

UNIVERSIDADE FEDERAL DO PARANÁ

**SISTEMA DE APOIO À DECISÃO DE EXAMES LABORATORIAIS DE URINA
UTILIZANDO REDES NEURONAIS ARTIFICIAIS**

CURITIBA
Julho / 2010

RAPHAEL DE ASSUMPÇÃO JAMIELNIAK

**SISTEMA DE APOIO À DECISÃO DE EXAMES LABORATORIAIS DE URINA
UTILIZANDO REDES NEURONAIAS ARTIFICIAIS**

Trabalho apresentado à disciplina de Trabalho de Conclusão de Curso, do Módulo VI, diurno, do curso de Tecnologia em Sistemas de Informação da Universidade Federal do Paraná, como requisito para obtenção da conclusão de curso.

Orientador : Roberto Tadeu Raittz, Dr.
Coorientadores: Jeroniza Marchaukoski, Dra.
Waldemar Volanski, Mestrando

CURITIBA
Julho / 2010

RESUMO

A grande quantidade de exames de urina realizados em laboratório com o auxílio de equipamentos ainda exige um cansativo e repetitivo processo de classificação do resultado dos exames, feitos um a um manualmente por um especialista, o que pode ocasionar eventualmente erros na interpretação dos dados. Este projeto visa a construção de um software capaz de fornecer com o mínimo de erro uma sugestão de diagnóstico para a análise de exames bioquímicos de urina realizados em laboratórios de grande escala. Para isso foi desenvolvido um estudo e implementação de Inteligência Artificial através do uso de Redes Neurais Artificiais para o aprendizado e sugestão de diagnósticos destes exames, que além de não exigir intervenção do usuário do sistema, o auxilia na interpretação massiva de dados minimizando seus erros ocasionais.

Palavras-chave: mlp, inteligência artificial, exame de urina, sistema de apoio à
decisão

ABSTRACT

The large amount of urine testing in laboratory with the aid of equipment still requires a tedious and repetitive process of classifying the test results, made one by one manually by an expert, which can eventually lead to errors in interpreting the data. This project involves the construction of a software capable of providing with a minimum error a suggested diagnosis for the analysis of urine biochemical tests performed in laboratories in large scale. In order to was developed a study and implementation of Artificial Intelligence through the use of Artificial Neural Networks for learning and suggestion of this diagnostic tests, which does not require user intervention in the system, and assists in massive data interpretation, minimizing their occasional mistakes.

Key words: mlp, artificial intelligence, urine testing, decision support system

LISTA DE ABREVIATURAS

IA - Inteligência Artificial

MLP - MultiLayer Perceptron

RP - Reconhecimento de padrões

LMC - Laboratório Municipal de Curitiba

EAS - Elementos Anormais e Sedimentares

FAN - Free Associative Neurons

RNA - Rede Neuronal Artificial

DARPA - Defense Advanced Research Projects Agency

OCR - Optical Character Recognition

INNS - International Neural Networks Society

IEEE - Institute of Electrical and Electronics Engineers

SQL - Structured Query Language

RBF - Radial Basis Function

SVM - Support Vector Machine

UML - Unified Modeling Language

GPL - General Public License

USB - Universal Serial Bus

RAM - Random Access Memory

HD - Hard Disk

LISTA DE FIGURAS

Figura 1 - Esquema do Sistema	3
Figura 2 - Rede de perceptrons proposta por Rosenblatt	5
Figura 3 - Redes Adaline e Madaline	6
Figura 4 - Estrutura do método Backpropagation (retropropagação)	7
Figura 5 -Esquema das partes constituintes de uma célula neuronal	9
Figura 6 - Unidade Processadora (neurônio artificial) do tipo linear (Adaline).....	11
Figura 7 - Neurônio artificial com função de transferência não linear.....	12
Figura 8 - Rede to tipo MLP (Multi Layer Perceptron) de 5 camadas.....	13
Figura 9 - Tira Reagente	15
Figura 10 - Comparação da Tira Reagente	16
Figura 11 - Aparelho Urisys 2400 – Sysmex	22
Figura 12 - Aparelho UF-100 – Sysmex	22
Figura 13 - Gráfico do progresso de Treinamento das Bases 1.....	34
Figura 14 - Gráfico do progresso de Treinamento das Bases 2.....	36
Figura 15 - Gráfico do Tempo de Aprendizado X Épocas de Treinamento.....	37
Figura 16 - Projeto do Sistema de Apoio à Decisão	44
Figura 17 - Definir Campos SQL	44
Figura 18 - Carregamento de Bases	45
Figura 19 - Resultados de Teste de Rede	45
Figura 20 - Carregar de SQL	45
Figura 21 - Estados de BaseSQL	46
Figura 22 - Carregar Arquivo de Treino	47
Figura 23 - Carregar Arquivo de Teste	47
Figura 24 - Carregar Arquivo de Validação	48
Figura 25 - Carregar Arquivo de Classificação	48
Figura 26 - Treinamento da Rede	49
Figura 27 - Salvar Base Classificada em SQL	50
Figura 28 - Salvar Esquema.....	51
Figura 29 - Atividades: Salvar Rede	52
Figura 30 - Classificar Base de Classificação	53

SUMÁRIO

RESUMO	i
ABSTRACT	ii
LISTA DE ABREVIATURAS	iii
LISTA DE FIGURAS	iv
1 – INTRODUÇÃO	1
1.1 – JUSTIFICATIVA	2
1.2 - OBJETIVO	3
1.3 - OBJETIVOS ESPECÍFICOS	3
2 - REVISÃO BIBLIOGRÁFICA	4
2.1 – HISTÓRICO DAS RNA's.....	4
2.2 – O NEURÔNIO HUMANO	8
2.3 – REDES NEURONAIS	10
2.4 – REDE NEURONAL MLP	13
2.5– EXAME PARCIAL DE URINA	15
3 - MATERIAIS E MÉTODOS	23
3.1 – HARDWARE	23
3.2 – SOFTWARE	23
3.3 - REDES NEURONAIS	24
3.4 - PASSO A PASSO DO SISTEMA DE APOIO À DECISÃO.....	26
3.5- TREINAMENTO E TESTES	31
4 - RESULTADOS	34
5 - CONCLUSÕES	40
6 - PROJETOS FUTUROS	42
7 - DIAGRAMAS	43
7.1 - DIAGRAMAS DE CLASSES	43
7.2 - DIAGRAMA DE CASOS DE USO	44
7.3 - DIAGRAMA DE ESTADO	46
7.4 - DIAGRAMAS DE SEQUÊNCIA	47

7.5 - DIAGRAMAS DE ATIVIDADES	52
8 - DESCRIÇÃO DOS CASOS DE USO	54
8.1 - CARREGAR BASE DE TREINO	54
8.2 - CARREGAR BASE DE TESTE	55
8.3 - CARREGAR BASE DE VALIDAÇÃO	56
8.4 - CARREGAR BASE DE CLASSIFICAÇÃO	57
8.5 - CLASSIFICAR BASE DE CLASSIFICAÇÃO	58
8.6 - VER DESEMPENHO	59
8.7 - TREINAR REDE	60
8.8 - DEFINIR CAMADAS DE REDE	61
8.9 - CARREGAR REDE	62
8.10 - SALVAR REDE	63
8.11 - CARREGAR DE SQL	64
8.12 - SALVAR BASE CLASSIFICADA EM SQL	65
8.13 - DEFINIR CONEXÃO SQL	66
8.14 - DEFINIR CAMPOS SQL	67
8.15 - SALVAR ESQUEMA	68
8.16 - CARREGAR ESQUEMA	69
8.17 - CRIAR EXECUTÁVEL	70
9 - REFERÊNCIAS BIBLIOGRÁFICAS	72

1 - INTRODUÇÃO

A Inteligencia Artificial (IA) é uma área da informática que tem contribuído com importantes avanços em todas as áreas do conhecimento humano, especialmente com a popularização e o avanço tecnológico dos computadores, que tem sua capacidade de processamento com taxas de crescimento exponenciais. Ela vem sendo utilizada em diferentes tipos de aplicações, como jogos, diagnósticos, logística, robótica, reconhecimento de padrões, entre outras funções. PAULA(2000) e MEDEIROS(2006)

Uma das principais ferramentas utilizadas na área de reconhecimento de padrões (RP) são as RNAs. A MLP (MultiLayer Perceptron) é uma RNA que utiliza o conceito de multicamadas de neurônios e foi uma das primeiras redes a serem idealizadas, sendo desenvolvida e obtendo resultados notáveis. A partir de uma base de dados de treinamento é possível treinar essa RNA, então esta “aprende” a responder a novos problemas a partir de um “conhecimento” já obtido através de treinamento. MEDEIROS(2006)

A partir de uma base de dados de treinamento obtida dos resultados de exames de urina, pode submetê-la a treinamento uma RNA, que assim classificaria os resultados de novos exames e auxiliaria um especialista como um sistema de apoio à decisão, principalmente quando a quantidade de avaliações é grande e fica sujeita à erros por ser uma tarefa repetitiva e realizada manualmente.

O exame de urina pode detectar problemas no paciente como inflamações no sistema urinário, infecções, cálculo renal e glomerulonefrite(grupo de doenças renais causados por infecção e destruição gradual e progressiva das estruturas internas dos rins), problemas estes que diagnosticados com rapidez podem ser devidamente solucionados ou amenizados.

1.1 - JUSTIFICATIVA

Em seu trabalho de mestrado, Waldemar Volanski estuda diferentes maneiras de diagnósticos e classificação de exames de urina em grandes quantidades. Atualmente, no Laboratório Municipal de Curitiba, este trabalho é realizado manualmente, onde as amostras são inseridas nos equipamentos que transmitem os resultados para uma base de dados e estes são analisados por um especialista um a um para a verificação da necessidade de uma reanálise microscópica mais minuciosa quando o resultado está fora do normal. Devido à grande quantidade de resultados de exames a serem interpretados, eventuais erros podem ocorrer, ocasionando reanálises desnecessárias ou liberação de exames com resultados alterados sem a reanálise necessária.

Seu estudo compara análises realizadas de diferentes modos: por um analista clínico em diferentes condições de trabalho, por outro analista clínico, por um sistema baseado em regras e por dois sistemas utilizando RNAs, um utilizando uma RNA MLP (MultiLayer Perceptron) e outro utilizando uma RNA FAN (Free Associative Neurons). VOLANSKI(2010)

Por ser um trabalho manual e repetitivo, que ocorre através da análise dos resultados de esterase leucocitária, leucócitos, hemoglobina e eritrócitos, além do sexo e a idade do paciente, ficando então sujeito à erros devido à grande quantidade de exames avaliados, a necessidade de um sistema de apoio à decisão, que lhe forneceria uma sugestão de diagnóstico, seria de extremo auxílio, minimizando os erros e com isso diminuindo o trabalho de reanálise em alguns exames e a liberação equivocada sem a devida reanálise minuciosa em outros exames.

1.2 - OBJETIVO

Este projeto visa o desenvolvimento de um sistema computacional de apoio à decisão no processo de análises clínicas de exames de urina, utilizando técnicas de Reconhecimento de Padrões e Redes Neurais Artificiais.

1.3 - OBJETIVOS ESPECÍFICOS

- Treinar uma rede MLP com diferentes configurações por uma base de conhecimento - composta por uma quantidade considerável de amostras já analisadas e armazenadas em uma base de treinamento - e utilizar aquela que apresentar melhor desempenho nos testes e validações;

- Buscar em um banco de dados os resultados de novos exames bioquímicos de níveis de leucócitos e eritrócitos, submetê-los a uma classificação desta rede e devolver esta classificação ao banco de dados com a mínima intervenção externa, que funcionará como auxiliar de apoio à decisão ao analista clínico, como descrito no esquema da Figura 1.

- Criar um método para que o sistema faça automaticamente a classificação destes novos exames no banco de dados, sempre que solicitado e conforme suas configurações.

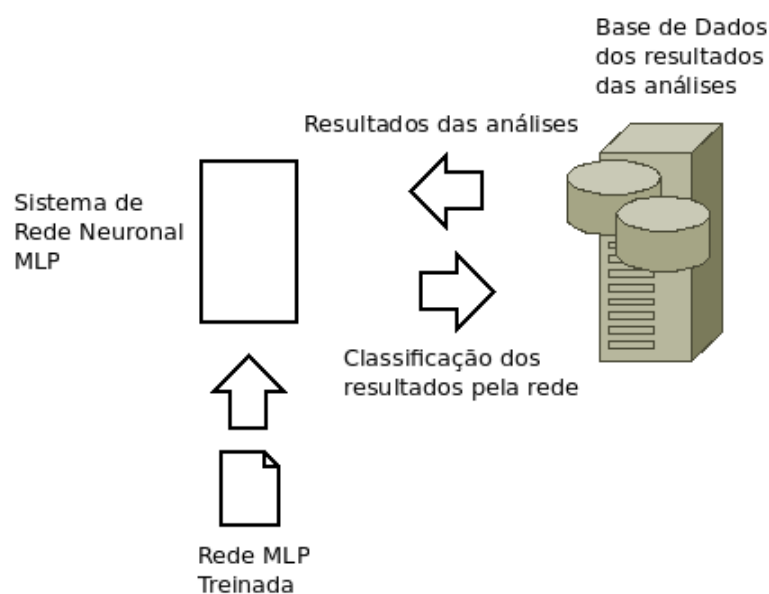


Figura 1 – Esquema do Sistema *fonte: própria*

2 - REVISÃO BIBLIOGRÁFICA

2.1 - Histórico

ZIPSER e RUMELHART(1985) citam a história das RNAs com o surgimento das RNA's na década de 40, mais precisamente em 1943, quando o neurofisiologista Warren McCulloch e o matemático Walter Pitts, da Universidade de Illinois, fizeram uma analogia entre as células nervosas e o processo eletrônico num artigo publicado no Bulletin of Mathematical Biophysics com o título: A Logical Calculus of the Ideas Immanent in Nervous Activity.

Em 1949, o biólogo e psicólogo HEBB (1949), que estudava o comportamento dos animais, escreveu o livro "The Organization of Behavior"(A Organização do Comportamento), que reforçava as teorias de que o condicionamento psicológico estava presente em qualquer parte dos animais, pelo fato de que esta é uma propriedade de neurônios individuais. As ideias de Hebb não eram pioneiras, mas ele propôs um princípio de aprendizado em sistemas nervosos complexos, ou seja, uma lei que descreve o funcionamento quantitativo da sinapse e do processo de treinamento humano.

Desde então, vários outros pesquisadores que entusiasmados com as novas descobertas, voltaram-se para esta linha de pesquisa.

Em 1951, Marvin Minsky, co-fundador do Laboratório de Inteligência Artificial do MIT, construiu o Snark, o primeiro simulador de cadeia neuronal. O Snark trabalhava com êxito e podia ajustar seus pesos sinápticos automaticamente. Ele nunca chegou a executar alguma função de processamento de informação interessante, servindo somente de fator motivador para ideias que surgiram posteriormente.

Em meados dos anos 50, a ideia de filtros lineares adaptativos usada em *processamento de sinais* é trabalhada por Gabor em 1954. O conceito de memória associativa iniciado por Taylor, em 1956, propôs novas ideias a respeito de aprendizado.

Em 1956 no Dartmouth College nasceram os dois paradigmas da Inteligência Artificial, a simbólica e o conexionista. A Inteligência Artificial Simbólica tenta simular o comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal. Já a Inteligência Artificial Conexionista acredita que construindo-se um sistema que simule a estrutura do cérebro, este sistema apresentará inteligência, ou seja, será capaz de aprender, assimilar, errar e aprender com seus erros. Neste mesmo ano, na Dartmouth Conference, foi apresentado um modelo de RNA artificial pelo pesquisador da IBM Nathaniel Rochester. Seu modelo consistia numa simulação de centenas de neurônios interconectados através de um sistema que verificaria como a rede responderia aos estímulos ambientais.

ROSENBLATT(1959) na Universidade de Cornell, criou uma rede de múltiplos neurônios do tipo discriminadores lineares e denominou-a Perceptron Multicamada (MLP - MultiLayer Perceptron). Rosenblatt baseou-se nas linhas de pensamento de McCulloch para desenvolver o seu modelo matemático de sinapse humana. Devido as suas complexas pesquisas e inúmeras contribuições técnicas, muitos o consideram como fundador da neurocomputação.

Enquanto Rosenblatt trabalhava no perceptron, WIDROW (1990) da Universidade de Stanford com a ajuda de alguns estudantes desenvolveu um novo modelo de processamento de RNAs chamado de Adaline (Adaptive Linear Element)(Figura 2), a qual se destacava pela sua poderosa lei de aprendizado.

