

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

O Uso de Redes Neurais Artificiais no Diagnóstico de
Doenças Reumatológicas

Cristiano Gonçalves e Souza

Lelia de Mello Tostes e Souza

Florianópolis, Novembro de 2004

Cristiano Gonçalves e Souza

Lelia de Mello Tostes e Souza

**O Uso de Redes Neurais Artificiais no Diagnóstico de
Doenças Reumatológicas**

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientação

Prof.º Dr. Jovelino Falqueto

Banca Examinadora

Prof.º Vidal de Souza

Prof.ª Silvia Modesto Nassar

Florianópolis, Novembro de 2004

O Uso de Redes Neurais Artificiais no Diagnóstico de Doenças Reumatológicas

Cristiano Gonçalves e Souza

Lelia de Mello Tostes e Souza

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Sistemas de Informação, e aprovado em sua forma final pela Coordenadoria do Curso de Bacharelado em Sistemas de Informação.

Orientador – Prof. Dr. Jovelino Falqueto

Co-orientador – Prof. M.Sc. Vidal de Souza

Membro da Banca - Profa. Dra. Silvia Nassar

AGRADECIMENTOS

Primeiramente, a Deus,

A nossos familiares e amigos que nos apoiaram e entenderam quando não pudemos estar presentes em alguns momentos;

Ao nosso orientador Prof. Dr. Jovelino Falqueto, por seu acompanhamento, auxílio e compreensão durante toda a confecção deste trabalho;

Ao médico reumatologista Prof. Msc Vidal de Souza, nosso pai e sogro, pelos ensinamentos, explicações e incentivo ao longo deste trabalho, e de todo o curso;

Aos médicos reumatologistas João Elias, Udson Piazza e Williams Willrich por disporem de seu tempo para nos auxiliar no recolhimento dos dados necessários;

À Profa. Dra. Sílvia Modesto Nassar pelas valiosas contribuições.

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE SIGLAS	9
RESUMO	10
ABSTRACT	11
1. INTRODUÇÃO	12
1.1 Motivação	12
1.2 Objetivos	14
2. REDES NEURAIS ARTIFICIAIS	15
2.1 O modelo de McCulloch-Pitts	15
2.2 Redes Neurais Lineares	17
2.3 Perceptron	17
2.4 Mapas de Kohonen - Redes SOM (Self Organizing Maps)	18
2.5 Redes Multi-Layer Perceptron (MLP) treinadas com Retropropagação	19
2.6 Tipos de aprendizado de Redes Neurais	22
3. O DIAGNÓSTICO MÉDICO	24
3.1 Artrite Reumatóide	25
3.2 Ostoartrose	26
3.3 Gota	27
3.4 Fibromialgia	28
4. ESTUDOS ANTERIORES	30
5. METODOLOGIA E IMPLEMENTAÇÃO	32
5.1 Coleta de Dados	32
5.2 Pré-processamento, Limpeza e Tratamento dos Dados	34

5.3	Escolha da Rede Neural	38
5.3.1	Análise da Rede Kohonen.....	38
5.3.2	Análise da Rede MLP treinada com retropropagação.....	40
5.4	Ferramenta Desenvolvida	42
6.	ANÁLISE DOS RESULTADOS	47
6.1.	Testes	48
6.2.	Validação	56
7.	CONCLUSÕES.....	59
8.	Trabalhos Futuros.....	60
8.	BIBLIOGRAFIA.....	61
9.	ANEXOS	63
9.1	Anexo 1 – Ficha do Paciente (preenchida).....	63
9.2	Anexo 2 – SQL para montagem da tabela com padrões para a RNA	64
9.3	Anexo 3 – Dados para a Validação	68
9.4	Anexo 4 – Resultados da Validação	72
9.5	Anexo 5 – Código Fonte	74
9.6.	Anexo 6 - Artigo	85

LISTA DE FIGURAS

Figura 1: Neurônio	16
Figura 2: Modelo ER do Banco de Dados	33
Figura 3: Histograma da Idade	36
Figura 4: Histograma da Idade Normalizada	36
Figura 5: Mapa Topológico da Rede Treinada	39
Figura 6: Dados de Teste Aplicados à Rede MLP treinada com retropropagação	41
Figura 7: Tela de Visualização de Consulta – dados de sintomas	43
Figura 8: Detalhe - Exames	43
Figura 9: Detalhe - Diagnósticos	43
Figura 10: Tela de Treinamento da Rede	44
Figura 11: Simulação da Rede	46
Figura 12: Detalhe - Exames	46
Figura 13: Detalhe - Diagnósticos	46

LISTA DE TABELAS

Tabela 1: Diagnósticos	34
Tabela 2: Classes de Diagnósticos	35
Tabela 3: Configurações da Rede para o Teste 1	48
Tabela 4: Resultados para o Teste 1	49
Tabela 5: Configurações da Rede para o Teste 2	50
Tabela 6: Resultados para o Teste 2	51
Tabela 7: Configurações da Rede para o Teste 3	51
Tabela 8: Resultados para o Teste 3	52
Tabela 9: Configurações da Rede para o Teste 4	53
Tabela 10: Resultados para o Teste 4	53
Tabela 11: Configurações da Rede para o Teste 5	54
Tabela 12: Resultados para o Teste 5	55
Tabela 13: Configurações da Rede após os Testes	55
Tabela 14: Estatísticas sobre o Erro Superior	56
Tabela 15: Estatísticas sobre o Erro Inferior	56
Tabela 16: Erros de Saída encontrados	56
Tabela 17: Registros de Consultas Médicas utilizadas na Validação	71
Tabela 18: Resultados da Validação	73

LISTA DE SIGLAS

ER Entidade Relacionamento

IFD Interfalangiana Distal

IFP Interfalangiana Proximal

MLP Multi-Layer Perceptron

OA Osteoartrose

RNA Redes Neurais Artificiais

SOM Self Organizing Map

RESUMO

O presente trabalho tem como principal objetivo propor o diagnóstico de doenças reumáticas utilizando-se de Redes Neurais Artificiais (RNA) . Para isto, analisamos o comportamento das redes de Kohonen e Multi-Layer Perceptron (MLP) a fim de estabelecer a que apresenta o melhor comportamento, dados, como entradas, uma série de sintomas, exames e características de pacientes e tendo como saída o diagnóstico. Os dados foram coletados de prontuários médicos, primando pela privacidade do paciente. Analisando os prontuários, resolvemos abordar quatro diagnósticos: Artrite Reumatóide, Fibromialgia, Gota e Osteoartrose.

Considerando os resultados obtidos, optamos por implementar uma Rede MLP treinada com Retropropagação e analisamos o seu desempenho em diversas situações nas quais os resultados foram bastante satisfatórios.

Palavras-chave: Redes Neurais Artificiais, Multi-Layer Perceptron, Diagnóstico Reumatológico.

ABSTRACT

The objective of this study is to propose a diagnosis for a given rheumatic illness by using Artificial Neural Networks. In order to accomplish it, we analyzed the Kohonen Self-Organizing Map and the Multi-Layer Perceptron (MLP) in order to establish which is more appropriate to this task. As input, we shall use a series of symptoms, exams and personal related characteristics; the output will be the diagnostic. The data were collected from real medical cases, priming for the identity of the patient. Considering the collected data, four diagnostics were chosen.

Considering the results, we chose to implement a MLP with Back-propagation and study it's performance in different situations achieving good results.

Keywords: Artificial Neural Networks, Multi-Layer Perceptron, Rheumatologic Diagnosys.

1. INTRODUÇÃO

1.1 Motivação

Vam Bemmel (MASSAD, 2003) classifica o conhecimento médico em duas categorias básicas.

O primeiro tipo de conhecimento médico baseia-se na literatura médica. O segundo tipo, o experiencial deriva de informações constantes de bases de dados dos pacientes, tais como o prontuário médico, ou em protocolos clínicos de procedimentos.

A partir destes surge um terceiro tipo de conhecimento que deriva dos dois primeiros, ou seja, o conhecimento baseado em evidências disponíveis nas bases de conhecimento médico universal. Difere do primeiro nas evidências acumuladas ao longo do tempo e sumarizadas pelos processos adequados.

Contudo difere do segundo tipo ao passo que o peso da experiência do especialista individualmente é minimizado e a decisão se baseia na associação entre as observações primárias e os resultados de certas decisões.

Tendo em vista os pontos levantados anteriormente, viemos propor o estudo do processo de descoberta do conhecimento em base de dados, contudo o objetivo deste não se reduz ao conhecimento médico *me si*.

Este estudo trata da aplicação de recursos tecnológicos de inteligência artificial, em específico, redes neurais artificiais, aplicado à ciência médica no diagnóstico médico baseado no conhecimento experiencial de um conjunto de

reumatologistas, registrado em prontuários de pacientes, os quais colaboraram com o presente estudo.

1.2 Objetivos

- OBJETIVO GERAL

O principal objetivo do presente estudo é propor e avaliar modelos de redes neurais artificiais para o diagnóstico de doenças reumáticas.

- OBJETIVOS ESPECÍFICOS

1. Estudar e analisar os algoritmos de treinamento para o desenvolvimento de uma Rede Neural Artificial (RNA);
2. Comparar o desempenho de redes de Kohonen com MLP utilizando ferramenta existente com suporte ao uso de RNAs;
3. Desenvolver uma RNA para o diagnóstico de doenças reumatológicas;
4. Planejar e realizar a coleta de dados de consultas para avaliar os padrões que levam a determinação de um diagnóstico médico;
5. Utilizando casos reais, avaliar o desempenho da RNA desenvolvida.

2. REDES NEURAIIS ARTIFICIAIS

A construção de Redes Neurais Artificiais (RNA) tem inspiração nos neurônios biológicos e nos sistemas nervosos. Porém, as semelhanças entre as RNA e as Redes Neurais Naturais (RNN) são pequenas.

O primeiro modelo de neurônio, foi proposto por McCulloch e Pitts em 1943, e tratava-se de um modelo simples, já que a idéia era imitar a realidade biológica. Porém, essa idéia não é utilizada pelos pesquisadores atuais. Atualmente, dois fatores motivam a pesquisa:

- Modelar o sistema nervoso com suficiente precisão de tal modo a poder observar um comportamento emergente que, sendo semelhante ao comportamento do ser vivo modelado, possa servir às hipóteses usadas na modelagem;
- Construir computadores com alto grau de paralelismo.

2.1 O modelo de McCulloch-Pitts

Apesar da crescente diversidade de variantes em redes neurais, todas têm algumas feições similares. Portanto todas elas, como o cérebro humano, consistem de uma grande quantidade de elementos duplicados – neurônios artificiais, os quais imitam os neurônios do cérebro humano, conectados uns aos outros (STARIKOV, 2001).

Warren S. McCulloch interpretou o funcionamento do neurônio como sendo um circuito binário.

As entradas do neurônio são binárias, e são combinadas por uma soma ponderada produzindo a entrada efetiva do neurônio:

$$S = \sum w_i \cdot x_i$$

A figura 1 mostra um neurônio artificial que, de forma semelhante ao natural, consiste de sinapses conectando as entradas do neurônio com o núcleo; o núcleo do neurônio processa os sinais de entrada e possui axônios que o conecta com os neurônios da próxima camada. Cada sinapse tem seu próprio peso, o qual determina como a entrada de um determinado neurônio influencia na sua condição.

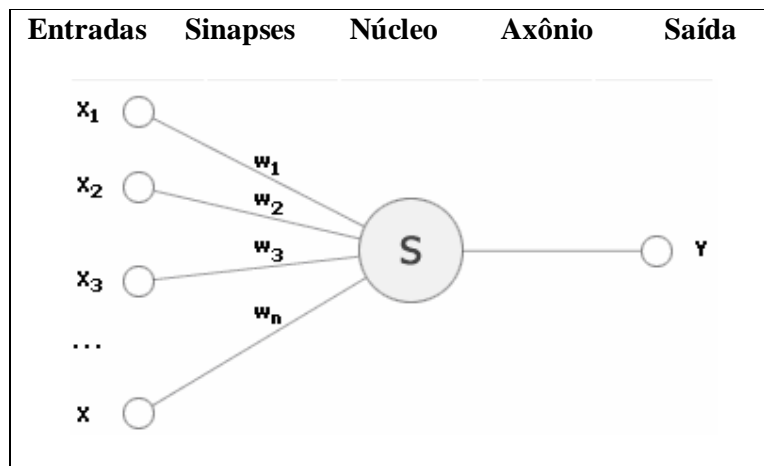


Figura 1: Neurônio

Através deste modelo, McCulloch e seu aluno Walter H. Pitts provaram sua equivalência com a máquina de Turing. Porém, a complexidade da demonstração não satisfaz os estudiosos da época.

2.2 Redes Neurais Lineares

As redes neurais lineares são consideradas o tipo de RNA mais simples que existe. Seus elementos de processamento são chamados “neurônios artificiais”.

A função de saída é linear, com valores de ativação proporcionais à soma das ativações. Assim, por um neurônio i com p entradas do tipo $w_{ij}\mu_j$ têm uma saída y_i dada por:

$$Y_i = \lambda(\sum w_{ij} \cdot \mu_j)$$

onde λ é uma constante de proporcionalidade, y é a saída e μ é a entrada do neurônio. O número real w sumariza a conexão sináptica do neurônio j para o neurônio i correspondente.

2.3 Perceptron

O Perceptron se constitui num dispositivo com três tipos de unidades: sensoriais, associativas e unidades de resposta. Este tipo de RNA pode ter diversas camadas escondidas, no entanto com apenas uma camada adaptativa, onde os pesos podem ser modificados por um algoritmo de aprendizado.

O algoritmo de aprendizado do Perceptron apresenta uma função net_i para cada neurônio i com p entradas do tipo $w_{ij}\mu_j$ dada por:

$$net_i = \sum_j w_{ij} \cdot \mu_j + \theta$$

onde w_{ij} é o peso da conexão do neurônio j para o neurônio i , μ_j são as entradas para o neurônio i , e θ é um termo chamado *bias*.

$$Y_i = \lambda(net_i)$$

onde y é a saída do neurônio i e λ é a função de saída. Neste caso será considerada a função *heaveside* para λ

$$y_i = 1 \text{ se } net_i > \theta$$

$$y_i = 0 \text{ se } net_i \leq \theta$$

Este tipo de rede não é capaz de aprender a classificar funções não linearmente separáveis.

2.4 Mapas de Kohonen - Redes SOM (Self Organizing Maps)

Em 1984 Teuvo Kohonen desenvolveu uma família de redes com a capacidade de reconhecer relações entre padrões apresentados à entrada da mesma, após um treinamento não-supervisionado competitivo.

As redes SOM seguem a arquitetura de aprendizado competitivo, onde além de os neurônios de uma camada se conectarem com os da próxima camada; existem conexões entre unidades da mesma camada.

O algoritmo de treinamento das redes SOM segue a arquitetura de aprendizado onde os neurônios de camadas diferentes se comunicarem por conexões excitatórias e neurônios de uma mesma camada por conexões inibitórias.

O algoritmo *winner-takes-all* (o vencedor leva tudo) é comumente usado como algoritmo de treinamento da camada competitiva da rede de Kohonen. Ele inicialmente atribui pesos aleatórios às conexões entre os neurônios das camadas de entrada e saída, onde a soma dos quadrados dos valores dos pesos das conexões de entrada é idêntica para cada neurônio de saída. Então o vetor padrão de entrada é apresentado à rede e é calculada a distância euclidiana d_j entre cada neurônio j e o vetor de entrada.

$$d_j = \sum (\mu_i(t) - w_{ij}(t))^2$$

onde $\mu_i(t)$ é a entrada do neurônio i no tempo t e $w_{ij}(t)$ é o peso entre o neurônio de entrada i e o neurônio de saída j , no tempo t .

O neurônio com menor d_j será o que melhor representa o padrão de entrada e é denominado o vencedor. Apenas os pesos das conexões entre os neurônios de entrada e a região composta pelo neurônio vencedor são atualizados a cada iteração. Essa vizinhança é inicializada com uma largura inicial que vai diminuindo a cada iteração.

Quando o número de padrões apresentados à rede for suficiente, ou o número de iterações máximo for atingido a rede será considerada treinada.

2.5 Redes Multi-Layer Perceptron (MLP) treinadas com Retropropagação

As redes MLP treinadas com retropropagação são ferramentas poderosas na busca de regras, predição e análise qualitativa. Elas são chamadas de redes de retropropagação por causa do algoritmo de aprendizado que estas usam, na qual um erro se move da camada de saída para a camada de entrada, ou seja, na direção contrária ao sinal original durante a operação normal da rede (STARIKOV, 2001).

As redes MLP consistem de uma série de camadas, onde cada neurônio de uma camada i está conectado com cada neurônio da camada $i + 1$, portanto uma rede *completamente conectada*.

Em geral, a tarefa de treinamento da rede neural se resume a achar uma dependência funcional $Y = f(X)$, onde X é um vetor de entrada e Y é um vetor de saída. Geralmente, esta tarefa, com um conjunto limitado de dados de entrada, tem um conjunto infinito de soluções. Para limitar o espaço de busca durante o treinamento, a tarefa consiste em minimizar a função de eficiência do erro da rede neural através do cálculo dos quadrados mínimos.

Vamos agora considerar o algoritmo completo para o treinamento da rede neural:

1. Aplicar um dos padrões requeridos à entrada da rede neural e definir os valores de saída dos neurônios da rede.
2. Calcular $\delta^{(N)}$ para a camada de saída da rede neural e calcular $\Delta w_{ij}^{(N)}$ as mudanças de pesos na camada de saída N dada a fórmula

$$\Delta w_{ij}^{(n)} = - \eta \cdot \delta_j^{(n)} \cdot x_i^j$$

onde:

η – taxa de aprendizagem

x_i – valor de entrada no neurônio n

$$\delta_o^{(N)} = (d_o - y_o) \cdot \lambda'_o(net_o)$$

para as sinapses entre a camada de saída e a camada escondida e

$$\delta_i^{(N)} = \lambda'_i(net_i) \cdot \sum_h \delta_h^{(N)} w_{hi}$$

para as sinapses pertencentes a uma camada escondida

onde:

λ'_o – derivada da função de saída da camada de saída 'o'

λ'_h – derivada da função de saída da camada escondida 'h'

d – valor de saída esperado

y_o – valor de saída real

net_o – somatório dos valores de entrada de cada neurônio de saída ‘o’

net_i – somatório dos valores de entrada de cada neurônio da camada ‘i’

w_{hi} – peso da sinapse entre os neurônios i e h

3. Calcular através da fórmula

$$\Delta w_{ij}^{(n)} = - \eta \cdot \delta_j^{(n)} \cdot x_i^n$$

correspondentemente para as camadas da rede, $n = N-1..1$, isto é entre cada duas camadas, até a camada de entrada.

