

## **Resumo**

Pesquisas de laboratório, especialmente as que necessitam de sacrifício de animais, normalmente envolvem práticas demoradas que exigem muito trabalho manual por parte do pesquisador. Além disso, com as práticas convencionais é muito difícil simular características desejadas nos animais para analisar qual o resultado da modificação daquelas características em outro aspecto relacionado.

Este trabalho apresenta uma ferramenta que utiliza computação inteligente, notadamente redes neurais artificiais, para auxiliar pesquisadores da área de Biologia na realização de seus experimentos. A ferramenta permite que o experimento seja modelado em uma rede neural artificial, que deve ser treinada com os novos dados do problema específico. Com esta ferramenta é possível classificar parâmetros de entrada e prever valores com base nestes parâmetros.

As simulações feitas com a ferramenta podem servir de complemento para as práticas laboratoriais. Espera-se que o uso desta ferramenta em pesquisas de Biologia possibilite ao pesquisador comprovar mais rapidamente suas conjecturas e possibilite que novas hipóteses sejam elaboradas mais rapidamente.

Os resultados experimentais obtidos e a avaliação favorável feita por uma pesquisadora de Biologia sobre as potencialidades de uma ferramenta como esta, demonstram a viabilidade e utilidade de seu uso por profissionais em laboratórios de Biologia.

## Abstract

Laboratory researches, especially those which require animal sacrifice, usually involve practices that demand much time to be concluded and a lot of manual work by the researcher. Moreover, with the conventional practices it is too hard to simulate desired characteristics in the animals to analyse which result would be generated in other related aspect.

This work presents a tool that uses intelligent computing, notoriously artificial neural networks, to assist researchers in Biology to do their experiments. The tool permits modeling the experiment in an artificial neural network which must be trained with the data of the specific problem. With this tool it is possible to classify input parameters and predict values based on them.

The simulations done with this tool can be used as complement to the laboratorial practices. It is expected that the use of this tool in Biology researches makes possible for the researcher to prove his conjectures and elaborate his hypothesis faster.

The experimental results obtained and the evaluation done by a researcher about the potentialities of a tool like this demonstrates the viability and utility of its usage by the target public.

# Sumário

<b>Índice de Figuras</b>	<b>v</b>
<b>Índice de Tabelas e Equações</b>	<b>vii</b>
<b>Tabela de Símbolos e Siglas</b>	<b>viii</b>
<b>1 Introdução</b>	<b>10</b>
1.1 Objetivos e Metas	11
1.2 Organização do Trabalho	11
<b>2 Referencial Teórico</b>	<b>13</b>
2.1 Inteligência Artificial	13
2.1.1 Definição	13
2.1.2 Técnicas	14
2.1.3 Aplicações	15
2.2 Redes Neurais Artificiais	16
2.2.1 Fatos Históricos	16
2.2.2 Fundamentos Biológicos	16
2.2.3 Neurônio Artificial	18
2.2.4 Caracterização de RNA	20
2.2.5 Classificação de RNA	21
2.2.6 Topologias de RNA	21
2.2.7 Aprendizado	22
2.2.8 <i>Perceptron</i>	24
2.2.9 <i>Multi-Layer Perceptron (MLP)</i>	24
2.3 InteliColheita	25
2.3.1 Descrição do projeto	25
2.3.2 Motor Neural	25
2.3.3 Módulo de Manipulação de Arquivos	26
<b>3 Ferramenta Desenvolvida</b>	<b>28</b>
3.1 Caracterização do problema utilizado	28
3.1.1 Animais	28
3.1.2 Dieta Experimental	28
3.1.3 Modelos Experimentais	29
3.1.4 Impactos da DBR observados no organismo dos animais	29
3.2 Análise de Requisitos	29
3.2.1 Problema e Causas	29
3.2.2 Requisitos	30
3.3 Características	31
3.4 Funcionalidades	31
3.5 Integração com o InteliColheita	39
3.6 Ferramentas e Tecnologias Utilizadas	39

<b>4</b>	<b>Testes e Resultados</b>	<b>40</b>
4.1	Dados Disponíveis	40
4.2	Experimentos	43
4.2.1	Classificação	44
4.2.2	Previsão	44
4.3	Simulações	45
4.3.1	Pré-processamento dos dados de treinamento	45
4.3.2	Criação das redes neurais	47
4.3.3	Treinamento das redes criadas	48
4.3.4	Escolha da melhor rede	49
4.3.5	Simulações	50
4.4	Resultados	50
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>55</b>
5.1	Contribuições	55
5.2	Discussão	56
5.3	Trabalhos Futuros	56

# Índice de Figuras

Figura 1. Neurônio: célula do sistema nervoso [21]	17
Figura 2. Esquema da conexão entre dois neurônios	18
Figura 3. Neurônio artificial de McCulloch e Pitts	19
Figura 4. Generalização do modelo de neurônio de McCulloch e Pitts	19
Figura 5. Não linearidades freqüentemente usadas no modelo de neurônios de uma RNA	19
Figura 6. Exemplo de uma rede neural organizada em camadas	20
Figura 7. Rede direta com 4 camadas [25]	21
Figura 8. Exemplo típico de rede com ciclos	22
Figura 9. Exemplo de rede simétrica	22
Figura 10. Ilustração do aprendizado supervisionado	23
Figura 11. Ilustração do aprendizado não-supervisionado	23
Figura 12. Diagrama de classes resumido do motor neural	26
Figura 13. Diagrama de classes resumido do módulo de manipulação de arquivos	27
Figura 14. Análise de Problemas e Causas	30
Figura 15. Diagrama de casos de uso da ferramenta desenvolvida	32
Figura 16. Tela principal do InteliLab	32
Figura 17. Menu principal do InteliLab	32
Figura 18. Opção de menu para realizar simulações	33
Figura 19. Tela de realização de simulações	34
Figura 20. Opção de menu para carregar simulações	34
Figura 21. Tela de carregar simulações	35
Figura 22. Opção de menu para criar novas redes neurais	35
Figura 23. Tela de criação de novas redes	36
Figura 24. Opção de menu para carregar as informações de um arquivo de rede neural	36
Figura 25. Tela de alteração dos parâmetros de uma rede neural	37
Figura 26. Opção de menu para treinar uma rede neural	37
Figura 27. Tela de treinamento de uma rede neural	38
Figura 28. Diagrama de classes resumido do InteliLab	39
Figura 29. Exemplo de arquivo de dados para pré-processamento do experimento de classificação	46
Figura 30. Exemplo de arquivo de dados pré-processado do experimento de classificação	46

Figura 31. Exemplo de arquivo de dados para pré-processamento do experimento de predição 47

Figura 32. Exemplo de arquivo de dados pré-processado do experimento de predição 47

# Índice de Tabelas e Equações

Tabela 1. Dados disponíveis de animais nutridos	41
Tabela 2. Dados disponíveis de animais desnutridos	41
Tabela 3. Dados de animais nutridos selecionados para uso nos treinamentos	42
Tabela 4. Dados de animais desnutridos selecionados para uso nos treinamentos	42
Tabela 5. Dados dos animais virtuais criados para uso nos treinamentos	42
Tabela 6. Dados dos animais nutridos não utilizados nos treinamentos	42
Tabela 7. Dados dos animais desnutridos não utilizados nos treinamentos	43
Tabela 8. Configuração de rede 1 para o experimento de classificação	44
Tabela 9. Configuração de rede 2 para o experimento de classificação	44
Tabela 10. Configuração de rede 1 para o experimento de predição	44
Tabela 11. Configuração de rede 2 para o experimento de predição	45
Tabela 12. Configuração das redes neurais criadas para cada experimento	47
Tabela 13. Resultado da classificação utilizando a configuração 1 para novos padrões	50
Tabela 14. Resultado da classificação utilizando a configuração 1 para padrões conhecidos	51
Tabela 15. Resultado da classificação utilizando a configuração 2 para novos padrões	51
Tabela 16. Resultado da classificação utilizando a configuração 2 para padrões conhecidos	52
Tabela 17. Resultado da predição utilizando a configuração 1 para novos padrões	52
Tabela 18. Resultado da predição utilizando a configuração 1 para padrões conhecidos	52
Tabela 19. Resultado da predição utilizando a configuração 2 para novos padrões	53
Tabela 20. Resultado da predição utilizando a configuração 2 para padrões conhecidos	53
Tabela 21. Resultado da predição utilizando a configuração 1 para a rede específica de animais nutridos	53
Tabela 22. Resultado da predição utilizando a configuração 1 para a rede específica de animais desnutridos	54
Equação 1. Cálculo da entrada efetiva do neurônio artificial	18

# Tabela de Símbolos e Siglas

IA – Inteligência Artificial  
RNA – Redes Neurais Artificiais  
DSC – Departamento de Sistemas Computacionais  
POLI – Escola Politécnica de Pernambuco  
UPE – Universidade de Pernambuco  
UFPE – Universidade Federal de Pernambuco  
STI – Sistemas Tutores Inteligentes  
IAS – Inteligência Artificial Simbólica  
IAC – Inteligência Artificial Conexionista  
MLP – Multi-Layer Perceptron (Perceptron Multi-Camada)  
DBR – Dieta Básica Regional  
IDE – Integrated Development Environment (Ambiente Integrado de Desenvolvimento)  
UML – Unified Modeling Language (Linguagem Unificada de Modelagem)  
cm – centímetro  
g – grama  
SNNS – Stuttgart Neural Network Simulator  
MSE – Mean Squared Error (Erro Médio Quadrático)

# Agradecimentos

Agradeço a Deus pelo meu anjo da guarda e por me dar a oportunidade de gozar plenamente das minhas faculdades físicas e intelectuais, sem as quais o caminho que percorri até aqui teria sido bem mais complicado. Agradeço a Ele pelos meus pais, minha irmã, minha namorada, meus amigos e parentes, pessoas maravilhosas com as quais é maravilhoso compartilhar minha vida.

Agradeço aos meus pais por seu esforço e dedicação para me fazer forte, por seu amor, por sua atenção e por serem sempre a minha fortaleza. Agradeço ao meu pai (João) por ser meu amigo mais sincero e fiel, por acreditar em mim e me ensinar lições tão valiosas. Agradeço a minha mãe (Elaine) por sua paciência, compreensão, cuidado e pelo dengo que tantas vezes me confortou em momentos difíceis. Joãozinho e Dona Elaine amo muito vocês!

Agradeço a minha irmã (Bella) por tanta coisa, especialmente por me amar, me aturar e cuidar de mim de um jeito tão especial que vai ser difícil um dia viver sem. Agradeço pela ajuda nesta reta final da monografia, Bellinha você é espetacular e eu te amo muito!

Agradeço a minha namorada (Marcela) que me manteve fiel ao objetivo de concluir a monografia, que foi sempre compreensiva e amável nesta reta final tão estressante. Namorada, amo você!

Agradeço aos amigos do trabalho, especialmente a Allysson Moraes, a Flávio Valença, a Hugo Brasileiro e a Odomar Brito, que possibilitaram que eu conseguisse 15 dias de férias para me dedicar exclusivamente à monografia.

Agradeço aos demais amigos de Moreno e da POLI, verdadeiros irmãos que Deus colocou em minha vida, agradeço especialmente a Ananias, Léo, Danilo, Felipe, Dani, Ericles, Daniel Gomes, Erick, Zoby e Câmara.

Agradeço ao meu orientador e professor, Fernando Buarque, pelos conselhos, ensinamentos, apoio e incentivo sem os quais eu não teria chegado tão longe. Extendo este agradecimento a todos os professores do DSC que foram fundamentais na minha formação, não citarei o nome de todos, mas guardarei com carinho o valor de tudo que me ensinaram.

Agradeço à minha co-orientadora, a Prof<sup>a</sup>. Silvia Regina, cujo apoio e estímulo foram decisivos na reta final de conclusão desta monografia.

Por fim, agradeço novamente a Deus por ter tanto o que agradecer, pela oportunidade de ingressar numa Universidade e estar agora concluindo este ciclo.

# Capítulo 1

## Introdução

Pesquisas científicas no campo da Biologia que envolvem a utilização de animais de laboratório estão sujeitas às normas de um Comitê de Ética, são dispendiosas e possuem uma duração elevada (*i.e.* alguns meses). Dentre as normas adotadas pela maioria dos Comitês de Ética se destaca a recomendação de que as cobaias não devem ser sacrificadas desnecessariamente, assim como deve haver o esforço para que sejam desenvolvidos métodos alternativos que minimizem o uso daqueles animais [1].

Quando em uma pesquisa científica existe a necessidade do sacrifício de cobaias, as boas práticas fundamentadas na Ética prezam para que o máximo de informações seja coletado nesse processo. A manutenção das cobaias é feita em biotérios – acomodação de animais vivos, para estudos laboratoriais, produção de soros, vacinas, etc. Eles têm custo elevado porque envolvem climatização do ambiente, veterinário para acompanhamento da vida dos animais, materiais de limpeza e alimentação adequada.

Os argumentos acima destacam a importância de se aumentar a produtividade e a eficiência na coleta dos dados provenientes do uso de animais de laboratório, especialmente quando o sacrifício desses animais é necessário. Isto porque é normalmente nas etapas que sucedem os sacrifícios que ocorrem problemas e erros mais significativos, como danificação ou perda do material coletado [2]. Portanto conseguir reunir mais informações realizando menor número de sacrifícios implica utilizar menos animais na pesquisa, diminuindo assim o custo direto e indireto do projeto e ainda, permite preservar vidas animais em acordo com as normas éticas.

O material coletado em um sacrifício pode ser utilizado para analisar aspectos macroscópicos ou microscópicos de alguma alteração provocada por um agente externo ao organismo do animal. Por exemplo, alterações orgânicas podem ser observadas em cobaias submetidas à administração de fármacos como a fluoxetina [3], ou de dietas específicas. São exemplos dessas alterações: retardo no desenvolvimento motor, perda de massa corpórea, diminuição da densidade óssea, redução do comprimento de ossos, má formação de órgãos e alterações comportamentais. Estudos considerando o fornecimento de uma dieta pobre em proteínas por exemplo, indicam a ocorrência de alterações no número de neurônios presente no plexo entérico de animais [4] [5].