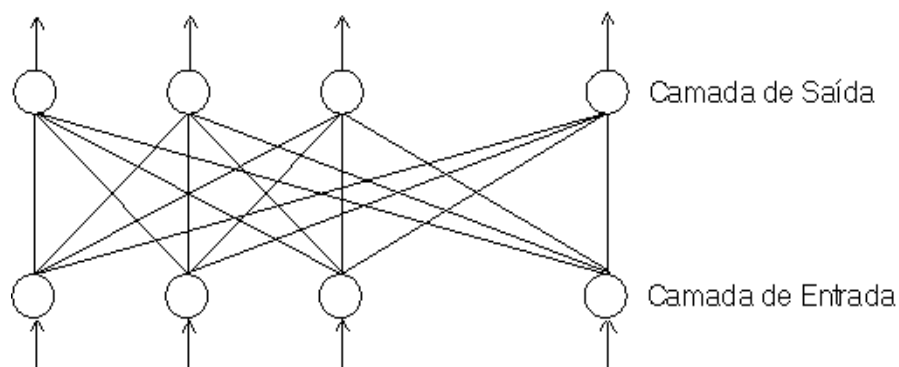


Figura 2 - Rede de perceptrons proposta por Rosenblatt

fonte:ZSOLUTIONS(2010)

O princípio de treinamento para as redes Adalines (Figura 3) ficou conhecido como a Regra Delta, que foi mais tarde generalizada para redes com modelos neuronais mais sofisticados. Posteriormente, Widrow criou a Madaline, que era uma generalização multidimensional do Adaline.

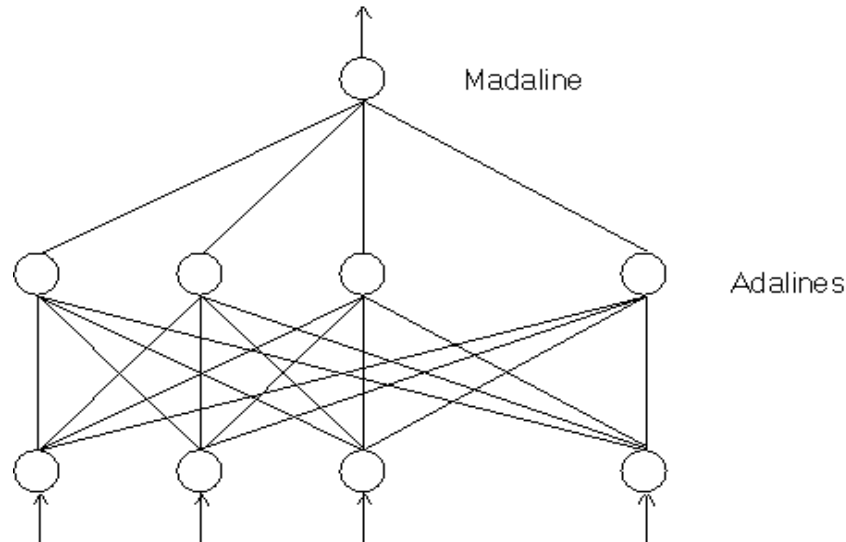


Figura 3 - Redes Adaline e Madaline

fonte:ZSOLUTIONS(2010)

Nos anos seguintes, muitos artigos foram publicados, e várias previsões exageradas e pouco confiáveis para a época foram anunciadas. A maioria destas suposições falava de computadores com um poder de raciocínio e/ou processamento igual ou superior ao do cérebro humano.

Após MINSKY e PAPERT (1969) lançarem em uma obra chamada Perceptron, a qual demonstrava que o modelo apresentado por Rosenblatt não era muito promissor, onde demonstravam matematicamente a limitação do perceptron em resolver certos problemas devido ao uso de técnicas empíricas, das grandes dificuldades da matemática envolvida e dos poucos recursos computacionais disponíveis na época, os estudos e a credibilidade na área foram desmotivados.

No início da década de 80, muitos pesquisadores publicaram inúmeras propostas para a exploração de desenvolvimento e pesquisa em RNAs. Foi quando o administrador de programas da DARPA (Defense Advanced Research Projects Agency) Ira Skurnick resolveu dar atenção às proposições da neurocomputação, contrariando todos os preceitos, fundando em 1983 as pesquisas em

neurocomputação da DARPA. Este fato acabou abrindo novos horizontes para a neurocomputação.

O físico e biólogo de reputação mundial HOPFIELD(1982) também se interessou pela neurocomputação e escreveu vários artigos em 1982 que levaram vários cientistas a se unirem nesta nova área emergente. Hopfield reascendeu as pesquisas em neurocomputação, criticando fortemente as teorias apresentadas por Minsky e Papert na década de 60.

Este campo de pesquisa explodiu mesmo em 1986, quando o professor de psicologia da Universidade de Stanford, David E. Rumelhart, e seu colega James L. McClelland, professor de psicologia da Universidade de CarnegieMellon, publicaram o livro *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (vol.1: Foundations, vol.2: Psychological and Biological Models). Nesse livro, eles apresentam um modelo matemático e computacional que propicia o treinamento supervisionado dos neurônios artificiais. Surgia, então, o algoritmo de backpropagation (retropropagação) (Figura 4), um algoritmo de otimização global sem restrições. Várias aplicações em RNAs em MLP vem surgindo desde então, como classificação de padrões, reconhecimento de fala, reconhecimento óptico de caracteres (OCR – Optical Character Recognition), entre outros. Também outros tipos de treinamento de redes têm sido propostos, como o método de gradiente conjugado, proposto inicialmente por FLETCHER e REEVES(1964), utilizado por KRAMER(1990) para acelerar o treinamento do perceptron, bem como o método de Newton.

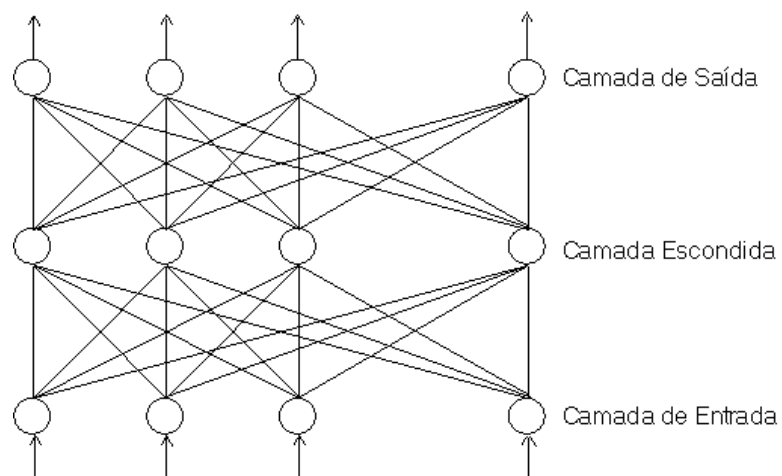


Figura 4 - Estrutura do método Backpropagation (retropropagação)

fonte: ZSOLUTIONS(2010)

Em 1987 ocorreu a Primeira Conferência de RNAs. Também foi formada a Sociedade Internacional de Redes Neurais (INNS - International Neural Networks Society) juntamente com o INNS Journal em 1989, do Neural Computation e do IEEE (Institute of Electrical and Electronics Engineers) Transactions on Neural Networks em 1990.

De acordo com a Sociedade Brasileira de Redes Neurais, em 1994 ocorreu o primeiro congresso de Redes Neurais no Brasil. Atualmente, pelo menos dois congressos brasileiros específicos em RNAs regularmente (Congresso Brasileiro de Redes Neurais e Simpósio Brasileiro de Redes Neurais), além de inúmeros eventos internacionais (entre eles o International Joint Conference on Neural Networks e o International Conference on Artificial Neural Networks), bem como eventos nacionais de abrangência mais geral que contam com seções específicas em RNAs. Além disso, uma gama ampla de periódicos científicos trazem os principais avanços da área, como o IEEE Transactions on Neural Networks, publicado pelo IEEE, e o periódico Neural Networks, publicado pela editora Elsevier.

2.2 - O Neurônio Humano

FERNEDA(2006) - O cérebro humano é considerado o mais fascinante processador baseado em carbono existente, sendo composto por aproximadamente 10 bilhões neurônios. Todas as funções e movimentos do organismo estão relacionados ao funcionamento destas pequenas células. Os neurônios estão conectados uns aos outros através de sinapses, e juntos formam uma grande rede, chamada Rede Neural. As sinapses transmitem estímulos através de diferentes concentrações de Na^+ (Sódio) e K^+ (Potássio), e o resultado disto pode ser estendido por todo o corpo humano. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação.

O sistema nervoso é formado por um conjunto extremamente complexo de neurônios. Nos neurônios a comunicação é realizada através de impulsos. Quando um impulso é recebido, o neurônio o processa, e passado um limite de ação, dispara um segundo impulso que produz uma substância neurotransmissora o qual flui do corpo celular para o axônio (que por sua vez pode ou não estar conectado a um

dendrito de outra célula). O neurônio que transmite o pulso pode controlar a frequência de pulsos aumentando ou diminuindo a polaridade na membrana pós-sináptica. Eles tem um papel essencial na determinação do funcionamento, comportamento e do raciocínio do ser humano. Ao contrário das RNAs artificiais, RNAs naturais não transmitem sinais negativos, sua ativação é medida pela frequência com que emite pulsos. As redes naturais não são uniformes como as redes artificiais, e apresentam uniformidade apenas em alguns pontos do organismo. Seus pulsos não são síncronos ou assíncronos, devido ao fato de não serem contínuos, o que a difere de redes artificiais.

Os principais componentes dos neurônios (Figura 5) são:

- os dentritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios;
- o corpo de neurônio, também chamado de soma, que é responsável por coletar e combinar informações vindas de outros neurônios;
- e o axônio, que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outras células.

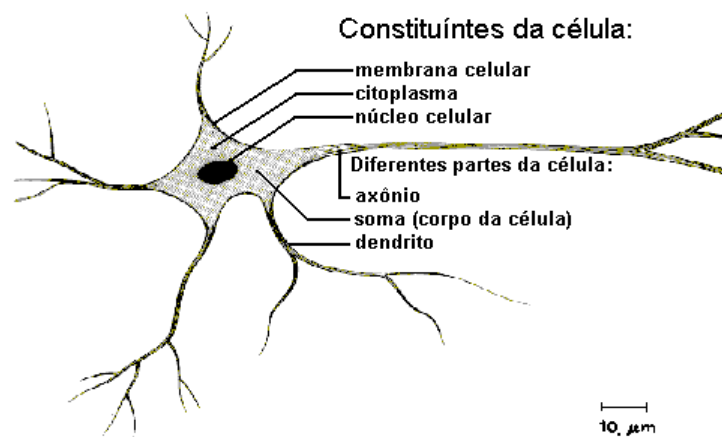


Figura 5 – Esquema das partes constituintes de uma célula neuronal

fonte: FERNEDA (2006)

– 2.3 - Redes Neurais

HAYKIN(2001) - A tecnologia de RNA tem como finalidade solucionar problemas envolvendo reconhecimento de padrões baseando-se em um conjunto de informações previamente conhecido. Esse conjunto geralmente é dividido em dois outros conjuntos: treinamento e teste (eventualmente há um terceiro conjunto de dados utilizados para validação). A rede é então submetida ao treinamento através do primeiro conjunto e para a avaliação deste treinamento é submetido o segundo conjunto à rede, que o classifica conforme seu treinamento e então são comparados os resultados das classificações da rede treinada e das classificações do segundo conjunto.

Hoje a pesquisa em RNA's busca a compreensão das capacidades da natureza humana, que possibilitam às pessoas a construção de soluções para problemas que não podem ser resolvidos através dos métodos tradicionais.

As RNA's objetivam a solução de problemas de Inteligência Artificial (IA), modelando sistemas através de circuitos (conexões) que possam simular o sistema nervoso humano, que possui a capacidade de aprender e agir perante as mais adversas situações, bem como adquirir conhecimento através da experiência e da observação.

A “imitação da biologia e do sistema nervoso” é feita pelas RNAs tanto no nível do processamento microscópico realizado por uma célula do sistema nervoso individualmente, quanto no nível macroscópico, seja em termos de padrões de interconexão entre elementos de processamento, seja em termos dos comportamentos coletivos que emergem em uma rede biológica com vários neurônios.

Segundo o paradigma vigente em neurofisiologia, a informação é representada no sistema nervoso através de pulsos elétricos gerados pelos neurônios, denominados potenciais de ação. No que se refere ao campo de redes de neurônios artificiais, a forma específica na qual a atividade neuronal é representada no modelo matemático pode assumir diversas simplificações. Isso depende do grau de refinamento na

representação de fenômenos biológicos que se tem em mente, o que por sua vez varia grandemente de grupo de pesquisa a grupo de pesquisa, de acordo com a orientação e a ênfase dos seus trabalhos.

Basicamente, todos os tipos de RNA apresentam a mesma unidade de processamento: um neurônio artificial, que simula o comportamento do neurônio biológico. Esse neurônio artificial possui várias entradas, que correspondem às conexões sinápticas com outras unidades similares a ele, e uma saída, cujo valor depende diretamente da somatória ponderada de todas as saídas dos outros neurônios a esse conectado. Neste cálculo de integração dos estímulos das várias entradas, a ponderação é dada pelos pesos que cada conexão possui (Figura 6).

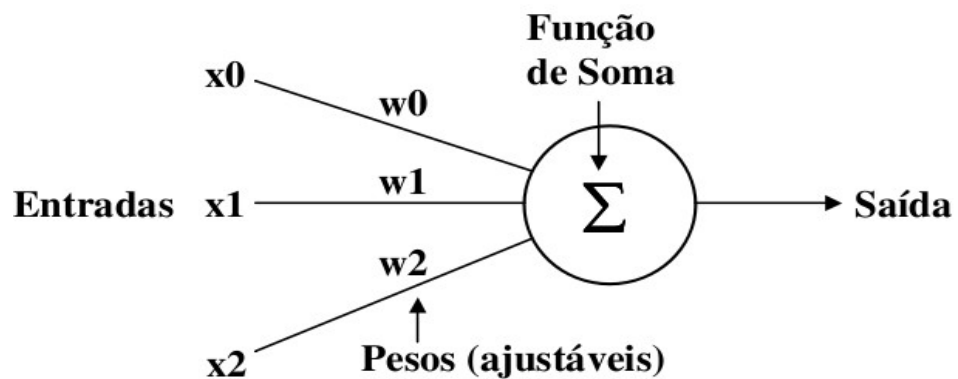


Figura 6 - Unidade Processadora (neurônio artificial) do tipo linear (Adaline).

Fonte: HAYKIN(2001)

Assim, a somatória que na figura corresponde à saída do modelo neuronal, é dada por:

$$Soma = \sum_{i=1}^n x_i w_i$$

As constantes de ponderação W_i representam as conexões sinápticas biológicas que intermedeiam a interação entre neurônios vizinhos, X_i é o estímulo proveniente do neurônio vizinho i , e o resultado Soma representa o estímulo global (com as devidas ponderações relativas) recebido pelo neurônio de todos os seus vizinhos. O neurônio artificial acima representado corresponde ao modelo matemático

conhecido como Adaline (Adaptive Linear Element). É importante notar que esse modelo somente incorpora transformações lineares; como consequência, temos que além de o modelo não permitir a implementação de computações complexas (há pois limitação do poder matemático), impede a representação de relações estímulo/resposta não lineares que de fato são observadas com frequência no neurônio biológico (há pois limitação do poder de modelagem de sistemas biológicos). Uma pequena alteração no modelo neuronal é necessária para a viabilizar as funcionalidades não lineares, através da introdução de uma função de transferência como representado na figura 7.

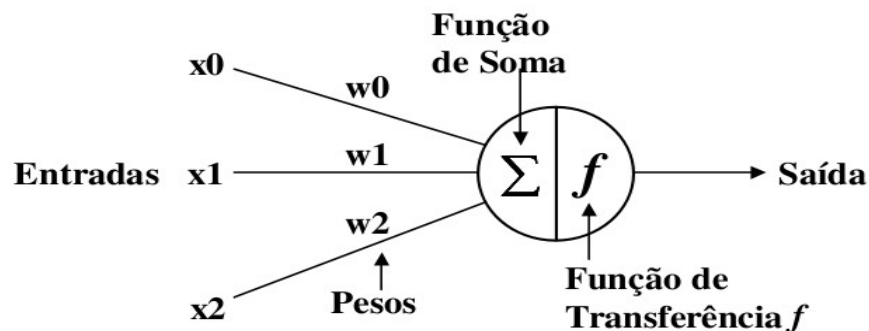


Figura 7 - Neurônio artificial com função de transferência não linear.

Fonte: HAYKIN(2001)

De acordo com o valor da variável Soma, e de acordo com a função de transferência à qual essa variável *Soma* é aplicada (ou também conhecida como função de ativação), o novo modelo de neurônio, não linear, fornecerá uma saída em resposta aos estímulos x_i , conforme representado na seguinte equação:

$$\text{Saída} = f(\text{Soma})$$

Na sua forma mais simples, a função de transferência não linear f pode ser binária (discreta), como no caso do modelo conhecido como Perceptron de ROSENBLATT(1958). Neste caso, o modelo tendo uma saída binária permite representar funcionalidades de natureza lógica realizadas pelos neurônios biológicos. É o caso da tomada de decisões (uma funcionalidade digital por natureza). Por outro lado, para o caso de modelagem de funcionalidades do sistema

nervoso em que não apenas grandezas digitais estão em jogo, mas uma gama de valores analógicos são necessários para a representação da atividade neuronal, devemos empregar uma função de transferência f com saída contínua, como por exemplo a função tangente hiperbólica (neste caso o modelo pode ser chamado de sigmoidal). Em ambos os casos, a ideia é que essa função de transferência possa imitar, em algum grau, a relação entrada / saída observada nos neurônios biológicos.