4. Ajustar todos os erros da rede neural

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)$$

onde:

t – variável de tempo, indicando a interação atual; $(t-1)$ indica a interação anterior

w_{ij} – peso da sinapse entre os neurônios i e j

5. Se o erro for significativo, retornar ao passo 1. O Erro é calculado pela seguinte fórmula:

$$E(w) = \frac{1}{2} \sum_{j=1}^P (y_j - d_j)^2$$

onde:

y_j – j-ésimo valor de saída da rede

d_j – valor esperado para a j-ésima saída

p – número de neurônios na camada de saída

2.6 Tipos de aprendizado de Redes Neurais

O treinamento supervisionado necessita de um par de vetores composto do vetor de entrada e do vetor alvo que se deseja como saída. Juntos, estes vetores são chamados de par de treinamento ou vetor de treinamento, sendo interessante ressaltar que geralmente a rede é treinada com vários vetores de treinamento.

O procedimento de treinamento funciona da seguinte forma: o vetor de entrada é aplicado. A saída da rede é calculada e comparada com o correspondente vetor alvo. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo determinado, visando minimizar este erro. Este processo de treinamento é repetido até que o erro para os vetores de treinamento tenha alcançado níveis bem baixos.

O treinamento não-supervisionado, por sua vez, não requer vetor alvo para as saídas e, obviamente, não faz comparações para determinar a resposta ideal. O conjunto de treinamento modifica os pesos da rede de forma a produzir saídas que sejam consistentes, isto é, tanto a apresentação de um dos vetores de treinamento, como a apresentação de um vetor que é suficientemente similar, irá produzir o mesmo padrão nas saídas. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes. A aplicação de um vetor de uma determinada classe à entrada da rede irá produzir um vetor de saída específico, mas não

existe maneira de se determinar, antes do treinamento, qual o padrão que será produzido na saída para um vetor de entrada de uma determinada classe. Desta forma, a saída de algumas RNAs deve ser transformada em uma forma compreensiva após o processo de treinamento, o que é um simples problema de identificação das relações entrada-saída estabelecidas pela rede.

No que diz respeito aos algoritmos de treinamento usados, existe uma grande variedade, tanto para o treinamento supervisionado, como para o não-supervisionado.

Muitos deles baseiam-se no algoritmo proposto por D. O. Hebb em 1961. Entretanto, os algoritmos mais eficientes foram desenvolvidos nos últimos trinta anos, baseados nos trabalhos de Rosenblatt (1962), Windrow (1959) e Windrow e Hoff (1960). Entre estes, o mais difundido com certeza foi o algoritmo de *retropropagação* ("Backpropagation") (ABELÉM, 1994).

3. O DIAGNÓSTICO MÉDICO

MASSAD nos ajuda a entender o significado do diagnóstico médico:

O Dicionário Médico Dorland define diagnóstico ou diagnose como sendo a arte de distinguir uma doença de outra e ainda, a determinação da causa de uma doença. Para o Oxford Medical Companion, diagnose é o processo de identificação de uma doença ou outras circunstâncias responsáveis pelas queixas de um paciente, ou seja, de sua doença.

Na verdade, a complexidade do processo diagnóstico deriva da extraordinária variedade de modos de apresentação das entidades produtoras de sintomas e esta complexidade é amplificada pelo impacto do paciente sobre sua expressão. O diagnóstico médico baseia-se em observações clínicas e pode beneficiar-se de técnicas auxiliares como laboratório clínico, imagenologia, etc. Podemos definir o processo diagnóstico tal como Sackett: *um conjunto de técnicas que rotula o paciente e classifica sua doença, identifica o prognóstico provável e define o melhor tratamento disponível.* Do ponto de vista da lógica quantitativa, o diagnóstico é um processo de redução de incertezas. O ato do diagnóstico clínico é, portanto, um processo de classificação, isto é, um esforço para se reconhecer a classe à qual determinado paciente pertence.

Várias tentativas têm sido feitas para se identificar os caminhos cognitivos que levam ao diagnóstico médico: reconhecimento de padrões, estratégias de arborização, estratégia de exaustão ou estratégia hipotético-dedutiva.

Esta última tem sido considerada como a mais apropriada para o processo diagnóstico por ser econômica (é mais rápida) e tem maior precisão.

A estratégia hipotético-dedutiva, aparentemente, é a adotada por clínicos mais experientes.

Em suma, o processo diagnóstico pode ser descrito como a combinação de um ou mais dos seguintes fatores: a abordagem de reconhecimento de padrões pelo clínico experiente, o método de múltiplas ramificações do residente, o método de exaustão do aprendiz, e a técnica aparentemente mais usada, a abordagem hipotético-dedutiva.

No presente estudo visamos propor um modelo, com o uso de redes neurais, para categorizar um conjunto de pacientes em um quadro diagnóstico provável. Por motivos de simplicidade na implementação desta tarefa, nos restringiremos a um subconjunto de diagnósticos da área de conhecimento da medicina denominada reumatologia.

3.1 Artrite Reumatóide

Segundo TAVARES, podemos definir artrite reumatóide da seguinte forma:

A artrite reumatóide é uma doença inflamatória, crônica e de distribuição universal. É a artrite inflamatória mais freqüente, com uma prevalência em torno de 1% a 3% na população mundial.

Seu pico de incidência é maior por volta dos 35 a 45 anos, sendo as mulheres duas vezes e meia mais acometidas que os homens. Em contrapartida os homens, geralmente, têm um curso da doença pior em relação às mulheres.

Ocorre um processo inflamatório complexo resultando em proliferação celular, angiogênese e apoptose (morte celular programada) responsável pela destruição celular.

À medida que os eventos inflamatórios acontecem, os sinais e sintomas clínicos vão surgindo. A dor articular, o edema e a rigidez matinal são os sintomas principais. Tardiamente, há evidente invasão do pannus no

osso subcondral, erosão óssea e distorção da arquitetura articular. Os sintomas incluem dor, edema, instabilidade e deformidades articulares, além de complicações extra-articulares.

3.2 Ostoartrose

FELICE nos ensina sobre osteoartrose:

A osteoartrose (OA) é a doença reumatológica ambulatorial mais comum. Representa 40% do total das consultas e 7,5% dos casos de afastamento do trabalho, incluindo aposentadoria por invalidez. Em 1994 o INSS computou 1.997.906 benefícios e 92.339 aposentadorias por OA, contra 35.912 benefícios e 14.054 aposentadorias por artrite reumatóide.

A diversidade de manifestações da OA justifica diferentes tendências na maneira de abordar sua classificação. Porém, sempre se procurou uma forma que englobasse em um mesmo quadro, os múltiplos tipos de apresentação da doença. Tendências mais recentes, com bases anatômicas, consideram separadamente, com critérios próprios, cada articulação principalmente mãos, joelhos e quadris.

Em mulheres, acima de 50 anos, a incidência de artrose é muito alta. Joelhos e IFDs (interfalangianas distais) são as articulações mais atingidas. Além disso, a hereditariedade, dentro da complexa patogenia da OA, desempenha importante papel, predispondo certos grupos de indivíduos a determinadas formas da doença.

Na clínica, a anamnese e os sinais articulares são elementos básicos para o diagnóstico, que se complementa com o auxílio indispensável da radiologia. Os sinais e sintomas comumente encontrados são: dor articular protocinética, rigidez, diminuição ou perda de função, crepitação, deformidades e, ocasionalmente, inflamação.

O que traz o paciente com OA ao médico é a dor ou algum tipo de deformidade. A dor tem características mecânicas, aparece ou se exacerba

no início dos movimentos (dor protocinética), melhora após alguma movimentação, podendo voltar, se o uso da articulação persistir por tempo prolongado (fadiga articular).

3.3 Gota

De acordo FELLET, Gota pode ser descrita da seguinte maneira:

Gota é uma doença articular inflamatória causada pelo depósito de cristais de monourato de sódio no tecido articular e periarticular.

A doença é mais comum no sexo masculino, no adulto jovem, sendo rara a gota primária na mulher jovem.

O diagnóstico da gota, naturalmente, depende do cuidadoso exame clínico e laboratorial. Embora o quadro clínico possa variar extensamente, quando se realizam os exames adequadamente o diagnóstico em geral é claro.

As características clínicas do paciente típico de gota são: excesso de peso, sexo masculino, meia-idade, consumidor de bebidas alcoólicas (principalmente de fermentação), alimentos ricos em purinas, alimentos com excesso de gordura, que se expõe ao sol, usa medicamentos uricogênicos (diuréticos, corticóides, etc.), com histórico de ataques anteriores, histórico familiar de gota. O ataque da gota é de início rápido, com inflamação aguda que ocorre da noite para o dia, sendo que o primeiro ataque em geral é monoarticular, geralmente nas articulações dos pés, mas, às vezes, nos joelhos, nos ombros e em outras grandes articulações; pode ocorrer durante a primeira semana após uma operação, podendo ocorrer graves lesões renais ou certo grau de disfunção renal, com nefrolitíase por ácido úrico. Confirma-se o diagnóstico pela resposta clássica e rápida à colchicina.

O diagnóstico de laboratório se dá por elevação do nível sérico ou plasmático de uratos, confirmação por método enzimático, método auto-analisador, uricosúria elevada, cristais de monourato de sódio em tofos do

pavilhão auricular, cristais de urato, especialmente os fagocitados pelo PMN no líquido aspirado da articulação inflamada e cristais de urato no espécime de biópsia ou aspirados dos nódulos dos cotovelos.

3.4 Fibromialgia

SCOTTON nos ajuda a definir fibromialgia:

Os dados epidemiológicos referentes à prevalência da fibromialgia são variáveis, de acordo com os diferentes estudos, dependendo da população avaliada, dos autores e da metodologia aplicada.

Em estudos norte-americanos e europeus a prevalência encontrada foi de 2,1% na prática clínica de família, 5,7% na clínica geral, 5% a 8% em pacientes hospitalizados e chegando a 14% a 20% dos atendimentos em reumatologia.

Existe forte predominância do sexo feminino (80% a 90% dos casos), com um pico de incidência entre os 30 e os 50 anos de idade, podendo manifestar-se em crianças, adolescentes e indivíduos mais idosos.

O diagnóstico deve ser feito através de uma boa anamnese e um detalhado exame físico. Enquanto os exames complementares assumem um papel de exclusão de doenças que podem simular um quadro de fibromialgia.

No exame clínico, deveremos investigar os 18 "trigger points" (pontos gatilho) elaborados pelo Colégio Americano de Reumatologia para facilitar o diagnóstico da fibromialgia.

A combinação de dor difusa pelo corpo associada a pelo menos 11 dos 18 "trigger points" proporciona uma sensibilidade de 88,4% e uma especificidade de 81,1%.

Devemos lembrar que alguns pacientes fibromiálgicos apresentam menos de 11 "trigger points" ou dor em outros pontos, além dos

especificados pelo Comitê do Colégio Americano de Reumatologia, sempre excluindo doença sistêmica ou localizada.

Exames laboratoriais de rotina são em geral normais, servindo apenas para exclusão de outras doenças. Os estudos mostram que estes exames estão alterados na mesma proporção da população saudável.

Existe um grupo de diagnósticos possíveis bem maior do que o apresentado; contudo para este estudo nos restringiremos a este conjunto de doenças reumáticas.

4. ESTUDOS ANTERIORES

Atualmente, observamos o uso de RNAs nos mais diversos tipos de aplicação: diagnóstico de doenças, identificação de compostos químicos, análise de testes médicos, controle de processo, predição de horas de trabalho necessárias para procedimentos industriais, identificação bacteriana, análise de custos, diagnóstico diferencial, otimizando resultados de experimentos biológicos, análise nutricional, experimentos agrícolas, gerenciamentos de recursos hídricos, etc. (GUAHYBA, 2001)..

Um estudo envolvendo os Cursos de Ciência da Computação da UPF/RS, Pós-Graduação em Ciência da Computação e Engenharia Biomédica da UFSC observou a aplicação *Neuscience* – *ActiveX*, que utiliza o algoritmo padrão de Kohonen com a aprendizagem competitiva e não-supervisionada no diagnóstico clínico das crises convulsivas. Neste estudo, a taxa de acerto na predição chegou a 90% (CARVALHO, 2002).

A corporação Kaman de Ciências (Colorado Springs, CO) criou uma rede a qual pode prever o câncer de próstata. Este sistema foi apresentado na "CNN" (canal de televisão) e no "Jornal Wall Street" como um grande avanço tecnológico. Redes foram treinadas para prever resultados baseados em teste de PSA ("Prostate Specific Antigen") e para prever a recorrência de câncer depois do tratamento (GUAHYBA, 2001).

O Laboratório Nacional do Noroeste do Pacífico desenvolveu um "nariz artificial" que é estimulado ao entrar em contato um vapor químico sobre seu sensor. Os sinais do sensor então alimentam a rede neural, a qual identifica o elemento químico. O

nariz pode ser usado para identificar elementos químicos em ambientes perigosos ou para monitorar qualidade do ar. Outras aplicações em potencial incluem usar o nariz para avaliar odores corporais para identificar possíveis problemas médicos e monitorar qualidade na indústria de alimentos (GUAHYBA, 2001).

Em 2002, MOSIMANN propôs o uso de redes de Kohonen no auxílio do diagnóstico de doenças reumatológicas. Foram usados como casos práticos os diagnósticos de doenças reumatológicas, apresentando, para os dados obtidos as possibilidades: artrite reumatóide, espondilite anquilosante, artrite gotosa (gota) e saudável. O estudo realizado ressalta as qualidades das redes de Kohonen no processamento de sinais tais como o diagnóstico médico. Observa sua precisão em comparação a outros métodos, porém enfatiza o alto custo de treinamento.

MOSIMANN questiona o desempenho das redes de Kohonen se o universo amostral for muito grande; contudo recomendaria no caso de uma aplicação crítica onde poucos erros têm uma significância grande pois, apesar de bastante lenta e complexa mostrou-se bastante precisa, evitando alguns erros comuns encontrados em alguns algoritmos de agrupamento de dados.

5. METODOLOGIA E IMPLEMENTAÇÃO

Através de algoritmos de aprendizagem o computador pode apresentar um comportamento inteligente no reconhecimento de padrões.

O uso destes algoritmos na área da saúde não é uma exceção; usado desde a locação de recursos hospitalares até a classificação de crises convulsivas ou ainda no diagnóstico médico de doenças.

Demonstraremos agora como foi realizado o presente estudo; desde a coleta de dados até a apresentação de um novo padrão de teste a uma RNA treinada. Os padrões serão compostos de um quadro sintomático, resultados de exames e dados de pacientes para o diagnóstico de doenças reumáticas.

O projeto foi dividido em várias etapas, como segue, visando dividir o problema em partes distintas, cada qual com sua devida importância e necessidade.

5.1 Coleta de Dados

Os dados médicos, sejam eles gerados através de entrevista com o paciente, ou de pesquisas de natureza clínica, epidemiológica ou fisiopatológica, só contribuem para o processo de conhecimento se organizados e interpretados adequadamente, gerando, desta forma, informação.

O prontuário médico do paciente desempenha papel essencial na geração do conhecimento médico. O conhecimento médico obtido de um paciente individual no momento de sua entrevista, consiste fundamentalmente, na etapa primordial do processo

de tomada de decisão, inicialmente na elaboração de um diagnóstico, seguido da proposta terapêutica e do prognóstico do paciente (MASSAD).

Fundamentando-se nesta idéia, os dados utilizados foram coletados através da aplicação de um questionário com questões abertas sobre dados do prontuário médico referentes a consultas de pacientes com um diagnóstico conclusivo. O modelo do questionário se encontra no Anexo 1.

A coleta dos dados foi realizada entre abril e julho de 2004. Para tal foram distribuídos os questionários entre um grupo de médicos reumatologistas, as quais foram preenchidas e posteriormente analisadas pelo co-orientador do presente estudo e também reumatologista, Vidal de Souza, e armazenadas num banco de dados conforme o modelo da figura 2.