Aplicações de Inteligência Artificial (IA) [6] têm sido utilizadas com sucesso nas mais diversas áreas, desde Engenharia à Medicina. O emprego de técnicas de IA que se baseiam no uso de conhecimento prévio para então classificar e prever valores futuros a cerca de um problema,

podem suprir satisfatoriamente muitas das deficiências e dificuldades encontradas por esses pesquisadores que necessitam realizar experimentos laboratoriais utilizando cobaias.

Este trabalho consiste no desenvolvimento de uma ferramenta computacional que utilize técnicas de computação inteligente, para ser utilizada em pesquisas de laboratório na área de Biologia, possibilitando que os pesquisadores realizem simulações utilizando modelos virtuais do problema, em substituição parcial às técnicas puramente laboratoriais.

## 1.1 Objetivos e Metas

O principal objetivo deste trabalho foi desenvolver uma ferramenta computacional, utilizando técnicas de IA apropriadas para o problema, notadamente Redes Neurais Artificiais (RNA) [7], visando auxiliar pesquisadores da área de Biologia no desenvolvimento de suas pesquisas.

Este trabalho utilizou dados de uma pesquisa que avalia os impactos da desnutrição em ratos. Foram definidos dois experimentos para a avaliação da ferramenta desenvolvida, o primeiro é classificatório, nele a ferramenta precisa identificar se um animal é nutrido ou desnutrido de acordo com alguns parâmetros informados. No segundo experimento, a ferramenta será utilizada na predição do valor de uma característica referente às cobaias utilizadas na pesquisa em questão, especificamente a quantidade de neurônios encontrada no plexo entérico dos animais.

O segundo objetivo do trabalho se refere aos experimentos realizados, conforme o parágrafo anterior, e consiste em calibrar a classificação dos animais e a predição da quantidade de neurônios para valores suficientemente próximos aos observados em laboratório, nas condições experimentais investigadas.

Para o usuário-pesquisador, as informações geradas pela ferramenta possibilitarão avaliar o impacto da dieta pobre em proteínas em uma determinada população de cobaias. Esta é uma condição extremamente interessante, pois permite restringir o uso das cobaias apenas para validar os resultados obtidos com a ferramenta computacional desenvolvida.

Uma importante meta também é estender a aplicação da ferramenta desenvolvida para outros problemas similares explorando ao máximo o potencial que uma inovação deste tipo pode oferecer. Finalmente, o projeto vai contribuir para estabelecer uma cooperação científica entre o DSC-POLI/UPE e o Departamento de Anatomia da UFPE, o que pode resultar em benefícios mútuos para ambas as Instituições e para a Sociedade.

## 1.2 Organização do Trabalho

O conteúdo desta monografia está distribuído em cinco capítulos resumidos a seguir:

No Capítulo 2, são explicados todos os conceitos e técnicas necessários para entender o trabalho como um todo, sejam eles de Biologia ou de Computação. Aqui esperamos abordar e esclarecer o leitor, antecipadamente, sobre os principais conceitos e assuntos tratados neste trabalho.

No Capítulo 3, são apresentadas a teoria e a prática utilizadas no desenvolvimento da ferramenta, é nele que o leitor vai conhecer a estrutura da solução desenvolvida, sua integração com outros sistemas, o modo de funcionamento e os recursos que foram implementados.

O Capítulo 4 diz respeito aos experimentos de validação da ferramenta que foram concebidos e executados. Portanto, é neste capítulo que o leitor encontrará todas as simulações e suas descrições, bem como a documentação da execução de cada simulação e os resultados que foram obtidos. Finalmente, no Capítulo 5 é feita a conclusão do trabalho, avaliando-se a

consecução dos objetivos e as novas perspectivas que se abriam. Isto é feito ao se apresentar as contribuições, quando fazemos também algumas considerações discursivas e propomos trabalhos futuros.

# Capítulo 2

## Referencial Teórico

Neste trabalho houve a cooperação entre duas grandes áreas da ciência: a Biologia que ofereceu um problema para atuarmos a fim de desenvolvermos uma possível solução computacional; e a Computação que, através da Inteligência Artificial, constituiu a estrutura fundamental sob a qual implementamos nossa contribuição para amenizar dificuldades e expandir os horizontes das pesquisas de colegas de outras áreas.

Como forma de auxiliar o leitor a entender melhor os conceitos e técnicas que serão apresentados nos próximos capítulos e prepará-lo para as discussões que virão, abordaremos nas seções a seguir os temas necessários para o completo entendimento do trabalho no que diz respeito às ciências envolvidas em seu desenvolvimento.

Partiremos apresentando a IA como uma área da Ciência da Computação, explicando sua história e seu desenvolvimento, até chegarmos na técnica específica que foi escolhida para compor a solução proposta. Mostraremos também um outro projeto que utiliza IA para o suporte à decisão na colheita de cana-de-açúcar, cujos resultados foram fundamentais para a realização deste trabalho. Logo após, detalharemos o problema de Biologia sob o qual atuamos, descrevendo sua metodologia e as informações que dela resultaram.

### 2.1 Inteligência Artificial

#### 2.1.1 Definição

A definição do termo Inteligência Artificial, utilizado pela primeira vez por John McCarthy [8] há aproximadamente 3 décadas, evoluiu muito ao longo dos anos à medida que as discussões sobre o que de fato é um comportamento inteligente avançaram. Segundo o Prof. Dr. Jorge Muniz Barreto, o conceito de IA varia com o tempo, com a época e com o contexto [9].

Em geral, definições que tentam explicar o termo IA fazendo referência à palavra “inteligência” acabam não sendo aceitas universalmente [10] [11] [12]; provavelmente porque inteligência é algo abstrato que ainda não podemos quantificar, mas que é normalmente definida como a habilidade de: aprender, adaptar-se, decidir apropriadamente, comunicar-se e entender [13].

Segundo estudiosos de Psicologia e Cognição, a inteligência ajuda a identificar a parte correta do conhecimento para executar determinadas tarefas [14] [15]. Partindo desta

formalização do termo “inteligência”, Amit Konar definiu IA como sendo a simulação da inteligência humana em uma máquina, para então fazer a máquina eficiente para identificar e usar a parte certa do conhecimento em um certo passo da resolução de um problema [16].

### 2.1.2 Técnicas

Existem muitas abordagens diferentes de IA, nenhuma é absolutamente melhor que outra, assim como não existe uma abordagem que seja a ideal para todos os casos. O fato é que algumas técnicas são normalmente mais adequadas que outras na resolução de determinados problemas.

Dentre as técnicas de IA mais conhecidas, destacam-se:

- Agentes e Sistema Multi-agentes: corresponde a um sistema em que cada agente é uma entidade computacional com um comportamento autônomo, o que lhe permite decidir suas próprias ações. Os agentes possuem um conjunto de objetivos e um conjunto de capacidades comportamentais que definem sua competência, é justamente a autonomia que os agentes possuem que lhes permite utilizar suas capacidades comportamentais a fim de alcançar seus objetivos. A decisão de qual ação deve ser realizada é determinada pelo agente, considerando as mudanças acontecidas no ambiente em que atua e o desejo de alcançar seus objetivos. A idéia principal em um sistema multiagente é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual dos agentes [17].
- Redes Bayesianas: é um modelo de representação do conhecimento que trabalha com o conhecimento incerto e incompleto através da Teoria da Probabilidade Bayesiana, publicada pelo matemático Thomas Bayes em 1763. Essa representação tem como uma das suas principais características a adaptabilidade, podendo, a partir de novas informações, e com base em informações de fundos verdadeiros, gerar alterações nas dependências e nos seus conceitos. Permite, dessa forma, que as probabilidades não sejam meros acasos, podendo confirmar e criar novos conceitos. É importante observar que o modelo baseia-se em probabilidades prévias mais algumas informações com um certo nível de incerteza, para representar o conhecimento. Um modelo totalmente incerto não seria de grande utilidade e não teria ponto de partida para a manipulação de suas situações e a geração de novas representações [18].
- Árvores de Decisão: as árvores de decisão consistem de nodos que representam atributos do problema, de arcos provenientes destes nodos e que recebem os valores possíveis para os atributos correspondentes, e de nodos folha que representam as diferentes classes de um conjunto de treinamento [19]. Uma árvore de decisão tem a função de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe. Para atingir esta meta, a técnica de árvores de decisão examina e compara a distribuição de classes durante a construção da árvore. O resultado obtido, após a construção de uma árvore de decisão, são dados organizados de maneira compacta, que são utilizados para classificar novos casos.
- Computação Evolutiva: propõe um paradigma alternativo ao processamento de dados convencional. Este novo paradigma não exige, para resolver um problema, o conhecimento prévio de uma maneira de encontrar uma solução. A Computação

Evolutiva é baseada em mecanismos evolutivos encontrados na natureza, tais como a auto-organização e o comportamento adaptativo [6].

- Redes Neurais Artificiais: conforme será visto na seção 2.2, esta técnica consiste em um método de solucionar problemas de IA construindo um sistema que tenha circuitos inspirados nos circuitos encontrados no cérebro humano, incluindo seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. Podemos dizer também que são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.
- Sistemas Especialistas: são programas constituídos por regras que analisam informações sobre uma classe específica de problemas, normalmente aqueles que seriam atendidos por um especialista humano. Os especialistas fornecem, explicitamente, as regras gerais de como analisariam o problema e um analista de sistemas experiente, também pode fazer uma análise implícita e derivar regras a partir da observação de um caso de testes [6].
- Sistemas com Lógica Difusa: são sistemas baseados em uma generalização da lógica clássica conhecida como lógica difusa ou lógica *fuzzy*. A lógica tradicional admite apenas pares opostos do tipo falso/verdadeiro, enquanto que a lógica difusa é uma tentativa de implementar níveis intermediários de verdade, chamados de tons de cinza. Isto quer dizer que as implementações da lógica difusa permitem que estados indeterminados como morno e médio possam ser tratados por dispositivos de controle [6].
- Sistemas Tutores Inteligentes (STI): são sistemas computacionais com modelos de conteúdo instrucionais que especificam o que ensinar e estratégias de ensino que especificam como ensinar. É importante que um STI seja capaz de fornecer instrução individualizada, ou seja, deve ser capaz de adaptar suas instruções para satisfazer os estilos individuais de cada aprendiz. Este é o princípio pedagógico geral que rege a concepção de um STI [20].

### 2.1.3 Aplicações

Em geral, quando se fala em IA, a maioria das pessoas imagina robôs com certas habilidades inteligentes realizando tarefas tipicamente humanas. Na verdade, as aplicações que fazem uso das técnicas de programação inteligente vão muito além da construção de robôs.

Na lista abaixo é possível observar algumas situações em que a IA vem obtendo êxito:

- Detecção de intrusão;
- Diagnósticos médicos;
- Jogos;
- Previsão de séries temporais;
- Reconhecimento de padrões;
- Robótica;
- Sistemas de navegação;
- Visão computacional.

## 2.2 Redes Neurais Artificiais

Esta seção dedica-se a detalhar as Redes Neurais Artificiais, uma técnica de IA mencionada na subseção 2.1.2, que foi utilizada no desenvolvimento deste trabalho. A técnica que será apresentada tem inspiração nos circuitos cerebrais, de maneira que ao invés de programar um computador digital para que ele imite um comportamento inteligente, procura-se construir uma estrutura que tenha circuitos modelando os circuitos cerebrais. Isto é feito a fim de se obter o comportamento inteligente desejado, como: aprendizado, capacidade de generalização e de fazer descobertas.

### 2.2.1 Fatos Históricos

Em 1956, após a *Summer Workshop* que aconteceu no *Dartmouth College* [21], considerado o primeiro esforço conjunto para estudar IA, foi publicado um artigo no livro “*Automata Studies*” que tratava redes neurais como um paradigma da arquitetura computacional [22]. Neste mesmo encontro foram discutidos pela primeira vez os dois paradigmas da IA: Inteligência Artificial Simbólica (IAS) e Inteligência Artificial Conexionista (IAC) do qual, atualmente, redes neurais artificiais fazem parte.

Cada paradigma tem uma proposta bem diferente para fazer uma máquina mostrar algum comportamento inteligente. Na IAS, o comportamento inteligente global é simulado sem considerar os mecanismos responsáveis por este comportamento. Na IAC, acredita-se que em se construindo um sistema que simule a estrutura do cérebro, este sistema apresentará comportamento inteligente como a capacidade de aprender com os próprios erros.

Cada paradigma evoluiu de maneira diferente, inicialmente as pesquisas em IAS foram mais aceleradas e algumas de suas contribuições se tornaram muito populares, por exemplo, o uso de sistemas especialistas que fazem apoio à decisão. No entanto, hoje já se espera de IAC um melhor desempenho em relação a IAS quando se trata de resolver problemas mal definidos, onde falta conhecimento explícito de como realizar uma tarefa.

### 2.2.2 Fundamentos Biológicos

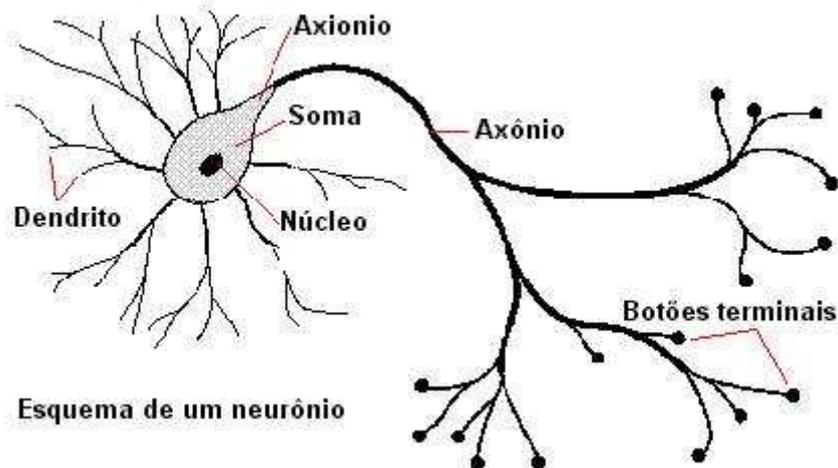
RNA é uma técnica computacional que apresenta um modelo inspirado na estrutura neural de organismos inteligentes que adquirem conhecimento através da experiência. Ou seja, a estrutura e o funcionamento de uma RNA são baseados naquilo que encontramos no sistema nervoso.