2.4 - Rede Neuronal MLP

HAYKIN(2001) - Naturalmente, uma estrutura de cálculo tão simples como a do Perceptron não atende a todas as necessidades de classificação de padrões no caso geral. À medida que bases de dados são mais complexas e que a classificação com separabilidade linear é necessária, empregamos uma estrutura com diversos neurônios encadeados. Com isso, temos o conceito do MultiLayer Perceptron (MLP), cujas características principais são as seguintes:

- 1) possui múltiplas entradas / múltiplas saídas / múltiplas camadas;
- 2) entradas e saídas podem ser tanto analógicas quanto digitais;
- 3) não há mais a restrição de separabilidade linear entre classes.

A figura 8 representa uma arquitetura do tipo MLP com 20 neurônios organizados em 5 camadas. Cada nó (círculo) representa um neurônio e cada linha uma conexão entre saída de um neurônio e uma entrada de neurônio da camada seguinte.

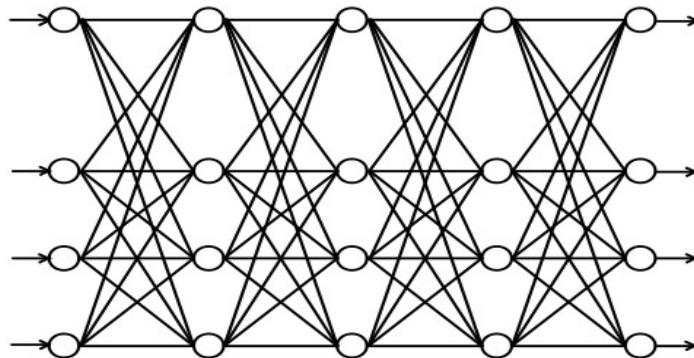


Figura 8 - Rede do tipo MLP (Multi Layer Perceptron) de 5 camadas. Cada um dos 20 círculos representa um neurônio artificial, e cada linha representa uma conexão sináptica, com seu peso programável. *Fonte: HAYKIN(2001)*

Como pode ser verificado, o fluxo de informação ocorre em camadas de neurônios, sem que haja realimentação (retorno da informação da saída para a entrada de algum dos neurônios da rede). Esta organização em camadas e a ausência de realimentação é própria da arquitetura MLP, não se observando tais características no entanto em muitas outras arquiteturas neuronais importantes como, por exemplo, a arquitetura de Hopfield, que é densa em realimentações e não possui qualquer organização em camadas.

Note que no caso do MLP, a função de transferência adotada tipicamente é do tipo sigmoideal (tangente hiperbólica por exemplo), de forma que com escalamentos adequados do argumento da função de transferência (ou dos pesos que indiretamente escalam as entradas) é possível passar de mapeamentos entrada-saída aproximadamente lineares a mapeamentos de natureza essencialmente digital, sem mudar a função f propriamente dita, mas apenas com o recurso de escalamento do argumento. Esta transição entre comportamento analógico e digital é possível porque a função $\text{sign}(x)$ pode ser vista como o caso limite da função tangente hiperbólica com constante escalamento tendendo a infinito.

Entre as aplicações do MLP, podemos citar a classificação de padrões sem separabilidade linear já mencionada, a aproximação de funções genéricas (inclusive analógicas) e a fusão não linear de grandezas analógicas multidimensionais, além da previsão de séries temporais não lineares. O aprendizado através de exemplos do MLP permite que ele realize as funções acima sem a necessidade de um modelo matemático conhecido / confiável .

Existe ainda um leque muito grande de arquiteturas e aplicações correspondentes, como as arquiteturas de HOPFIELD(1982), as arquiteturas com modelos pulsados, as arquiteturas com funções de base radial (radial basis functions – RBFs), as arquiteturas do tipo ART (Adaptive Resonance Theory), e os mapas auto organizáveis de Kohonen, entre outras.

2.5 - Exame Parcial de Urina (EAS)

LOPES(2004) classifica o EAS como um dos mais simples exames. Colhe-se de 40 a 50 ml de urina, normalmente a primeira urina da manhã e que se despreze o primeiro jato. Essa pequena quantidade serve para eliminar as impurezas que possam estar na uretra. Em seguida enche-se o recipiente com o jato do meio. Quanto mais fresca estiver, mais confiável são seus resultados. Um intervalo de mais de 2 horas entre a coleta e a avaliação normalmente invalidam qualquer resultado, principalmente se urina não tiver sido mantida sob refrigeração.

O EAS é dividido em 2 partes: a primeira é feita através de reações químicas e a segunda por visualização de gotas da urina pelo microscópio.

Na primeira parte mergulha-se na urina uma fita, denominada Tira Reagente (Figura 9). Cada fita possui vários quadradinhos coloridos compostos por substâncias químicas que reagem com determinados elementos da urina.



Figura 9 – Tira Reagente *fonte: LOPES(2004)*

Após alguns minutos, compara-se a cores dos quadradinhos com uma tabela de referência que costuma vir na embalagem das próprias fitas do EAS (Figura 10).



Figura 10 – Comparação da Tira Reagente *fonte: LOPES(2004)*

Através destas reações e com o complemento do exame microscópico, podemos detectar a presença e a quantidade dos seguintes dados da urina:

- Densidade (gravidade específica) ;
- pH ;
- Glicose ;
- Proteínas ;
- Hemácias (sangue) ;
- Leucócitos ;
- Cetonas ;
- Urobilinogênio e bilirrubina ;
- Nitrito ;
- Cristais ;
- Células epiteliais e cilindros.

Os resultados da Tira Reagente são qualitativos e não quantitativos. A fita identifica a presença dessas substâncias, mas a quantificação é apenas aproximada. O resultado é normalmente fornecido em uma graduação de cruzes de 0 a 4.

a) Densidade:

A densidade da água pura é igual a 1000. Quanto mais próximo deste valor, mais diluída está a urina. Do mesmo modo, quanto mais afastado, mais concentrada ela está. Os valores de referência variam de 1005 a 1035.

A densidade indica a concentração das substâncias sólidas diluídas na urina. Quanto menos água houver na urina, menos diluída ela estará e maior será sua densidade. Urina com densidade próxima de 1030 indica desidratação. São muito amareladas e normalmente possuem odor característico.

b) pH:

A urina é naturalmente ácida, pois o rim é o principal meio de eliminação dos ácidos do organismo. Enquanto que o pH do sangue está em torno de 7,4, o pH da urina varia entre 5,5 e 7,0.

Valores maiores ou igual 7 podem indicar presença de bactérias que alcalinizam a urina. Valores menores que 5,5 podem indicar acidose no sangue ou doença nos túbulos renais.

O valor mais comum é um pH por volta de 5, porém, mesmo valores acima ou abaixo dos descritos podem não necessariamente indicar alguma doença. Este resultado deve ser interpretado pelo médico.

c) Glicose:

Toda a glicose que é filtrada nos rins, é reabsorvida e volta para o sangue pelo túbulos renais. Deste modo, o normal é não apresentar evidências de glicose na urina.

Os doentes com diabetes mellitus costumam apresentar glicose na urina. Como a quantidade de açúcar no sangue está muito elevada, o rim acaba recebendo e filtrando também uma grande quantidade deste. O problema é que a partir de valores de glicemia acima de 150 mg/dL (ainda variando de indivíduo para indivíduo) a capacidade de reabsorção do túbulo renal é ultrapassada, e o paciente acaba

perdendo glicose na urina.

Já a presença de glicose na urina sem que haja diabetes, costuma ser um sinal de doença nos túbulos renais. Portanto, só há glicose na urina se houver excesso desta no sangue ou se houver doença nos rins.

d) Proteínas:

A maioria das proteínas não são filtradas pelo rim, e por isso, em situações normais, não devem estar presentes na urina. Na verdade, existe apenas uma pequena quantidade de proteínas na urina, mas são tão poucas que não costumam ser detectadas pelo teste da fita. Portanto, o normal é a ausência de proteínas. Existem 2 maneiras de se apresentar o resultado: em cruces ou uma estimativa em mg/dL:

Ausência = menos que 10 mg/dL ou 0,05 g/L (valor de referência habitual)

Traços = entre 10 e 30 mg/dL

1+ = 30 mg/dl

2+ = 40 a 100 mg/dL

3+ = 150 a 350 mg/dL

4+ = Maior que 500 mg/dL

A presença de proteínas na urina se chama proteinúria, pode indicar doença renal e deve ser sempre investigada.

e) Hemácias (sangue):

Assim como nas proteínas, a quantidade de hemácias (glóbulos vermelhos) na urina é desprezível e não consegue ser detectada pelo exame da fita. Mais uma vez, os resultados costumam ser fornecidos em cruces. O normal é haver ausência de hemácias (hemoglobina).

A urina após o teste da fita, é examinada também em microscópio. Desta maneira, pode-se contar a quantidade de hemácias que estão presentes. Essa avaliação por microscópio é chamada de sedimentoscopia.

Através do microscópio consegue-se detectar qualquer presença de sangue, mesmo aquelas não detectadas pela fita. Neste caso os valores normais são descritos de 2 maneiras: menos que 3 a 5 hemácias por campo ou menos que 10 células/ μL .

A presença de sangue na urina chama-se hematúria, e pode ocorrer por diversos motivos, desde um falso positivo devido a menstruação, até infecções, cálculos nos rins e doenças renais graves.

Uma vez detectada a hematúria, o próximo passo é avaliar a forma das hemácias em um exame chamado de dismorfismo eritrocitário. Nem todo laboratório tem gente capacitada para executar esse exame. Por isso, muitas vezes ela não é feito automaticamente. É preciso o médico solicitar especificamente essa avaliação.

A presença de hemácias dismórficas, principalmente se em mais de 50%, indica uma doença dos glomérulos.

f) Leucócitos

Os leucócitos são os glóbulos brancos, nossas células de defesa. A presença de leucócitos na urina costuma indicar que há atividade inflamatória nas vias urinárias. Em geral sugere infecção urinária, mas pode estar presente em várias outras situações, como traumas, drogas irritativas ou qualquer outra inflamação não causada por uma agente infeccioso.

Como também são células, os leucócitos podem ser contados na sedimentoscopia. Valores de referência estão abaixo dos 10 células/ μL .

Algumas Tiras Reagentes apresentam um quadradinho para detecção de leucócitos, normalmente o resultado vem descrito como esterase leucocitária. O resultado normal é estar negativo.

g) Cetonas ou corpos cetônicos:

Os corpos cetônicos são produtos da metabolização das gorduras. Normalmente não estão presentes na urina. A sua detecção pela Tira Reagente pode indicar

diabetes descompensado ou jejum prolongado.

f) Urobilinogênio e bilirrubina

Também normalmente ausentes na urina, podem indicar doença hepática (fígado) ou hemólise (destruição anormal das hemácias). A bilirrubina só costuma aparecer na urina quando os seus níveis sanguíneos ultrapassam 1,5 mg/dL.

g) Nitritos

A urina é rica em nitratos. A presença de bactérias na urina transforma esses nitratos em nitritos. Logo, a Tira Reagente com nitrito positivo é um sinal da presença de bactérias. Nem todas as bactérias tem a capacidade de metabolizar o nitrato, por isso, nitrito negativo de forma alguma descarta infecção urinária.

Na verdade, o EAS apenas sugere infecção. A presença de hemácias, associado a leucócitos e nitritos positivos, fala muito a favor de infecção urinária, porém, o exame de certeza é a urinocultura .

A pesquisa do nitrito é feita através da reação de Griess, que é o nome dado a reação do nitrito com um meio ácido. Por isso, alguns laboratórios fornecem o resultado como Griess positivo ou Griess negativo, que é igual a nitrito positivo e nitrito negativo, respectivamente.

h) Cristais

Esse é talvez o resultado mais mal interpretado, tanto por pacientes como por alguns médicos. A presença de cristais na urina, principalmente de oxalato de cálcio, não tem nenhuma importância clínica. Ao contrário do que se possa imaginar, a presença de cristais não indica uma propensão a formação de cálculos renais.

Os únicos cristais com relevância clínica são os de cistina, magnésio-amônio-fosfato (estruvita), tirosina, bilirrubina e de colesterol.

i) Células epiteliais e cilindros

A presença de células epiteliais é normal. São as próprias células do trato urinário que descamam. Elas só tem valor quando presente em cilindros.

Como os túbulos renais são cilíndricos, toda vez que temos alguma substância em grande quantidade na urina, elas se agrupam em forma de um cilindro. A presença de cilindros indica que esta substância veio dos túbulos renais e não de outros pontos do trato urinário como a bexiga, ureter, próstata, entre outros. Isto é muito relevante, por exemplo, nos casos de sangramento, onde um cilindro hemático indica o glomérulo como origem.

Os cilindros que podem indicar alguma alteração são:

- Cilindros hemáticos (sangue) = Indica glomerulonefrite
- Cilindros leucocitários = Indicam inflamação dos rins
- Cilindros epiteliais = indicam lesão dos túbulos
- Cilindros gordurosos = indicam proteinúria

Cilindros hialinos não indicam doença, mas pode ser um sinal de desidratação. A presença de muco é inespecífica e normalmente ocorre pelo acúmulo de células epiteliais com cristais e leucócitos. Tem pouquíssima utilidade clínica. É mais uma observação.

Em relação ao EAS (urina tipo I) é importante salientar que esta é uma análise que deve ser sempre interpretada. Os falsos positivos e negativos são muito comuns e não dá para se concluir qualquer diagnóstico apenas comparando os resultados com os valores de referência.

Em uma segunda metodologia, que envolve a análise em grandes quantidades de exames, VOLANSKI(2006) descreve o setor de Urinálise da LMC (Laboratório Municipal de Curitiba), que é equipado com um aparelho Urisys 2400(ROCHE, que analisa aproximadamente 240 amostras/hora) para análise de parâmetros físico-químicos (Figura 11) e dois aparelhos Sysmex UF-100 (ROCHE, com velocidade de

análise de 100 amostras/hora) para quantificação e avaliação dos diferentes tipos de células e partículas contidas na amostra de urina(Figura 12).



Figura 11 – Urisys 2400 ROCHE *fonte: Volanski (2006)*



Figura 12 – Sysmex UF – 100 *fonte: Volanski (2006)*

Após a chegada dos tubos no setor de urinálise, é realizada uma triagem separando os tubos para exame parcial EAS (Elementos Anormais e Sedimentares) daqueles com solicitação para outros exames. Em seguida, os tubos são colocados nas racks que serão encaminhadas aos equipamentos Urisys 2400 e , em seguida, ao Sysmex UF-100. Os resultados saem diretamente dos equipamentos para um banco de dados e então são analisados os resultados por uma interface onde o especialista define quais exames apresentam resultados alterados e devem passar por uma confirmação microscópica, ou seja, reanalisados pelo especialista em microscópio um a um. Após essas revisões os exames são aprovados e liberados.

3 - MATERIAIS E MÉTODOS

3.1 - HARDWARE

Computadores pessoais com as seguintes configurações:

- Processador Pentium 2.2GHz DualCore
- Memória RAM 2GiB
- HD180G
- Conexão à internet
- Entradas USB

3.2 - SOFTWARE

3.2.1 - Desenvolvimento

- Python 2.6.5 – é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Possui uma gama considerável de bibliotecas disponíveis, principalmente no campo científico. Disponibilizado sob a licença GPL (*General Public License* – Licença Geral Pública).
- Biblioteca ffnet 0.6.2 – criada por Marek Wojciechowski, o autor a descreve como uma solução de RNA com retroalimentação rápida e fácil de usar. Possui várias facilidades como interface com a biblioteca matplotlib, que facilita a plotagem de gráficos e resultados, além de vários esquemas de algoritmos de otimização de treinamento. Disponibilizado sob a licença GPL.

3.2.2 - Documentação

- BrOffice.org 3.2 - é o principal software de código-fonte aberto de suíte de escritório para processamento de texto, planilhas, apresentações, gráficos, bases de dados e muito mais. Ele está disponível em vários idiomas e funciona em todos os computadores comuns. Ele armazena todos os dados em um formato padrão internacional aberto e pode também ler e escrever arquivos de outros pacotes de software comum de escritório. Ele pode ser baixado e utilizado totalmente gratuito para qualquer finalidade. Disponibilizado sob a licença GPL.
- Umbrello UML Modeller 2.4.2 – O Umbrello auxilia no processo de desenvolvimento de software através do uso do padrão industrial da Linguagem de Modelagem Unificada (UML) para permitir-lhe criar diagramas para o desenvolvimento e documentação de seus sistemas. Disponibilizado sob a licença GPL.

3.3 - REDES NEURONAIS

Para trabalhar com RNA, existem algumas pré-condições e necessidades estabelecidas. Para a viabilidade do projeto, foram realizadas as seguintes etapas:

- Levantamento da base de dados a ser utilizada para treinamento, teste e validação da RNA;
- Estudo da RNA para definição de sua configuração e esquema de treinamento;
- Treinamento da rede e avaliação dos resultados da submissão das bases de teste e validação à rede

As RNA necessitam de uma base de dados ou base de conhecimento iniciais para serem treinadas e posteriormente testadas e validadas. Essa base inicial deve ser submetida ao treinamento da rede e deve possuir amostras suficientes de todas as classes para um treinamento satisfatório.