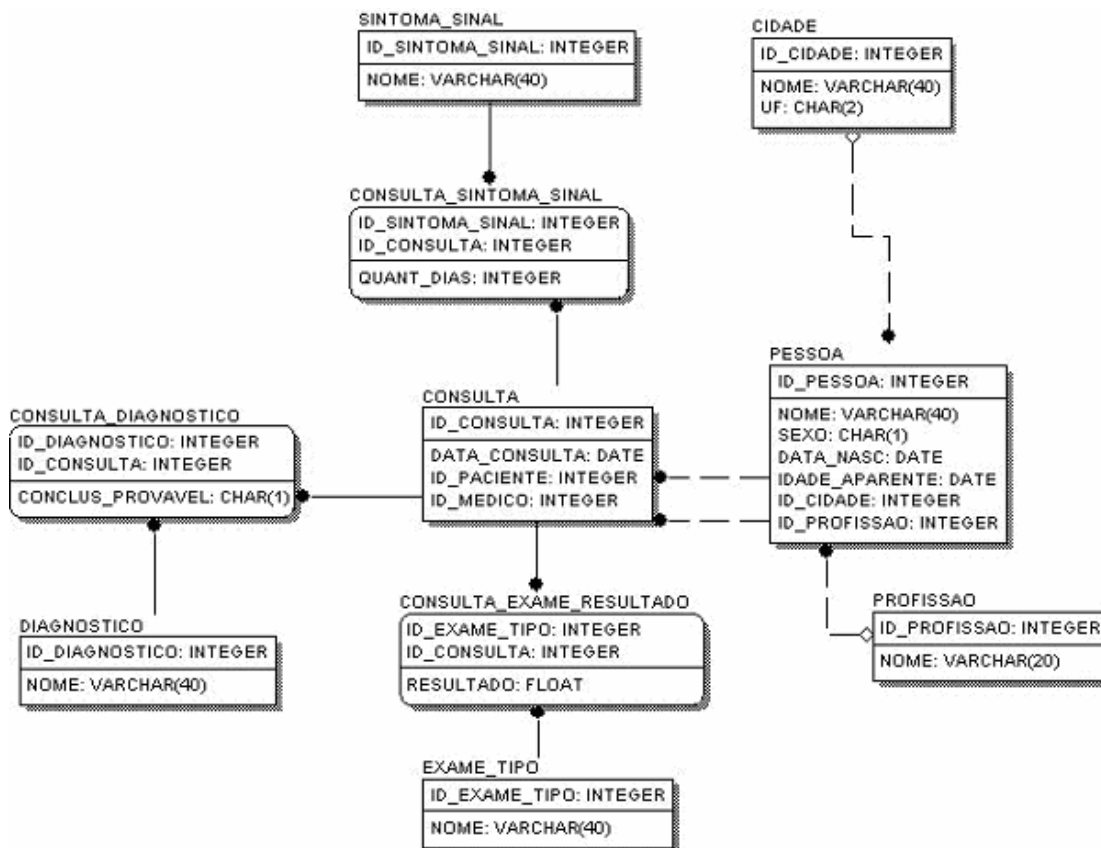


Figura 2: Modelo ER do Banco de Dados.

A partir do modelo relacional, foi montada uma tabela não normalizada, com granularidade referente a uma consulta com apenas um diagnóstico por consulta, com a coleção de sintomas e exames referentes a estes diagnósticos. Esta tabela foi montada a partir do SQL no Anexo 2.

5.2 Pré-processamento, Limpeza e Tratamento dos Dados

Inicialmente foi gerada uma tabela com todos os registros encontrados no banco de dados. A partir desta, foi necessária uma preparação dos dados, que consistiu na escolha dos diagnósticos alvos (tabela 1), baseado no número de ocorrências destes; bem como a escolha dos dados do prontuário médico associados a essas doenças.

DIAGNÓSTICO	OCORRÊNCIA
Artrite Reumatóide	8
Osteoartrose	6
Gota	6
Fibromialgia	6
Lupus Eritematoso Sistêmico	3
Síndrome do Túnel do Carpo	3
Espondilite Anquilosante	3
Tenossinovite Dupuitren (dedo em gatilho)	2
Epicondilite Lateral	2
Outras Doenças	13

Tabela 1: Diagnósticos

Como podemos observar os dados apresentam-se muito dispersos, sendo que apenas quatro diagnósticos possuem mais de três registros em nossa base de dados, portanto os escolhidos foram: Artrite Reumatóide, Osteoartrose, Gota e Fibromialgia.

Uma outra abordagem seria a de verificar a distribuição por Classe de Diagnóstico (fornecidas por um médico reumatologista), como observado na tabela 2:

CLASSE DO DIAGNÓSTICO	OCORRÊNCIA
Partes Moles	22
Colagenosa	11
Degenerativa	6
Microcristalinas	6
Espondiloartropatias Sero-Negativa	3
Inespecífica	3
Osteoporose	1

Tabela 2: Classes de Diagnósticos

Neste caso, a divisão se dá por uma classificação médica, onde não necessariamente as várias doenças seguem um padrão distinguível como entrada de dados, portanto o uso destas categorias é questionável. Por este fato optou-se em manter o diagnóstico como variável alvo.

Os sintomas e resultados de exames relativos a cada uma das consultas escolhidas, associadas a cada diagnóstico, foram selecionados como relevantes para as entrada escolhidas.

Outras informações merecedoras de atenção são a idade, o sexo do paciente e sua ocupação profissional, para verificar se estes devem ou não entrar no modelo de predição. Observaremos somente o conjunto selecionado até então.

Iniciaremos pela idade, observando sua distribuição nos diversos diagnósticos. A média calculada para a distribuição da idade é de 48 anos, com desvio padrão 12,08 em 26 casos registrados. Sua distribuição é próxima da normal, com um pequeno desvio à direita. Podemos observar esta distribuição com a figura 3; contudo, ao normalizarmos a idade temos uma distribuição mais próxima da normal.

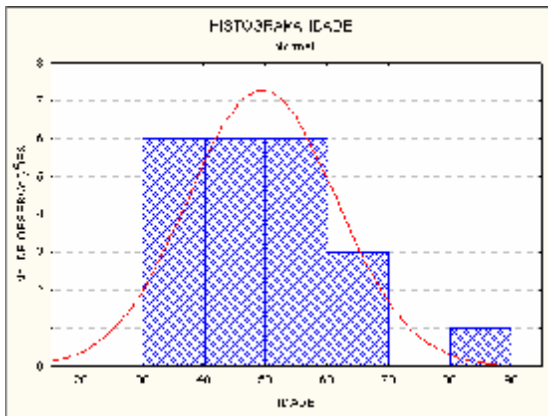


Figura 3: Histograma da Idade

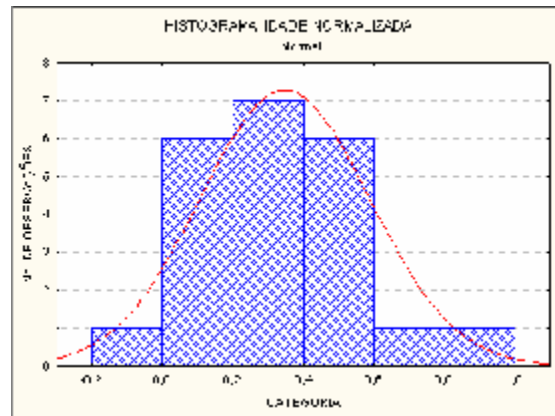


Figura 4: Histograma da Idade Normalizada

As redes neurais requerem que os dados de entrada fiquem no intervalo [0..1], mas os dados do problema variam de acordo com a idade, extrapolando esses valores. Vamos supor que os dados de um parâmetro variam no intervalo [Min..Max]. Então o melhor meio de normalização seria:

$$\tilde{x} = \frac{x - \text{Min}}{\text{Max} - \text{Min}}$$

onde

x – valor inicial do parâmetro

\tilde{x} - valor dado para a entrada da rede

Este método de codificação tem suas desvantagens. Num caso onde a saída não segue a normal, a distribuição dos parâmetros de entrada será extremamente desigual, o que causará uma queda na qualidade do treinamento. Por esta razão em tais casos, como também no caso onde o valor entrada pertence ao intervalo $[0, \infty)$, a normalização pode usar uma função sigmoideal.

No caso da idade, como a saída da função de normalização apresentou uma distribuição normal, optou-se pelo primeiro método.

Para o sexo, observamos que 7 dos 8 casos de ‘artrite reumatóide’ ocorreram em mulher, levando-nos a constatar uma forte ligação entre o sexo e este relatório em específico. Outro caso foi o de ‘gota’, onde 5 dos 6 casos diagnosticados foram com homens. Isso indica a existência de uma relação do sexo na predição do diagnóstico.

Antes de analisarmos a relação do diagnóstico do paciente com sua ocupação, devemos levar em consideração a relação da sua ocupação com sua idade, já que usaremos a idade como uma das variáveis em nosso modelo de predição. A ocupação ‘aposentado’, especialmente, tem uma relação grande com a idade do paciente e não necessariamente com sua atividade profissional até o momento de sua aposentadoria.

Analisando a relação que a profissão ‘DO LAR’ tem com os vários diagnósticos, observamos que esta categoria responde por 6 dos 8 casos de artrite reumatóide e a 3 dos 6 casos de fibromialgia. Isso poderia sugerir que a profissão ‘DO LAR’ tem alguma relação com o diagnóstico, porém para todos os casos, o paciente era mulher indicando uma relação mais forte com o sexo do que com a ocupação profissional.

Dados os problemas citados, desconsideraremos a profissão como variável de entrada no modelo preditivo.

Alguns sintomas e exames da base de dados original não apareceram em nenhum diagnóstico e foram excluídos dos dados que serão usados na modelagem. Outros campos foram excluídos por não contribuírem com a solução do problema tais como dados do médico, local de residência e local da consulta.

5.3 Escolha da Rede Neural

Vários estudos tratam o problema do diagnóstico médico através do uso de métodos de agrupamento e classificação. Visando observar o comportamento de redes neurais para esta tarefa, foram selecionadas duas possíveis abordagens: redes de Kohonen e redes MLP treinadas com retropropagação.

Ambos MOSIMANN e CARVALHO citam o uso de redes de Kohonen como uma abordagem boa para o diagnóstico de doenças. FALQUETO e STARIKOV propõem o uso de redes MLP treinadas com retropropagação para o mesmo problema.

Com o auxílio do software Statistica 6.0, da StatSoft, realizamos uma breve comparação de aspectos favoráveis ou contra a adoção de cada uma das duas abordagens. Ambas tiveram os mesmos padrões de treinamento.

5.3.1 Análise da Rede Kohonen

As redes de Kohonen funcionam basicamente da seguinte maneira: quando um padrão de entrada x é apresentado, a rede procura na camada de neurônios de saída a unidade mais parecida com x . Durante seu treinamento, a rede aumenta a semelhança do neurônio escolhido e de seus vizinhos ao padrão x . Desta forma, a rede constrói um mapa topológico onde neurônios que estão topologicamente próximos respondem de forma semelhante a padrões de entrada semelhante (MOSIMANN).

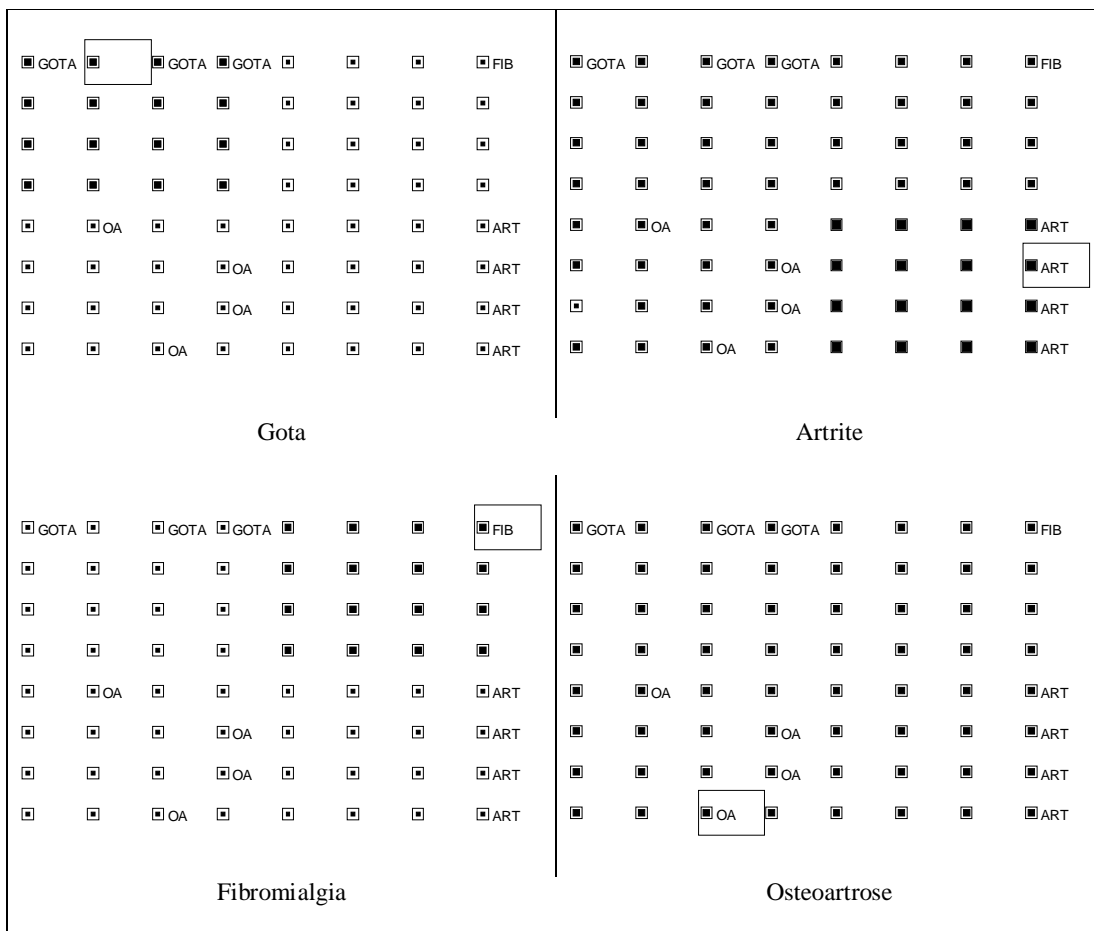


Figura 5: Mapa Topológico da Rede Treinada.

A figura 5 mostra a camada de saída da rede treinada, onde o ponto circulado nos detalhes indica o neurônio vencedor para cada caso, ou seja, o diagnóstico apontado

dados um vetor de entrada correspondente. Os pontos escuros próximos ao neurônio vencedor indicam a vizinhança ativada, e denotam diagnósticos semelhantes.

Ao testarmos os resultados com padrões de teste, obtivemos resultados coerentes com as pesquisas de outros autores.

O tempo gasto com treinamento foi menor que o apresentado pela rede MLP treinada com retropropagação.

Contudo, a rede não indicava claramente quais neurônios de saída estavam relacionados a cada diagnóstico. Além disso, a rede sempre categorizava o padrão de entrada, mesmo sendo ele um paciente saudável, ou com outra doença desconhecida por ela.

Em casos onde o padrão de entrada era de um paciente com doenças associadas (gota e artrite reumatóide, por exemplo), a rede somente categorizava-o em no máximo um dos diagnósticos corretos.

Os dados coletados apresentavam o diagnóstico do paciente, e esta informação era desprezada, pelo fato das redes de Kohonen utilizarem um método de treinamento não-supervisionado.

5.3.2 Análise da Rede MLP treinada com retropropagação

A Rede MLP treinada com retropropagação, geralmente apresenta 3 camadas de neurônios. Os valores de entrada na primeira camada são pesados e passados para a segunda camada, a qual passa valores à camada de saída da mesma forma. A camada de saída apresenta os resultados desejados, sendo que cada neurônio da camada de saída indica a presença ou não de um determinado fator de predição ou classificação.

	FIBROM	ART	GOTA	AO	T. FIBROM	T. ART	T. GOTA	T. AO
01	0.0432512	0.01314	0.04071	0.9751962	0	0	0	1
02	0.8953557	0.04185	0.05481	0.06751	1	0	0	0
03	0.9067177	0.0437	0.0525365	0.0629609	1	0	0	0
04	0.03865	0.0726	0.8961117	0.0778731	0	0	1	0
05	0.08969	0.03851	0.8611091	0.08089	0	0	1	0
06	0.02273	0.9900779	0.02996	0.008681	0	1	0	0
07	0.0545	0.1036545	0.02147	0.8511675	0	0	0	1
08	0.04031	0.03754	0.1033761	0.8512948	0	0	0	1
09	0.02046	0.9170769	0.03302	0.08549	0	1	0	0
10	0.01503	0.9758734	0.02379	0.0424	0	1	0	0
11	0.04863	0.05525	0.9345978	0.04851	0	0	1	0

Figura 6: Dados de Teste Aplicados à Rede MLP treinada com retropropagação

A figura 6 mostra as saídas apresentadas pela rede MLP treinada com retropropagação quando apresentada a padrões de entrada, comparadas com suas respectivas saídas esperadas. Analisando o caso número 1, por exemplo, podemos observar que as saídas da rede para fibromialgia – ‘FIBROM’ (0,0432512), artrite reumatóide – ‘ART’ (0,01314), Gota – ‘GOTA’ (0,04071) estão próximos de 0, que eram as respostas esperadas para estas saídas. Enquanto isso o valor apresentado por Osteoartrose – ‘AO’ (0,9751962) está bem próximo de 1, exatamente o resultado esperado para este caso. Os resultados dos testes foram satisfatórios, com um número de acertos superior aos apresentados pelas redes de Kohonen.

O tempo gasto com treinamento foi muito grande pois é processo de atualização dos pesos é muito custoso. Em casos com muitos neurônios, ou ainda muitos padrões de treinamento, o custo pode ser proibitivo.

Porém, como cada saída da rede corresponde a um diagnóstico, fica fácil interpretar sua saída. Para um paciente saudável, bem como para doenças não treinadas pela rede, os valores para todas as saídas se distanciavam de 1, ou seja, negavam os diagnósticos.

Ainda para o caso de doenças associadas, o resultado apresentava os múltiplos diagnósticos do paciente.

Dadas as diferenças entre os dois modelos apresentados, optamos por implementar uma rede MLP treinada com retropropagação e analisar o seu desempenho. A linguagem de programação escolhida foi o Delphi, usando a ferramenta de desenvolvimento da Borland Delphi 6.0.

5.4 Ferramenta Desenvolvida

O programa desenvolvido para o Diagnóstico Reumatológico, o qual chamaremos de *Diagnoser*, tem por objetivo treinar uma rede neural baseado em consultas médicas cadastradas numa base Firebird 1.5 (utilizamos a ferramenta IbExpert, da HK-Software para o cadastramento dos casos); dados estes coletados com médicos reumatologistas, conforme visto anteriormente.

As figuras 7, 8 e 9 mostram os dados que foram coletados de cada paciente, bem o diagnóstico de cada consulta. Foram cadastradas somente consultas com um único diagnóstico.