O sistema nervoso proporciona muitas das funções de controle do corpo como as reações rápidas, por exemplo, uma contração muscular (função motora). Partes do corpo que têm as funções controladas pelo sistema nervoso têm tempo de resposta relativamente rápido, observa-se também que o processamento das informações captadas do meio ambiente se dá quase que instantaneamente. Exemplos disso são o processamento das imagens captadas pelos olhos e o processamento dos sons que chegam ao cérebro através do ouvido.

O principal órgão do sistema nervoso é o cérebro, que desempenha tarefas complexas, mesmo quando possui uma estrutura mais primitiva ou menor que a do cérebro humano. É neste órgão que se encontra a estrutura na qual as redes neurais artificiais são baseadas.

A estrutura básica do cérebro é o neurônio (ver Figura 1), a ele são atribuídas as principais funções cerebrais. O neurônio tem um corpo celular chamado soma e diversas ramificações

conhecidas como dendritos, que são responsáveis por conduzir sinais de suas extremidades para o corpo celular. Existe ainda uma ramificação, geralmente única, chamada axônio, que transmite um sinal do corpo celular para suas extremidades. As extremidades do axônio se ligam aos dendritos de outros neurônios pelas sinapses (ver Figura 2), que têm um papel fundamental na memorização das informações.



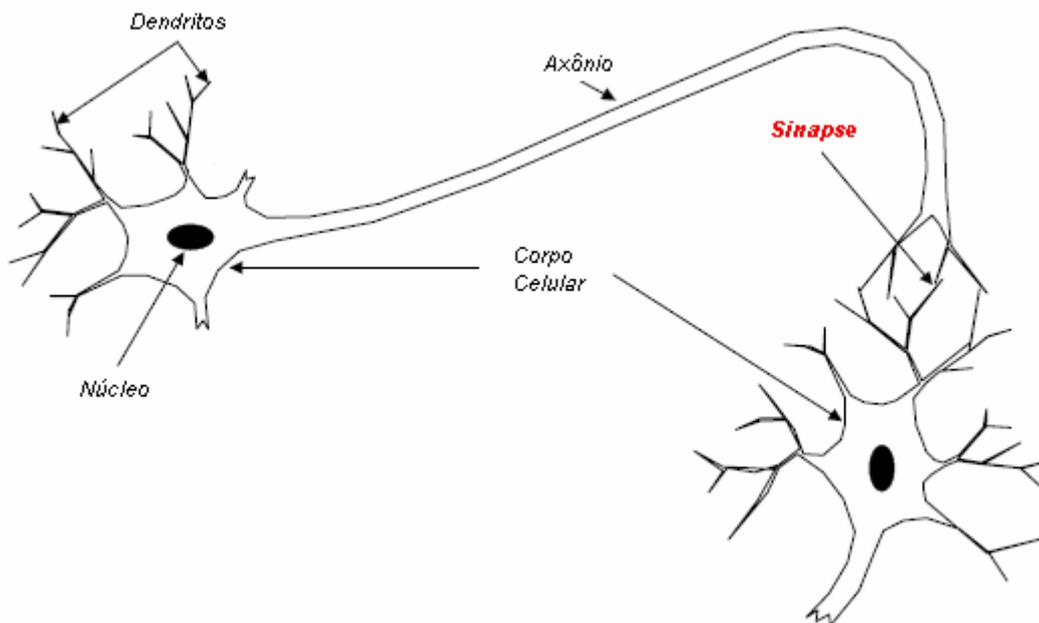
**Figura 1.** Neurônio: célula do sistema nervoso [23]

Entender como funciona o neurônio e o mecanismo das sinapses é extremamente importante para entender o funcionamento das RNA. O neurônio, assim como a maioria das células, é composto de água, eletrólitos, proteínas, lipídios e carboidratos. Podemos dividir sua configuração em duas partes: núcleo e citoplasma, que fazem parte do meio intracelular. A membrana citoplasmática é a estrutura que divide o meio intracelular do meio extracelular.

A membrana citoplasmática do neurônio tem a propriedade de permitir o transporte de eletrólitos, capazes de modificar o potencial elétrico entre os meios interno e externo da célula nervosa. A diferença de potencial criada pelos eletrólitos faz a célula nervosa gerar pulsos de frequência, que se propagam através do axônio para os dendritos de outros neurônios. Estes pulsos de frequência são na verdade estímulos contendo as informações codificadas, que posteriormente serão decodificadas por dendritos de outras células.

No interior da membrana citoplasmática existe uma alta concentração de moléculas específicas de neurotransmissores, que ficam armazenados em vesículas especiais denominadas vesículas sinápticas. Quando ocorre a despolarização da membrana, isto é, quando o potencial do citoplasma torna-se maior que o potencial de repouso, íons fluem para dentro da célula através dos canais da membrana, ativando a liberação dos neurotransmissores. Assim acontece a sinapse nervosa, que é a ligação entre a terminação axônica e os dendritos e que permite a propagação dos impulsos nervosos de uma célula à outra.

As sinapses podem ser classificadas em excitatórias ou inibitórias. São consideradas excitatórias quando permitem que a informação passe de um neurônio para outro e são ditas inibitórias quando impedem ou dificultam a passagem da informação.



**Figura 2.** Esquema da conexão entre dois neurônios

A Lei de Hebb, que é base de muitos algoritmos de aprendizagem de RNA, é inspirada no mecanismo das sinapses descrito nos dois parágrafos anteriores, e diz que: “A intensidade de uma conexão sináptica entre dois neurônios aumenta quando os dois neurônios estão excitados simultaneamente”.

A modificação da intensidade com que um neurônio é capaz de excitar ou inibir um outro neurônio, quando analisado globalmente, cria o comportamento da rede de neurônios. Portanto, o que muda o comportamento da rede são as mudanças dos valores das conexões e o aprendizado da rede está justamente nestas mudanças de comportamento.

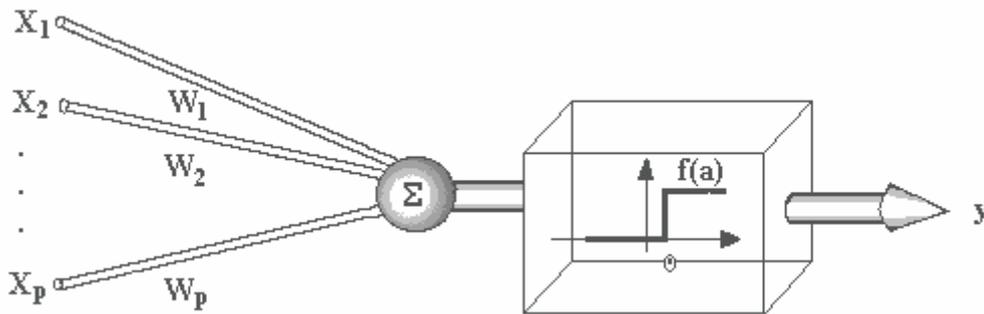
### 2.2.3 Neurônio Artificial

Assim como no cérebro, a estrutura fundamental para o funcionamento de uma RNA são neurônios. O modelo de neurônio artificial proposto em 1943 pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts (ver Figura 3) [24], representa o neurônio como um circuito binário contendo  $p$  ( $x_1$  à  $x_p$ ) entradas binárias, cada uma com um peso ( $w_1$  à  $w_p$ ) específico. A entrada efetiva do neurônio é obtida fazendo a soma ponderada de todas as entradas e seus respectivos pesos, de acordo com a fórmula (1) apresentada abaixo:

$$\text{Entrada efetiva} = \sum_{j=1}^p x_j \cdot w_j$$

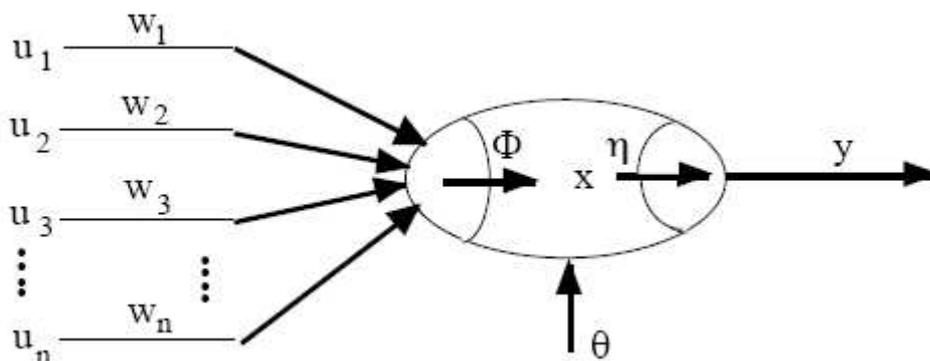
**Equação 1.** Cálculo da entrada efetiva do neurônio artificial

A saída da entrada efetiva é o argumento de uma função de transferência  $f$  (também binária) cujo resultado corresponde à saída  $y$  do neurônio.



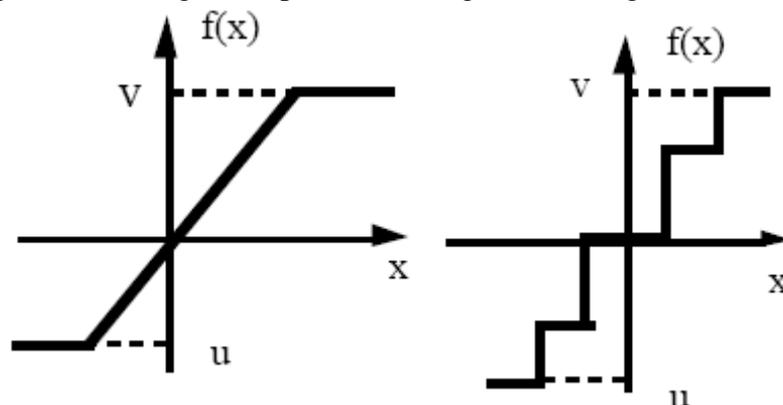
**Figura 3.** Neurônio artificial de McCulloch e Pitts

Uma generalização do neurônio artificial de McCulloch e Pitts de acordo com Dayhoff [25] é mostrada na Figura 4, neste modelo as entradas  $u_i w_i$  são combinadas usando uma função  $\Phi$  para produzir um estado de ativação do neurônio que através da função  $\eta$  vai produzir a saída do neurônio (este comportamento pode ser comparado à frequência de descarga do neurônio biológico). Existe ainda um valor  $\theta$  que pode ser usado para representar um valor abaixo do qual a saída é nula.



**Figura 4.** Generalização do modelo de neurônio de McCulloch e Pitts

Assim como no neurônio biológico, nos modelos de neurônio artificial mostrados (ver Figuras 3 e 4) as ligações sinápticas são externas ao neurônio. A função  $\Phi$  corresponde normalmente a soma das entradas, algumas vezes o produto delas, outras funções também podem ser aplicadas apesar disso não ser muito comum. Para introduzir a não-linearidade no neurônio são utilizadas funções como tangente hiperbólica e degraus (ver Figura 5).



**Figura 5.** Não linearidades freqüentemente usadas no modelo de neurônios de uma RNA

Em resumo as operações realizadas pelo neurônio artificial são:

- Receber sinais de entrada;
- Multiplicar o sinal de entrada recebido por um peso que representa a influência daquela entrada na saída da unidade;
- Aplicar uma função sobre os sinais de entrada produzindo um nível de atividade;
- Produzir uma saída quando o nível produzido exceder um determinado limiar (*threshold*).

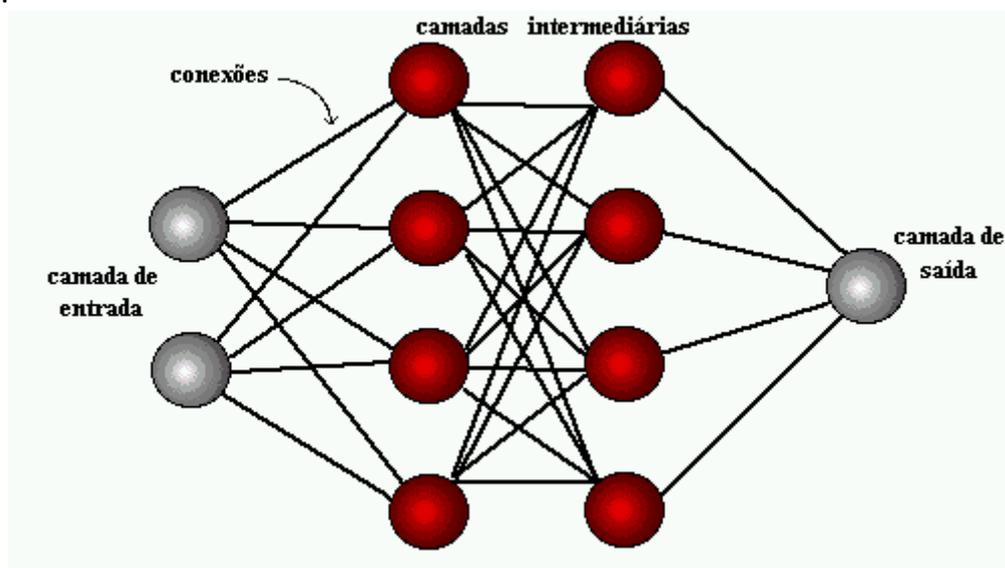
## 2.2.4 Caracterização de RNA

As redes neurais artificiais fazem uma representação distribuída da informação utilizando para isso as conexões entre neurônios artificiais. Operacionalmente, a rede neural, através de seu conjunto de neurônios, executa uma transformação entre a entrada e a saída do sistema. As RNA permitem a representação do processamento da informação conforme acontece nos neurônios biológicos.