Do Laboratório Municipal de Curitiba, foi colhida uma base contendo cerca de

100.000 amostras de análises de exames de urina, cujas classes de classificação são os resultados dos exames que foram avaliados como resultado normal e resultado anormal. Esta base possui linhas com os dados dos exames de urina contendo colunas com as seguintes informações dos pacientes:

- 1 – idade do paciente;
- 2 – sexo do paciente;
- 3 – esterase leucocitária;
- 4 – leucócitos;
- 5 – hemoglobina;
- 6 – eritrócitos;
- 7 – classificação do exame entre normal e anormal.

Inicialmente esta base de dados foi dividida igualmente em três partes, formando a base de treinamento, a base de teste e a base de validação. Então submeteu-se à base de treinamento diferentes tipos de RNA: MLP(MultiLayer Perceptrons), FAN(Free Associative Neurons), RBF(Radial Basis Function) e SVM(Support Vector Machines).

Foram utilizados os seguintes softwares para as RNA: biblioteca ffnet e Python para RNA do tipo MLP, o software EasyFAN para a RNA do tipo FAN e para as RNA do tipo RBF e SVM foi utilizado o software MATLAB® R2009a. Devido aos resultados superiores, as facilidades de uso e flexibilidade, este estudo deu ênfase às duas primeiras redes, porém foi implementado utilizando a RNA do tipo MLP. Pela portabilidade, por dispor de biblioteca nativa em Python, pela maior disseminação e conhecimento e predominantemente por apresentar os melhores resultados no estudo, a utilização da MLP representa a melhor escolha, porém nada impede a utilização de outras redes devidamente estudadas e configuradas para este ou outros projetos semelhantes.

A biblioteca `ffnet`, de WOJCIECHOWSKI(2007), foi desenvolvida em Python com funções de portabilidade para FORTRAN. Possui diferentes algoritmos de otimização de treinamento e algumas funções auxiliares, como a de algoritmos genéticos. Por ser desenvolvida para fins de estudos, não possui muitas facilidades para visualização de resultados ou automatização de processos, sendo então utilizada como parte desta solução.

Foi então desenvolvido este programa que, com a utilização desta biblioteca, possui várias funcionalidades para estudo e apresentação de resultados, e o mais importante, para a automatização da atividade de busca dos resultados dos exames, classificação e armazenamento do resultado conforme desejado.

3.4 - PASSO A PASSO DO SISTEMA DE APOIO À DECISÃO

Dentro do prompt do python e no mesmo diretório em que estiver o arquivo do sistema de apoio à decisão (`sad.py`), importa a classe `sad` para o ambiente e a instancia:

```
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from sad import sad
>>> sad = sad()
```

Após, carrega-se os arquivos de treino, teste e validação, passando como argumento o nome do arquivo e o separador dos campos, neste caso é a tabulação (`'\t'`):

```
>>> sad.carregarTreino('reduzido/treino.txt','\t')
-> Base de treino carregada de arquivo
>>> sad.carregarTeste('reduzido/teste.txt','\t')
-> Base de teste carregada de arquivo
>>> sad.carregarValidacao('reduzido/validacao.txt','\t')
-> Base de validacao carregada de arquivo
```

Exemplo de algumas linhas de um dos arquivos, onde cada um dos campos refere-se a uma característica: idade, sexo, esterase leucocitária, leucócitos, hemoglobina, eritrócitos e classe.

```
78    0    10    5.90  100    674.10 1
69    0    0     3.10  500    202.6  0
46    1    25    57.50  0     6.90  0
24    1    25    13.30  100    1024.20 1
56    1    0     3.10  0     37.70  1
76    1    10    16.00  100    21.60  0
48    1    0     5.10  25     10.5  0
24    1    10    9.40  500    257.1  0
2     1    0     5.10  0     10.30  1
77    1    50    8.80  25     27.30  1
```

Define-se as camadas ocultas da rede passando como parâmetro um array com a quantidade de perceptrons de cada camada. A primeira camada possui perceptrons conforme o número de características e é definido automaticamente quando se carrega o arquivo de treino e a última camada é sempre um, pois é o perceptron que apresenta a classificação.

```
>>> sad.defCamadas([15,15])
>>> sad.camadas
[6, 15, 15, 1]
```

Depois dessas configurações básicas vem o treinamento da rede, que pode ser interrompido a qualquer momento. Durante o treinamento é apresentado a taxa de acertos da rede em relação à classificação dos arquivos:

```
>>> sad.treinar()
início: 2010-06-23 10:15:55.867856
TREINANDO REDE.(pressione ENTER para parar o treinamento)
épocas: 1000 | acertos - Treino: 96.08% Teste: 95.74% Validação: 90.70% /tempo: 0:13:20.107251
épocas: 2000 | acertos - Treino: 96.09% Teste: 95.73% Validação: 90.29% /tempo: 0:26:42.369600
épocas: 3000 | acertos - Treino: 96.11% Teste: 95.75% Validação: 90.50% /tempo: 0:40:05.227460
épocas: 4000 | acertos - Treino: 96.09% Teste: 95.73% Validação: 90.50% /tempo: 0:53:30.732171
parando treinamento...
```

As matrizes de confusão, onde o sistema mostra os erros e acertos nas classificações, são mostradas através da função desempenho():

```
>>> sad.desempenho()
-----
BASE DE TREINO com 17200 amostras
acertos: 16665 96.89% (erros: 535 3.11%)
Matriz de confusão:
      0      1
0  8185( 47.59%)  415( 2.41%)
1   120( 0.70%) 8480( 49.30%)
-----
BASE DE TESTE com 17200 amostras
acertos: 16606 96.55% (erros: 594 3.45%)
Matriz de confusão:
      0      1
0  8175( 47.53%)  425( 2.47%)
1   169( 0.98%) 8431( 49.02%)
-----
BASE DE VALIDAÇÃO com 17210 amostras
acertos: 16550 96.17% (erros: 660 3.83%)
Matriz de confusão:
      0      1
0  8131( 47.25%)  475( 2.76%)
1   185( 1.07%) 8419( 48.92%)
-----
```

Para carregar dados para uma base de um banco de dados (no caso MySQL), primeiro define-se as informações para conexão com o servidor:

```
>>> sad.defConSQL()
Entre com o servidor SQL: localhost
Entre com o usuário: bioinfo
Entre com a senha: bioinfo
Entre com a base de dados: bioinfo
-> Conexão com banco de dados ok !
```

Com o teste de conexão funcionando, define-se então a tabela e os campos a serem buscados do banco de dados:

```
>>> sad.defCamposSQL()
Nome da Tabela: TB_Resul
Já existe uma rede carregada com 6 características
Coluna da característica 1/6: DATE_FORMAT(Date, '%Y')-DATE_FORMAT(DN, '%Y')-
(DATE_FORMAT(Date, '00-%m-%d')< DATE_FORMAT(DN, '00-%m-%d'))
Coluna da característica 2/6: SX
```

```

Coluna da característica 3/6: campo10
Coluna da característica 4/6: campo11
Coluna da característica 5/6: campo3
Coluna da característica 6/6: campo12
Entre com a coluna da classe: Campo22
Entre com a coluda do identificador: Sample_NR
Caso deseje, entre com algum filtro SQL: where Campo22 = NULL or Campo22 = ''
-> OK, query válida

```

Note que não há um campo com a idade do paciente na tabela. Então define-se a característica 1 com uma query que traz o cálculo deste valor. A coluna na classe é onde está a classificação o exame ou é o campo onde será armazenada a classificação realizada pela rede posteriormente. A coluna do identificador é o campo de identificação de cada exame para que seja corretamente atualizado. No caso do filtro SQL, foi solicitado apenas os exames que ainda não foram classificados. Também pode passar como parâmetro se deseja que os dados carregados sejam definidos como base de classificação (parâmetro definido por omissão), treino, teste ou validação.

O comando `carregarDeSQL` importa os dados do banco de dados conforme os parâmetro de conexão e de campos definidos:

```

>>> sad.carregarDeSQL()
-> Dados carregados do banco de dados na base de classificacao

```

Após carregar a base de classificação, executa-se a classificação da base:

```

>>> sad.classificar()
-> Base de classificação classificada !

```

E, finalmente para salvar essa classificação no banco de dados, onde servirá como sugestão de diagnóstico:

```

>>> sad.salvarClassificadoSQL()
->Classificações salvas com sucesso !

```

Então, para que o sistema salve as configurações já executadas:

```
>>> sad.salvarEsquema('esquema')
->Rede salva em esquema.net
->Esquema salvo em esquema.esq
```

Toda vez que o esquema salvo for carregado, ele carregará todas as configurações salvas e executará a busca de dados, classificação e salvamento dos dados no bando de dados:

```
>>> sad.carregarEsquema('esquema')
->Base de treino carregada de arquivo
->Base de teste carregada de arquivo
->Base de validacao carregada de arquivo
->Rede carregada
->Dados carregados do banco de dados na base de classificada
->Base de classificação classificada !
->Classificações salvas com sucesso !
```

Para automatizar ainda mais, a função de criar executável monta um arquivo que basta apenas executá-lo para realizar toda a operação:

```
>>> sad.criarExecutavel('exe')
-> OK - Executável exe.py criado !
```

Exemplo do executável criado:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from sad import sad
sad = sad()
sad.carregarEsquema('esquema')
```

3.5 - TREINAMENTO E TESTES

Inicialmente foi utilizada a base completa dividida nas três partes necessárias para o estudo (treinamento, teste e validação), ficando cada uma das bases com aproximadamente 34.300 amostras. Estudando a base completa, verificou-se a proporção de 8.600 linhas com pacientes diagnosticados com o exame de urina com resultado anormal e o restante como resultado normal, que representa uma proporção de 25% de amostras classificadas na classe resultado anormal e 75% das amostras classificadas como resultado normal para o referido exame.

Foram utilizadas diferentes épocas de treinamento, sendo que quanto maior a quantidade de épocas, maior é o tempo e o nível de treinamento da rede. Na rede MLP, foram utilizadas várias formas de montagem, neste estudo resumiremos nas duas formas que apresentaram os melhores resultados: uma com uma camada oculta com 30 perceptrons e outra com duas camadas ocultas com 15 perceptrons cada.

Os resultados desse experimento foram satisfatórios, porém aparentemente não suficientes para que se usa-se como um sistema de apoio à decisão. Foi então realizado o estudo com alterações nas bases de dados de treinamento, teste e validação, alterando-se as proporções dos arquivos.

Nesse novo estudo, os arquivos foram dimensionados da forma que a proporção entre as diferentes classes fossem iguais. Cada um dos arquivos foi então reduzido para um total de 17.200 amostras de exames com 8.600 amostras de exames com resultados normais e outros 8.600 amostras com exames que apresentaram como diagnóstico o resultado anormal, ficando então a proporção de 50% de uma classe e 50% de outra classe em cada uma das três bases. Submeteu-se então novamente às mesmas etapas de testes realizadas no primeiro estudo.