Diagnoser

Consulta | Treinamento da Rede | Simulação da Rede

NOME: HMP RESIDENCIA: FLORIANÓPOLIS

IDADE: 49 Sexo: M F MEDICO: HUDSON PIAZZA LOCAL CONSULTA: FLORIANÓPOLIS

Sintomas | Exames | Diagnósticos

<input type="checkbox"/> Ansiedade/Depressão	<input type="checkbox"/> Inchaço ou edema
<input type="checkbox"/> Artrite de Curta Duração	<input type="checkbox"/> Lombalgia
<input type="checkbox"/> Artrite de Mãos/Dedos	<input checked="" type="checkbox"/> Manobras/sinais compatíveis com osteoartrose artic
<input type="checkbox"/> Artrite de Punhos	<input type="checkbox"/> Monoartrite de início súbito
<input type="checkbox"/> Aumento Vol Art IFD e IFP de dedos (mãos)	<input type="checkbox"/> Nódulos de Heberden/Buchard
<input type="checkbox"/> Astenia	<input type="checkbox"/> Oligoartralgia
<input type="checkbox"/> Cervicalgia	<input type="checkbox"/> Oligoartrite
<input type="checkbox"/> Dificuldade em fechar as mãos	<input type="checkbox"/> Parestesia antebraço/mão/dedos da mão
<input type="checkbox"/> Dor a Palpação - Trigger Points	<input type="checkbox"/> Poliartralgia
<input type="checkbox"/> Dor aos Movimentos/Esforços	<input type="checkbox"/> Poliartralgia simétrica
<input type="checkbox"/> Dor em Arts IFD e IFP de Dedos (mãos)	<input type="checkbox"/> Poliartrite (Aditiva)
<input type="checkbox"/> Dor em Cotovelo - aperto de mãos/segura objetos	<input type="checkbox"/> Rigidez em coluna lombar/dorsal
<input checked="" type="checkbox"/> Dor em Região Cervical aos Movimentos (c/ ou s/ crepitado)	<input type="checkbox"/> Rigidez articular ao iniciar movimentos
<input type="checkbox"/> Dor em Repouso	<input type="checkbox"/> Rigidez articular matinal curta duração
<input type="checkbox"/> Dor Generalizada	<input type="checkbox"/> Rigidez articular matinal prolongada
<input type="checkbox"/> Dor Matinal	<input checked="" type="checkbox"/> Sinais envolvimento pulmonar/cardíaco/renal
<input type="checkbox"/> História Familiar Osteoporose	<input type="checkbox"/> Síndrome do Sono
<input type="checkbox"/> História Progressiva Gota	<input type="checkbox"/> Travamento/Dor Flexo-Extensão da mão

Figura 7: Tela de Visualização de Consulta – dados de sintomas

Sintomas | Exames | Diagnósticos

RX/RNM/Outro compatível com artrose

Fator reumatóide negativo

Fator reumatóide positivo

Hiperuricemia ou hiperuricosúria

Prova de atividade inflamatória negativa

Prova de atividade inflamatória positiva

RX com dados sugestivos de osteoartrose

RX mãos sugestivos de osteoartrose

Figura 8: Detalhe - Exames

Sintomas | Exames | Diagnósticos

DIAGNOST

OSTEDARTROSE (ARTROSE, OSTEDARTRITE) PRIMÁRIA

CLASSEDIA

DEGENERATIVA

Figura 9: Detalhe - Diagnósticos

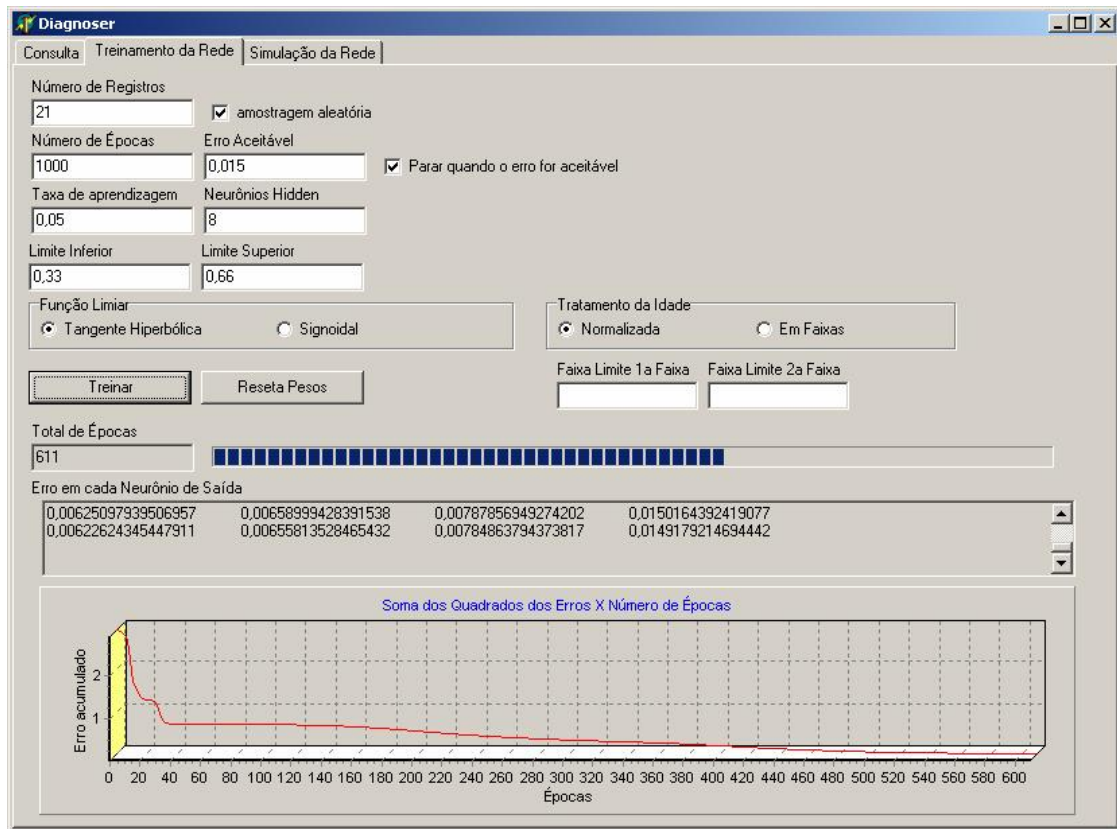


Figura 10: Tela de Treinamento da Rede

Na figura 10 temos a tela de configuração de parâmetros para treinamento da rede. Nesta pode ser informado o número de registros que serão utilizados para o treinamento, permitindo assim que se possa reservar uma parte para testes.

Podemos também indicar se os dados para treinamento serão escolhidos de forma aleatória, ou se serão trazidos da base sequencialmente. Definimos o número máximo de épocas a ser usada, bem como se a parada será dada por atingir o número de épocas máximo, ou quando o erro acumulado em cada saída chegar ao valor proposto pelo usuário.

A quantidade de neurônios da camada *Hidden* (escondida) também é definida pelo usuário. FALQUETO propõe, entre outras maneiras para definir este valor, o uso

da regra de Eberhart: “O número de neurônios da camada escondida é igual à raiz quadrada da quantidade de neurônios da camada de entrada mais a quantidade da camada de saída”.

Outros valores que o usuário pode configurar são:

- a taxa de aprendizagem, que segundo vários autores varia entre 0,05 e 0,25;
- o limite inferior e superior – saídas abaixo do limite inferior não são consideradas diagnósticos válidos; abaixo do limite superior são considerados diagnósticos possíveis, e acima disto, prováveis;
- A função de saída (*threshold*) usada para o treinamento;
- O tratamento da idade – se for normalizada, a idade será transformada pela fórmula de normalização, citada anteriormente. Se for por faixas, a primeira faixa vai de zero ao valor definido em “Faixa Limite 1ª Faixa”; a segunda faixa, entre este valor e o informado em “Faixa Limite 2ª Faixa” e a terceira, idades a partir deste último valor.

No final do treinamento, será apresentado o número de épocas efetivamente executado, o erro acumulado por época, por neurônio de saída e um gráfico com a evolução do treinamento (soma dos quadrados dos erros por número de épocas).

As figuras 11, 12 e 13 mostram a simulação da rede, onde o usuário pode apresentar os dados de um paciente com seu quadro sintomático e resultado de exames, solicitando então para a rede informar as possibilidades de diagnósticos baseado no seu aprendizado.

Para o exemplo apresentado nas figuras, baseado no limite inferior, descartaríamos as hipóteses de diagnósticos: Artrite Reumatóide, Fibromialgia e

Osteoartrose, e considerariamos Gota como diagnóstico provável baseado no limite superior.

Diagnoser
 Consulta | Treinamento da Rede | Simulação da Rede

IDADE: 45
 SEXO: M F

Sintomas | Exames | Diagnósticos

- Ansiedade/Depressão
- Artrite de Curta Duração
- Artrite de Mãos/Dedos
- Artrite de Punhos
- Aumento Vol Art IFD e IFP de dedos (mãos)
- Astenia
- Cervicalgia
- Dificuldade em fechar as mãos
- Dor a Palpação - Trigger Points
- Dor aos Movimentos/Esforços
- Dor em Arts IFD e IFP de Dedos (mãos)
- Dor em Cotovelo - aperto de mãos/segura objetos
- Dor em Região Cervical aos Movimentos (c/ ou s/ crepitado)
- Dor em Repouso
- Dor Generalizada
- Dor Matinal
- História Familiar Osteoporose
- História Pregressa Gota
- Inchaço ou edema
- Lombalgia
- Manobras/sinais compatíveis com osteoartrose artic
- Monoartrite de início súbito
- Nódulos de Heberden/Buchard
- Oligoartralgia
- Oligoartrite
- Parestesia antebraço/mão/dedos da mão
- Poliartralgia
- Poliartralgia simétrica
- Poliartrite (Aditiva)
- Rigidez em coluna lombar/dorsal
- Rigidez articular ao iniciar movimentos
- Rigidez articular matinal curta duração
- Rigidez articular matinal prolongada
- Sinais envolvimento pulmonar/cardíaco/renal
- Síndrome do Sono
- Travamento/Dor Flexo-Extensão da mão

Figura 11: Simulação da Rede

Sintomas | Exames | Diagnósticos

- RX/RNM/Outro compatível com artrose
- Fator reumatóide negativo
- Fator reumatóide positivo
- Hiperuricemia ou hipericosúria
- Prova de atividade inflamatória negativa
- Prova de atividade inflamatória positiva
- RX com dados sugestivos de osteoartrose
- RX mãos sugestivos de osteoartrose

Figura 12: Detalhe - Exames

Sintomas | Exames | Diagnósticos

Diagnóstico	
Artrite Reumatóide	Descartado - (8,29%)
Fibromialgia	Descartado - (7,94%)
Gota	Provável - (75,66%)
Osteoartrose	Descartado - (14,43%)

Figura 13: Detalhe - Diagnósticos

6. ANÁLISE DOS RESULTADOS

Para a análise que se segue, foram coletados 28 prontuários relativos a consultas médicas. Este conjunto de dados foi dividido em duas partes: um subconjunto de dados para testes e outro subconjunto para a validação da rede.

Cada teste consistirá na aplicação de 13 consultas médicas a uma rede treinada, onde serão avaliados os diagnósticos propostos pela rede. Durante os testes serão selecionados os parâmetros de configuração para criar as RNAs a serem verificadas na validação.

A validação consistirá no treinamento de 10 RNAs, cuja configuração segue a proposta apresentada durante os testes. Ao final de cada treinamento, serão aplicadas as 15 consultas médicas restantes a cada RNA, e então verificado seu desempenho.

Para ambos os casos, a aplicação de um padrão (uma consulta médica específica) à rede resultará numa saída, a qual estará classificada numa das três categorias seguintes:

- Diagnóstico provável - acima do limite superior;
- Diagnóstico possível – entre o limite inferior e superior;
- Diagnóstico descartado – abaixo do limite inferior.

A rede terá acertado o diagnóstico quando responder afirmativamente às questões que se seguem:

- Quando o diagnóstico era único considerou como provável o diagnóstico correto?

- Quando haviam diagnósticos associados, considerou como provável ou possível os diagnósticos corretos?
- Para todos os casos, descartou corretamente os diagnósticos adequados?

6.1. Testes

Para o primeiro teste, tentaremos ajustar um valor de erro aceitável para a rede neural que proporemos.

Teste 1: Erro Aceitável

Configurações da Rede			
Parar quando o erro for aceitável	ü	Função de Saída	Tangente Hiperbólica
Erro Aceitável	?	Tratamento da Idade	Normalizada
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	-

Tabela 3: Configurações da Rede para o Teste 1

Casos de teste:

1. Erro Aceitável = 0,015
2. Erro Aceitável = 0,0015
3. Erro Aceitável = 0,00015

Erro Aceitável = 0,015	
Acertos	11

Limite superior mínimo encontrado	73,03%
Limite inferior máximo encontrado	15,26%
Número de épocas realizadas	504
Erro Aceitável = 0,0015	
Acertos	11
Limite superior mínimo encontrado	90,37%
Limite inferior máximo encontrado	6,02%
Número de épocas realizadas	1188
Erro Aceitável = 0,00015	
Acertos	12
Limite superior mínimo encontrado	88,76%
Limite inferior máximo encontrado	4,04%
Número de épocas realizadas	7506

Tabela 4: Resultados para o Teste 1

O caso de teste 3 obteve o maior número de acertos, portanto utilizaremos o erro aceitável de 0,00015 para os próximos testes.

Teste 2: Taxa de Aprendizagem

Configurações da Rede

Parar quando o erro for aceitável	\dot{u}	Função de Saída	Tangente Hiperbólica
Erro Aceitável	0,00015	Tratamento da Idade	Normalizada
Taxa de aprendizagem	?	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	-

Tabela 5: Configurações da Rede para o Teste 2

Casos de teste:

1. Taxa de Aprendizagem = 0,05
2. Taxa de Aprendizagem = 0,10
3. Taxa de Aprendizagem = 0,25

Taxa de Aprendizagem = 0,05	
Acertos	12
Limite superior mínimo encontrado	88,76%
Limite inferior máximo encontrado	4,04%
Número de épocas realizadas	8921
Taxa de Aprendizagem = 0,10	
Acertos	11
Limite superior mínimo encontrado	96,92%
Limite inferior máximo encontrado	1,88%
Número de épocas realizadas	4606
Taxa de Aprendizagem = 0,25	
Acertos	11
Limite superior mínimo encontrado	97,01%

Limite inferior máximo encontrado	3,48%
Número de épocas realizadas	1745

Tabela 6: Resultados para o Teste 2

O caso de teste 1 obteve o maior número de acertos, portanto utilizaremos o valor da taxa de aprendizagem de 0,05 para os próximos testes.

Teste 3: Neurônios *Hidden*

Configurações da Rede			
Parar quando o erro for aceitável	\dot{u}	Função de Saída	Tangente Hiperbólica
Erro Aceitável	0,00015	Tratamento da Idade	Normalizada
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	?	Faixa Limite 2ª Faixa	-

Tabela 7: Configurações da Rede para o Teste 3

Casos de teste:

1. Neurônios *Hidden* = 8
2. Neurônios *Hidden* = 4
3. Neurônios *Hidden* = 23

Neurônios <i>Hidden</i> = 8	
Acertos	12

Limite superior mínimo encontrado	88,76%
Limite inferior máximo encontrado	4,04%
Número de épocas realizadas	8921
Neurônios <i>Hidden</i> = 4	
Acertos	12
Limite superior mínimo encontrado	95,55%
Limite inferior máximo encontrado	3,33%
Número de épocas realizadas	8581
Neurônios <i>Hidden</i> = 23	
Acertos	11
Limite superior mínimo encontrado	92,51%
Limite inferior máximo encontrado	2,5%
Número de épocas realizadas	10878

Tabela 8: Resultados para o Teste 3

Os casos 1 e 2 apresentaram mesmo número de acertos; no entanto, ao analisarmos os resultados de cada uma das 13 simulações de consultas, observamos que o caso de teste 1 apresentou melhor resultado quando o diagnóstico era associado, errando com uma margem menor em relação ao caso 2. Tendo este quesito como embasamento, optaremos por utilizar 8 neurônios na camada *hidden* para os próximos testes.

Teste 4: Função de Saída

Configurações da Rede

Parar quando o erro for aceitável	\dot{U}	Função de Saída	?
Erro Aceitável	0,00015	Tratamento da Idade	Normalizada
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	-

Tabela 9: Configurações da Rede para o Teste 4

Casos de teste:

1. Função de Saída = Tangente Hiperbólica
2. Função de Saída = Sigmoidal

Função de Saída = Tangente Hiperbólica	
Acertos	12
Limite superior mínimo encontrado	88,76%
Limite inferior máximo encontrado	4,04%
Número de épocas realizadas	8921
Função de Saída = Sigmoidal	
Acertos	12
Limite superior mínimo encontrado	90,89%
Limite inferior máximo encontrado	2,06
Número de épocas realizadas	8582

Tabela 10: Resultados para o Teste 4

Ambos os casos apresentaram mesmo número de acertos e comportamento diante de diagnósticos associados, então usaremos como critério, o caso que obteve o maior limite superior e o menor limite inferior, ou seja, o caso 2. Observamos também

que para este teste, o caso 2 também teve o menor tempo de treinamento, com 8582 épocas, 339 menos que o primeiro caso. Portanto, passaremos a utilizar a função sigmoidal para o próximo teste.

Teste 5: Tratamento da Idade

Configurações da Rede			
Parar quando o erro for aceitável	\dot{u}	Função de Saída	Sigmoidal
Erro Aceitável	0,00015	Tratamento da Idade	?
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	?
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	?

Tabela 11: Configurações da Rede para o Teste 5

Casos de teste:

1. Tratamento da Idade = Normalizada
2. Tratamento da Idade = Em Faixas
 - Faixa Limite 1ª Faixa = 45 anos
 - Faixa Limite 2ª Faixa = 60 anos

Tratamento da Idade = Normalizada	
Acertos	12
Limite superior mínimo encontrado	90,89%

Limite inferior máximo encontrado	2,06
Número de épocas realizadas	8582
Tratamento da Idade = Em Faixas (até 45anos; de 45 a 60anos; acima de 60 anos)	
Acertos	12
Limite superior mínimo encontrado	82,88 %
Limite inferior máximo encontrado	5,74%
Número de épocas realizadas	7197

Tabela 12: Resultados para o Teste 5

Novamente obtivemos acertos iguais, então utilizaremos o mesmo critério usado para o teste anterior – do maior limite superior e menor limite inferior – e adotaremos a idade normalizada.