Um sistema que utiliza RNA é capaz de reconhecer a informação aprendida ainda que ela esteja incompleta. Consegue fazer generalizações e associações para resolver um determinado problema. Sendo assim, pode-se dizer que RNA é uma técnica aplicada a IA, inspirada em processos cognitivos, para a resolução de problemas complexos [26].

Numa RNA existem neurônios que recebem excitações do mundo exterior e são chamados de neurônios de entrada. Outros têm suas respostas utilizadas para modificar ou interferir de alguma maneira no mundo exterior e são chamados de neurônios de saída. Os neurônios internos à rede, aqueles que não são nem de entrada nem de saída, comumente chamados de neurônios escondidos, são extremamente importantes por processarem de fato a informação, extraindo características.

Podemos organizar uma RNA em camadas, onde a camada de entrada possui todos os neurônios que recebem sinais de excitação, a camada de saída é formada pelos neurônios cujas saídas correspondem às saídas da rede. Os neurônios que não pertencem as camadas de entrada ou saída, formam a camada escondida que pode ser composta por várias camadas internas (ver Figura 6).



**Figura 6.** Exemplo de uma rede neural organizada em camadas

### 2.2.5 Classificação de RNA

Uma rede neural é normalmente desenvolvida para ser utilizada sobre um domínio específico. Sendo assim, a estrutura da Rede Neural e a escolha de outras características peculiares, como forma de conexão entre os neurônios e tipo de treinamento, são todos escolhidos visando otimizar o resultado do uso daquela rede para obter a melhor resposta diante de uma determinada classe de problemas.

Do ponto de vista estrutural, a arquitetura de Redes Neurais pode ser classificada como fixa ou variável, e também pode ser classificada de acordo com o número de camadas, sendo de uma única camada ou com múltiplas camadas.

Outras diferenças surgem também de acordo com a forma como são feitas as conexões entre os neurônios, nos permitindo classificá-las como: estritamente no sentido de ida, no sentido de ida e volta, lateralmente conectadas, topologicamente ordenadas ou híbridas.

Quanto às aplicações de Redes Neurais podemos classificá-las em: Reconhecimento de Padrões e Classificação; Processamento de Imagem e Visão; Identificação de Sistema e Controle e Processamento de Sinais [7].

### 2.2.6 Topologias de RNA

Têm haver com a forma como os neurônios se conectam. Qualquer grafo dirigido de conectividade entre os neurônios pode dar lugar a uma topologia de RNA [27]. Algumas topologias receberam maior atenção dos pesquisadores e são conhecidas com nomes específicos.

A organização das RNA em diferentes topologias se dá de acordo com a natureza do problema, algumas topologias comuns são:

- Redes diretas: é a topologia mais popular hoje em dia, também conhecida como redes *feedforward*. Seu grafo de conectividade não possui ciclos e elas podem ser representadas em camadas. Na Figura 7 é mostrado um exemplo de rede direta com 4 camadas;

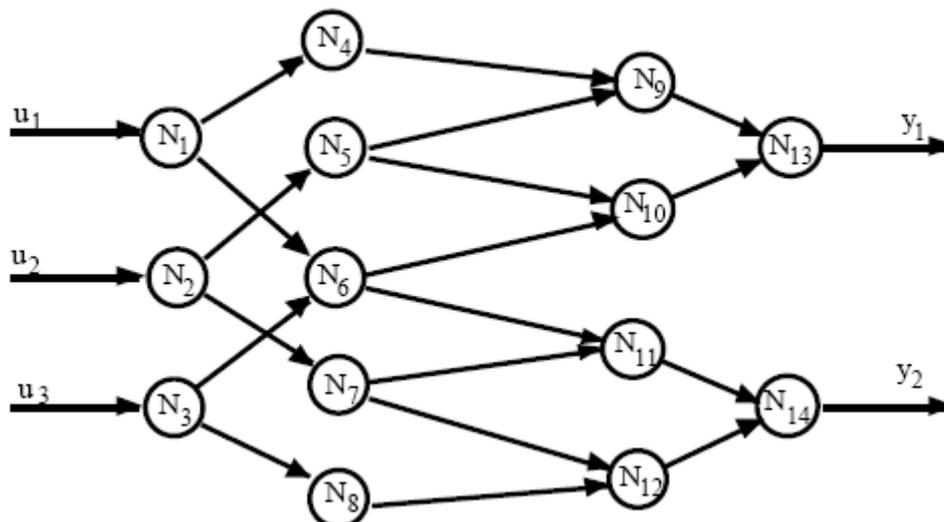
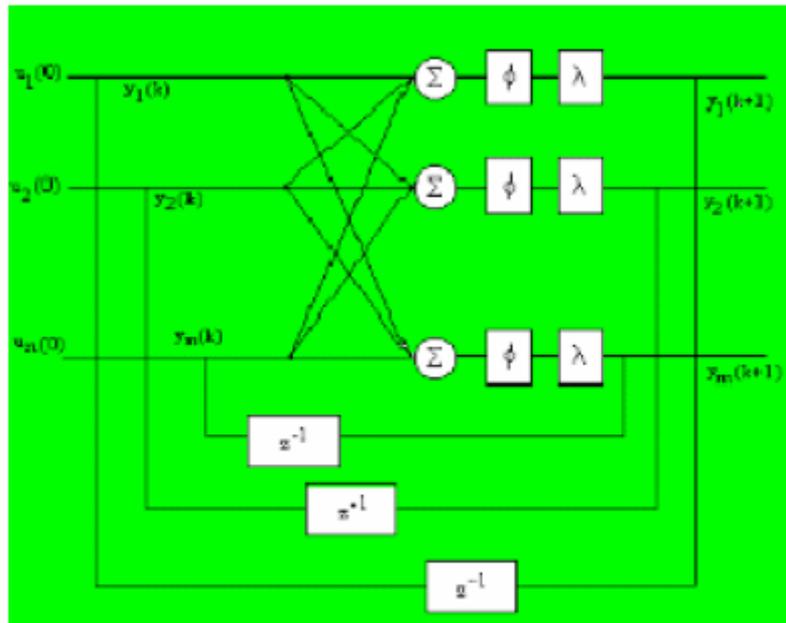


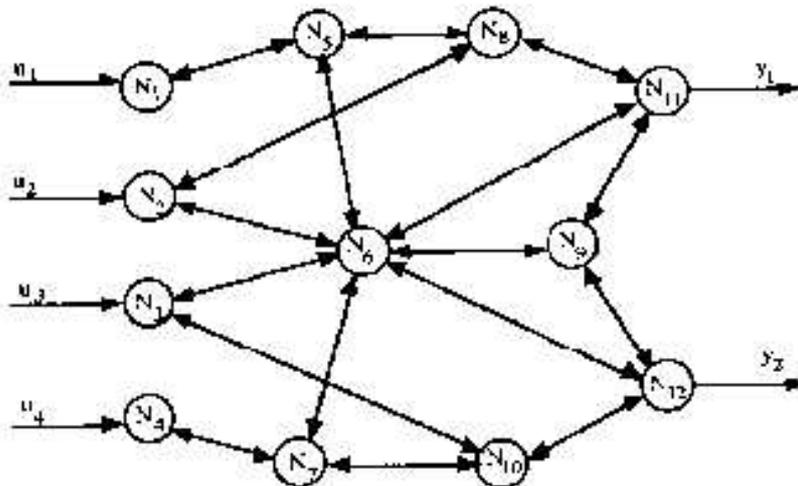
Figura 7. Rede direta com 4 camadas [28]

- Redes com realimentação: também chamadas de redes com feedback ou redes com ciclos. Nelas, o grafo de conectividade contém pelo menos um ciclo. Um exemplo de rede com realimentação é mostrado na Figura 8, extraído de [29];



**Figura 8.** Exemplo típico de rede com ciclos

- Redes simétricas: são um caso especial de redes com realimentação, nela o grafo de conectividade é simétrico, ou seja, estão nos dois sentidos e são iguais (ver Figura 9).



**Figura 9.** Exemplo de rede simétrica

### 2.2.7 Aprendizado

A principal habilidade de uma rede neural é a capacidade de aprender a partir de informações provenientes do ambiente, este aprendizado que acontece na fase de treinamento da rede neural é fundamental para se obter o seu melhor desempenho. Podemos dizer que uma rede neural “aprendeu” a resolver um problema, quando ela produz uma solução para instâncias do problema não utilizadas no treinamento. Neste caso, pode-se afirmar que houve generalização.

O mecanismo pelo qual uma rede neural adquire conhecimento é determinado por um algoritmo de aprendizado. Existem algoritmos que são mais adequados para cada um dos diferentes modelos de redes neurais existentes.

O conhecimento da rede neural está armazenado no conjunto dos valores dos pesos das conexões entre os neurônios; desta forma, quanto mais eficientemente o algoritmo de aprendizado conseguir ajustar os pesos das conexões de acordo com as informações extraídas dos padrões apresentados, maior será a generalização da rede para um determinado problema.

Existem alguns paradigmas de aprendizado que se referem à maneira como a rede neural interage com o ambiente durante a fase de treinamento, são eles:

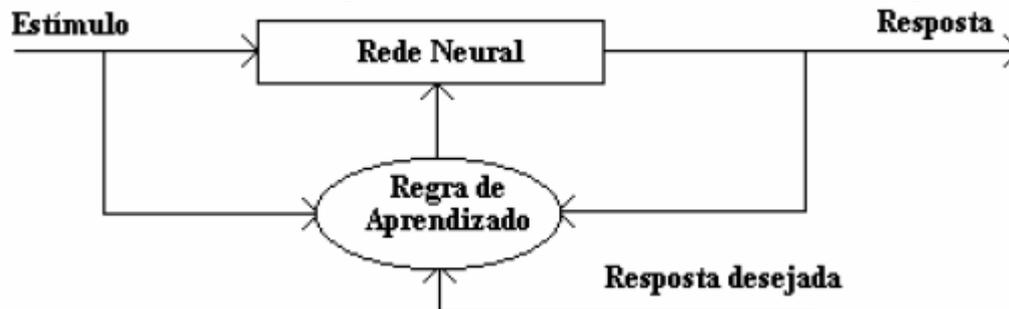
1. Por independência de quem aprende:

A rede aprende por meio de memorização, através de exemplos, comparações e também por descoberta;

2. Por retroação do mundo:

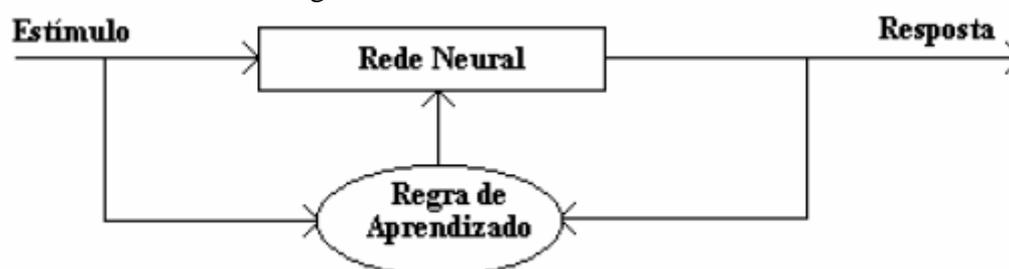
Neste caso um agente do mundo exterior vai assinalar em determinados intervalos de tempo os acertos e erros da rede durante o aprendizado, neste contexto o aprendizado pode ser:

a. Supervisionado: quando o agente indica o bom ou mau comportamento da rede de acordo com o padrão de entrada, conforme ilustrado na Figura 10.



**Figura 10.** Ilustração do aprendizado supervisionado

b. Não-supervisionado: utiliza exemplos semelhantes, a fim de que a rede se organize para responder de forma semelhante à apresentada, conforme ilustrado na Figura 11.



**Figura 11.** Ilustração do aprendizado não-supervisionado

3. Por finalidade do aprendizado:

a. Auto-associador: a rede deve memorizar um conjunto de exemplos e quando for apresentado um elemento do conjunto de exemplos contendo agora um erro, a rede mostra o exemplo original.

b. Hetero-associador: a rede memoriza um conjunto de pares e aprende a reproduzir o segundo elemento do par mesmo quando o primeiro sofre alguma modificação.

### 2.2.8 *Perceptron*

O *perceptron* é um modelo de RNA que foi proposto por Roseblatt em 1959 [30] para reconhecimento de letras maiúsculas do alfabeto. É uma rede direta consistindo de uma camada de entrada e outra de saída formada por unidades binárias que aprendem a classificar padrões através de aprendizado supervisionado. Modelam o neurônio fazendo a soma ponderada de suas entradas e enviam o resultado 1 se a soma for maior do que algum resultado inicial ajustável e o valor zero caso contrário.

Os padrões são classificados usando vetores binários com valores iguais a 0 e 1, a unidade de processamento do *perceptron* executa um somatório dos pesos por seus valores de entrada. O *perceptron* testa o valor da soma ponderada dos pesos em relação ao valor de polarização, gerando resultado 1 quando o valor da soma ponderada for maior que a polarização e 0 caso contrário.

Em 1969, Minsky e Pappert demonstraram que o *perceptron* não consegue classificar padrões não linearmente separáveis. Para se obter esta capacidade seria preciso adicionar camadas escondidas no modelo dos *perceptrons*, o problema é que na época não haviam algoritmos de treinamento preparados para um modelo deste tipo. Com o avanço nas pesquisas sobre redes neurais e o desenvolvimento das linguagens de programação utilizadas em IA, surgiu o modelo de RNA mostrado na subseção 2.2.9 que vem a suprir as deficiências do modelo dos *perceptrons*.