```
>>> sad.treinar(100)
início: 2010-06-20 02:30:39.765489
TREINANDO REDE.(pressione ENTER para parar o treinamento)
épocas: 100 | acertos - Treino: 94.77%  Teste: 94.52%  Validação: 93.99%  /tempo: 0:00:44.881297
épocas: 200 | acertos - Treino: 94.76%  Teste: 94.58%  Validação: 94.00%  /tempo: 0:01:32.004454
épocas: 300 | acertos - Treino: 94.75%  Teste: 94.60%  Validação: 94.03%  /tempo: 0:02:16.324216
parando treinamento...
```

Exemplo de mostra de resultados durante o treinamento da rede

Os resultados desse segundo estudo foram superiores e de maior qualidade em relação ao primeiro, o que determinou a escolha da rede a ser implementada em produção.

Os estudos de VOLANSKI ainda envolvem outros sistemas de apoio à decisão e também tem várias comparações dos resultados de diagnósticos realizados pelos especialistas laboratoriais. Estes estudos comparam resultados entre um mesmo especialista avaliando a mesma base em diferentes situações ambientais (a primeira em condições ambientais físicas ideais de trabalho e outra com condições ambientais físicas adversas), entre dois diferentes especialistas avaliando a mesma base de dados e entre os especialistas e as RNA desenvolvidas para o apoio à decisão dos exames.

Foi então adicionado ao estudo uma nova base para comparação, contendo 484 amostras. Esta base então foi classificada por todos os classificadores utilizados no estudo. Todos estes resultados foram reunidos e comparados.

Outra análise, que foi levantada durante o projeto, é a da matriz de confusão. A matriz de confusão apresenta dados mais detalhados da classificação dos exames na rede comparando as classificações corretas e incorretas em cada classe e também as classificações dos erros ou onde a RNA se confundiu ao classificar o exame.

 BASE DE VALIDAÇÃO com 484 amostras
 acertos: 433 89.46% (erros: 51 10.54%)

Matriz de confusão:

	0	1
0	392(80.99%)	44(9.09%)
1	7(1.45%)	41(8.47%)

Exemplo de matriz de confusão apresentada pelo sistema

Esse novo item de análise levou o estudo a considerar não apenas o quanto a rede erra, mas onde está errando. Nesse caso podem existir dois tipos de erros: confundir um resultado normal com um anormal e confundir um resultado anormal com normal. O primeiro caso é menos relevante, pois rever um resultado de um exame diagnosticado como normal não afeta diretamente o desempenho do resultado final, pois será necessário apenas a confirmação de resultado normal através da reanálise da amostra causando apenas retrabalho desnecessário. Porém, no segundo caso, o sistema de apoio à decisão classifica um exame anormal como normal, o que induziria o especialista ao erro de diagnosticar sem uma reanálise um exame anormal. Assim evidenciou-se a necessidade maior de minimizar a confusão do sistema referente a exames anormais classificados como normais do que o contrário.

4 - RESULTADOS

Durante o treinamento da RNA utilizando as bases de treinamento, teste e validação com 34300 amostras cada, obteve-se o seguintes resultados:

Tabela 1 – Progresso do Treinamento (taxa de acerto em %)

Épocas	Tempo de Treino	Bases		
		Treino	Teste	Validação
1000	8m22s	86,37	86,02	85,53
5000	41m46s	90,69	90,43	90,16
10000	1h23m04s	91,88	91,74	91,61
15000	2h04m20s	92,14	91,94	91,81
20000	2h45m35s	92,28	92,14	91,99
25000	3h26m45s	92,43	92,29	92,14
30000	4h16m16s	92,66	92,56	92,41
35000	4h49m25s	92,77	92,71	92,56
40000	5h30m32s	93,01	92,95	92,78
45000	6h11m37s	93,09	93,06	92,90
50000	6h52m45s	93,25	93,17	93,01

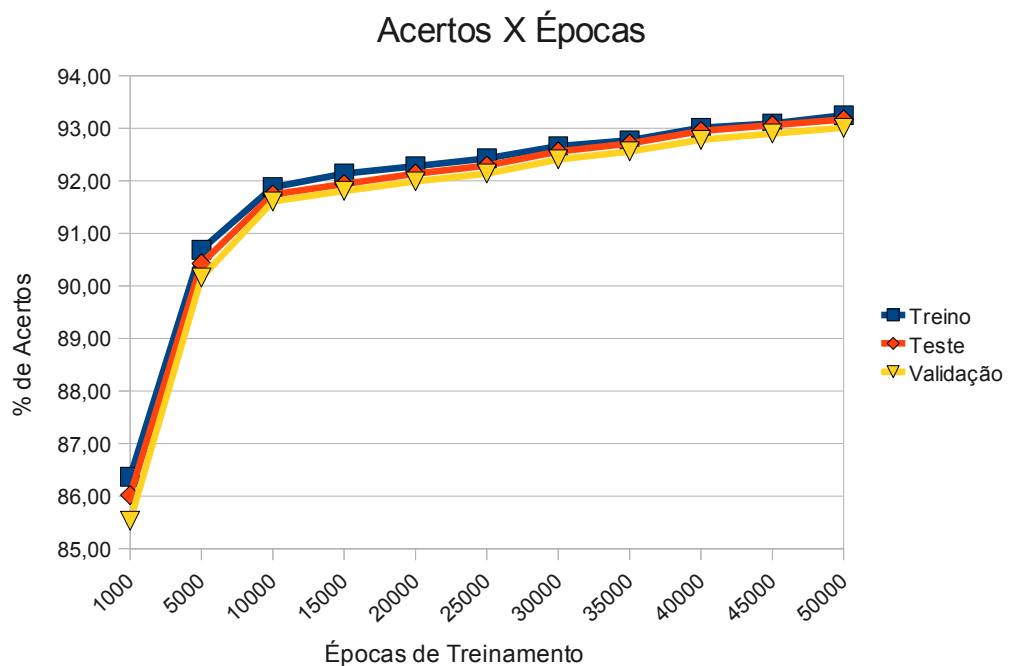


Figura 13 - Gráfico do progresso de Treinamento das Bases

Suas matrizes de confusão finais foram estas:

 BASE DE TREINO com 34292 amostras
 acertos: 32249 94.04% (erros: 2043 5.96%)

Matriz de confusão:

	0	1
0	24855(72.48%)	837(2.44%)
1	1206(3.52%)	7394(21.56%)

 BASE DE TESTE com 34292 amostras
 acertos: 32178 93.84% (erros: 2114 6.16%)

Matriz de confusão:

	0	1
0	24813(72.36%)	879(2.56%)
1	1235(3.60%)	7365(21.48%)

 BASE DE VALIDAÇÃO com 34296 amostras
 acertos: 32171 93.80% (erros: 2125 6.20%)

Matriz de confusão:

	0	1
0	24846(72.45%)	846(2.47%)
1	1279(3.73%)	7325(21.36%)

Já durante o treinamento da RNA com as bases de treinamento, teste e validação reduzidas, obteve-se os seguintes resultados:

Tabela 2 - Progresso do Treinamento (taxa de acerto em %)

Épocas	Tempo de Treino	Bases		
		Treino	Teste	Validação
1000	4m27s	83,06	82,65	82,06
5000	22m27s	89,83	89,93	89,02
10000	44m58s	91,47	91,90	90,97
15000	1h07m32s	92,17	92,72	91,67
20000	1h30m07s	92,53	93,24	92,23
25000	1h52m39s	93,04	93,60	92,71
30000	2h15m12s	93,30	93,78	92,88
35000	2h37m43s	93,48	93,90	93,06
40000	3h00m16s	93,74	94,06	93,25
45000	3h22m53s	94,21	94,12	93,65
50000	3h45m29s	94,52	94,33	93,88
55000	4h08m03s	94,81	94,59	94,05

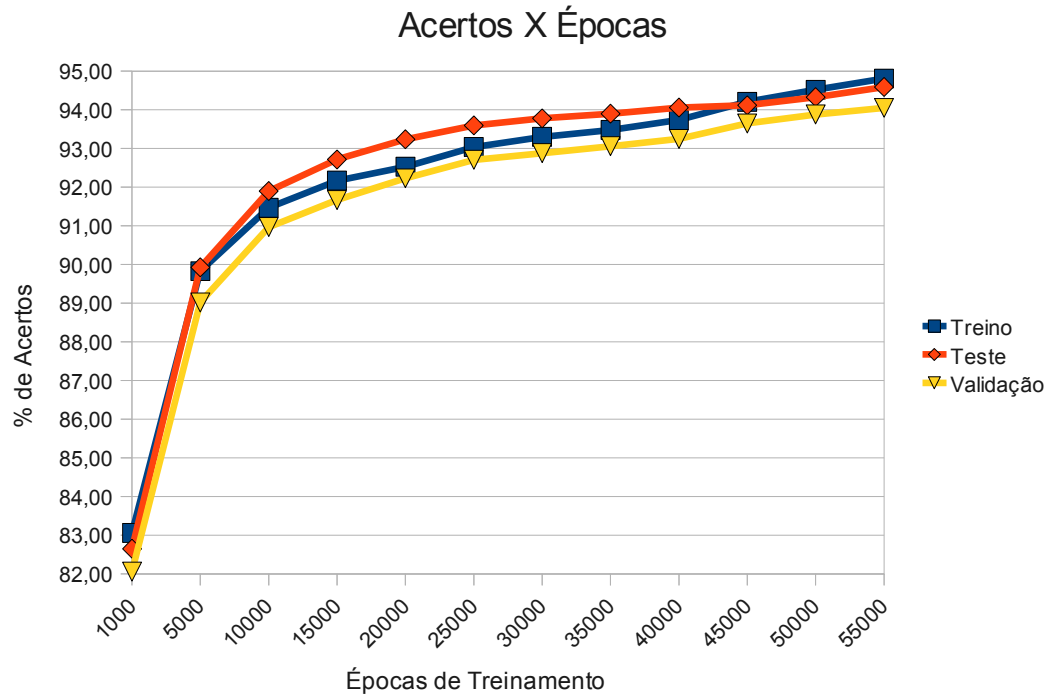


Figura 14 - Gráfico do progresso de Treinamento das Bases

E suas matrizes de confusão foram:

 BASE DE TREINO com 17200 amostras
 acertos: 16308 94.81% (erros: 892 5.19%)
 Matriz de confusão:

	0	1
0	7842(45.59%)	758(4.41%)
1	134(0.78%)	8466(49.22%)

 BASE DE TESTE com 17200 amostras
 acertos: 16269 94.59% (erros: 931 5.41%)
 Matriz de confusão:

	0	1
0	7877(45.80%)	723(4.20%)
1	208(1.21%)	8392(48.79%)

 BASE DE VALIDAÇÃO com 17210 amostras
 acertos: 16180 94.02% (erros: 1030 5.98%)
 Matriz de confusão:

	0	1
0	7796(45.30%)	810(4.71%)
1	220(1.28%)	8384(48.72%)

Além de possuir um melhor desempenho nas bases de treinamento, teste e validação, a segunda rede também teve um melhor resultado na matriz de confusão da base de comparação, além de apresentar os melhores resultados em um menor tempo de treinamento:

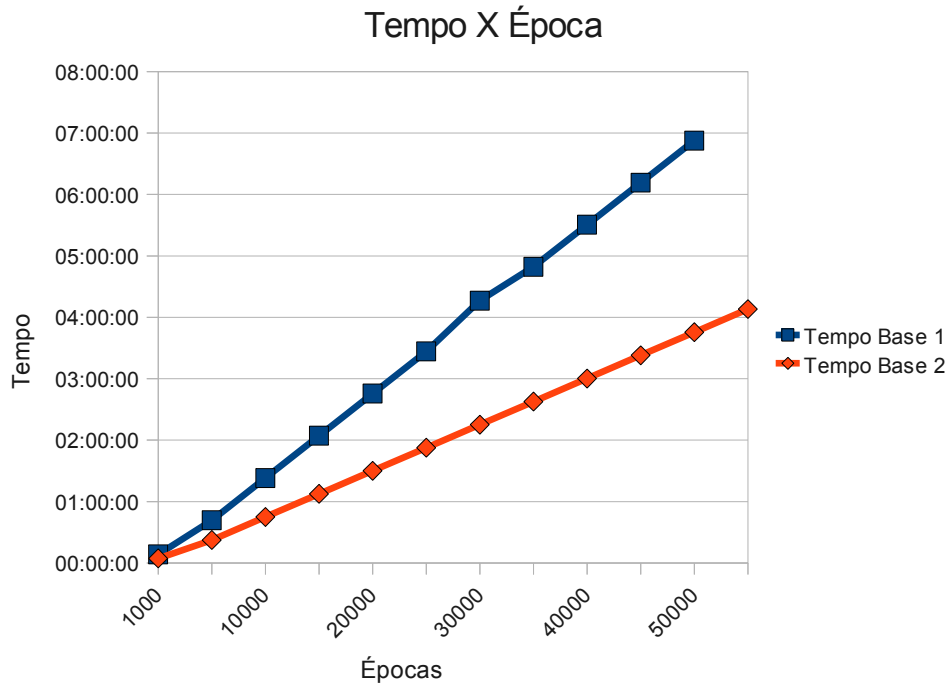


Figura 15 - Gráfico do Tempo de Aprendizado X Épocas de Treinamento

Matriz de confusão da base de comparação submetida à primeira rede:

 BASE DE COMPARAÇÃO com 484 amostras
 acertos: 455 94.01% (erros: 29 5.99%)

Matriz de confusão:

	0	1
0	418(86.36%)	18(3.72%)
1	11(2.27%)	37(7.64%)

Matriz de confusão da base de comparação submetida à segunda rede:

 BASE DE TESTE - Não há base de teste carregada
 BASE DE VALIDAÇÃO com 484 amostras
 acertos: 440 90.91% (erros: 44 9.09%)

Matriz de confusão:

	0	1
0	399(82.44%)	44(9.09%)
1	0(0.00%)	41(8.47%)

Finalmente, após a definição da rede a ser utilizada e submeter a base de comparação à rede, obteve-se então a seguinte matriz de divergência, que apresenta a diferença de classificação entre os classificadores:

Tabela 3 – Taxa de Divergência entre Classificadores

Classificador	Especialista 1-1°	Especialista 1-2°	MLP	Especialista 2	Regras	FAN
Especialista 1-1°		5,17%	9,09%	8,06%	2,89%	17,15%
Especialista 1-2°	5,17%		10,54%	11,16%	7,23%	19,42%
MLP	9,09%	10,54%		12,19%	10,33%	15,08%
Especialista 2	8,06%	11,16%	12,19%		9,71%	16,94%
Regras	2,89%	7,23%	10,33%	9,71%		16,32%
FAN	17,15%	19,42%	15,08%	16,94%	16,32%	
média divergências	8,47%	10,70%	11,45%	11,61%	9,30%	16,98%

Sendo:

Especialista 1 – 1° : Primeira classificação do especialista;

Especialista 1 - 2° : Segunda classificação do mesmo especialista em outra oportunidade, em diferente situação.

MLP: Classificação da RNA do tipo MLP

Especialista 2: Classificação realizada por outro especialista

Regras: Classificação realizada por um sistema baseado em regras

FAN: Classificação realizada pela RNA FAN

Média divergência: A média de divergência entre um classificador e todos os outros classificadores.

Tendo a classificação do Especialista 1-1° como a melhor por tratar-se da realizada em melhores condições, foi comparada a matriz de confusão entre esta e os outros classificadores.

Tabela 4 – Taxa de Divergência de Diagnósticos Anormal entre Especialista 1 - 2° e outros