Após os testes realizados, obtivemos a seguinte configuração de rede para treinamento:

Configurações da Rede			
Parar quando o erro for aceitável	Ü	Função de Saída	Sigmoidal
Erro Aceitável	0,00015	Tratamento da Idade	Normalizada
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	-

Tabela 13: Configurações da Rede após os Testes

A partir dos valores para o ‘Limite Superior Mínimo’ e ‘Limite Inferior Máximo’ encontrados nos testes, podemos determinar qual seriam valores de limite mais adequados para a interpretação das saídas da rede treinada.

Para encontrarmos valores adequados para os limites, em cada teste pegamos o maior erro de saída apresentado para ambos os valores. A partir destes valores (conforme tabela 16), calculamos a média e o desvio padrão. Aplicou-se então o teste *t* de Student (BARBETTA, 2002) para calcular os valores de corte com um grau de confiabilidade de 95% para que o valor fosse aceito quando superior ao limite superior ou inferior ao limite inferior respectivamente, como observado nas tabelas 14 e 15.

Erro Superior	Erro Inferior
0,27	0,15
0,10	0,06
0,11	0,04
0,03	0,02
0,03	0,03
0,04	0,03
0,07	0,03
0,09	0,02
0,17	0,06

Tabela 16: Erros de Saída encontrados

Média do Erro Superior:	0,10
Desvio Padrão (Erro Superior):	0,0771
Valor de corte	0,23
Limite Superior Estabelecido:	0,77

Tabela 14: Estatísticas sobre o Erro Superior.

Média do Erro Inferior:	0,10
Desvio Padrão (Erro Inferior):	0,0414
Valor de corte	0,17
Limite Inferior Estabelecido:	0,17

Tabela 15: Estatísticas sobre o Erro Inferior.

6.2. Validação

A etapa de validação consistiu na aplicação de alguns padrões ou consultas médicas reais não previamente cadastrados na base de dados utilizada, listados no anexo 3, como entrada para a RNA proposta ao final da etapa de testes e na avaliação dos dados de saída com o diagnóstico esperado.

Os critérios de avaliação da saída foram os mesmos utilizados para o teste.

A RNA foi treinada com os parâmetros apresentados na tabela 13, com Limite Superior de 0,77 (ou seja; caso o valor apresentado pelo neurônio da camada de saída for igual ou maior que 0,77, o diagnóstico é provável) e Limite Inferior de 0,17 (caso o valor apresentado pelo neurônio da camada de saída for igual ou menor que 0,17, a possibilidade de apresentar o diagnóstico é descartada). Dada esta rede treinada, foram aplicados e verificados os 15 padrões apresentados no anexo 3 – tabela 17.

Este processo se repetiu dez vezes e seus resultados foram registrados no anexo 4 – tabela 18; onde é possível observarmos as saídas da rede para os 15 padrões propostos no anexo 3 - tabela 17, para os dez treinamentos da validação.

Para uma análise mais objetiva, consideraremos cada uma aplicação de um padrão a um treinamento específico da rede como um caso de validação. Desta forma, como temos 15 padrões aplicados a 10 treinamentos, temos 150 casos de validação distintos.

Destes 150 casos de validação, 90 referem-se a consultas onde o médico chegou a um único diagnóstico; 30 onde o médico não chegou a nenhum diagnóstico dos previstos pela RNA e outros 30 casos onde o médico percebeu a presença de dois diagnósticos associados.

Ao avaliarmos os 90 casos de validação, onde o paciente apresentava apenas um dos diagnósticos conhecidos pela RNA proposta, verificamos que a rede conseguiu prever o diagnóstico correto para todos os casos.

Quando foi apresentado um padrão referente a um diagnóstico não conhecido pela rede, foi apontado que para 11 dos casos existiam diagnósticos válidos; sendo 6 deles de doenças associadas e 5 com um diagnóstico único. Todos apontavam para Fibromialgia, sendo que em dois deles, o único sintoma apresentado pelo paciente era dor generalizada', um sintoma importante para o diagnóstico de Fibromialgia e para os outros 9 casos, a confirmação da inexistência de atividade inflamatória ou de fator reumatóide positivo indicavam que dos quatro diagnósticos conhecidos pela rede, o mais provável seria o do Fibromialgia.

Os 30 casos de validação restantes apresentaram 8 respostas diferentes do esperado. Apesar de não indicar a existência de uma associação de diagnósticos para estes 8 casos de validação, todos indicaram a existência de ao menos um dos diagnósticos corretos.

Considerando todos os casos de validação, obtemos uma média geral de acerto de 87,33%.

7. CONCLUSÕES

O propósito principal do presente estudo – propor e avaliar modelos de redes neurais artificiais para o diagnóstico de doenças reumáticas – foi plenamente alcançado, o que levou a algumas considerações.

A definição do escopo do problema bem como a escolha do tratamento dos dados é de suma importância para a realização adequada do trabalho, já que permite focar no que realmente interessa para o estudo considerando que a área estudada é tão vasta e abrangente.

A escolha da topologia da rede também foi de grande importância para o sucesso do trabalho, visto que com a gama de redes existentes, foi necessário um estudo aprofundado para estabelecer a que melhor se encaixaria ao propósito do trabalho.

Os resultados obtidos mostraram que o modelo proposto apresentou-se bastante eficiente para a previsão do diagnóstico médico quando este for único, ou seja, não apresentar doenças associadas. A abordagem adotada não foi muito apropriada para esta abordagem bem como para o caso de diagnósticos não conclusivos.

Uma maneira de abranger estas situações, seria acrescentar aos dados de entrada, consultas com diagnósticos associados ou não conclusivos. Deste modo, a rede poderia aprender estes casos, e conseguiria prever resultados quando fosse lhe apresentadas situações semelhantes.

Avaliando o aspecto tecnológico conseguiu-se verificar que o desenvolvimento de sistemas no auxílio à decisão baseada em redes neurais artificiais é

uma alternativa com desempenho e confiabilidade aceitáveis, especialmente no caso de diagnóstico médico.

De um ângulo diferente, a medicina pode se beneficiar dos sistemas de informação, como os estudados aqui, inclusive como ferramenta de suporte em consultório e clínicas médicas auxiliando e reafirmando o diagnóstico.

Ainda assim existem aspectos que somente o ser humano é capaz de identificar, os quais não foram levados em conta pelas RNAs.

Portanto, tem-se que os sistemas de informação abordados existem para adicionar conhecimento à área da saúde e não substituir o médico.

8. Trabalhos Futuros

Durante a realização deste estudo verificou-se a possibilidade de melhorias para possíveis estudos futuros, tais como:

- Questionários com diagnósticos definidos e entradas padronizadas;
- Treinar a rede com diagnósticos associados e não conclusivos;
- Tratamento do sexo como uma entrada única;
- Estudar a viabilidade do sistema desenvolvido como ferramenta de ensino para graduandos da área de medicina.

8. BIBLIOGRAFIA

ABELÉM, Antônio Jorge Gomes. Redes Neurais Artificiais na Previsão de Séries Temporais. Dissertação de Mestrado, PUC – RJ, Dept^o de Engenharia Elétrica. Set/1994

AZEVEDO, Fernando M. et al. Redes neurais com aplicações em controle e sistemas especialistas. Visual Books, Ago/2000.

BARBETTA, Pedro A. Estatística aplicada às ciências sociais. 5^a edição revisada, Editora da UFSC, 2002

CARVALHO, Lucimar F. et al. Redes neurais artificiais auto-organizáveis utilizadas na classificação de crises convulsivas. Universidade de Passo Fundo/RS, 2002.

FALQUETO, Jovelino. Inspiração biológica em IA. Tese de Doutorado, UFSC, Dept^o de Informática e Estatística, Mar/2002

FELICE, José Carlos et al. Temas de Reumatologia Clínica – Ostoartrose (AO). CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=2031

FELLET, Aloysio J. Elementos Básicos do Diagnóstico de Gota. CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=1308

GUAHYBA, Adriano. Utilização de inteligência artificial (redes neurais artificiais) no gerenciamento de reprodutoras pesadas. Tese de Doutorado, PPGCV – UFRGS, 2001.

MASSAD, Marin. O prontuário eletrônico do paciente na assistência, informação e conhecimento médico. Azevedo Neto editores OPAS/OMS 2003.

MOSIMANN, João P. Uma análise comparativa entre a utilização de Redes de Kohonen e o método PAM no agrupamento de dados para o auxílio do diagnóstico de doenças reumatológicas. UFSC, 2002

SCOTTON, Antônio Scafuto. Como diagnosticar e tratar fibromialgia. CiberSaude, 2004. http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=583

STARIKOV, Alexey. Neural networks - mathematical apparatus. BaseGroup Lab Data Analysis Technology, 2001.

TAVARES, Laura Nascimento et al. Elementos básicos de diagnóstico da doença (artrite) reumatóide. CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=128

9. ANEXOS

9.1 Anexo 1 – Ficha do Paciente (preenchida)

FICHA DO PACIENTE	
I - IDENTIFICAÇÃO	
NOME (as letras iniciais): JJC	
SEXO: M(x) F()	IDADE ou DATA NASC: 49 anos
PROFISSÃO: comerciário	CIDADE ONDE RESIDE: Lages
Para os quesitos II a V , favor preceder o dado com as seguintes informações: data do evento e se o evento é uma consulta ou retorno ; por exemplo: consulta(01/01/2001), retorno(14/01/2001)	
II – SINTOMAS CLÍNICOS (Descrever sucintamente apenas dados clínicos retirados da anamnese, tais como: dores articulares, dores musculares, inflamação articular, rigidez articular, etc.) (28/01/04) poliartralgia simétrica, artrite em mãos, punhos, há 2 meses, rigidez matinal	
III – SINAIS CLÍNICOS (Descrever os achados do exame físico que sejam importantes para o diagnóstico, tais como: artrite de punho, dor e diminuição da amplitude da art. do ombro do exame, cianose de mãos e dedos, etc.) Exame Físico: Edema em tornozelos, edema em arts. Metacarpo-falangianas, fácies de sofrimento	
IV – RESULTADOS DE EXAMES COMPLEMENTARES (Oferecer resultados de exames complementares, tais como: Látex = 80, FAN = 1/180, VHS = 35, Teste patergia +, Estenose de volva mitral ao ecocardiograma, Presença de cristais de monurato de sódio no líquido sinovial, etc.) Ex. complementares: (03/01/04) VHS 38, látex 80, Muco = 6,2; PCR = 24	
V – DIAGNÓSTICO (Oferecer um ou mais conforme o caso e esclarecer de conclusivo ou provável. Os casos em que não haja um diagnóstico conclusivo ou provável, declarar como inconclusível) HD = Artrite reumatóide	
Obs: Se possível oferecer um telefone para contato (do médico) para o caso de necessitar esclarecer algum dado não compreendido pelo analista: <i>Contato: Nome: Vidal Fone:</i>	

9.2 Anexo 2 – SQL para montagem da tabela com padrões para a RNA

```
select
  c.id_consulta,
  pac.nome, (c.data_consulta - pac.idade_aparente) / (365) as idade, pac.sexo,
  med.nome as medico,
  cid.nome as residencia, cid.id_cidade as id_residencia,
  cidm.nome as localCons, cidm.id_cidade as id_loacalCons,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 3) as sintoma3,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 4) as sintoma4,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 5) as sintoma5,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 6) as sintoma6,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 7) as sintoma7,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 8) as sintoma8,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 10) as sintoma10,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 14) as sintoma14,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 15) as sintoma15,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 17) as sintoma17,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 18) as sintoma18,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 21) as sintoma21,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 22) as sintoma22,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 26) as sintoma26,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 27) as sintoma27,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 28) as sintoma28,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 32) as sintoma32,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 34) as sintoma34,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 36) as sintoma36,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 38) as sintoma38,
  (select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
  css.id_sintoma_sinal = 39) as sintoma39,
```



```

(select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
css.id_sintoma_sinal = 84) as sintoma84,
(select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
css.id_sintoma_sinal = 85) as sintoma85,
(select count(id_sintoma_sinal) from consulta_sintoma_sinal css where css.id_consulta = c.id_consulta and
css.id_sintoma_sinal = 87) as sintoma87,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 2) as exame2,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 3) as exame3,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 4) as exame4,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 5) as exame5,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 6) as exame6,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 7) as exame7,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 8) as exame8,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 12) as exame12,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 13) as exame13,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 14) as exame14,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 16) as exame16,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 17) as exame17,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 20) as exame20,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 22) as exame22,
(select count(id_exame_tipo) from consulta_exame_resultado css where css.id_consulta = c.id_consulta and
css.id_exame_tipo = 26) as exame26,
diag.id_diagnostico,
diag.nome as diagnost,
dcla.id_diagnostico_classe,
dcla.nome as classeDiag
from consulta c, pessoa pac, pessoa med, cidade cid, cidade cidm,
diagnostico diag, consulta_diagnostico cdiag, diagnostico_classe dcla
where pac.id_pessoa = c.id_paciente and
med.id_pessoa = c.id_medico and
cid.id_cidade = pac.id_cidade and
cidm.id_cidade = med.id_cidade and
diag.id_diagnostico = cdiag.id_diagnostico and
c.id_consulta = cdiag.id_consulta and
dcla.id_diagnostico_classe = diagnostico.id_diagnostico_classe and
(diag.id_diagnostico = 15 or
diag.id_diagnostico = 20 or

```

```
diag.id_diagnostico = 19 or  
diag.id_diagnostico = 8 )  
order by c.id_consulta
```

9.3 Anexo 3 – Dados para a Validação

Padrão	Idade	Sexo	Sintomas/Sinais/Exames	Diagnóstico
1	59	F	Rigidez ao iniciar movimentos Poliartalgia Inchaço Dor aos movimentos Manobras/sinais OA Nódulos de Heberden RX compatível com OA Prova de Atividade Inflamatória +	Osteoartrose
2	65	M	Dor matinal Rigidez matinal prolongada Inchaço Poliartrite Artrite mão/dedos Fator Reumatóide + Prova de Atividade Inflamatória +	Artrite Reumatóide
3	49	M	Dor Generalizada	Inconclusivo
4	51	F	Lombalgia Poliartralgia Cervicalgia Aumento volume artic IFD/IFP Nódulos de Heberden RX compatível com OA	Osteoartrose

5	13	M	Inchaço Fator Reumatóide – Prova de Atividade Inflamatória -	Monoartrite Inespecífica
6	65	F	Dor aos movimentos Fator Reumatóide – Prova de Atividade Inflamatória -	Ciatalgia
7	48	F	Dor generalizada Síndrome do sono Dor à palpação – <i>Trigger Points</i> Fator Reumatóide – Prova de Atividade Inflamatória -	Fibromialgia
8	60	F	Dor artic IFD/IFP Nódulos de Heberden RX compatível com OA	Osteoartrose
9	45	M	Inchaço Monoartrite de início súbito História pregressa de Gota Hiperuricemia Prova de Atividade Inflamatória +	Gota
10	42	M	Inchaço História pregressa de Gota Hiperuricemia Prova de Atividade Inflamatória +	Gota
11	66	M	Poliartrite Dor artic IFD/IFP	Artrite Reumatóide

			<p>Dificuldade em fechar as mãos</p> <p>Rigidez ao iniciar movimentos</p> <p>Artrite de punhos</p> <p>Artrite de mãos</p> <p>Prova de Atividade Inflamatória +</p>	
12	36	F	<p>Dor Generalizada</p> <p>Astenia</p> <p>Ansiedade</p> <p>Dor movimentos</p> <p>Prova de Atividade Inflamatória -</p>	Fibromialgia
13	36	F	<p>Artrite mãos/dedos</p> <p>Artrite de punhos</p> <p>Astenia</p> <p>Dor à palpação – <i>Trigger Points</i></p> <p>Dor generalizada</p> <p>Poliartalgia simétrica</p> <p>Rigidez matinal de longa duração</p> <p>Síndrome do sono</p> <p>Fator Reumatóide +</p> <p>Prova de Atividade Inflamatória +</p>	<p>Fibromialgia +</p> <p>Artrite Reumatóide</p>
14	52	M	<p>Dor aos movimentos</p> <p>História pregressa de Gota</p> <p>Monoartrite de início súbito</p> <p>Nódulos de Heberden</p> <p>Oligoartrite</p>	<p>Gota +</p> <p>Osteoartrose</p>

			RX compatível com OA Hiperuricemia Prova de Atividade Inflamatória +	
15	40	M	Artrite mãos/dedos Aumento de volume artic IFD/IFP História pregressa de Gota Inchaço Poliartralgia simétrica Poliartrite Hiperuricemia Fator Reumatóide + Prova de Atividade Inflamatória +	Gota + Artrite Reumatóide

Tabela 17: Registros de Consultas Médicas utilizadas na Validação.