### 2.2.9 *Multi-Layer Perceptron (MLP)*

Modelo concebido para resolver problemas mais complexos impossíveis de serem resolvidos no modelo dos *perceptrons* da subseção 2.2.8, o princípio é montar uma rede de *perceptrons* dispostos em camadas, neste modelo temos neurônios na camada de entrada, camada oculta e camada de saída.

Além da camada oculta existente no modelo MLP, existe uma outra diferença marcante em relação ao modelo dos *perceptrons*, que se dá na estrutura dos neurônios que passam a contemplar a não-linearidade.

O comportamento inteligente em uma rede MLP vem das interações entre as unidades de processamento da rede. O processo de treinamento da rede é feito utilizando-se dados reais conhecidos, a partir dos quais a rede neural será capaz de extrair conhecimento adquirindo assim a sistemática para executar o processo desejado.

Na década de 80, Rumelhart [31] definiu um algoritmo capaz de treinar as redes MLP, que ficou conhecido como *backpropagation of error* [32]. Este algoritmo tem duas fases definidas, uma “*forward*” (para frente) e outra “*backward*” (para trás), que são executadas durante o treinamento para cada padrão (valor de entrada e resposta desejada) apresentado a rede.

Durante a fase *forward*, um padrão é apresentado à rede que executa o processamento até ser produzida a saída da rede; na fase *backward*, os pesos das conexões entre os neurônios da camada oculta vão sendo ajustados com base nos valores de erro calculados para cada uma destas unidades. Isto é feito visando minimizar o erro que a saída da rede oferece em relação à saída que se deseja obter.

O treinamento da rede MLP com o algoritmo *backpropagation* continua até se chegar a um critério de parada estabelecido, como número de ciclos de treinamento ou uma diferença mínima entre erros sucessivos calculados.

## 2.3 InteliColheita

Esta seção dedica-se a fazer uma breve introdução a respeito do projeto InteliColheita [33], desenvolvido no Departamento de Sistemas Computacionais da Escola Politécnica de Pernambuco, do qual alguns módulos produzidos foram fundamentais para o desenvolvimento da ferramenta proposta neste trabalho como é documentado nos Capítulos 3 e 4.

### 2.3.1 Descrição do projeto

Concebido a partir da Dissertação de Mestrado do Professor Doutor Fernando Buarque [33] de Lima e Neto apresentada ao departamento de Informática da Universidade Federal de Pernambuco em meados 1998. Intitulada Suporte a Decisão Gerencial Baseado em Redes Neurais Artificiais – nDSS, propõe um modelo de apoio gerencial a gestores de empresas agrícolas, com a utilização de computação inteligente para predição de indicadores de produtividade.

Percebendo a necessidade de técnicas mais eficientes para o auxílio na tomada de decisão, o projeto utiliza-se de módulos que se interligam formando o sistema de apoio gerencial. Os módulos externos correspondem à obtenção de dados das empresas agrícolas no relevante à informações de produtividades e fatores climáticos e ambientais, e ainda, na interface de auxílio aos gerentes. O módulo interno corresponde a um motor implementado com computação inteligente, em especial redes neurais artificiais, que processa os dados e com sua capacidade exibe informações preditivas.

Com o sistema, as técnicas utilizadas anteriormente, em especial informações estatísticas, poderão ter maior fundamentação e assim melhorar na tomada de decisão dos gestores de empresa agrícolas.

Além do módulo que incorpora o motor neural, existe um outro módulo capaz de manipular arquivos de dados de acordo com parâmetros específicos a fim de possibilitar que estes sejam utilizados pelas redes (pré-processamento) em fases como treinamento e simulação. Permite também resgatar os valores para seu formato original, anterior ao procedimento de manipulação aplicado sobre eles (pós-processamento).

### 2.3.2 Motor Neural

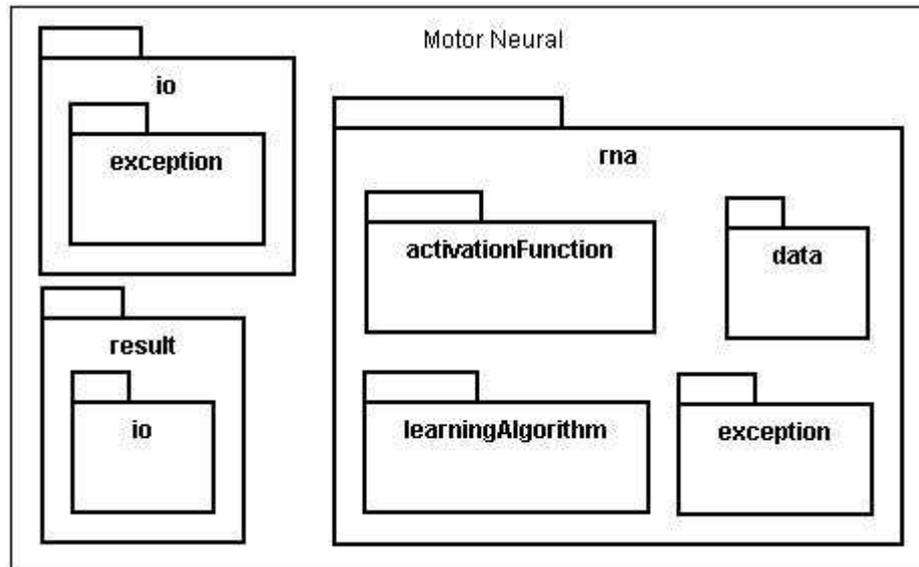
É um mecanismo inteligente com propriedades preditivas baseadas nas redes MLP, oferece recursos para a criação de redes neurais de forma parametrizada na definição da estrutura e da topologia da rede.

O motor também oferece o recurso de treinamento das redes neurais criadas, bastando para isso informar os conjuntos de dados para treinamento e a configuração do treinamento, por exemplo, taxa de aprendizado da rede neural e o critério de parada. É possível ainda utilizar o motor para realizar simulações, que consistem em aproveitar redes neurais já treinadas e utilizá-las na predição de valores para novas entradas definidas em tempo de execução.

O motor neural foi concebido para ser um módulo reutilizável, foi desenvolvido na linguagem Java e faz uso das principais características disponíveis nesta tecnologia, como: portabilidade, polimorfismo e herança. Este motor é utilizado como mecanismo inteligente da ferramenta desenvolvida neste trabalho, oferecendo satisfatoriamente os recursos preditivos desejados.

Na Figura 12 é apresentada a estrutura de pacotes do motor neural. No pacote *io* estão as classes que tratam da manipulação de arquivos (leitura e gravação) e inclui o pacote *exception* para tratamento de exceções. No pacote *result* estão as classes referentes ao arquivo onde são salvos os resultados de uma simulação. No pacote *rna* estão todas as classes necessárias para a

manipulação das redes neurais, tanto no aspecto de criação como no de treinamentos e simulações.



**Figura 12.** Diagrama de classes resumido do motor neural

Um aspecto interessante quanto a classe *rna* é o fato de que ela contém pacotes onde podem ser definidas técnicas de função de ativação (*activationFunction*) ou diferentes algoritmos de aprendizagem (*learningAlgorithm*), permitindo assim escolher a técnica a ser utilizada em tempo de execução.

O motor neural foi incorporado a este trabalho (ver seção 3.5) e seu desempenho foi suficiente para atender as propostas estabelecidas. Assim como, a arquitetura em que foi desenvolvido está em completo acordo com o desejo de se obter uma ferramenta flexível como produto deste trabalho.

### 2.3.3 Módulo de Manipulação de Arquivos

Consiste num conjunto de classes organizadas em dois pacotes principais para a manipulação de arquivos de dados, a fim de se obter o pré ou pós-processamento dos mesmos para sua devida utilização nas redes neurais.

A manipulação dos arquivos de dados é feita definindo-se parâmetros de formatação e disposição dos dados no arquivo e no tipo de manipulação que se deseja realizar.

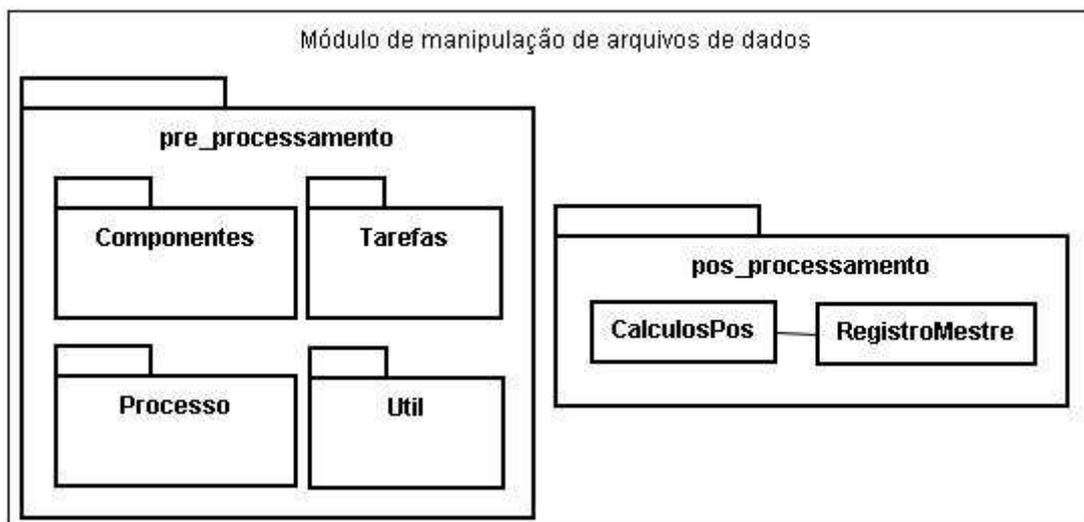
No caso de pré-processamento dos dados os parâmetros determinam basicamente o tipo de normalização que será feita para cada entrada da rede neural e em alguns casos a faixa de valores que estas entradas irão assumir.

No pós-processamento os parâmetros dizem respeito ao tipo de pré-processamento que foi realizado e permitem resgatar os valores normalizados para seu formato original. Um exemplo de parâmetro de pós-processamento é o valor máximo de uma determinada entrada que foi utilizado do conjunto de treinamento.

A Figura 13 apresenta a estrutura de pacotes do módulo de manipulação de arquivos de dados. No pacote *pre\_processamento* estão as classes que tratam da manipulação de arquivos de dados em formato texto e sua codificação de acordo com parâmetros específicos. De acordo com suas características as classes estão organizadas em outros quatro pacotes: no pacote *Componentes* estão as classes que definem a estrutura dos dados e as configurações dos arquivos

a serem lidos e da codificação desejada; no pacote Tarefas estão as classes que implementam o pré-processamento propriamente dito e a verificação de compatibilidade entre os dados apresentados e as configurações definidas; no pacote Util estão muitas das constantes utilizadas pela tarefa de pré-processamento e no pacote Processo estão as classes responsáveis pela início da execução do pré-processamento.

O pacote pos\_processamento possui duas classes principais para a execução do pós-processamento: a classe RegistroMestre define as configurações dos dados e a classe CalculosPos realiza o pós-processamento propriamente dito com base nas informações da classe RegistroMestre.



**Figura 13.** Diagrama de classes resumido do módulo de manipulação de arquivos

O módulo de manipulação de arquivos de dados foi incorporado na ferramenta para tratamento dos dados utilizados nas simulações que a ferramenta se dispõe a fazer e possibilitou um ganho significativo na facilidade de uso do programa para o usuário.

# Capítulo 3

## Ferramenta Desenvolvida

Este capítulo dedica-se à descrição da ferramenta desenvolvida, partindo da análise dos requisitos, descrevendo as características da solução proposta, as funcionalidades implementadas, a integração com outros sistemas e as ferramentas e tecnologias utilizadas.

### 3.1 Caracterização do problema utilizado

O problema utilizado neste projeto considera os dados de uma pesquisa científica desenvolvida no Laboratório de Morfologia dos Plexos Entéricos, do Departamento de Anatomia da UFPE. Esta pesquisa verificou os efeitos da desnutrição, durante o período crítico do desenvolvimento de cobaias de laboratório, sobre o crescimento do intestino delgado. Também foram avaliados os aspectos morfológicos e morfométricos dos neurônios do plexo submucoso do jejuno das cobaias. A desnutrição induzida nos animais é resultado da administração da chamada Dieta Básica Regional (DBR).

A pesquisa foi chefiada pela Prof<sup>ª</sup>. Dr<sup>ª</sup>. Sílvia Regina Arruda de Moraes e teve muitos de seus resultados publicados em revistas científicas da área [5]. Um aspecto interessante da pesquisa é que a DBR, dieta utilizada no experimento, simula a dieta básica do nordestino mais pobre. Conseqüentemente, os resultados da pesquisa têm bastante relevância social, dada a possibilidade de comparação entre o que ocorreria com as cobaias e o que poderia acontecer aos seres humanos vítimas de semelhante desnutrição.

#### 3.1.1 Animais

Os animais utilizados como cobaias foram ratos albinos da linhagem *Wistar*, provenientes da colônia de criação do Departamento de Nutrição da UFPE. Os ratos foram inicialmente divididos em dois grupos: Nutrido e Desnutrido. Cada grupo foi composto pela mesma quantidade de animais, todos do sexo masculino, e foi submetido às mesmas condições de temperatura ambiente, luminosidade e acesso livre à água e à ração.

#### 3.1.2 Dieta Experimental

Consiste em uma ração própria para as cobaias utilizadas no experimento, que foi desenvolvida pelo Departamento de Nutrição da UFPE. Proposta para ser multicarenal [34], a

DBR é deficiente em gordura, vitaminas e sais minerais como o sódio, e principalmente em proteínas (8%). A DBR promove uma desnutrição em ratos semelhante àquela observada em humanos do Nordeste Brasileiro [35].