Classificador	Especialista 1-1°	Especialista 1-2°	MLP	Especialista 2	Regras	FAN
Especialista 1-1°		0,62%	0,00%	4,13%	2,07%	5,17%

Na tabela 4, temos o resultado dos que foram classificados como resultado anormal de exame pelo Especialista 1 na primeira classificação e que os outros classificadores classificaram de maneira diferente, diagnosticando o exame como resultado normal.

Tabela 5 – Taxa de Divergência de Diagnósticos Normal entre Especialista 1 - 2º e outros

Classificador	Especialista 1-1º	Especialista 1-2º	MLP	Especialista 2	Regras	FAN
Especialista 1-1º		4,55%	9,09%	3,93%	0,83%	11,98%

Na tabela 5, temos o resultado dos que foram classificados como resultado normal de exame pelo Especialista 1 na primeira classificação e que os outros classificadores classificaram de maneira diferente, diagnosticando o exame como resultado anormal.

Outro exemplo de comparação entre um sistema de classificação em RNA e um sistema de classificação por regras estudado por HUAN, LIU, uma base de dados sobre câncer de mama obteve um máximo de 93% de acertos utilizando o sistema baseado em regras, enquanto que utilizando RNA do tipo MLP, as mesmas bases para treino e teste, chega a 97% de acertos com cerca de 30 segundos de treinamento.

5- CONCLUSÕES

Através dos resultados dos testes e comparações entre diferentes métodos de avaliação dos exames, pode-se extrair algumas conclusões quanto à decisão de qual rede utilizar na implementação do sistema de apoio à decisão. Além de simplesmente avaliar os resultados absolutos deve-se também utilizar a análise dos erros apresentados e suas comparações.

Durante o treinamento das redes, foram obtidos resultados significativos de acertos nas bases de teste e validação que chegaram a 95%. Considerando que a rede nunca aprendeu com estas duas bases, que a diferença de classificação entre um mesmo especialista em ocasiões diferentes chega a pouco mais de 5% e que a diferença de classificação entre dois especialistas diferentes chega a 8%, o nível de acerto de quase 95% é extremamente satisfatório.

Outro resultado bastante considerável foi a concordância de 100% entre a melhor classificação do especialista e a MLP na parte de classificação de resultados anormais, ou seja, comparam-se na detecção de exames que realmente precisam ser reavaliados microscopicamente. Vê-se também a necessidade da melhora da rede MLP ao classificar os exames de resultados normais em relação a primeira classificação do especialista, o que gera um índice pouco maior de reanálise microscópica desnecessária.

Comparando-se as diferenças entre diferentes especialistas, diferentes situações ambientais e diferentes sistemas de apoio à decisão, vê-se proporções de divergências semelhantes, o que indica o sistema de apoio à decisão de exames laboratoriais de urina utilizando RNA é de grande auxílio na decisão de especialistas, pois apresenta uma sugestão de diagnóstico para cada um dos exames, principalmente quando se trata de uma grande quantidade de exames a serem diagnosticados, uma tarefa manual e repetitiva e altamente sujeita à erros.

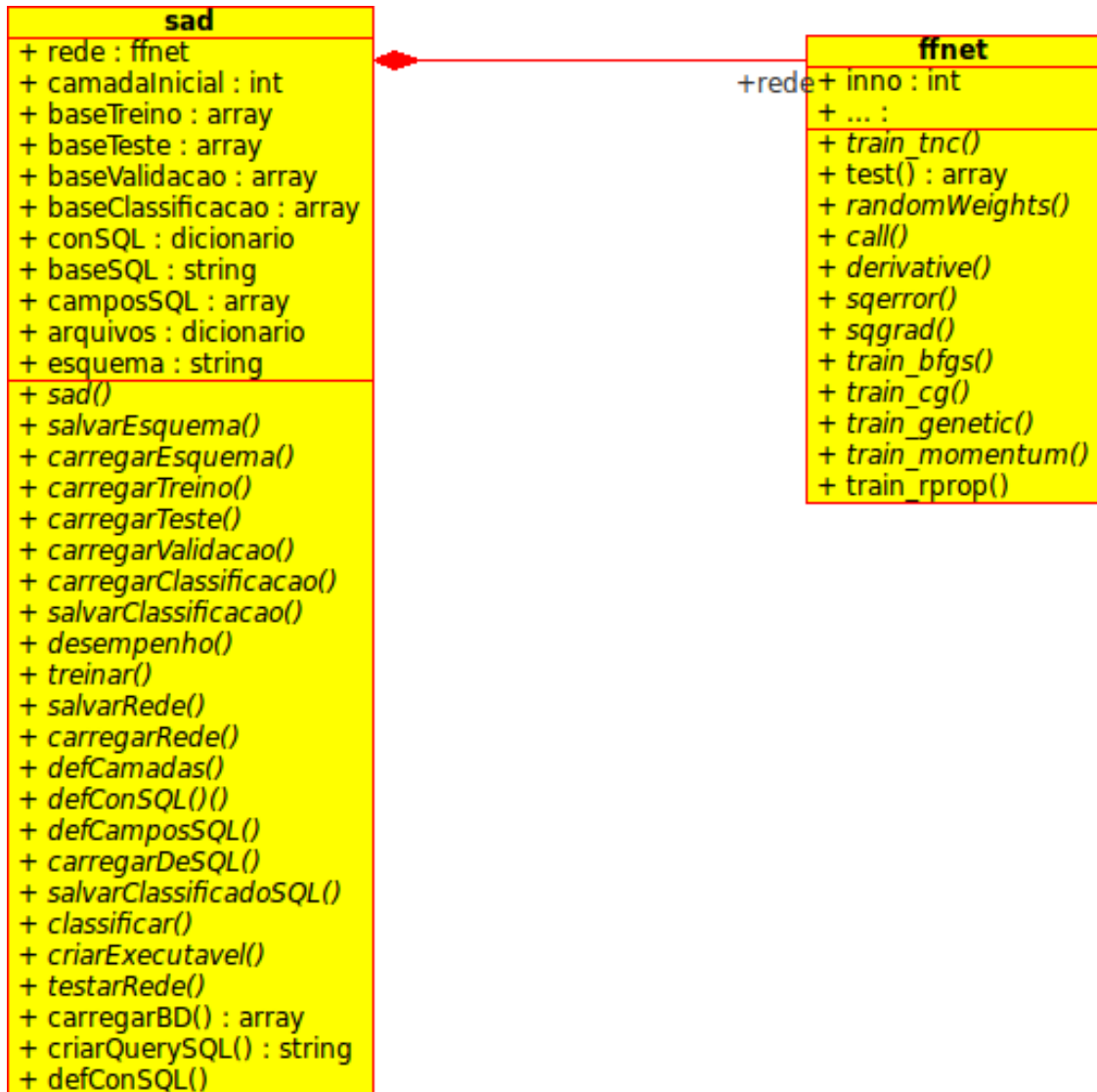
Mais um item a ser levado em consideração é que para o desenvolvimento desse tipo de ferramenta ainda se exige um conhecimento razoável das RNA, o que limita consideravelmente a capacidade de estudiosos de outras áreas desenvolverem soluções nessa linha de estudo. A exigência de conhecimento de configurações e particularidades das RNA deve encorajar especialistas da área de informática a desenvolver ferramentas mais práticas e que exijam menor conhecimento da área por especialistas de outras áreas.

6 - PROJETOS FUTUROS

- Estudo dos resultados de desempenho da rede e verificação na necessidade ou não de continuidade no treinamento e, caso positivo, o desenvolvimento das ferramentas para fazê-lo;
- Estudos mais aprofundados dos variados tipos de rede para melhores configurações, testes e utilização das mesmas;
- Inclusão de normalização e substituição automáticos de valores de string para números inteiros ou pontos flutuantes, pois a rede só aceita este tipo de valores para treinamento;
- Pesquisa na aplicação do sistema para outros tipos de resultados de exames.

7 - DIAGRAMAS

7.1 - DIAGRAMA DE CLASSES



7.2 - DIAGRAMA DE CASOS DE USO

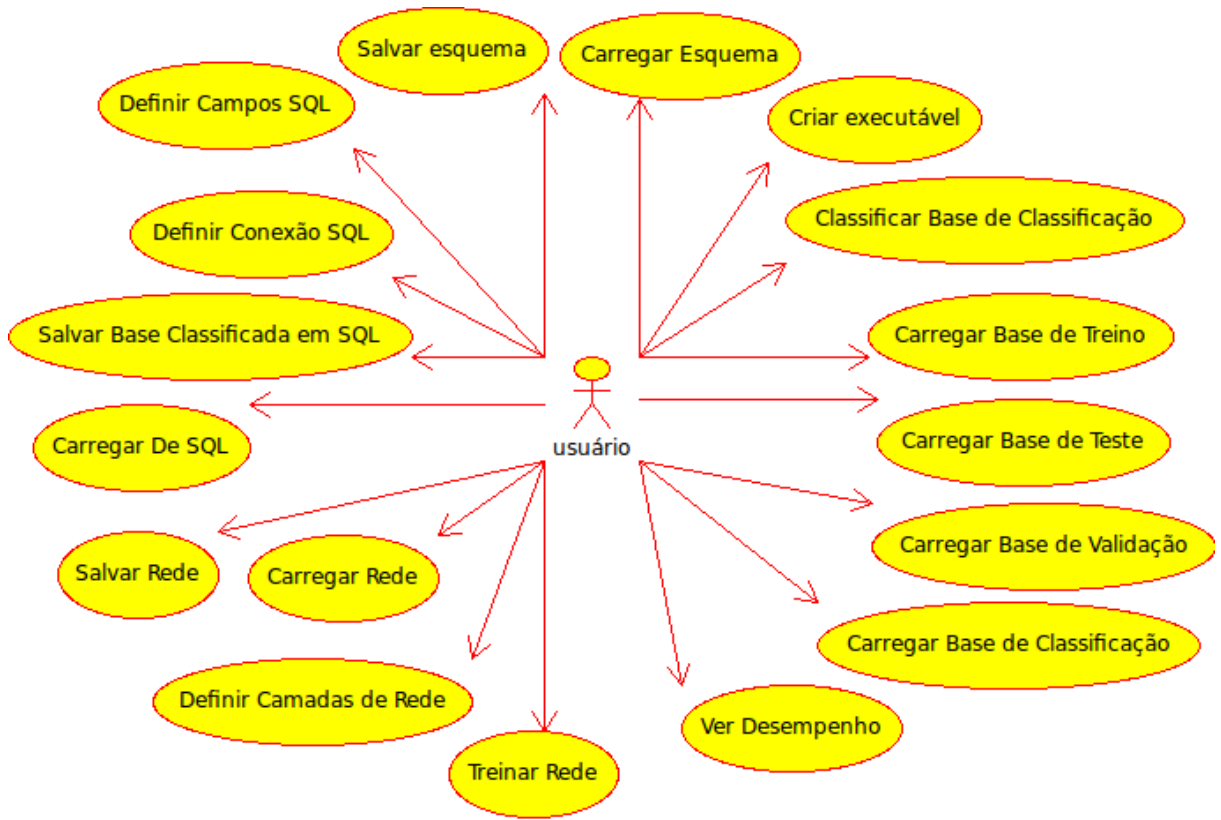


Figura 16 – Projeto do Sistema de Apoio à Decisão



Figura 17 – Definir Campos SQL

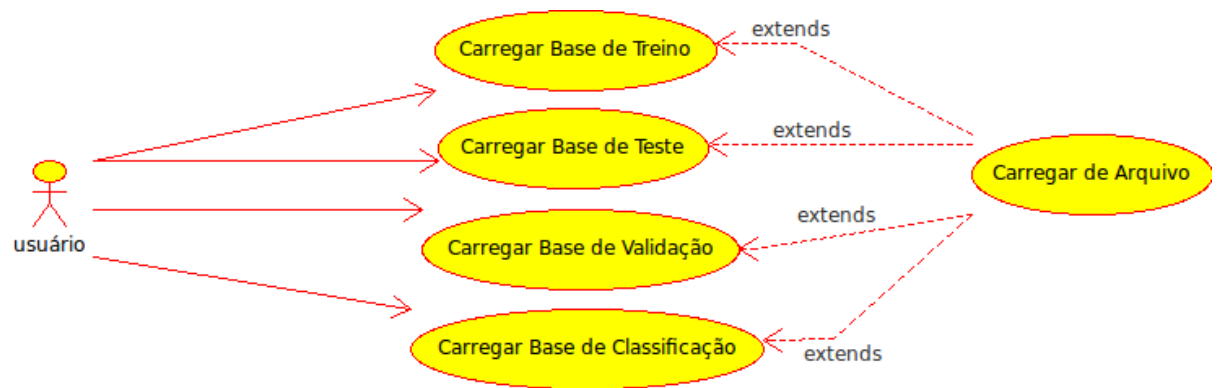


Figura 18 – Carregamento de Bases

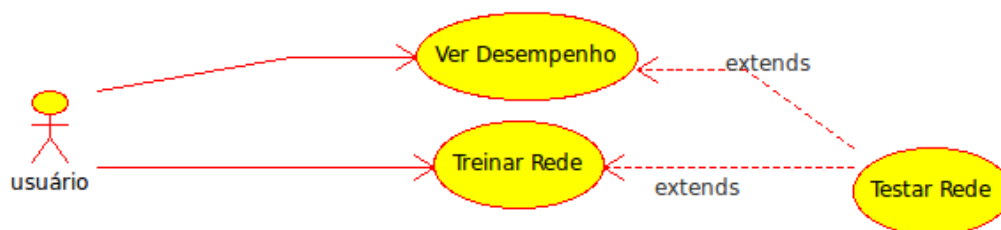


Figura 19 – Resultados de Teste de Rede

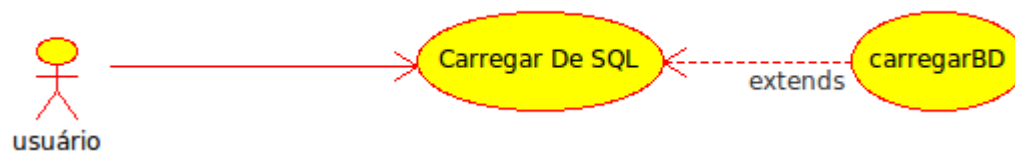


Figura 20 – Carregar de SQL

7.3 - DIAGRAMA DE ESTADO

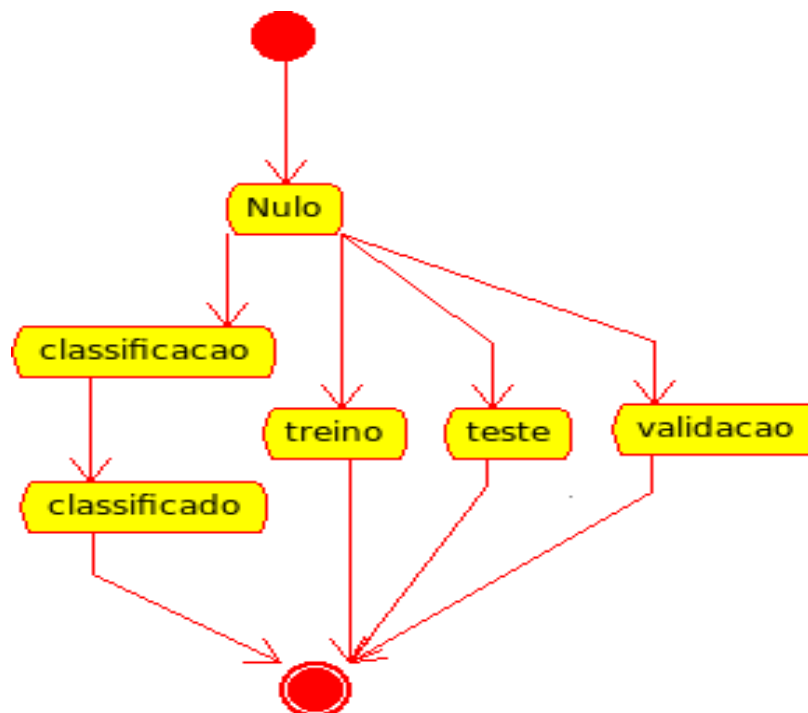


Figura 21 – Estados do atributo BaseSQL da classe SAD

7.4 - DIAGRAMAS DE SEQUÊNCIA

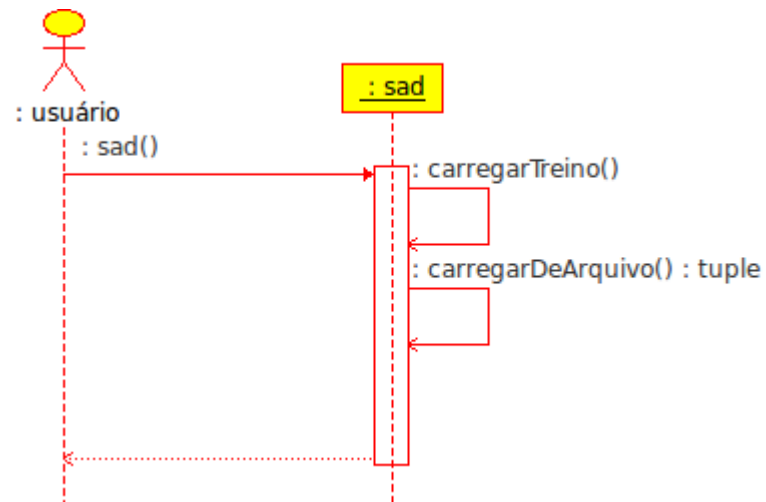


Figura 22 – Carregar Arquivo de Treino

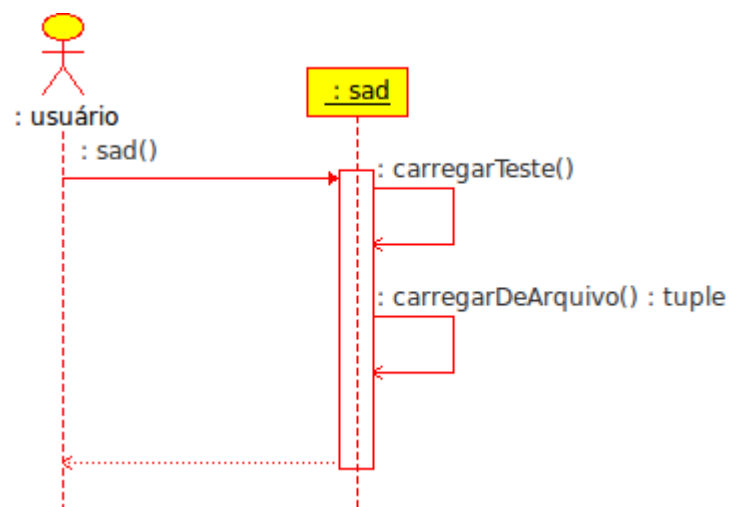


Figura 23 – Carregar Arquivo de Teste

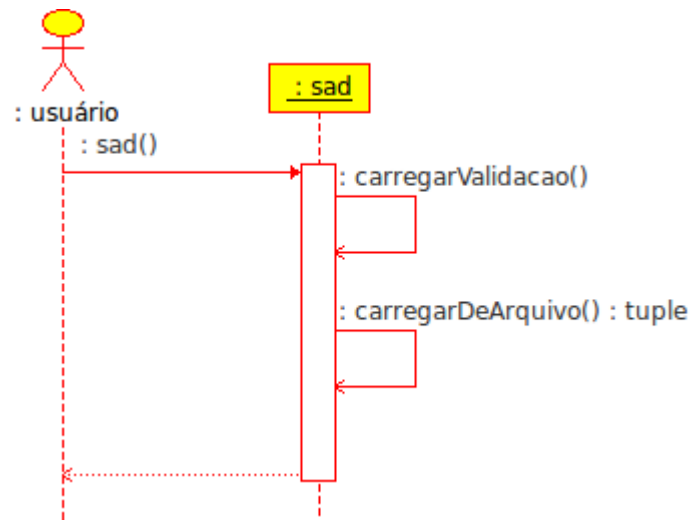


Figura 24 – Carregar Arquivo de Validação

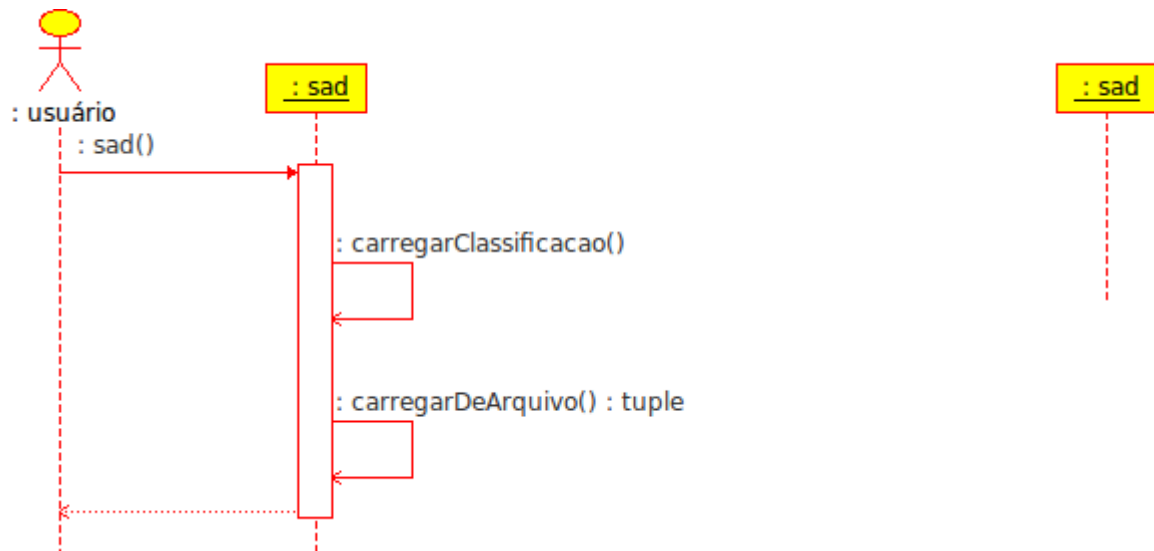


Figura 25 – Carregar Arquivo de Classificação

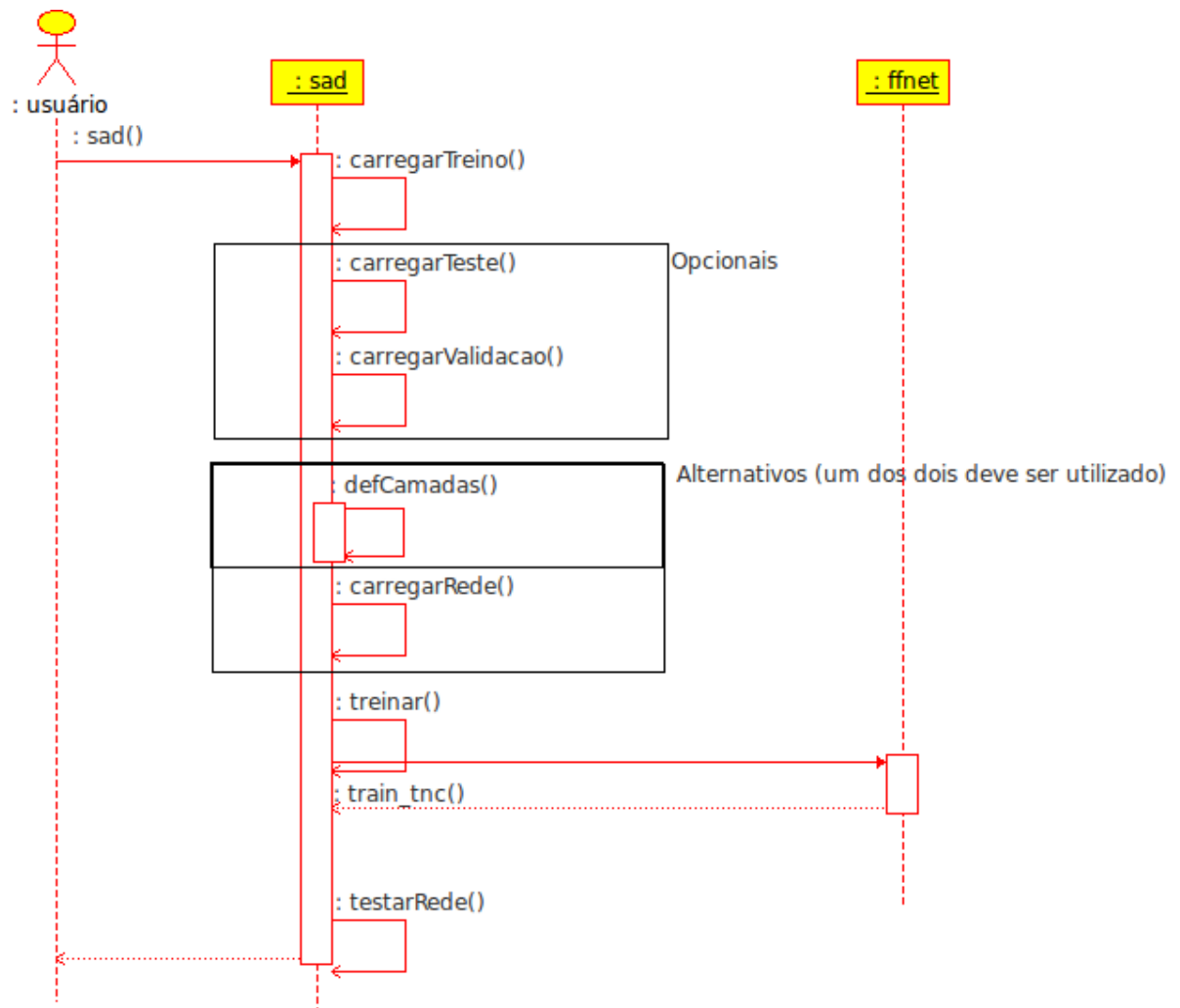


Figura 26 – Treinamento da Rede

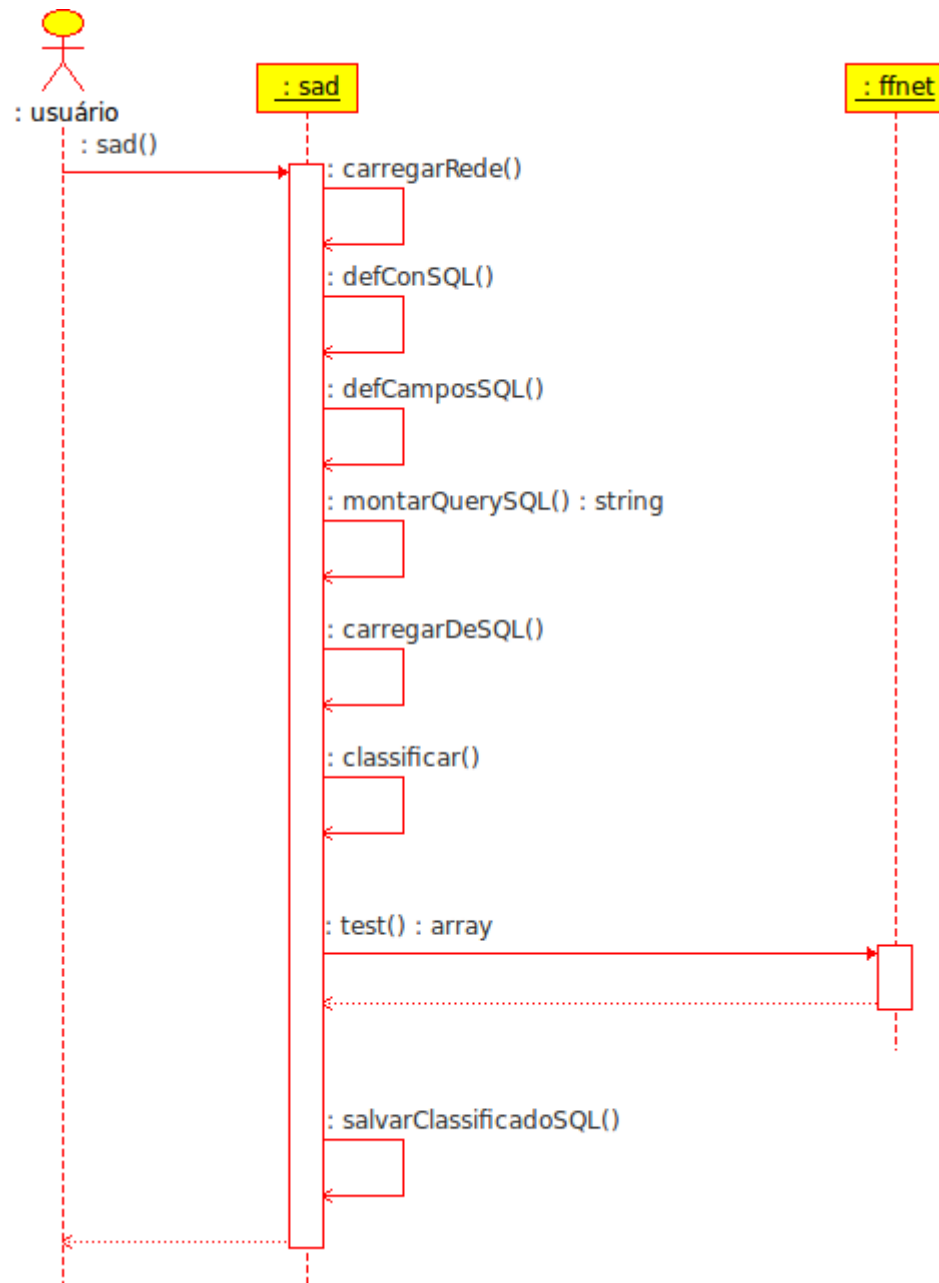


Figura 27 – Salvar Base Classificada em SQL

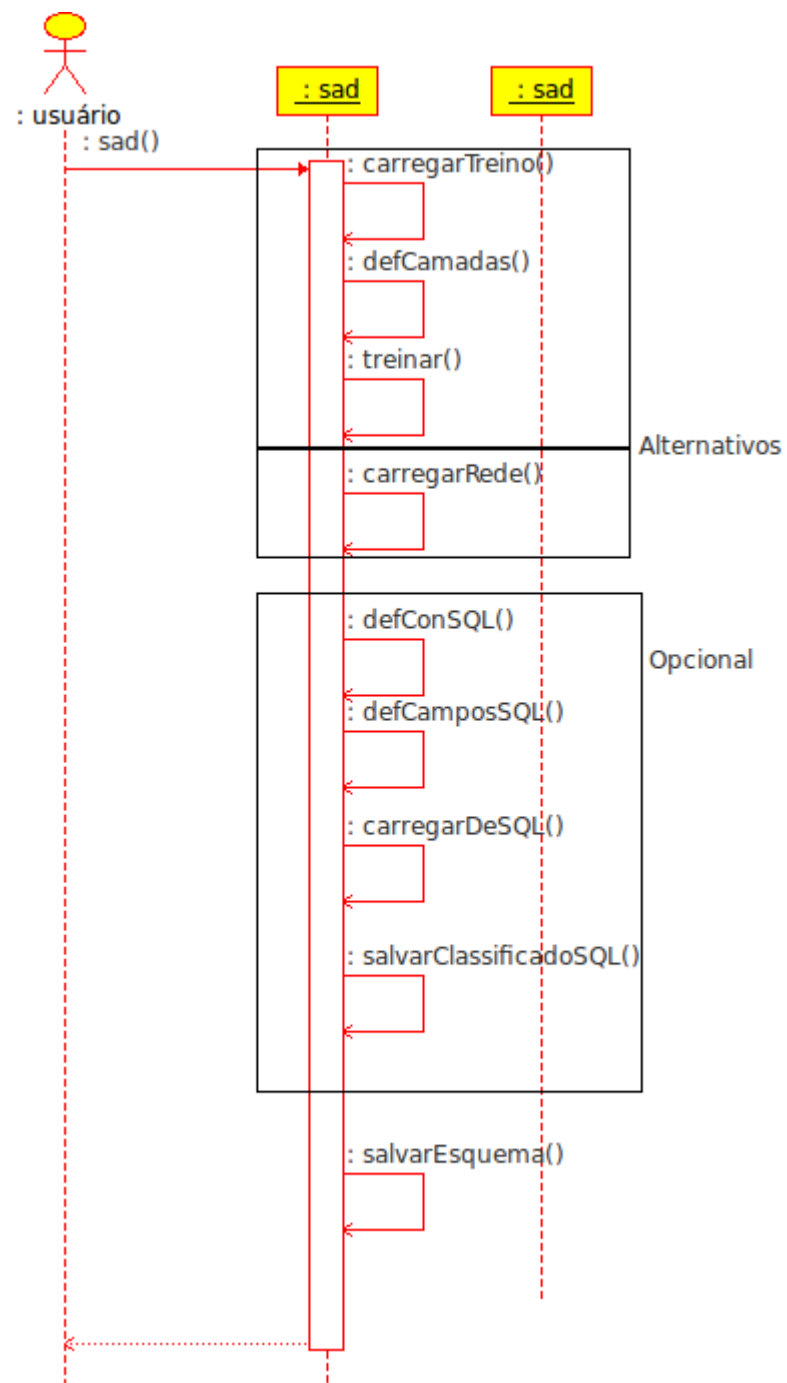


Figura 28 – Salvar Esquema

7.5– DIAGRAMAS DE ATIVIDADES

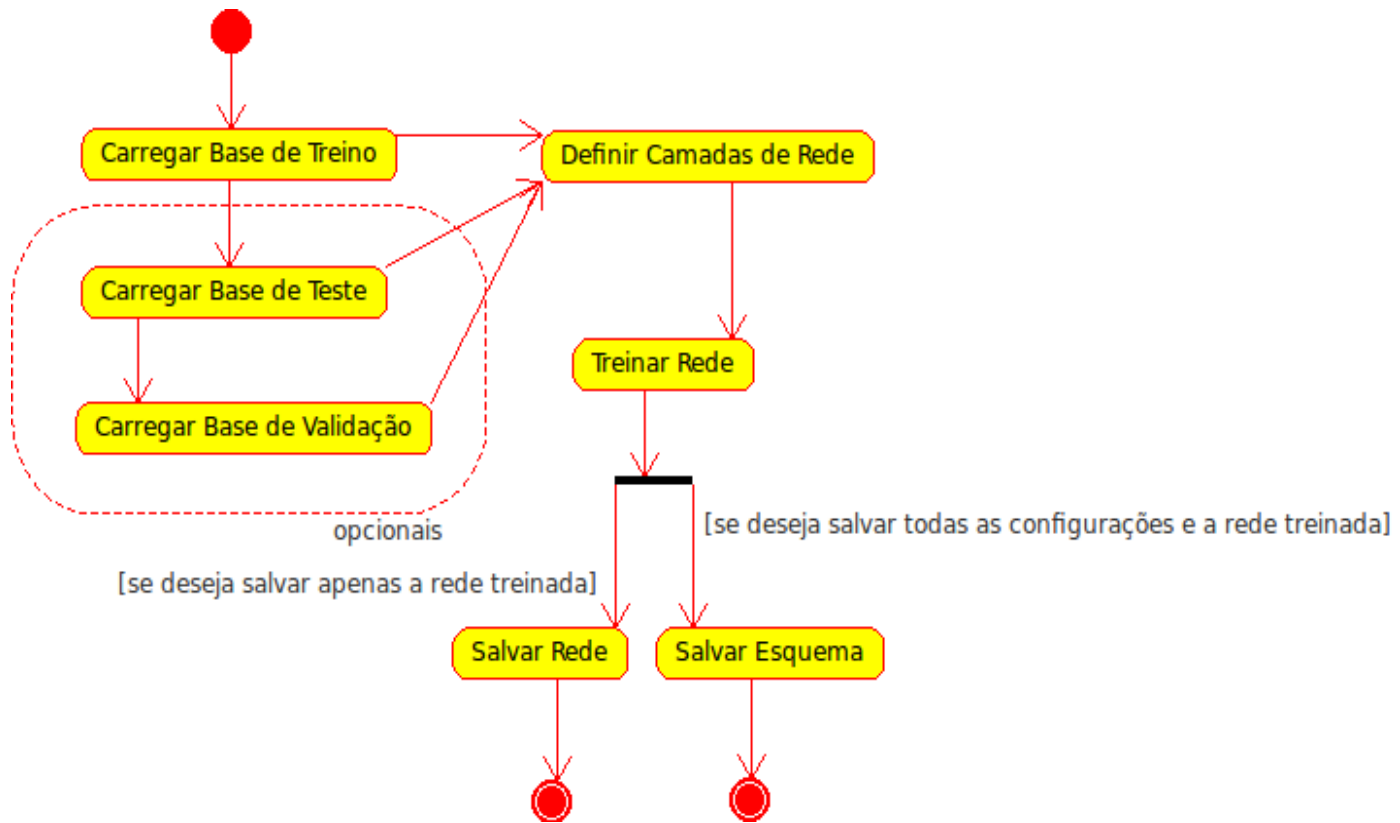


Figura 29 – Atividades: Salvar Rede

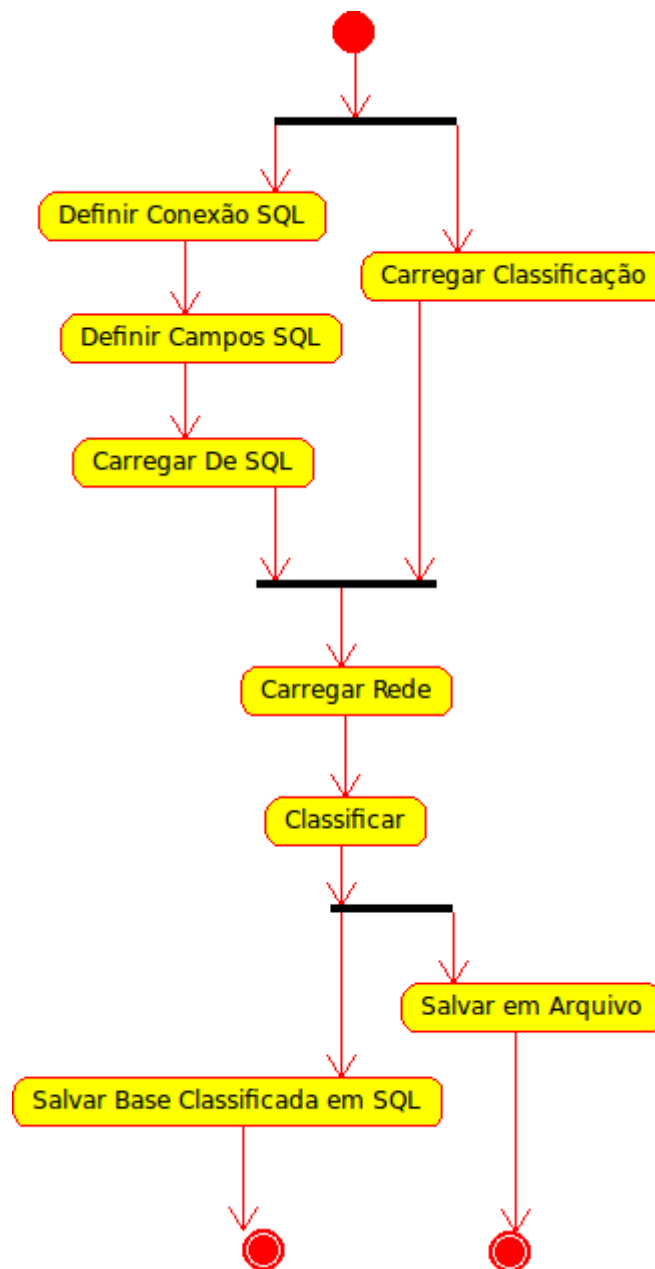


Figura 30 – Classificar Base de Classificação

8 - DESCRIÇÃO DOS CASOS DE USO

8.1 - CASO DE USO CARREGAR BASE DE TREINO

8.1.1 - Descrição Resumida

O usuário poderá carregar a base de treino a partir de um arquivo.

8.1.2 - Fluxo de Eventos

8.1.2.1 - Pré-condições

Existir o arquivo e ter corretamente definido o separador dos campos.

8.1.2.2 - Fluxo Principal

1. O usuário realiza o comando para carregar o arquivo;
2. O sistema carrega o arquivo na base de treino.

8.1.3 - Fluxo de exceção

Caso o arquivo não exista, o separador de campos não seja o correto ou já existir uma rede carregada com parâmetros diferentes de camada inicial, uma mensagem de erro será apresentada.

8.1.4 - Cenário Principal

O usuário instancia a classe e chama a função de carregar arquivo de treino.

8.2 - CASO DE USO CARREGAR BASE DE TESTE

8.2.1 - Descrição Resumida

O usuário poderá carregar a base de teste a partir de um arquivo.

8.2.2 - Fluxo de Eventos

8.2.2.1 - Pré-condições

Existir o arquivo e ter corretamente definido o separador dos campos e existir uma base de treino carregada.

8.2.2.2 - Fluxo Principal

1. O usuário realiza o comando para carregar o arquivo;
2. O sistema carrega o arquivo na base de treino.

8.2.3 - Fluxo de exceção

Caso o arquivo não exista, o separador de campos não seja o correto, já existir uma rede carregada com parâmetros diferentes de camada inicial ou a base de treino já carregada tiver parâmetros de quantidade de características diferente, uma mensagem de erro será apresentada.

8.2.4 - Cenário Principal

O usuário instancia a classe e chama a função de carregar arquivo de teste.

8.3 - CASO DE USO CARREGAR BASE DE VALIDAÇÃO

8.3.1 - Descrição Resumida

O usuário poderá carregar a base de validação a partir de um arquivo.

8.3.2 - Fluxo de Eventos

8.3.2.1 - Pré-condições

Existir o arquivo e ter corretamente definido o separador dos campos e existir uma base de treino carregada.

8.3.2.2 - Fluxo Principal

1. O usuário realiza o comando para carregar o arquivo;
2. O sistema carrega o arquivo na base de treino.

8.3.3 - Fluxo de exceção