9.4 Anexo 4 – Resultados da Validação

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Treinamento 1															
AR	1,95	98,41	7,99	1,22	5,17	7,30	1,75	7,78	0,72	1,96	97,94	1,34	95,01	0,07	73,85
FIB	0,64	0,51	58,46	1,05	86,03	56,27	98,34	1,52	0,58	0,36	0,59	97,06	17,59	0,18	0,13
GOT	0,44	1,40	3,21	0,35	1,52	0,93	0,65	0,47	99,11	98,77	1,04	0,83	0,48	89,15	60,82
OST	97,90	0,75	9,42	98,13	3,18	6,29	1,24	90,39	0,41	0,47	0,96	1,11	0,11	44,30	0,54
Treinamento 2															
AR	1,78	98,56	6,23	1,39	5,95	6,52	1,96	1,39	0,80	1,47	95,19	1,45	93,90	0,01	73,56
FIB	0,36	0,81	59,61	0,52	62,17	71,57	98,55	1,30	0,30	0,30	0,77	97,86	34,17	0,52	0,07
GOT	1,08	1,07	6,19	1,14	6,25	1,35	0,66	2,94	98,70	98,31	1,52	0,53	0,60	95,36	76,72
OST	98,03	0,65	9,00	97,74	1,63	2,95	0,51	93,18	0,77	0,75	1,29	1,27	0,06	51,02	0,36
Treinamento 3															
AR	4,10	98,75	4,82	1,25	4,87	5,32	1,34	3,38	0,53	1,42	96,79	0,54	93,31	0,19	66,90
FIB	0,55	0,68	56,95	1,41	73,76	63,36	98,44	1,97	0,69	0,71	0,88	97,52	40,80	0,07	0,11
GOT	0,33	0,68	5,43	0,60	5,40	1,05	0,80	0,84	99,27	98,83	0,64	0,67	0,39	95,11	66,22
OST	98,75	0,88	7,73	96,59	1,89	14,54	0,65	91,07	0,38	0,35	1,38	2,48	0,07	14,72	0,25
Treinamento 4															
AR	3,06	98,41	11,04	1,23	6,37	3,03	1,20	2,45	0,77	1,39	98,45	1,03	97,72	0,42	92,70
FIB	0,72	0,48	53,06	1,50	79,09	87,79	98,85	3,50	0,76	0,68	0,71	97,82	7,59	0,55	0,31
GOT	0,30	0,82	4,50	0,32	2,95	1,32	1,79	0,85	98,85	98,12	0,40	1,25	0,43	89,83	33,43
OST	97,65	0,97	6,85	97,91	2,34	5,68	0,58	82,26	0,55	0,67	1,10	1,11	0,15	3,20	0,09
Treinamento 5															
AR	1,13	98,50	8,52	0,61	6,99	2,66	1,59	2,20	0,52	1,18	96,82	0,52	93,95	0,08	72,56
FIB	0,91	1,05	61,08	0,97	73,92	69,38	98,13	2,00	0,31	0,33	1,38	97,72	55,91	0,59	0,11
GOT	0,77	0,61	3,01	1,00	6,30	1,45	0,67	1,06	99,20	98,88	0,58	0,76	0,08	35,43	40,21
OST	98,76	0,46	6,03	98,56	1,12	19,31	0,51	93,79	1,33	1,08	0,81	2,21	0,06	93,49	0,77
Treinamento 6															
AR	5,27	98,66	2,11	1,00	20,91	5,97	3,82	0,39	1,04	2,17	94,52	0,45	96,19	0,90	18,95
FIB	1,01	0,33	59,05	1,72	68,55	60,85	98,12	2,54	0,82	0,59	0,37	98,30	2,81	0,20	0,81
GOT	0,08	1,10	3,33	0,57	1,67	2,18	0,76	4,03	98,90	98,52	0,57	0,40	0,33	64,34	52,10
OST	99,21	0,79	32,37	95,70	1,17	2,14	0,14	93,41	0,64	0,76	4,68	2,96	1,11	93,87	0,87
Treinamento 7															
AR	1,35	99,21	4,29	0,89	3,90	1,38	0,53	1,71	0,90	1,36	98,68	0,51	95,58	0,20	79,49
FIB	0,39	0,61	59,26	1,91	45,73	69,12	98,59	2,05	0,32	0,42	0,57	98,30	21,73	0,13	0,43
GOT	0,86	0,58	5,75	0,55	7,11	0,87	0,84	3,21	99,21	98,80	0,74	0,70	0,29	93,03	14,39
OST	97,74	0,98	6,80	98,27	3,19	25,11	1,03	90,90	0,70	0,74	1,00	1,24	0,15	32,85	0,49
Treinamento 8															

AR	1,28	98,46	7,29	1,00	8,58	2,40	0,92	2,66	1,28	1,81	97,56	0,77	95,50	0,27	63,40
FIB	0,82	0,67	62,72	1,14	86,93	62,34	98,32	3,69	0,54	0,72	0,90	97,69	8,48	0,50	0,53
GOT	0,78	0,79	3,89	0,84	1,52	0,94	1,22	0,96	98,73	98,40	0,78	0,52	0,31	71,33	37,56
OST	97,93	0,85	8,81	98,25	3,72	20,87	0,71	93,71	0,42	0,42	1,21	2,42	0,53	38,83	0,31
Treinamento 9															
AR	0,72	98,54	4,95	0,84	4,59	2,63	1,48	0,93	0,86	1,84	95,98	0,77	94,35	0,14	64,14
FIB	1,21	0,67	57,96	1,58	92,37	65,14	98,37	2,29	0,54	0,56	0,84	97,69	4,04	0,24	0,06
GOT	0,62	0,79	5,73	0,54	2,02	0,87	0,67	2,69	99,33	98,90	0,69	0,52	0,17	89,79	23,16
OST	97,77	0,85	7,83	98,93	1,32	12,15	0,57	91,79	0,34	0,34	1,55	2,91	0,58	35,47	0,62
Treinamento 10															
AR	1,98	98,63	18,25	1,17	6,90	2,19	1,44	2,14	0,57	4,20	98,34	0,95	94,65	0,40	49,53
FIB	0,84	0,61	59,79	0,88	47,07	72,93	98,68	1,75	0,29	0,19	0,63	97,74	55,43	0,60	0,36
GOT	0,46	0,91	3,68	0,56	3,95	1,49	0,44	1,25	99,26	98,80	0,81	0,71	0,18	88,93	77,68
OST	97,61	0,71	3,76	98,34	5,57	4,36	1,11	92,63	0,61	0,33	0,93	1,25	0,13	11,26	0,10

Tabela 18: Resultados da Validação.

Legenda da Tabela 18:

	AR – Artrite Reumatóide;
	FIB – Fibromialgia;
	GOT – Gota;
	OST – Osteoartrose
	Erros na Predição do Diagnóstico

9.5 Anexo 5 – Código Fonte

Código do Diagnoser – Delphi

```
//Diagnoser
unit Diagnoser;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, DataModule, StdCtrls, Grids, DBGrids, DBCtrls, urede, Mask, ExtCtrls, ComCtrls;
type
  TFormDiagnostico = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    eNumReg: TEdit;
    btTreinar: TButton;
    GroupBox2: TGroupBox;
    DBCheckBox1: TDBCheckBox;
    DBCheckBox2: TDBCheckBox;
    DBCheckBox3: TDBCheckBox;
    DBCheckBox5: TDBCheckBox;
    DBCheckBox6: TDBCheckBox;
    DBCheckBox7: TDBCheckBox;
    DBCheckBox8: TDBCheckBox;
    DBCheckBox9: TDBCheckBox;
    DBCheckBox10: TDBCheckBox;
    DBCheckBox11: TDBCheckBox;
    DBCheckBox12: TDBCheckBox;
    DBCheckBox13: TDBCheckBox;
    DBCheckBox14: TDBCheckBox;
    DBCheckBox15: TDBCheckBox;
    DBCheckBox16: TDBCheckBox;
    DBCheckBox17: TDBCheckBox;
    DBCheckBox18: TDBCheckBox;
    DBCheckBox19: TDBCheckBox;
    DBCheckBox26: TDBCheckBox;
    DBCheckBox4: TDBCheckBox;
    GroupBox3: TGroupBox;
    DBCheckBox20: TDBCheckBox;
    DBCheckBox21: TDBCheckBox;
    DBCheckBox22: TDBCheckBox;
    Label3: TLabel;
    chbmasc: TDBCheckBox;
    chbfem: TDBCheckBox;
    Label2: TLabel;
    eNumEpic: TEdit;
    Label4: TLabel;
    Memo1: TMemo;
    eDiagnostico: TEdit;
    Button1: TButton;
    Label5: TLabel;
    DBEdit1: TDBEdit;
    Label6: TLabel;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    ProgressBar1: TProgressBar;
    procedure btTreinarClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
```

```

FReDe: TReDe;
FEscolhidos: Set of byte;
function MontaPadroes: TPadroes;
public
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;

  { Public declarations }
end;

implementation

uses Math, DB;
{$R *.dfm}
constructor TFormDiagnostico.Create(AOwner: TComponent);
begin
  inherited;
  FReDe := TReDe.Create;
end;

destructor TFormDiagnostico.Destroy;
begin
  FReDe.Free;
  inherited;
end;

procedure TFormDiagnostico.btTreinarClick(Sender: TObject);
var
  i, j : Integer;
  aErros: TSinais;
  astring: string;
begin
  FReDe.TaxaAprendizagem := 0.05;
  aErros := FReDe.treinar(MontaPadroes, StrToInt(eNumEpoc.Text), ProgressBar1);
  Memo1.Clear;
  for i := 0 to Length(aErros) - 1 do begin
    astring := ";";
    for j := 0 to Length(aErros[i]) - 1 do begin
      astring := astring + ' ' + FloatToStr(aErros[i][j]);
    end;
    Memo1.Lines.Add(astring);
  end;
end;

function TFormDiagnostico.MontaPadroes: TPadroes;
var
  APadroes: TPadroes;
  APadrao: TPadrao;
  i, j, n: integer;
begin
  SetLength(APadroes, 0);
  FEscolhidos := [];
  for i := 0 to StrToInt(eNumReg.Text) - 1 do begin // pega vinte amostras aleatórias da base
    randomize;
    n :=random(28) + 3;
    while n in FEscolhidos do begin // sorteia um que nao foi escolhido
      randomize;
      n :=random(28) + 3;
      FEscolhidos := FEscolhidos + [n];
    end;
    DataModule1.TableConsulta.First;
  end;
end;

```

```

for j := 0 to DataModule1.TableConsulta.RecordCount - 1 do begin
  if DataModule1.TableConsulta.FieldByName('COD_CONS').AsInteger = n then
    break;
  DataModule1.TableConsulta.Next;
end;
SetLength(APadrao.x, 23);
APadrao.x[0] :=
IfThen(DataModule1.TableConsulta.FieldByName('AFTAS_ORAL_GENITAL').AsBoolean, 1, 0);
APadrao.x[1] :=
IfThen(DataModule1.TableConsulta.FieldByName('ARTRITE_MAOS_PUNHOS').AsBoolean, 1, 0);
APadrao.x[2] :=
IfThen(DataModule1.TableConsulta.FieldByName('DORES_MOVIMENTOS').AsBoolean, 1, 0);
APadrao.x[3] :=
IfThen(DataModule1.TableConsulta.FieldByName('DORES_REPOUSO').AsBoolean, 1, 0);
APadrao.x[4] := IfThen(DataModule1.TableConsulta.FieldByName('ENTESOPATIA').AsBoolean, 1,
0);
APadrao.x[5] :=
IfThen(DataModule1.TableConsulta.FieldByName('EROSOES_AO_RX').AsBoolean, 1, 0);
APadrao.x[6] :=
IfThen(DataModule1.TableConsulta.FieldByName('ESCLERITE_OU_OVEITE').AsBoolean, 1, 0);
APadrao.x[7] :=
IfThen(DataModule1.TableConsulta.FieldByName('F_REUMATOIDE_MAIOR_80').AsBoolean, 1, 0);
APadrao.x[8] :=
IfThen(DataModule1.TableConsulta.FieldByName('LOMBALGIA_CRONICA').AsBoolean, 1, 0);
APadrao.x[9] :=
IfThen(DataModule1.TableConsulta.FieldByName('NODULOS_HEBERDEN_BURCHAT').AsBoolean,
1, 0);
APadrao.x[10] :=
IfThen(DataModule1.TableConsulta.FieldByName('OLIGO_ARTRALGIA').AsBoolean, 1, 0);
APadrao.x[11] := IfThen(DataModule1.TableConsulta.FieldByName('OSTEOFITOS').AsBoolean, 1,
0);
APadrao.x[12] :=
IfThen(DataModule1.TableConsulta.FieldByName('OSTEOPOROSE_VISIVEL_RX').AsBoolean, 1, 0);
APadrao.x[13] :=
IfThen(DataModule1.TableConsulta.FieldByName('POLIARTRALGIA').AsBoolean, 1, 0);
APadrao.x[14] :=
IfThen(DataModule1.TableConsulta.FieldByName('POLIARTRITE_6_SEMANAS').AsBoolean, 1, 0);
APadrao.x[15] :=
IfThen(DataModule1.TableConsulta.FieldByName('PROVA_PATERGIA_POSITIVA').AsBoolean, 1,
0);
APadrao.x[16] :=
IfThen(DataModule1.TableConsulta.FieldByName('PUSTULAS_CUTANEAS').AsBoolean, 1, 0);
APadrao.x[17] :=
IfThen(DataModule1.TableConsulta.FieldByName('RIGIDEZ_ESQ_AXIAL').AsBoolean, 1, 0);
APadrao.x[18] :=
IfThen(DataModule1.TableConsulta.FieldByName('RIGIDEZ_MATINAL').AsBoolean, 1, 0);
APadrao.x[19] := IfThen(DataModule1.TableConsulta.FieldByName('SACROILEITE').AsBoolean, 1,
0);
  if DataModule1.TableConsulta.FieldByName('I_INI_DOENCA_ATE_25_ANOS').AsBoolean then
begin
  APadrao.x[20] := 0;
  APadrao.x[21] := 0;
end
else
  if DataModule1.TableConsulta.FieldByName('I_INI_DOENCA_26_45_ANOS').AsBoolean then
begin
  APadrao.x[20] := 0;
  APadrao.x[21] := 1;
end
else begin
  APadrao.x[20] := 1;

```

```

    APadrao.x[21] := 0;
end;
APadrao.x[22] := IfThen(DataModule1.TableConsulta.FieldByName('SEXO_MASC').AsBoolean, 1,
0);
SetLength(APadrao.y, 3);
case DataModule1.TableConsulta.FieldByName('COD_DIAG').AsInteger of
1: begin
    APadrao.y[0] := 1;
    APadrao.y[1] := 0;
    APadrao.y[2] := 0;
end;
2: begin
    APadrao.y[0] := 0;
    APadrao.y[1] := 1;
    APadrao.y[2] := 0;
end;
3: begin
    APadrao.y[0] := 1;
    APadrao.y[1] := 1;
    APadrao.y[2] := 0;
end;
4: begin
    APadrao.y[0] := 0;
    APadrao.y[1] := 0;
    APadrao.y[2] := 1;
end;
5: begin
    APadrao.y[0] := 1;
    APadrao.y[1] := 0;
    APadrao.y[2] := 1;
end;
else
    APadrao.y[0] := 1;
    APadrao.y[1] := 1;
    APadrao.y[2] := 1;
end;
SetLength(APadrao, Length(APadrao) + 1);
APadrao[Length(APadrao) - 1] := APadrao;
end;
Result := APadrao;
end;

procedure TFormDiagnostico.Button1Click(Sender: TObject);
var
    APadrao: TPadrao;
    ADiag: TSinal;
    i: Integer;
    ADiagNum: Integer;
begin
    SetLength(APadrao.x, 23);
    APadrao.x[0] :=
IfThen(DataModule1.TableConsulta.FieldByName('AFTAS_ORAL_GENITAL').AsBoolean, 1, 0);
    APadrao.x[1] :=
IfThen(DataModule1.TableConsulta.FieldByName('ARTRITE_MAOS_PUNHOS').AsBoolean, 1, 0);
    APadrao.x[2] :=
IfThen(DataModule1.TableConsulta.FieldByName('DORES_MOVIMENTOS').AsBoolean, 1, 0);
    APadrao.x[3] := IfThen(DataModule1.TableConsulta.FieldByName('DORES_REPOUSO').AsBoolean,
1, 0);
    APadrao.x[4] := IfThen(DataModule1.TableConsulta.FieldByName('ENTESOPATIA').AsBoolean, 1,
0);

```

```

APadrao.x[5] := IfThen(DataModule1.TableConsulta.FieldByName('EROSOES_AO_RX').AsBoolean,
1, 0);
APadrao.x[6] :=
IfThen(DataModule1.TableConsulta.FieldByName('ESCLERITE_OU_OVEITE').AsBoolean, 1, 0);
APadrao.x[7] :=
IfThen(DataModule1.TableConsulta.FieldByName('F_REUMATOIDE_MAIOR_80').AsBoolean, 1, 0);
APadrao.x[8] :=
IfThen(DataModule1.TableConsulta.FieldByName('LOMBALGIA_CRONICA').AsBoolean, 1, 0);
APadrao.x[9] :=
IfThen(DataModule1.TableConsulta.FieldByName('NODULOS_HEBERDEN_BURCHAT').AsBoolean,
1, 0);
APadrao.x[10] :=
IfThen(DataModule1.TableConsulta.FieldByName('OLIGO_ARTRALGIA').AsBoolean, 1, 0);
APadrao.x[11] := IfThen(DataModule1.TableConsulta.FieldByName('OSTEOFITOS').AsBoolean, 1,
0);
APadrao.x[12] :=
IfThen(DataModule1.TableConsulta.FieldByName('OSTEOPOROSE_VISIVEL_RX').AsBoolean, 1, 0);
APadrao.x[13] := IfThen(DataModule1.TableConsulta.FieldByName('POLIARTRALGIA').AsBoolean,
1, 0);
APadrao.x[14] :=
IfThen(DataModule1.TableConsulta.FieldByName('POLIARTRITE_6_SEMANAS').AsBoolean, 1, 0);
APadrao.x[15] :=
IfThen(DataModule1.TableConsulta.FieldByName('PROVA_PATERGIA_POSITIVA').AsBoolean, 1,
0);
APadrao.x[16] :=
IfThen(DataModule1.TableConsulta.FieldByName('PUSTULAS_CUTANEAS').AsBoolean, 1, 0);
APadrao.x[17] :=
IfThen(DataModule1.TableConsulta.FieldByName('RIGIDEZ_ESQ_AXIAL').AsBoolean, 1, 0);
APadrao.x[18] :=
IfThen(DataModule1.TableConsulta.FieldByName('RIGIDEZ_MATINAL').AsBoolean, 1, 0);
APadrao.x[19] := IfThen(DataModule1.TableConsulta.FieldByName('SACROILEITE').AsBoolean, 1,
0);
if DataModule1.TableConsulta.FieldByName('I_INI_DOENCA_ATE_25_ANOS').AsBoolean then
begin
APadrao.x[20] := 0;
APadrao.x[21] := 0;
end
else
if DataModule1.TableConsulta.FieldByName('I_INI_DOENCA_26_45_ANOS').AsBoolean then
begin
APadrao.x[20] := 0;
APadrao.x[21] := 1;
end
else begin
APadrao.x[20] := 1;
APadrao.x[21] := 0;
end;
APadrao.x[22] := IfThen(DataModule1.TableConsulta.FieldByName('SEXO_MASC').AsBoolean, 1,
0);
ADiag := FRede.testar(APadrao);
eDiagnostico.Text := "";
ADiagNum := ROUND(aDiag[0]) + (ROUND(aDiag[1]) * 2) + (ROUND(aDiag[2]) * 4);
case ADiagNum of
1: eDiagnostico.Text := 'Artrite Reumatóide';
2: eDiagnostico.Text := 'Espondilite Anquilosante';
3: eDiagnostico.Text := 'Osteoartrose';
4: eDiagnostico.Text := 'Doença de Beçet';
5: eDiagnostico.Text := 'Outras Doenças';
else
eDiagnostico.Text := 'Indefinido';
end;
end;