### **3.1.3 Modelos Experimentais**

Os filhotes do Grupo Nutrido foram amamentados por nutrízes que receberam uma dieta normoprotéica (23% de proteína), enquanto que os filhotes do Grupo Desnutrido foram amamentados por nutrízes que receberam a dieta experimental (DBR). Desta forma, o leite fornecido pela mãe do Grupo Desnutrido não continha o suprimento de nutrientes suficiente à devida nutrição dos filhotes.

Diariamente, foram realizadas aferições do peso corporal das cobaias até a sua eutanásia que ocorreu no 22º dia de vida do animal. A partir daí, foi realizada a dissecação e coleta do material, no caso o intestino delgado, que foi encaminhado para processamento histológico, e foi então realizada a contagem da quantidade de neurônios presentes no plexo submucoso do jejuno.

### **3.1.4 Impactos da DBR observados no organismo dos animais**

Raquitismo, menor densidade óssea, diminuição do comprimento e peso dos ossos, redução da massa corporal dos animais, atraso no crescimento e desenvolvimento dos animais, diminuição do comprimento do intestino delgado de ratos.

## **3.2 Análise de Requisitos**

### **3.2.1 Problema e Causas**

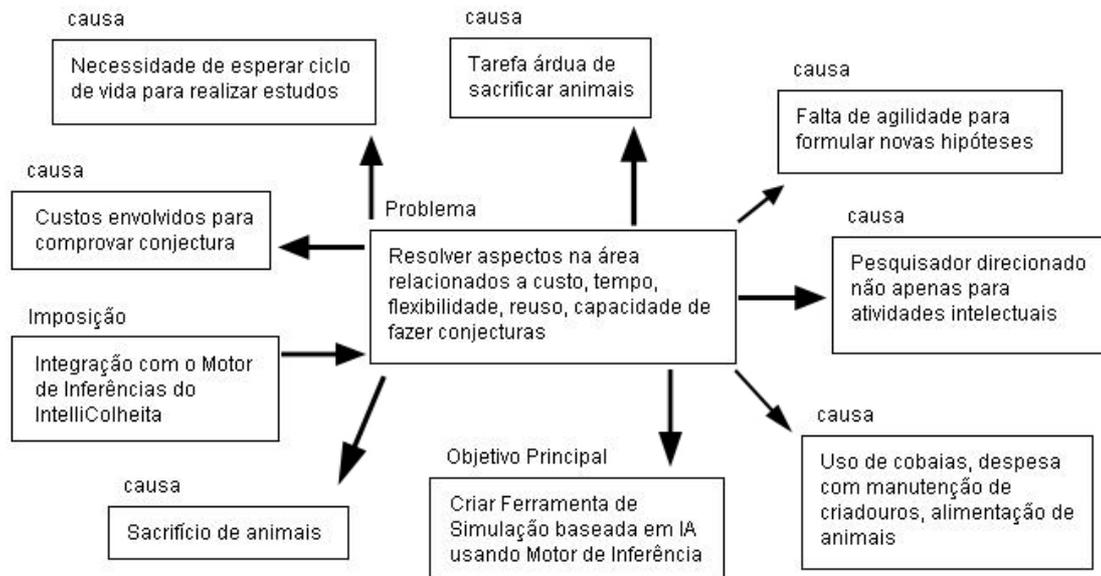
Identificar o problema e suas principais causas é fundamental para a formulação de uma solução adequada. Através das observações realizadas sobre a pesquisa de desnutrição utilizando cobaias na UFPE foi possível determinar algumas características que compõem o cenário do problema em questão.

O problema diz respeito a resolver aspectos da área de pesquisas de Biologia que utilizam cobaias, as principais causas encontradas foram:

- Uso de cobaias, que envolve custos com manutenção e esforço demasiadamente grande do pesquisador nesta tarefa;
- O pesquisador passa muito tempo da pesquisa se dedicando a atividades menos intelectuais, por precisar cuidar e monitorar diariamente o ambiente em que o animal está alojado;
- Necessidade de esperar até que os animais cumpram um determinado ciclo de vida que pode variar de acordo com a pesquisa, até que todos os dados necessários sejam coletados;
- A necessidade de sacrificar animais;
- Tarefa árdua de sacrificar os animais, que é uma etapa trabalhosa onde um erro pode significar a necessidade de refazer todo o experimento;
- Custos envolvidos para que as conjecturas formuladas pelo pesquisador sejam comprovadas, visto que o uso dos animais é inevitável com as práticas convencionais;

- Dificuldade para manipular as características dos animais a fim de se obter modelos experimentais que permitam formular novas hipóteses sobre um mesmo problema, fazendo com que esta atividade ocorra de forma mais lenta.

A Figura 14 ilustra o problema normalmente encontrado em pesquisas de laboratório na área de Biologia, apontando as causas identificadas, o objetivo deste trabalho e as imposições para a sua realização.



**Figura 14.** Análise de Problemas e Causas

### 3.2.2 Requisitos

Através da observação do processo de realização da pesquisa sobre impactos da administração da DBR em cobaias de laboratório, que tem início na criação dos animais e termina com as conclusões extraídas dos resultados do experimento, identificamos que o trabalho dos pesquisadores poderia ser otimizado com o uso de uma ferramenta computacional. Para atingir este objetivo, seria necessário disponibilizar para o grupo de pesquisadores uma ferramenta que oferecesse uma interface simples, na qual o usuário precisasse apenas informar alguns parâmetros de entrada para então obter uma previsão do resultado do experimento, simulando o que seria obtido com dados reais.

A ferramenta deve utilizar uma RNA especialmente treinada com dados reais do problema em questão e oferecer resultados com percentuais de erros pequenos o suficiente para tornar viável o uso da ferramenta em substituição parcial ao uso de cobaias neste experimento. A redução do uso das cobaias e a possibilidade de explorar mais aspectos do problema por meio de simulações caracterizam a utilidade de uma ferramenta deste tipo em experimentos da mesma natureza do utilizado neste trabalho.

Os pesquisadores precisam de um *software* que tenha como entradas os mesmos dados que são coletados na pesquisa, neste caso: o peso corpóreo do animal antes do sacrifício e o comprimento da alça intestinal depois de coletado o material utilizado para análise. Passados os dados de entrada, o software deve fornecer um mecanismo para realizar a simulação que resultará em uma saída correspondente a informação resultante do experimento, neste caso a quantidade de neurônios no plexo entérico das cobaias.

É interessante, também, que o pesquisador tenha a possibilidade de criar novas redes variando a quantidade de entradas e saídas, fazer novos treinamentos em suas redes neurais, bem como gravar os parâmetros e os resultados das simulações permitindo assim, resgatar os trabalhos realizados que foram salvos para reavaliação ou estudo.

### 3.3 Características

A ferramenta proposta é um programa executável, independente de plataforma, concebido para auxiliar pesquisadores da área de Biologia na realização de suas pesquisas. É caracterizado pela facilidade de uso, abstraindo conceitos de RNA e possibilitando que mesmo usuários inexperientes consigam utilizar a ferramenta para fazer simulações.

*Java* foi a linguagem de programação utilizada no desenvolvimento deste projeto, uma linguagem muito difundida no meio acadêmico e profissional; possibilitando assim, fácil entendimento no futuro a quem se interessar pelo estudo do código-fonte desta ferramenta.

A portabilidade da linguagem *Java* também foi fator fundamental na sua escolha, visto que desejamos que a ferramenta desenvolvida possa funcionar em qualquer ambiente independente de plataforma.

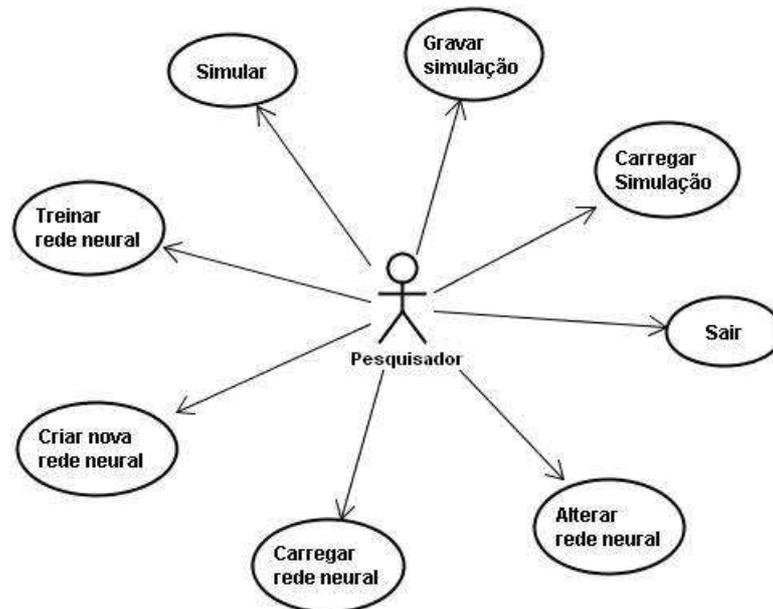
O fato do InteliColheita ter sido desenvolvido em *Java* também foi considerado, porque possibilitou uma integração mais rápida e fácil do que se a integração precisasse ter sido realizada com base em programas desenvolvidos em linguagens diferentes.

O mecanismo inteligente utilizado no projeto é proveniente do motor neural do InteliColheita, todas as redes neurais foram criadas utilizando este recurso, bem como todos os treinamentos e simulações foram realizados utilizando o mesmo motor que foi integrado à ferramenta desenvolvida neste trabalho.

As redes criadas pelo motor neural são do tipo MLP *feedforward*, com neurônios totalmente interligados. O treinamento utiliza o algoritmo *backpropagation of error* para ajustar os pesos das ligações entre os neurônios, além do número de ciclos e do algoritmo GL5 como critérios de parada.

### 3.4 Funcionalidades

Com base nos requisitos observados para o tipo de problema que este trabalho se propõe a resolver, definimos as funcionalidades da ferramenta desenvolvida, o InteliLab. No diagrama de casos de uso apresentado na Figura 15, são ilustradas as funcionalidades do InteliLab.



**Figura 15.** Diagrama de casos de uso da ferramenta desenvolvida

O diagrama de casos de uso nos permite observar as principais opções que o pesquisador, único ator do sistema, têm disponíveis. A seguir, descreveremos as etapas de utilização do programa, explorando seus recursos e a maneira de utilizá-los.

Assim que o programa é executado, a tela principal é carregada (ver Figura 16) permitindo o acesso às principais funções do programa através do menu principal (ver Figura 17).



**Figura 16.** Tela principal do Intelilab

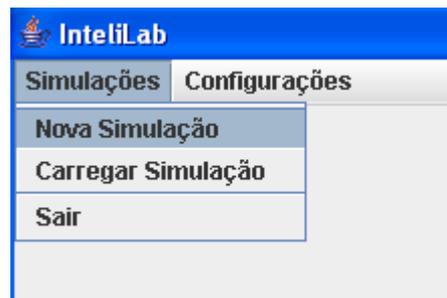


**Figura 17.** Menu principal do Intelilab

Abaixo está a descrição dos principais recursos do programa e o caminho para acessá-los através do menu principal.

1. Nova Simulação:

Permite a criação de uma nova simulação. Para acessar este recurso, basta clicar na opção de menu “Simulações→Nova Simulação” conforme mostrado na Figura 18.

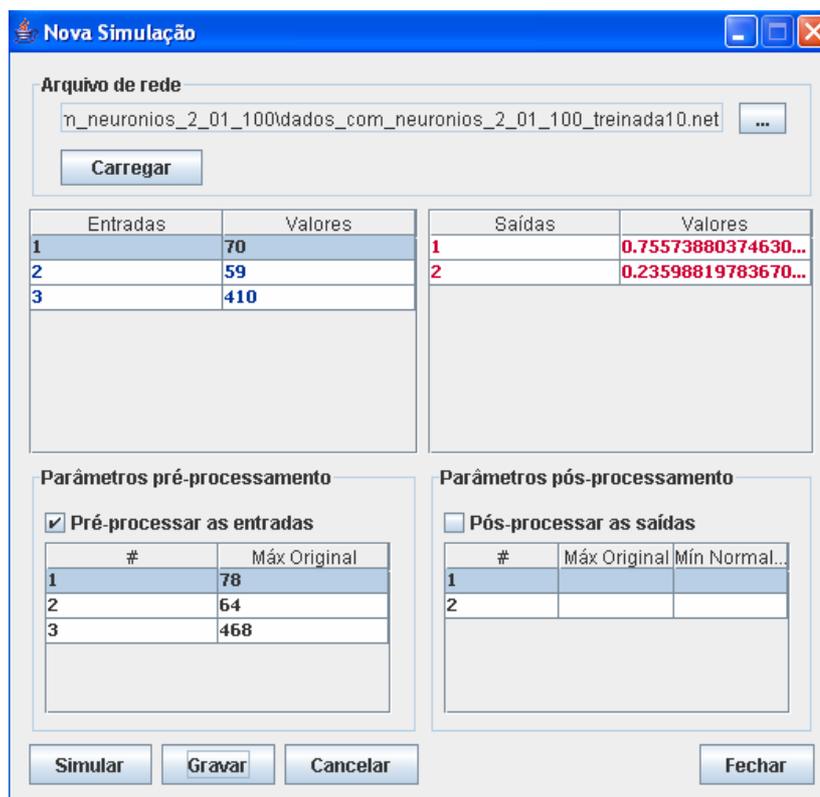


**Figura 18.** Opção de menu para realizar simulações

Na janela intitulada “Nova Simulação”, é possível configurar os parâmetros necessários à realização da simulação selecionando primeiramente o arquivo da rede neural que será utilizada e posteriormente informando o valor para cada entrada da rede. O arquivo de rede consiste num arquivo texto com extensão .net onde ficam registradas a quantidade de camadas da rede, a quantidade de neurônios em cada camada, a interligação entre os neurônios e o peso de cada ligação neuronal.