Caso o arquivo não exista, o separador de campos não seja o correto, já existir uma rede carregada com parâmetros diferentes de camada inicial ou a base de treino já carregada tiver parâmetros de quantidade de características diferente, uma mensagem de erro será apresentada.

8.3.4 - Cenário Principal

O usuário instancia a classe e chama a função de carregar arquivo de validação.

8.4 - CASO DE USO CARREGAR BASE DE CLASSIFICAÇÃO

8.4.1 - Descrição Resumida

O usuário poderá carregar a base de classificação a partir de um arquivo.

8.4.2 - Fluxo de Eventos

8.4.2.1 - Pré-condições

Existir o arquivo e ter corretamente definido o separador dos campos e existir uma base de treino ou uma rede carregada.

8.4.2.2 - Fluxo Principal

3. O usuário realiza o comando para carregar o arquivo.
4. O sistema carrega o arquivo na base de treino.

8.4.3 - Fluxo de exceção

Caso o arquivo não exista, o separador de campos não seja o correto, já existir uma rede carregada com parâmetros diferentes de camada inicial ou a base de treino já carregada tiver parâmetros de quantidade de características diferente, uma mensagem de erro será apresentada.

8.4.4 - Cenário Principal

O usuário instancia a classe e chama a função de carregar arquivo de validação.

8.5 - CASO DE USO CLASSIFICAR BASE DE CLASSIFICAÇÃO

8.5.1 - Descrição Resumida

Classifica as amostras da base de classificação.

8.5.2 - Fluxo de Eventos

8.5.2.1 - Pré-condições

Ter a base de classificação carregada e uma rede treinada ou carregada.

8.5.2.2 - Fluxo Principal

1. O usuário deve carregar ou treinar uma rede criada;
2. O usuário deve carregar a base de classificação;
3. O usuário deve chamar a função de classificar a base de classificação

8.5.3 - Fluxo de exceção

Caso não exista uma rede treinada o sistema retorna uma mensagem de erro.

8.5.4 - Cenário Principal

Após treinar ou carregar uma rede e carregar a base de classificação, o usuário chama a função de classificar base de classificação.

8.6 - CASO DE USO VER DESEMPENHO

8.6.1 - Descrição Resumida

Apresentação dos resultados de desempenho da rede em relação às bases de treino, teste e validação.

8.6.2 - Fluxo de Eventos

8.6.2.1 - Pré-condições

Ter carregado ou treinado uma rede e ter carregado alguma das bases (treino, teste ou validação)

8.6.2.2 - Fluxo Principal

1. O usuário treina ou carrega uma rede;
2. O usuário carrega uma das bases;
3. O usuário chama a função de ver desempenho.

8.6.3 - Fluxo de exceção

Caso não exista uma rede carregada, o sistema retorna uma mensagem de erro. Caso não exista uma base carregada, nenhum resultado será exibido.

8.6.4 - Cenário Principal

Após ter carregado uma rede treinada e carregar ao menos uma das bases desejadas, chamar a função de ver desempenho.

8.7 - CASO DE USO TREINAR REDE

8.7.1 - Descrição Resumida

Treinar uma rede a partir de uma base de treino.

8.7.2 - Fluxo de Eventos

8.7.2.1 - Pré-condições

Definir ou carregar uma rede e ter a base de treino carregada.

8.7.2.2 - Fluxo Principal

1. O usuário carrega a base de treino;
2. O usuário cria uma rede;
3. O usuário chama a função de treinar rede;
4. O usuário para o treino assim que desejar.

8.7.2.2 - Fluxos Alternativos

Alternativo ao passo 2:

1. O usuário carrega uma rede já existente

8.7.3 - Fluxo de exceção

Caso não exista uma rede definida ou não exista uma base de treino carregada, o sistema retorna uma mensagem de erro.

8.7.4 - Cenário Principal

Após ter definido uma rede e uma base de treino, o usuário chama a função treinar rede, e assim que desejar pode para o treinamento com um ENTER.

8.8 - CASO DE USO DEFINIR CAMADAS DE REDE

8.8.1 - Descrição Resumida

O usuário define as camadas intermediárias (ocultas) da rede.

8.8.2 - Fluxo de Eventos

8.8.2.1 - Pré-condições

Não há.

8.8.2.2 - Fluxo Principal

1. O usuário define as camadas de rede por um array.

8.8.3 - Fluxo de exceção

Caso o usuário não passe um array de inteiros como parâmetro, uma mensagem de erro será apresentada.

8.8.4 - Cenário Principal

Através da função definir camadas de rede, o usuário indica através de um array de inteiros os números de perceptrons de cada camada da rede.

8.9 - CASO DE USO CARREGAR REDE

8.9.1 - Descrição Resumida

A partir de um arquivo com os dados salvos de uma rede, o usuário pode carregar essa rede.

8.9.2 - Fluxo de Eventos

8.9.2.1 - Pré-condições

Ter a rede a ser carregada salva em um arquivo e caso exista uma base de treino já carregada, os parâmetros de camada inicial devem ser os mesmos.

8.9.2.2 - Fluxo Principal

1. O usuário carrega uma rede a partir de um arquivo.

8.9.3 - Fluxo de exceção

Caso uma base de treino já esteja carregada e os parâmetros de camada inicial da base e da rede não forem os mesmos, uma mensagem de erro será mostrada e a rede não será carregada.

8.9.4 - Cenário Principal

Através da função carregar rede, o usuário poderá carregar uma rede salva em um arquivo.

8.10 - CASO DE USO SALVAR REDE

8.10.1 - Descrição Resumida

O usuário pode salvar uma rede configurada ou treinada.

8.10.2 - Fluxo de Eventos

8.10.2.1 - Pré-condições

Existir uma rede definida.

8.10.2.2 - Fluxo Principal

1. O usuário define uma rede;
2. O usuário treina a rede a partir de uma base de treino;
3. O usuário salva a rede definindo o nome do arquivo.

8.10.2.2 - Fluxos Alternativos

Fluxo alternativo ao passo 1:

1. O usuário carrega uma rede;

8.10.3 - Fluxo de exceção

Caso não exista uma rede definida ou carregada, o sistema retorna uma mensagem de erro.

8.10.4 - Cenário Principal

Após carregar ou definir uma rede e treiná-la ou não, o usuário salva a rede chamando a função de salvar rede e definindo o nome do arquivo em que os dados da rede ficarão armazenados.

8.11 - CASO DE USO CARREGAR DE SQL

8.11.1 - Descrição Resumida

O usuário pode carregar dados de uma base SQL.

8.11.2 - Fluxo de Eventos

8.11.2.1 - Pré-condições

Ter definido os parâmetros de conexão com o banco de dados SQL e os parâmetros da query SQL, como nome da tabela, as colunas de dados e a base onde serão carregados os dados.

8.11.2.2 - Fluxo Principal

1. O usuário define os parâmetros de conexão com o banco SQL;
2. O usuário define os parâmetros da query SQL;
3. O usuário chama a função de carregar de SQL.
4. O sistema armazena os dados em uma base definida pelo usuário.

8.11.2.2 - Fluxos Alternativos

Fluxo alternativo aos passos 1, 2 e 3:

1. Usuário carrega esquema já salvo.

8.11.3 - Fluxo de exceção

Caso não estejam definidos os parâmetros de conexão SQL, query SQL ou o banco de dados esteja sem comunicação, uma mensagem de erro é apresentada.

8.11.4 - Cenário Principal

Após definir os parâmetros de conexão e query com o banco de dados, o usuário chama a função carregar de SQL e o sistema carrega os dados na base

definida.

8.1 - CASO DE USO SALVAR BASE CLASSIFICADA EM SQL

8.12.1 - Descrição Resumida

O usuário salva a classificação da base de classificação em um banco de dados SQL.

8.12.2 - Fluxo de Eventos

8.12.2.1 - Pré-condições

Existir uma rede de classificação já classificada.

8.12.2.2 - Fluxo Principal

1. O usuário carrega a base de classificação;
2. O usuário carrega a rede;
3. O usuário classifica a base de classificação;
4. O usuário chama a função de salvar base classificada em SQL;
5. O sistema salva as classificações da base de classificação no banco de dados definido.

8.12.2.2 - Fluxos Alternativos

1. O usuário carrega um esquema onde a base já esteja definida como classificada.

8.12.3 - Fluxo de exceção

Caso não exista uma base de classificação já classificada, o sistema retorna uma mensagem de erro.

8.12.4 - Cenário Principal

Após classificar uma rede, o usuário chama a função salvar base classificada em SQL e o sistema salva as classificações desta base no banco de dados SQL.