```

```

end;

end.

//Rede Neural

unit Urede;

interface

uses Math, ComCtrls, Dialogs, SysUtils;

type
  TSinal = array of Real;
  TSinais = array of TSinal;
  TPadrao = record
    x: TSinal;
    y: TSinal;
  end;
  TPadroes = array of TPadrao;

type
  TNeuronio = class
  private
  protected
    FErro, FSaida, FNet: Real;
  public
    function Limiar: Real;
    function LimiarDerivada: Real;
    property Erro: Real read FErro write FErro;
    property Net: Real read FNet write FNet;
    property Saida: Real read FSaida write FSaida;
  end;

  TAxonio = class
  protected
    FPeso: Real;
    FNeuronioOrigem,
    FNeuronioDestino: TNeuronio;
  public
    procedure AtribuiPesoRandomico;
    procedure Liga(POrigem, PDestino: TNeuronio);
    property Destino: TNeuronio read FNeuronioDestino;
    property Origem: TNeuronio read FNeuronioOrigem;
    property Peso: Real read FPeso write FPeso;
  end;

  TNeuronios = array of TNeuronio;
  TAxonios = array of TAxonio;

  TRede = class
  private
  protected
    FCamadaIn,
    FCamadaHidden,
    FCamadaOut: TNeuronios;
    FAxoniosInHidden,
    FAxoniosHiddenOut: TAxonios;
    FTaxa: Real;
    procedure MontaRede(PPadrao: TPadrao);
    Procedure IncErro(PErro, PIncremento: TSinal);

```

```

function Epoca(PPadros: TPadros): TSinal;
function ErroSaida: TSinal;
procedure AtribuiPadraoEntrada(PPadrao: TPadrao);
procedure AtribuiErroSaida(PPadrao: TPadrao);
procedure ZeraNetErro;
procedure FeedForward(POrigem, PDestino: TNeuronios;
  PAxonios: TAxonios);
procedure BackPropagationOutHidden;
procedure BackPropagationHiddenIn;
public
destructor destroy; override;
function treinar(PPadros: TPadros; PEpocas: Integer;
  PBar: TProgressBar): TSinai;
function testar(PPadrao: TPadrao): TSinal;
property TaxaAprendizagem: Real read FTaxa write FTaxa;
end;

implementation

{ TAxonio }

procedure TAxonio.AtribuiPesoRandomico;
begin
  Randomize;
  FPeso := Random;
end;

procedure TAxonio.Liga(POrigem, PDestino: TNeuronio);
begin
  FNeuronioOrigem := POrigem;
  FNeuronioDestino := PDestino;
end;

{ TNeuronio }

function TNeuronio.Limiar: Real;
begin
  Result := (1 - Exp(-Net)) / (1 + Exp(-Net));
  // Result := 1 / (1 + Exp(-Net));
end;

function TNeuronio.LimiarDerivada: Real;
begin
  Result := 1 - ((1 - Exp(-Net)) / (1 + Exp(-Net)) *
    (1 - Exp(-Net)) / (1 + Exp(-Net)));
  // Result := Limiar * (1 - Limiar);
end;

{ TRede }

procedure TRede.AtribuiErroSaida(PPadrao: TPadrao);
var
  i: Integer;
begin
  for i:= 0 to Length(FCamadaOut) - 1 do begin
    FCamadaOut[i].Erro := (PPadrao.y[i] - FCamadaOut[i].Saida);
  end;
end;

procedure TRede.AtribuiPadraoEntrada(PPadrao: TPadrao);
var

```



```

i: Integer;
begin
  for i:= 0 to Length(FCamadaIn) - 1 do begin
    FCamadaIn[i].Saida := PPadrao.x[i];
  end;
end;

function TRede.ErroSaida: TSinal;
var
  i: Integer;
begin
  SetLength(Result, Length(FCamadaOut));
  for i:= 0 to Length(FCamadaOut) - 1 do begin
    Result[i] := fCamadaOut[i].Erro;
  end;
end;

procedure TRede.FeedForward(POrigem, PDestino: TNeuronios;
  PAxonios: TAxonios);
var
  i, ANumDes, ANumAxo : Integer;
begin
  ANumDes := Length(PDestino);
  ANumAxo := Length(PAxonios);
  for i:= 0 to ANumAxo - 1 do begin
    PAxonios[i].Destino.Net := PAxonios[i].Destino.Net +
      (PAxonios[i].Origem.Saida * PAxonios[i].Peso);
  end;
  for i := 0 to ANumDes - 1 do begin
    PDestino[i].Saida := PDestino[i].Limiar;
  end;
end;

function TRede.Epoca(PPadros: TPadros): TSinal;
var
  i: Integer;
begin
  SetLength(Result, Length(FCamadaOut));
  for i := 0 to Length(Result) - 1 do begin
    Result[i] := 0;
  end;
  for i:= 0 to Length(PPadros) - 1 do begin
    AtribuiPadraoEntrada(PPadros[i]);
    ZeraNetErro;
    FeedForward(FCamadaIn, FCamadaHidden, FAxoniosInHidden);
    FeedForward(FCamadaHidden, FCamadaOut, FAxoniosHiddenOut);
    AtribuiErroSaida(PPadros[i]);
    IncErro(Result, ErroSaida);
    BackPropagationOutHidden;
    BackPropagationHiddenIn;
  end;
  for i := 0 to Length(Result) - 1 do begin
    Result[i] := Result[i] / Length(PPadros);
  end;
end;

procedure TRede.IncErro(PErro, PIncremento: TSinal);
var
  i: Integer;
begin
  for i:= 0 to Length(PErro) - 1 do begin

```

```

    PErro[i] := PErro[i] + (PIncremento[i] * PIncremento[i]);
end;
end;

procedure TRede.MontaRede(PPadrao: TPadrao);
var
    i, j, ANumIn, ANumHid, ANumOut: Integer;
begin
    ANumIn := Length(PPadrao.x);
    ANumOut := Length(PPadrao.y);
    ANumHid := 4;
    //monta camada in
    SetLength(FCamadaIn, ANumIn);
    for i := 0 to ANumIn - 1 do begin
        FCamadaIn[i] := TNeuronio.Create;
    end;
    //monta camada hidden
    SetLength(FCamadaHidden, ANumHid);
    SetLength(FAxoniosInHidden, 0);
    for i := 0 to ANumHid - 1 do begin
        FCamadaHidden[i] := TNeuronio.Create;
        for j := 0 to ANumIn - 1 do begin
            SetLength(FAxoniosInHidden, Length(FAxoniosInHidden) + 1);
            FAxoniosInHidden[Length(FAxoniosInHidden) - 1] := TAxonio.Create;
            FAxoniosInHidden[Length(FAxoniosInHidden) - 1].Liga(fCamadaIn[j], fCamadaHidden[i]);
            FAxoniosInHidden[Length(FAxoniosInHidden) - 1].AtribuiPesoRandomico;
        end;
    end;
    //monta camada out
    SetLength(FCamadaOut, ANumOut);
    SetLength(FAxoniosHiddenOut, 0);
    for i := 0 to ANumOut - 1 do begin
        FCamadaOut[i] := TNeuronio.Create;
        for j := 0 to ANumHid - 1 do begin
            SetLength(FAxoniosHiddenOut, Length(FAxoniosHiddenOut) + 1);
            FAxoniosHiddenOut[Length(FAxoniosHiddenOut) - 1] := TAxonio.Create;
            FAxoniosHiddenOut[Length(FAxoniosHiddenOut) - 1].Liga(fCamadaHidden[j], fCamadaOut[i]);
            FAxoniosHiddenOut[Length(FAxoniosHiddenOut) - 1].AtribuiPesoRandomico;
        end;
    end;
end;

function TRede.testar(PPadrao: TPadrao): TSinal;
var
    i, j: Integer;
begin
    SetLength(Result, Length(FCamadaOut));
    AtribuiPadraoEntrada(PPadrao);
    ZeraNetErro;
    FeedForward(FCamadaIn, FCamadaHidden, FAxoniosInHidden);
    FeedForward(FCamadaHidden, FCamadaOut, FAxoniosHiddenOut);
    for i := 0 to Length(Result) - 1 do begin
        Result[i] := (fCamadaOut[i]).Saida;
    end;
end;

function TRede.treinar(PPadros: TPadros; PEpocas: Integer;
    PBar: TProgressBar): TSinais;
var
    i : Integer;
    AErroAcumulado: TSinal;

```

```

begin
  SetLength(Result, PEpocas);
  MontaRede(PPadros[0]);
  PBar.Max := PEpocas;
  PBar.Position := 0;
  for i := 0 to PEpocas - 1 do begin
    AErroAcumulado := Epoca(PPadros);
    Result[i] := AErroAcumulado;
    if (Result[i][0] < 0.0015) and
      (Result[i][1] < 0.0015) and
      (Result[i][2] < 0.0015) then begin
      showMessage('Rede treinada em: ' + IntToStr(i));
      Break;
    end;
    PBar.Position := i;
    PBar.Refresh;
  end;
end;

destructor TRede.destroy;
var
  i: Integer;
begin
  for i := 0 to Length(FCamadaIn) - 1 do begin
    FCamadaIn[i].Free;
  end;
  for i := 0 to Length(FCamadaHidden) - 1 do begin
    FCamadaHidden[i].Free;
  end;
  for i := 0 to Length(FCamadaOut) - 1 do begin
    FCamadaOut[i].Free;
  end;
  for i := 0 to Length(FAxoniosInHidden) - 1 do begin
    FAxoniosInHidden[i].Free;
  end;
  for i := 0 to Length(FAxoniosHiddenOut) - 1 do begin
    FAxoniosHiddenOut[i].Free;
  end;
  inherited;
end;

procedure TRede.BackPropagationHiddenIn;
var
  i, ANumAxi: Integer;
  AErroPropag: Real;
begin
  ANumAxi := Length(FAxoniosInHidden);
  for i := 0 to ANumAxi - 1 do begin
    AErroPropag := FAxoniosInHidden[i].Destino.Erro *
      FAxoniosInHidden[i].Destino.LimiarDerivada;
    FAxoniosInHidden[i].Peso := FAxoniosInHidden[i].Peso +
      (TaxaAprendizagem * AErroPropag * FAxoniosInHidden[i].Origem.Saida);
  end;
end;

procedure TRede.BackPropagationOutHidden;
var
  i, ANumAxi: Integer;
  AErroPropag: Real;
begin
  ANumAxi := Length(FAxoniosHiddenOut);

```

```

for i := 0 to ANumAxi - 1 do begin
  AErroPropag := (FAxoniosHiddenOut[i].Destino.LimiarDerivada *
    FAXoniosHiddenOut[i].Destino.Erro);
  FAXoniosHiddenOut[i].Origem.Erro := FAXoniosHiddenOut[i].Origem.Erro +
    (AErroPropag * FAXoniosHiddenOut[i].Peso);
  FAXoniosHiddenOut[i].Peso := FAXoniosHiddenOut[i].Peso +
    (TaxaAprendizagem * AErroPropag * FAXoniosHiddenOut[i].Origem.Saida);
end;
end;

procedure TRede.ZeraNetErro;
var
  i: Integer;
begin
  for i := 0 to Length(FCamadaIn) - 1 do begin
    FCamadaIn[i].Net := 0;
    FCamadaIn[i].Erro := 0;
  end;
  for i := 0 to Length(FCamadaHidden) - 1 do begin
    FCamadaHidden[i].Net := 0;
    FCamadaHidden[i].Erro := 0;
  end;
  for i := 0 to Length(FCamadaOut) - 1 do begin
    FCamadaOut[i].Net := 0;
    FCamadaOut[i].Erro := 0;
  end;
end;
end.

```

9.6. Anexo 6 - Artigo

O USO DE REDES NEURAIS ARTIFICIAIS NO DIAGNÓSTICO DE DOENÇAS REUMATOLÓGICAS

Cristiano Gonçalves e Souza

Lelia de Mello Tostes e Souza

{csouza, [lelia](mailto:lelia@inf.ufsc.br)}@inf.ufsc.br

Universidade Federal de Santa Catarina

1. Introdução

Vam Bommel (MASSAD, 2003) classifica o conhecimento médico em duas categorias básicas.

O primeiro tipo de conhecimento médico baseia-se na literatura médica. O segundo tipo, o experiencial deriva de informações constantes de bases de dados dos pacientes, tais como o prontuário médico, ou em protocolos clínicos de procedimentos.

A partir destes surge um terceiro tipo de conhecimento que deriva dos dois primeiros, ou seja, o conhecimento baseado em evidências disponíveis nas bases de conhecimento médico universal. Difere do primeiro nas evidências acumuladas ao longo do tempo e sumarizadas pelos processos adequados.

A partir destes surge um terceiro tipo de conhecimento que deriva dos dois primeiros, ou seja, o conhecimento baseado em evidências disponíveis nas bases de conhecimento médico universal. Difere do primeiro nas evidências acumuladas ao longo do tempo e sumarizadas pelos processos adequados.

Contudo difere do segundo tipo ao passo que o peso da experiência do especialista individualmente é minimizado e a decisão se baseia na associação entre as observações primárias e os resultados de certas decisões.

Tendo em vista os pontos levantados anteriormente, viemos propor o estudo do processo de descoberta do conhecimento em base de dados, contudo o objetivo deste não se reduz ao conhecimento médico *me si*.

Este estudo trata da aplicação de recursos tecnológicos de inteligência artificial, em específico, redes neurais artificiais, aplicado à ciência médica no diagnóstico médico baseado no conhecimento experiencial de um conjunto de reumatologistas, registrado em prontuários de pacientes, os quais colaboraram com o presente estudo.

O principal objetivo do presente estudo é propor e avaliar modelos de redes neurais artificiais para o diagnóstico de doenças reumáticas.

Os objetivos específicos são: Estudar e analisar os algoritmos de treinamento para o

desenvolvimento de uma Rede Neural Artificial (RNA); comparar o desempenho de redes de Kohonen com MLP utilizando ferramenta existente com suporte ao uso de RNAs; desenvolver uma RNA para o diagnóstico de doenças reumatológicas; planejar e realizar a coleta de dados de consultas para avaliar os padrões que levam a determinação de um diagnóstico médico; utilizando casos reais, avaliar o desempenho da RNA desenvolvida.

2. Redes Neurais Artificiais

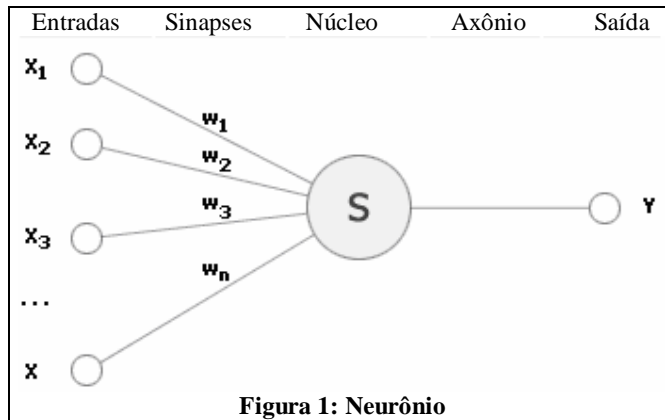
A construção de Redes Neurais Artificiais (RNA) tem inspiração nos neurônios biológicos e nos sistemas nervosos. Porém, as semelhanças entre as RNA e as Redes Neurais Naturais (RNN) são pequenas.

O primeiro modelo de neurônio, foi proposto por McCulloch e Pitts em 1943, e tratava-se de um modelo simples, já que a idéia era imitar a realidade biológica. Porém, essa idéia não é utilizada pelos pesquisadores atuais. Atualmente, dois fatores motivam a pesquisa:

- Modelar o sistema nervoso com suficiente precisão de tal modo a poder observar um comportamento emergente que, sendo semelhante ao comportamento do ser vivo modelado, possa servir às hipóteses usadas na modelagem;

- Construir computadores com alto grau de paralelismo.

A figura 1 mostra um neurônio artificial que, de forma semelhante ao natural, consiste de sinapses conectando as entradas do neurônio com o núcleo; o núcleo do neurônio processa os sinais de entrada e possui axônios que o conecta com os neurônios da próxima camada. Cada sinapse tem seu próprio peso, o qual determina como a entrada de um determinado neurônio influencia na sua condição.



2.1. Perceptron

O Perceptron se constitui num dispositivo com três tipos de unidades: sensoriais, associativas e unidades de resposta. Este tipo de RNA pode ter diversas camadas escondidas, no entanto com apenas uma camada adaptativa, onde os pesos podem ser modificados por um algoritmo de aprendizado.

2.2. Mapas de Kohonen – Redes SOM (Self-Organizing Maps)

Em 1984 Teuvo Kohonen desenvolveu uma família de redes com a capacidade de reconhecer relações entre padrões apresentados à entrada da mesma, após um treinamento não-supervisionado competitivo.

