuração inicial da rede neural era de 10 neurônios na camada de entrada, 3 neurônios na camada intermediária e uma saída. Para que a RNA responda-se corretamente a 100% dos padrões de treinamento, foi necessário aumentar o número de neurônios na camada intermediária de 3 para 12, caso contrário não teria sido possível atingir o erro total menor que 0,01.

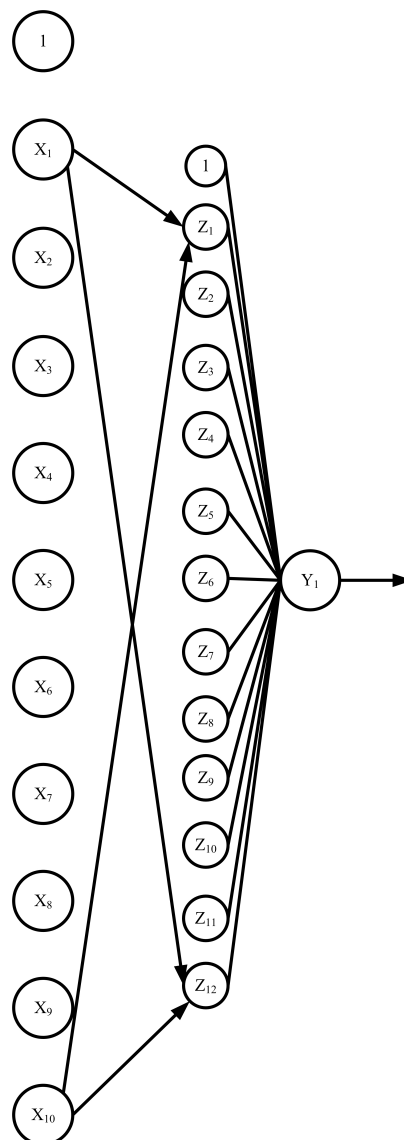


Figura 10 – Arquitetura final da Rede Neural Artificial.