8.13 - CASO DE USO DEFINIR CONEXÃO SQL

8.13.1 - Descrição Resumida

O usuário define os parâmetros de conexão com o banco SQL.

8.13.2 - Fluxo de Eventos

8.13.2.1 - Pré-condições

O banco de dados SQL deve estar comunicável e os parâmetros passados para a conexão devem ser válidos.

8.13.2.2 - Fluxo Principal

1. O usuário define os parâmetros de conexão com o banco de dados SQL através de um array;
2. O sistema testa a conexão com o banco de dados SQL e salva a configuração.

8.13.2.2 - Fluxos Alternativos

Fluxo alternativo ao passo 1 :

1. O usuário apenas chama a função de definir a conexão SQL sem passar os parâmetros;
2. O sistema solicita que o usuário entre com os parâmetros um a um;

8.13.3 - Fluxo de exceção

Caso os parâmetros de conexão sejam inválidos ou o banco de dados SQL estiver incomunicável, o sistema retorna uma mensagem de erro.

8.13.4 - Cenário Principal

O usuário define os parâmetros de conexão SQL para busca de dados.

8.14 - CASO DE USO DEFINIR CAMPOS SQL

8.14.1 - Descrição Resumida

O usuário define as colunas com os dados e a query para que sejam carregados do banco SQL.

8.14.2 - Fluxo de Eventos

8.14.2.1 - Pré-condições

O banco de dados SQL deve estar comunicável e os parâmetros de conexão com o banco devem estar definidos.

8.14.2.2 - Fluxo Principal

1. O usuário define os parâmetros de conexão com o banco SQL;
2. O usuário define os parâmetros da query SQL através de um array.
3. O sistema testa a conexão com o banco de dados e se a query é válida, e salva os parâmetros.

8.14.2.2 - Fluxos Alternativos

Fluxo alternativo ao passo 2:

1. O usuário chama a função definir campos SQL sem a passagem de parâmetros;
2. O sistema solicita um a um os parâmetros.

8.14.3 - Fluxo de exceção

Caso os parâmetros da query sejam inválidos ou o banco de dados esteja

sem comunicação, o sistema retorna uma mensagem de erro.

8.14.4 - Cenário Principal

Após definir os parâmetros da conexão com o banco de dados SQL, o usuário define e valida os campos da query SQL que irá carregar os dados do banco SQL.

8.15 - CASO DE USO SALVAR ESQUEMA

8.15.1 - Descrição Resumida

O usuário salva o esquema de configuração ou classificação da base de classificação em um arquivo.

8.15.2 - Fluxo de Eventos

8.15.2.1 - Pré-condições

Existir algum parâmetro da rede ou das bases definido.

8.15.2.2 - Fluxo Principal

1. O usuário carrega alguma base ;
2. O usuário define algum parâmetro;
3. O usuário chama a função salvar esquema, indicando o nome do arquivo onde serão salvas as configurações atuais do sistema;
4. O sistema salva as configurações atuais do esquema no arquivo indicado.

8.15.3 - Fluxo de exceção

Caso não exista nenhum parâmetro definido ou base carregada para serem salvos ou o usuário não defina o nome do esquema a ser salvo, o sistema retorna

uma mensagem de erro.

8.15.4 - Cenário Principal

Após carregar alguma base, rede ou definir algum parâmetros, o usuário utiliza a função salvar esquema para salvar em um arquivo todas as configurações atuais do sistema.

8.16 - CASO DE USO CARREGAR ESQUEMA

8.16.1 - Descrição Resumida

O usuário carrega um esquema a partir de um arquivo de esquema.

8.16.2 - Fluxo de Eventos

8.16.2.1 - Pré-condições

Os parâmetros definidos no esquema continuarem sendo válidos.

8.16.2.2 - Fluxo Principal

1. O usuário chama a função carregar esquema indicando o nome do arquivo de esquema previamente existente.

8.16.3 - Fluxo de exceção

Caso algum parâmetro da configuração do arquivo de esquema não seja mais válido, como não existirem mais os arquivos para carregamento das bases ou os parâmetros de conexão SQL não sejam mais válidos, o sistema retorna uma mensagem de erro.

8.16.4 - Cenário Principal

A partir de um arquivo de esquema previamente salvo, o usuário carrega as

configurações salvas neste arquivo através da função carregar esquema.

8.17 - CASO DE USO CRIAR EXECUTÁVEL

8.17.1 - Descrição Resumida

O usuário cria um executável para chamar um esquema apenas com a chamada deste arquivo.

8.17.2 - Fluxo de Eventos

8.17.2.1 - Pré-condições

Existir um esquema já salvo para ser carregado.

8.17.2.2 - Fluxo Principal

1. O usuário define os parâmetros das bases;
2. O usuário define os parâmetros da rede;
3. O usuário define os parâmetros conexão SQL;
4. O usuário define os parâmetros dos campos SQL;
5. O usuário define e classifica a base de classificação;
6. O usuário salva a base classificada em SQL;
7. O usuário salva o esquema;
8. O usuário chama a função criar executável passando o esquema como parâmetro;
9. O sistema cria um arquivo que ao ser executado, carrega o esquema passado como parâmetro;

8.17.2.2 - Fluxos Alternativos

Fluxo alternativo aos passos 1 a 7:

1. O usuário carrega um esquema;

8.17.3 - Fluxo de exceção

Caso tenha indicado um arquivo de esquema inexistente, o sistema retorna uma mensagem de erro

8.17.4 - Cenário Principal

Após definir os parâmetros e salvá-los em um esquema ou carregar um esquema a partir de um arquivo, o usuário chama a função criar executável para criar um arquivo que quando executado carrega o esquema indicado.

9 - REFERÊNCIAS BIBLIOGRÁFICAS

- 1 - WOJCIECHOWSKI, M., **Feed-forward Neural Network for Python**, Technical University of Lodz (Poland), Department of Civil Engineering, Architecture and Environmental Engineering, <http://ffnet.sourceforge.net/>, ffnet-0.6, Março de 2007
- 2 – HEBB, D. , **The Organization of Behavior**. Jon Wiley & Sons, New York, 1949.
- 3 – ZIPSER, D., RUMELHART, D.E., **Feature Discovery by Competitive Learning**, Cognitive Science, 1985
- 4 – ROSENBLATT, F. , **A probabilistic model for information storage and organization in the brain**, Psychological review, 1958
- 5 – MINSKY M. L., PAPERT S. A.,**Perceptrons**. Cambridge, MA: MIT Press, 1969
- 6 – WIDROW, B., LEHR, M. A., **30 years of adaptive neural networks: Perceptron, madaline, and backpropagation**, Proceedings of the IEEE, Citeseer, 1990
- 7 – HOPFIELD, J. J., **Neural networks and physical systems with emergent collective computational abilities**, National Academy Sciences, 1982
- 8 – FLETCHER, R., REEVES, C. M., **Function minimization by conjugate gradients**. The computer journal, 1964
- 9 – KRAMER, M. A., **Diagnosis using backpropagation neural networks -analysis and criticism**, Computers & chemical engineering, 1990
- 10 – SBRN - SOCIEDADE BRASILEIRA DE REDES NEURAIAS, <http://www.sbrn.org.br/> acessado em maio de 2010
- 11 – FERNEDA, E., **Redes neurais e sua aplicação em sistemas de recuperação de informação**, Ci. Inf. vol.35 no.1 Brasília Janeiro/Abril, 2006

- 12 - HAYKIN, S. S., **Livro: Redes Neurais, Princípios e Prática**, Bookman, 2001 - 900 páginas
- 13 – HUAN, LIU, **A Family of Efficient Rule Generators**, Department of Information Systems and Computer Science, National University of Singapore.
- 14 – PAULA, MAURÍCIO BRAGA DE, **Reconhecimento de palavras faladas utilizando Redes Neurais Artificiais**, Universidade Federal de Pelotas, dezembro de 2000.
- 15 – RAITZ, ROBERTO TADEU, **Free Associative Neurons – FAN: uma abordagem para reconhecimento de padrões**, Florianópolis, 1997.
- 16 – VOLANSKI, W. ET AL; **Automação em urinálise: uma solução para rotinas de grande porte**, Curitiba 2006.
- 17 – LOPES, HOMERO JACKSON DE JESUS; **O Laboratório Clínico na Avaliação da Função Renal**, Belo Horizonte, 2004.
- 18 – MEDEIROS, LUCIANO FRONTINO DE, **Redes Neurais em Delphi. 2. ed.**, Florianópolis: Visual Books, 2006
- 19 – ZSOLUTIONS; <http://www.zsolutions.com/>, acessado em junho/2010
- 20 – VOLANSKI, W., **Comparação entre diferentes métodos de apoio à decisão a resultados de exames de urina**, Curitiba 2010.