As redes SOM seguem a arquitetura de aprendizado competitivo, onde além de os neurônios de uma camada se conectarem com os da próxima camada; existem conexões entre unidades da mesma camada.

O algoritmo de treinamento das redes SOM segue a arquitetura de aprendizado onde os neurônios de camadas diferentes se comunicarem por conexões excitatórias e neurônios de uma mesma camada por conexões inibitórias.

O algoritmo *winner-takes-all* (o vencedor leva tudo) é comumente usado como algoritmo de treinamento da camada competitiva da rede de Kohonen. Ele inicialmente atribui pesos aleatórios às conexões entre os neurônios das camadas de entrada e saída, onde a soma dos quadrados dos valores dos pesos das conexões de entrada é idêntica para cada neurônio de saída.

2.3. Redes Multi-Layer Perceptron (MLP) treinadas com Retropropagação

As redes MLP treinadas com retropropagação são ferramentas poderosas na

busca de regras, predição e análise qualitativa. Elas são chamadas de redes de retropropagação por causa do algoritmo de aprendizado que estas usam, na qual um erro se move da camada de saída para a camada de entrada, ou seja, na direção contrária ao sinal original durante a operação normal da rede (STARIKOV, 2001).

As redes MLP consistem de uma série de camadas, onde cada neurônio de uma camada i está conectado com cada neurônio da camada $i + 1$, portanto uma rede completamente conectada.

2.4. Tipos de Aprendizado de Redes Neurais

O treinamento supervisionado necessita de um par de vetores composto do vetor de entrada e do vetor alvo que se deseja como saída. Juntos, estes vetores são chamados de par de treinamento ou vetor de treinamento, sendo interessante ressaltar que geralmente a rede é treinada com vários vetores de treinamento.

O procedimento de treinamento funciona da seguinte forma: o vetor de entrada é aplicado. A saída da rede é calculada e comparada com o correspondente vetor alvo. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo determinado, visando minimizar este erro. Este processo de treinamento é repetido até que o erro para os vetores de treinamento tenha alcançado níveis bem baixos.

O treinamento não-supervisionado, por sua vez, não requer vetor alvo para as saídas e, obviamente, não faz comparações para determinar a resposta ideal. O conjunto de treinamento modifica os pesos da rede de forma a produzir saídas que sejam consistentes, isto é, tanto a apresentação de um dos vetores de treinamento, como a apresentação de um vetor que é suficientemente similar, irá produzir o mesmo padrão nas saídas. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores

similares em classes. A aplicação de um vetor de uma determinada classe à entrada da rede irá produzir um vetor de saída específico, mas não existe maneira de se determinar, antes do treinamento, qual o padrão que será produzido na saída para um vetor de entrada de uma determinada classe. Desta forma, a saída de algumas RNAs deve ser transformada em uma forma compreensiva após o processo de treinamento, o que é um simples problema de identificação das relações entrada-saída estabelecidas pela rede.

No que diz respeito aos algoritmos de treinamento usados, existe uma grande variedade, tanto para o treinamento supervisionado, como para o não-supervisionado.

Muitos deles baseiam-se no algoritmo proposto por D. O. Hebb em 1961. Entretanto, os algoritmos mais eficientes foram desenvolvidos nos últimos trinta anos, baseados nos trabalhos de Rosenblatt (1962), Windrow (1959) e Windrow e Hoff (1960). Entre estes, o mais difundido com certeza foi o algoritmo de *retropropagação* ("Backpropagation") (ABELÉM, 1994).

3. O Diagnóstico Médico

O Dicionário Médico Dorland define diagnóstico ou diagnose como sendo a arte de distinguir uma doença de outra e ainda, a determinação da causa de uma doença. Para o Oxford Medical Companion, diagnose é o processo de identificação de uma doença ou outras circunstâncias responsáveis pelas queixas de um paciente, ou seja, de sua doença.

No presente estudo visamos propor um modelo, com o uso de redes neurais, para categorizar um conjunto de pacientes em um quadro diagnóstico provável. Por motivos de simplicidade na implementação desta tarefa, nos restringiremos a um subconjunto de diagnósticos da área de conhecimento da medicina denominada reumatologia. São estas doenças: Artrite Reumatóide, Osteoartrose, Gota e Fibromialgia.

4. Metodologia e Implementação

Através de algoritmos de aprendizagem o computador pode apresentar um comportamento inteligente no reconhecimento de padrões.

O uso destes algoritmos na área da saúde não é uma exceção; usado desde a locação de recursos hospitalares até a classificação de crises convulsivas ou ainda no diagnóstico médico de doenças.

Demonstraremos agora como foi realizado o presente estudo; desde a coleta de dados até a apresentação de um novo padrão de teste a uma

RNA treinada. Os padrões serão compostos de um quadro sintomático, resultados de exames e dados de pacientes para o diagnóstico de doenças reumáticas.

O projeto foi dividido em várias etapas, como segue, visando dividir o problema em partes distintas, cada qual com sua devida importância e necessidade.

4.1 Coleta de Dados

Os dados médicos, sejam eles gerados através de entrevista com o paciente, ou de pesquisas de natureza clínica, epidemiológica ou fisiopatológica, só contribuem para o processo de conhecimento se organizados e interpretados adequadamente, gerando, desta forma, informação.

O prontuário médico do paciente desempenha papel essencial na geração do conhecimento médico. O conhecimento médico obtido de um paciente individual no momento de sua entrevista, consiste fundamentalmente, na etapa primordial do processo de tomada de decisão, inicialmente na elaboração de um diagnóstico, seguido da proposta terapêutica e do prognóstico do paciente (MASSAD).

Fundamentando-se nesta idéia, os dados utilizados foram coletados através da aplicação de um questionário com questões abertas sobre dados do prontuário médico referentes a consultas de pacientes com um diagnóstico conclusivo. O modelo do questionário se encontra no Anexo 1.

A coleta dos dados foi realizada entre abril e julho de 2004. Para tal foram distribuídos os questionários entre um grupo de médicos reumatologistas, as quais foram preenchidas e posteriormente analisadas pelo co-orientador do presente estudo e também reumatologista, Vidal de Souza.

4.2 Pré-processamento, Limpeza e Tratamento dos Dados

Inicialmente foi gerada uma tabela com todos os registros encontrados no banco de dados. A partir desta, foi necessária uma preparação dos dados, que consistiu na escolha dos diagnósticos alvos (tabela 1), baseado no número de ocorrências destes; bem como a escolha dos dados do prontuário médico associados a essas doenças.

DIAGNÓSTICO	OCORRÊNCIA
Artrite Reumatóide	8
Osteoartrose	6
Gota	6
Fibromialgia	6
Outras Doenças	26

Tabela 1: Diagnósticos

Como podemos observar apenas quatro diagnósticos possuem mais de três registros em nossa base de dados, portanto os escolhidos foram: Artrite Reumatóide, Osteoartrite, Gota e Fibromialgia.

Os sintomas e resultados de exames relativos a cada uma das consultas escolhidas, associadas a cada diagnóstico, foram selecionados como dados de entrada para a rede neural, bem como a idade e o sexo do paciente.

4.3 Escolha da Rede Neural

Vários estudos tratam o problema do diagnóstico médico através do uso de métodos de agrupamento e classificação. Visando observar o comportamento de redes neurais para esta tarefa, foram selecionadas duas possíveis abordagens: redes de Kohonen e redes MLP treinadas com retropropagação.

O tempo gasto com treinamento na Rede de Kohonen foi menor que o apresentado pela rede MLP treinada com retropropagação.

Contudo, esta não indicava claramente quais neurônios de saída estavam relacionados a cada diagnóstico. Além disso, a rede sempre categorizava o padrão de entrada, mesmo sendo ele um paciente saudável, ou com outra doença desconhecida por ela.

Em casos onde o padrão de entrada era de um paciente com doenças associadas (gota e artrite reumatóide, por exemplo), a rede somente categorizava-o em no máximo um dos diagnósticos corretos.

Os dados coletados apresentavam o diagnóstico do paciente, e esta informação era desprezada, pelo fato das redes de Kohonen utilizarem um método de treinamento não-supervisionado.

Já com as redes MLP, o tempo gasto com treinamento foi muito grande pois é processo de atualização dos pesos é muito custoso. Em casos com muitos neurônios, ou ainda muitos padrões de treinamento, o custo pode ser proibitivo.

Porém, como cada saída da rede corresponde a um diagnóstico, fica fácil interpretar sua saída. Para um paciente saudável, bem como para doenças não treinadas pela rede, os valores para todas as saídas se distanciavam de 1, ou seja, negavam os diagnósticos.

Ainda para o caso de doenças associadas, o resultado apresentava os múltiplos diagnósticos do paciente.

Dadas as diferenças entre os dois modelos apresentados, optamos por implementar uma rede MLP treinada com retropropagação e analisar o seu desempenho.

4.4 Ferramenta Desenvolvida

Foi desenvolvido um programa para o Diagnóstico Reumatológico, o qual

chamaremos de *Diagnoser*, que tem por objetivo treinar uma rede neural baseado em consultas médicas fornecidas por médicos reumatologistas, conforme visto anteriormente.

O sistema possui uma tela de configuração de parâmetros para treinamento da rede. Nesta pode ser informado o número de registros que serão utilizados para o treinamento, permitindo assim que se possa reservar uma parte para testes.

Definimos o número máximo de épocas a ser usada, bem como se a parada será dada por atingir o número de épocas máximo, ou quando o erro acumulado em cada saída chegar ao valor proposto pelo usuário.

A quantidade de neurônios da camada *Hidden* (escondida) também é definida pelo usuário.

Outros valores que o usuário pode configurar são: a taxa de aprendizagem; o limite inferior e superior – saídas abaixo do limite inferior não são consideradas diagnósticos válidos, abaixo do limite superior são considerados diagnósticos possíveis, e acima disto, prováveis; a função de saída (*threshold*) usada para o treinamento; o tratamento da idade - normalizada ou por faixas.

No final do treinamento, será apresentado o número de épocas efetivamente executado, o erro acumulado por época, por neurônio de saída e um gráfico com a evolução do treinamento (soma dos quadrados dos erros por número de épocas).

5. Análise dos Resultados

Para a análise que se segue, foram coletados 28 prontuários relativos a consultas médicas. Este conjunto de dados foi dividido em duas partes: um subconjunto de dados para testes e outro subconjunto para a validação da rede.

Cada teste consistiu na aplicação de 13 consultas médicas a uma rede treinada, onde foram avaliados os diagnósticos propostos pela rede. Durante os testes foram selecionados os parâmetros de configuração para criar as RNAs a serem verificadas na validação. A validação consistirá no treinamento de 10 RNAs, cuja configuração segue a proposta apresentada durante os testes. Ao final de cada treinamento, serão aplicadas as 15 consultas médicas restantes a cada RNA, e então verificado seu desempenho.

Para ambos os casos, a aplicação de um padrão (uma consulta médica específica) à rede resultará numa saída, a qual estará classificada numa das três categorias seguintes:

- Diagnóstico provável - acima do limite superior;
- Diagnóstico possível - entre o limite inferior e superior;

- Diagnóstico descartado – abaixo do limite inferior.

A rede acerta o diagnóstico quando responde afirmativamente às questões que se seguem:

- Quando o diagnóstico era único considerou como provável o diagnóstico correto?

- Quando haviam diagnósticos associados, considerou como provável ou possível os diagnósticos corretos?

- Para todos os casos, descartou corretamente os diagnósticos adequados?

Após os testes realizados, obtivemos a seguinte configuração de rede para treinamento:

Configurações da Rede			
Parar quando o erro for aceitável	\bar{u}	Função Limiar	Sigmoidal
Erro Aceitável	0,00015	Tratamento da Idade	Normalizada
Taxa de aprendizagem	0,05	Faixa Limite 1ª Faixa	-
Neurônios <i>Hidden</i>	8	Faixa Limite 2ª Faixa	-

Tabela 2: Configurações da Rede após os Testes

Foi aplicado o teste *t* de Student (BARBETTA, 2002) para calcular os valores de corte com um grau de confiabilidade de 95% para que o valor fosse aceito quando superior ao limite superior ou inferior ao limite inferior respectivamente. O valor do limite superior O valor do limite superior encontrado foi 0,77 e inferior, 0,17.

Em seguida, foram aplicados os padrões de validação como entrada para a RNA proposta.

Os critérios de avaliação da saída foram os mesmos utilizados para o teste.

Para uma análise mais objetiva, considerou-se cada uma aplicação de um padrão a um treinamento específico da rede como um caso de validação. Desta forma, como temos 15 padrões aplicados a 10 treinamentos, temos 150 casos de validação distintos. Destes 150 casos de validação, 90 referem-se a consultas onde o médico chegou a um único diagnóstico; 30 onde o médico não chegou a nenhum diagnóstico dos previstos pela RNA e outros 30 casos onde o médico percebeu a presença de dois diagnósticos associados.

Ao avaliar os 90 casos de validação, onde o paciente apresentava apenas um dos diagnósticos conhecidos pela RNA proposta, verificou-se que a rede conseguiu predizer o diagnóstico correto para todos os casos.

Quando foi apresentado um padrão referente a um diagnóstico não conhecido pela rede, foi apontado que para 11 dos casos existiam diagnósticos válidos; sendo 6 deles de doenças associadas e 5 com um diagnóstico único. Todos apontavam para Fibromialgia, sendo que em dois deles, o único sintoma apresentado pelo paciente era dor generalizada, um sintoma importante para o diagnóstico de Fibromialgia e para os outros 9 casos, a confirmação da inexistência de atividade inflamatória ou de fator reumatóide positivo indicavam que dos quatro diagnósticos conhecidos pela rede, o mais provável seria o do Fibromialgia. Os 30 casos de validação restantes apresentaram 8

respostas diferentes do esperado. Apesar de não indicar a existência de uma associação de diagnósticos para estes 8 casos de validação, todos indicaram a existência de ao menos um dos diagnósticos corretos.

Considerando todos os casos de validação, obteve-se uma média geral de acerto de 87,33%.

6. Conclusão

O propósito principal do presente estudo – propor e avaliar modelos de redes neurais artificiais para o diagnóstico de doenças reumáticas – foi plenamente alcançado, o que levou a algumas considerações.

A definição do escopo do problema bem como a escolha do tratamento dos dados é de suma importância para a realização adequada do trabalho, já que permite focar no que realmente interessa para o estudo considerando que a área estudada é tão vasta e abrangente.

A escolha da topologia da rede também foi de grande importância para o sucesso do trabalho, visto que com a gama de redes existentes, foi necessário um estudo aprofundado para estabelecer a que melhor se encaixaria ao propósito do trabalho.

Os resultados obtidos mostraram que o modelo proposto apresentou-se bastante eficiente para a previsão do diagnóstico médico quando este for único, ou seja, não apresentar doenças associadas. A abordagem adotada não foi muito apropriada para esta abordagem bem como para o caso de diagnósticos não conclusivos.

Uma maneira de abranger estas situações, seria acrescentar aos dados de entrada, consultas com diagnósticos associados ou não conclusivos. Deste modo, a rede poderia aprender estes casos, e conseguiria prever resultados quando fosse lhe apresentadas situações semelhantes.

Avaliando o aspecto tecnológico conseguiu-se verificar que o desenvolvimento de sistemas

no auxílio à decisão baseado em redes neurais artificiais é uma alternativa com desempenho e confiabilidade aceitáveis, especialmente no caso de diagnóstico médico.

De um ângulo diferente, a medicina pode se beneficiar dos sistemas de informação, como os estudados aqui, inclusive como ferramenta de suporte em consultório e clínicas médicas auxiliando e reafirmando o diagnóstico.

Ainda assim existem aspectos que somente o ser humano é capaz de identificar, os quais não foram levados em conta pelas RNAs.

Portanto, tem-se que os sistemas de informação abordados existem para adicionar conhecimento à área da saúde e não substituir o médico.

7. Referências Bibliográficas

ABELÉM, Antônio Jorge Gomes. Redes Neurais Artificiais na Previsão de Séries Temporais. Dissertação de Mestrado, PUC – RJ, Dept^o de Engenharia Elétrica. Set/1994

AZEVEDO, Fernando M. et al. Redes neurais com aplicações em controle e sistemas especialistas. Visual Books, Ago/2000.

BARBETTA, Pedro A. Estatística aplicada às ciências sociais. 5ª edição revisada, Editora da UFSC, 2002

CARVALHO, Lucimar F. et al. Redes neurais artificiais auto-organizáveis utilizadas na classificação de crises convulsivas. Universidade de Passo Fundo/RS, 2002.

FALQUETO, Jovelino. Inspiração biológica em IA. Tese de Doutorado, UFSC, Dept^o de Informática e Estatística, Mar/2002

FELICE, José Carlos et al. Temas de Reumatologia Clínica – Ostoartrose (AO). CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=2031

FELLET, Aloysio J. Elementos Básicos do Diagnóstico de Gota. CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=1308

GUAHYBA, Adriano. Utilização de inteligência artificial (redes neurais artificiais) no gerenciamento de reprodutoras pesadas. Tese de Doutorado, PPGCV – UFRGS, 2001.

MASSAD, Marin. O prontuário eletrônico do paciente na assistência, informação e conhecimento médico. Azevedo Neto editores OPAS/OMS 2003

MOSIMANN, João P. Uma análise comparativa entre a utilização de Redes de Kohonen e o método PAM no agrupamento de dados para o auxílio do diagnóstico de doenças reumatológicas. UFSC, 2002

SCOTTON, Antônio Scafuto. Como diagnosticar e tratar fibromialgia. CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=583

SOUZA, Cristiano & SOUZA, Lelia. O Uso de Redes Neurais Artificiais no Diagnóstico de Doenças Reumatológicas. UFSC, 2004.

STARIKOV, Alexey. Neural networks - mathematical apparatus. BaseGroup Lab Data Analysis Technology, 2001.

TAVARES, Laura Nascimento et al. Elementos básicos de diagnóstico da doença (artrite) reumatóide. CiberSaude, 2004.

http://www.cibersaude.com.br/revistas.asp?fase=r003&id_materia=128