Para configurar o parâmetro de pré-processamento correspondente a cada entrada basta marcar o *checkbox* “Pré-processar as entradas” e digitar na tabela abaixo dele o valor máximo de cada campo que foi registrado dentre os dados utilizados para aquele campo no treinamento, na mesma ordem que o campo aparece na tabela de entradas, conforme a Figura 18.

Semelhante ao que é feito no pré-processamento das entradas, é possível pós-processar as saídas da rede, bastando para isso, marcar o *checkbox* “Pós-processar as saídas” e informar na tabela correspondente o maior valor (sem estar pré-processado) utilizado para aquele campo no treinamento e o valor mínimo utilizado em sua normalização, na mesma ordem que as saídas aparecem na tabela de saídas, conforme a Figura 19.

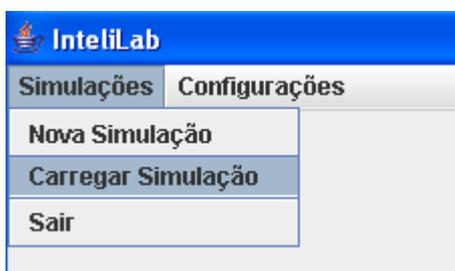


**Figura 19.** Tela de realização de simulações

Clicando em “Simular”, o programa executa uma simulação passando os dados de entrada informados para a rede neural previamente selecionada. O resultado da simulação é exibido na tabela de saídas ao lado da saída correspondente. É dada ao usuário a possibilidade de gravar os parâmetros da simulação realizada, caso o mesmo queira reavaliar a simulação ou executá-la novamente.

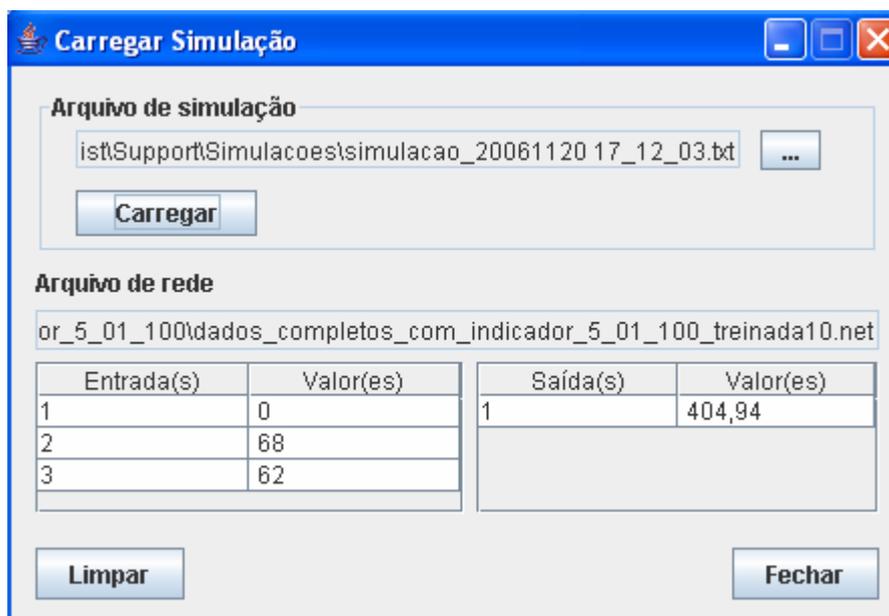
## 2. Carregar Simulação:

Permite que o usuário carregue uma simulação salva. Para acessar este recurso basta clicar na opção de menu “Simulações→Carregar Simulação”, conforme a Figura 20:



**Figura 20.** Opção de menu para carregar simulações

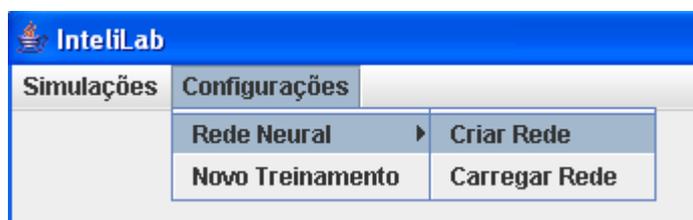
Na nova janela intitulada “Carregar Simulação”, o usuário pode selecionar um arquivo de simulação salvo anteriormente, sendo assim exibidas as informações daquela simulação, conforme a Figura 21.



**Figura 21.** Tela de carregar simulações

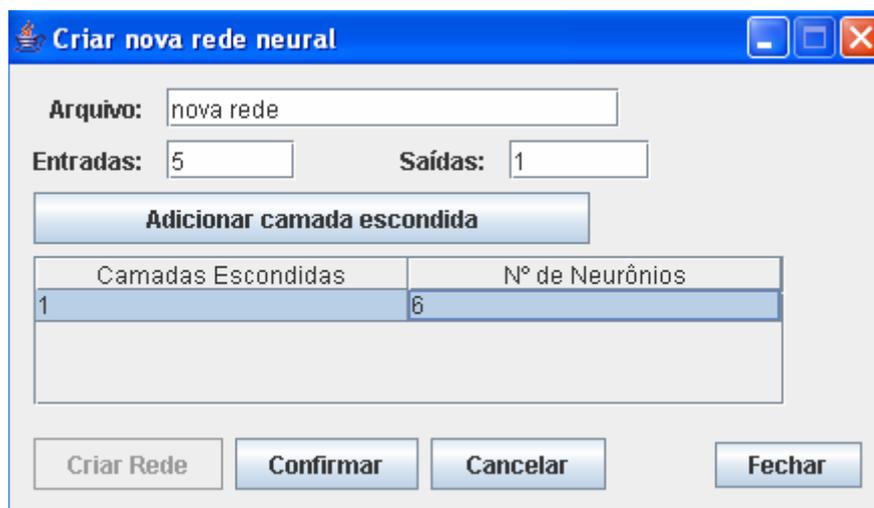
### 3. Criar Rede Neural:

Permite a criação de uma nova rede neural de acordo com as necessidades do usuário. Para acessar este recurso é preciso clicar na opção de menu “Configurações→Rede Neural→Criar Rede”, conforme a Figura 22.



**Figura 22.** Opção de menu para criar novas redes neurais

Na nova janela intitulada “Criar Rede”, o usuário deve preencher as informações necessárias para a criação da nova rede, são elas: o nome da rede, que será usado como nome do arquivo da rede neural; as entradas e as saídas da rede neural e também a quantidade de camadas escondidas e o número de neurônios em cada uma delas, de acordo com a Figura 23.

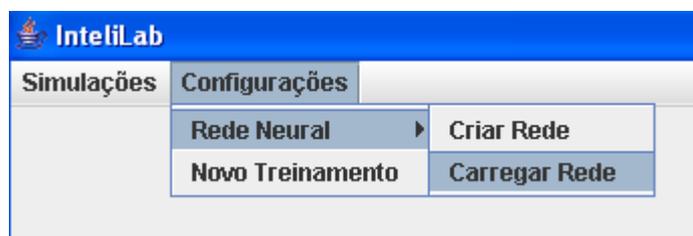


**Figura 23.** Tela de criação de novas redes

Clicando em “Confirmar”, a nova rede neural será salva na pasta “redes” que fica no mesmo diretório onde está instalado o programa.

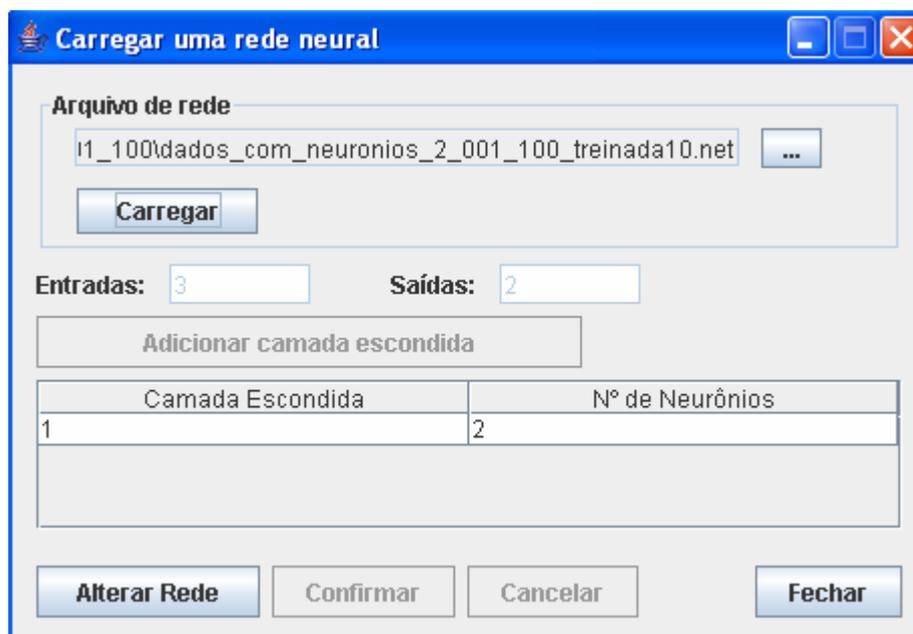
#### 4. Carregar Rede Neural:

Permite carregar e editar as configurações de uma rede neural de acordo com as necessidades do usuário. Para acessar este recurso, basta clicar na opção de menu “Configurações→Rede Neural→Carregar Rede” conforme mostrado na Figura 24.



**Figura 24.** Opção de menu para carregar as informações de um arquivo de rede neural

Na nova janela intitulada “Carregar Rede”, as informações do arquivo de rede salvo serão carregadas (ver Figura 25), são elas: o título da rede neural (no caso o nome do arquivo), suas entradas, saídas e camadas escondidas.



**Figura 25.** Tela de alteração dos parâmetros de uma rede neural

Depois de carregadas as informações da rede neural, clicando em “Alterar Rede” o programa permite alterar os parâmetros da rede neural. Clicando em “Confirmar” as alterações realizadas serão salvas num arquivo mantendo o nome original.

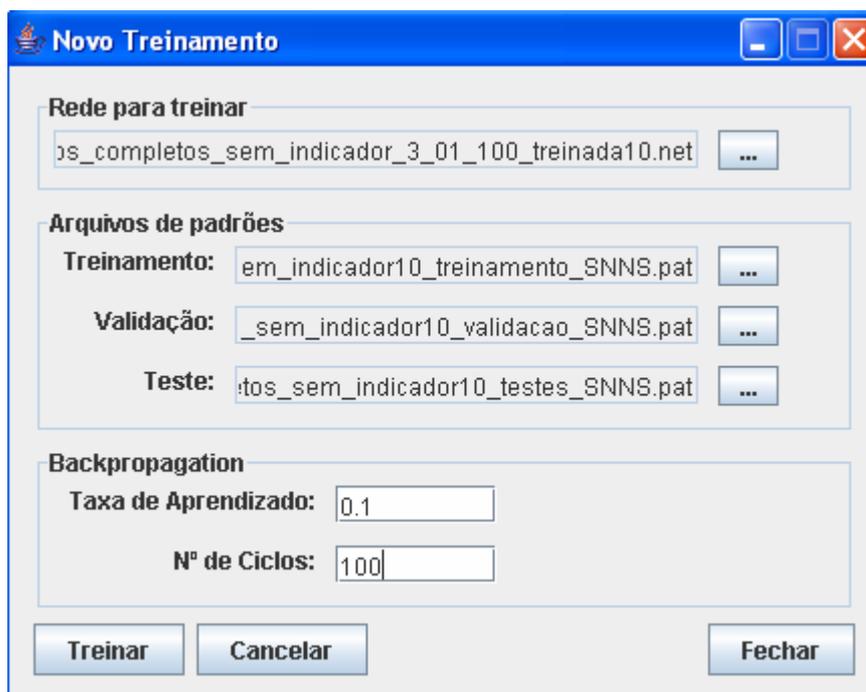
#### 5. Novo Treinamento:

Permite realizar um novo treinamento tanto em redes que já foram treinadas como em redes que ainda não receberam nenhum treinamento. Para acessar este recurso, basta clicar na opção de menu “Configurações→Novo Treinamento” conforme mostrado na Figura 26:



**Figura 26.** Opção de menu para treinar uma rede neural

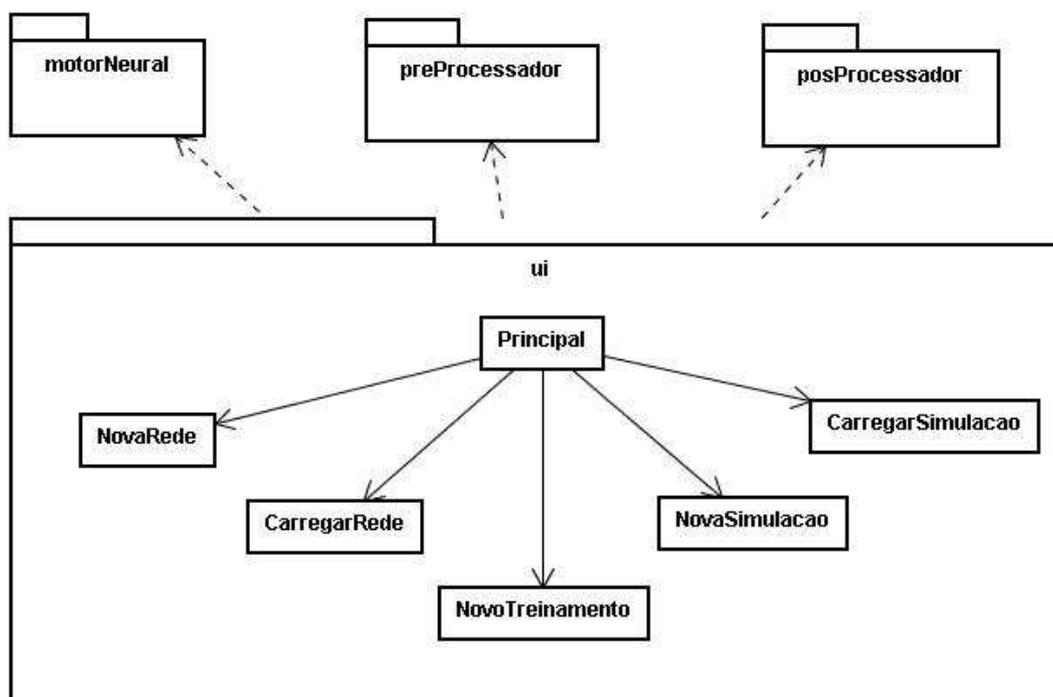
Na nova janela intitulada “Novo Treinamento” (ver Figura 27), o usuário escolhe a rede que deseja treinar e seleciona também os arquivos que serão utilizados para treinamento, validação e testes. Além disso, para realizar o treinamento é preciso definir os parâmetros para o algoritmo de aprendizado, que são: taxa de aprendizado e o número de ciclos para a finalização do treinamento.



**Figura 27.** Tela de treinamento de uma rede neural

Clicando em “Treinar”, o programa executa a rotina de treinamento e mostra no final os resultados obtidos.

O código-fonte está organizado em pacotes que modularizam o programa em relação às responsabilidades das diversas classes, agrupando classes comuns em um mesmo pacote. O diagrama de classes resumido é mostrado na Figura 28 que inclui os pacotes que foram importados do projeto InteliColheita e foram utilizados nas tarefas de predição (motorNeural), pré-processamento (preProcessador) e pós-processamento (posProcessador). Dentre os pacotes importados o que merece maior destaque é o do motor neural que foi descrito na subseção 2.3.2, os outros dois pacotes são mais simples no tocante a sua função, sendo a maior parte de suas classes usadas para a manipulação dos arquivos.



**Figura 28.** Diagrama de classes resumido do IntelliLab

### 3.5 Integração com o InteliColheita

Inicialmente, foi preciso identificar todas as classes do InteliColheita que seriam necessárias para o projeto e também o relacionamento entre elas. Feito isso, foram selecionados os pacotes das classes identificadas como necessárias para a mesma pasta onde ficam os pacotes no projeto IntelliLab; depois, foi preciso apenas importar estes pacotes através do NetBeans para este mesmo projeto.

A única adaptação necessária foi nas diretivas import das classes importadas do InteliColheita, apenas para ajustá-las à nova estrutura de pastas sob a qual estas classes seriam utilizadas. Reafirmando assim a possibilidade e a praticidade de integração do motor neural e dos módulos de processamento de dados do InteliColheita em outros projetos.

### 3.6 Ferramentas e Tecnologias Utilizadas

Utilizamos neste projeto algumas ferramentas disponíveis gratuitamente na *web* para nos auxiliar na implementação do código e na documentação das informações. Dentre as ferramentas e tecnologias que contribuíram para a execução deste trabalho, destacam-se:

- JDK 1.5.0\_06
- NetBeans 5.0;
- JUDE 3.1.1.

A escolha da linguagem Java já foi justificada na seção 3.3. Quanto a IDE NetBeans sua escolha se baseou na facilidade que este ambiente oferece na criação do ambiente visual do programa e por outros recursos que auxiliam o programador na implementação do código, como “*code-completion*” com *help* integrado. Para as modelagens UML utilizamos o programa JUDE, mais informações a respeito do JUDE e do NetBeans podem ser encontradas no ANEXO A.

# Capítulo 4

## Testes e Resultados

Para validar e avaliar os resultados obtidos com o desenvolvimento do InteliLab foi adotada a seguinte estratégia:

- Definir alguns experimentos que permitam explorar as possibilidades de uso da ferramenta sobre os dados disponíveis e, realizar algumas simulações comparando os resultados obtidos com dados reais, para assim verificar quão possível é utilizar a ferramenta proposta nesta ou em outras práticas laboratoriais;

- Fundamentar a análise dos resultados e dos aspectos subjetivos, resultantes do uso da ferramenta desenvolvida com o depoimento de usuários futuros do programa e a partir de um questionário de aplicabilidade do InteliLab (ver Apêndices A e B) justificando assim, a avaliação do trabalho como um todo.

### 4.1 Dados Disponíveis

Na aplicação atual, trabalhamos com uma quantidade muito reduzida de padrões se compararmos com a quantidade de dados normalmente utilizada em outros trabalhos envolvendo o uso de redes neurais. A pequena quantidade de dados disponível se deu por termos escolhido, na fase de elaboração do projeto, um experimento de alto custo de tempo e trabalho do pesquisador na fase de coleta e preparação dos dados. Isso não permite que uma quantidade maior de animais seja utilizada, observando-se que mais ratos podem ser utilizados para compensar erros, mortes prematuras e desequilíbrio nas condições experimentais, além de outros ratos serem utilizados numa mesma pesquisa para avaliar diferentes aspectos do mesmo problema.

Escolhemos este experimento porque na época era o mais avançado dentre todos os projetos em andamento no Departamento de Anatomia da UFPE, inclusive com artigos sendo escritos que hoje já se encontram publicados.

Uma peculiaridade a respeito dos dados disponíveis é que apenas em alguns o valor dos neurônios é encontrado, isto porque para a pesquisa não é necessário contabilizar a quantidade dos neurônios em todos os animais, sendo assim apenas alguns eram escolhidos aleatoriamente e utilizados neste processo.

Numa tentativa de contornar ou minimizar o impacto de utilizarmos uma quantidade pequena de padrões na fase de treinamento, adotamos a técnica de *crossvalidation* dos dados nesta fase. Esta técnica consiste em combinar os dados disponíveis para treinamento da rede de modo a se obter vários conjuntos de arquivos de treinamento, todos diferentes entre si,

possibilitando realizar mais treinamentos com as redes neurais utilizando sempre conjuntos diferentes de dados em cada fase do treinamento. Na Tabela 1, são exibidas as informações coletadas na pesquisa, referentes aos animais nutridos, cujos dados foram disponibilizados para este trabalho:

**Tabela 1.** Dados disponíveis de animais nutridos

	<b>Comprimento da alça intestinal (cm)</b>	<b>Peso corpóreo (g)</b>	<b>Nº de Neurônios</b>
1	85	50	-
2	77	56,5	-
3	78	64	429
4	76	58	376
5	71,4	57	-
6	67,7	58,5	-
7	72,9	60	-
8	72	57	-
9	67	55	-
10	52	44,5	-
11	75	65,5	468
12	64	53,5	379
13	72,8	48,5	-
14	70,2	41	-
15	72,2	54	-
16	70,4	43,5	-
17	68,7	48,5	-

Na Tabela 2, são exibidas as informações coletadas na pesquisa referentes aos animais desnutridos, cujos dados também foram disponibilizados para este trabalho:

**Tabela 2.** Dados disponíveis de animais desnutridos

	<b>Comprimento da alça intestinal (cm)</b>	<b>Peso corpóreo (g)</b>	<b>Nº de Neurônios</b>
1	46	17	-
2	49	24	-
3	53,5	21	-
4	55	19	-
5	55	22,5	-
6	53,5	22,5	-
7	52,6	22,5	-
8	53,6	25,5	-
9	45,9	21	-
10	56,6	27	329
11	48,3	23	324
12	45,7	32	-
13	46,5	28	-
14	62,2	30	316
15	55,5	27	-
16	51,3	30	-

17	47,5	28	358
----	------	----	-----

Apenas os animais com as informações completas foram utilizados nos experimentos que seguem, um detalhe é que acrescentamos um novo animal com a quantidade de neurônios igual à média obtida através dos dados coletados para este parâmetro, desta forma conseguimos mais dois padrões para serem usados nos treinamentos. Nesta fase dos treinamentos, o foco não é a utilização dos resultados das simulações para fins científicos na área de biologia, desta forma não estamos obrigados a trabalhar estritamente com dados reais, justificando assim a decisão tomada.

Os padrões de cada grupo selecionados para serem utilizados nos treinamentos são exibidos nas Tabelas 3 e 4.

**Tabela 3.** Dados de animais nutridos selecionados para uso nos treinamentos

	Comprimento da alça intestinal (cm)	Peso corpóreo (g)	Nº de Neurônios
1	78	64	429
2	76	58	376
3	75	65,5	468
4	64	53,5	379

**Tabela 4.** Dados de animais desnutridos selecionados para uso nos treinamentos

	Comprimento da alça intestinal (cm)	Peso corpóreo (g)	Nº de Neurônios
1	56,6	27	329
2	48,3	23	324
3	62,2	30	316
4	47,5	28	358

Um animal de cada grupo, que não pertencia as Tabelas 3 e 4, foi escolhido aleatoriamente para a criação dos padrões virtuais utilizados para aumentar a quantidade de dados na realização dos treinamentos, de modo que apenas o campo “Nº de neurônios” destes animais ficou sendo a média dos valores já conhecidos para este campo. Os dados resultantes deste procedimento encontram-se na Tabela 5.

**Tabela 5.** Dados dos animais virtuais criados para uso nos treinamentos

	Comprimento da alça intestinal (cm)	Peso corpóreo (g)	Nº de Neurônios	Classificação
1	72,9	60	413	Nutrido
2	53,6	25,5	332	Desnutrido

Os animais que restaram não foram utilizados nos treinamentos, porém alguns deles foram aproveitados na realização das simulações, abaixo seguem as Tabelas 6 e 7, contendo os dados dos animais de cada grupo não utilizados nos treinamentos:

**Tabela 6.** Dados dos animais nutridos não utilizados nos treinamentos

	Comprimento da alça intestinal (cm)	Peso corpóreo (g)
1	85	50
2	77	56,5

3	71,4	57
4	67,7	58,5
5	72,9	60
6	72	57
7	67	55
8	52	44,5
9	72,8	48,5
10	70,2	41
11	72,2	54
12	70,4	43,5
13	68,7	48,5

**Tabela 7.** Dados dos animais desnutridos não utilizados nos treinamentos

	<b>Comprimento da alça intestinal (cm)</b>	<b>Peso corpóreo (g)</b>
1	46	17
2	49	24
3	53,5	21
4	55	19
5	55	22,5
6	53,5	22,5
7	52,6	22,5
8	53,6	25,5
9	45,9	21
10	45,7	32
11	46,5	28
12	55,5	27
13	51,3	30

## 4.2 Experimentos

Inicialmente havíamos definido que o trabalho seria desenvolver uma ferramenta para ser utilizada na predição de neurônios em uma pesquisa específica do Departamento de Anatomia da UFPE. Durante o desenvolvimento do programa, verificamos que seria melhor construí-lo de maneira que fosse mais flexível, possibilitando assim ser utilizado em diferentes pesquisas ou ser utilizado com focos diferentes para um mesmo problema. Dessa forma, a ferramenta deveria possibilitar a construção de novas redes com diferentes características.

A fim de se explorar e avaliar esta flexibilidade, consideramos duas situações para a pesquisa em questão nas quais o uso de redes neurais seria interessante. A primeira delas diz respeito à classificação dos padrões informados em nutridos e desnutridos, baseado nas entradas informadas; e a outra consiste em prever quantos neurônios aproximadamente seriam encontrados no plexo entérico de um animal baseado em algumas entradas que seriam informadas, os dois problemas serão mais bem detalhados a seguir.

### 4.2.1 Classificação

Para a realização deste experimento utilizamos duas configurações de redes neurais: na primeira, (ver Tabela 8) os dados de entrada são o comprimento da alça intestinal, o peso do animal e a quantidade de neurônios encontrada em seu plexo entérico, e a saída é composta por dois neurônios que indicam uma chance (percentual) de que o padrão informado pertença ao grupo de nutridos (indicador nutrido) ou ao grupo de desnutridos (indicador desnutrido).

**Tabela 8.** Configuração de rede 1 para o experimento de classificação

<b>Entradas</b>	Comprimento da alça intestinal
	Peso corpóreo
	Quantidade de neurônios no plexo entérico
<b>Saídas</b>	Indicador nutrido
	Indicador desnutrido

Na segunda configuração utilizada (ver Tabela 9), retiramos um dos campos de entrada, especificamente, a quantidade de neurônios encontrada no plexo entérico do animal. No treinamento das redes para estas duas configurações utilizamos os dados que tinham as informações completas e os animais virtuais criados, sempre fazendo *crossvalidation* entre eles.

**Tabela 9.** Configuração de rede 2 para o experimento de classificação

<b>Entradas</b>	Comprimento da alça intestinal
	Peso corpóreo
<b>Saídas</b>	Indicador nutrido
	Indicador desnutrido

### 4.2.2 Previsão

Este é o experimento vislumbrado no início deste trabalho. Identificamos quatro configurações de redes para serem investigadas via simulações. Isto de acordo com as possibilidades que os dados disponíveis ofereciam. Na primeira configuração (ver Tabela 10), a rede possui quatro entradas, que são o comprimento da alça intestinal, o peso do animal e dois indicadores para determinar se aquele animal é nutrido (indicador nutrido) ou desnutrido (indicador desnutrido); esta rede tem apenas uma saída que corresponde ao valor predito da quantidade de neurônios presente no plexo entérico. No treinamento desta rede, foram utilizados todos os animais do grupo nutrido e desnutrido que tinham as informações completas mais os dois animais virtuais criados.

**Tabela 10.** Configuração de rede 1 para o experimento de previsão

<b>Entradas</b>	Indicador nutrido
	Indicador desnutrido
	Comprimento da alça intestinal
	Peso corpóreo
<b>Saída</b>	Valor predito da quantidade de neurônios presente no plexo entérico

Na segunda configuração (ver Tabela 11), utilizamos apenas duas entradas, o comprimento da alça intestinal e o peso do animal, e a saída corresponde ao valor predito da

quantidade de neurônios presente no plexo entérico. Os dados utilizados no treinamento foram exatamente os mesmos da primeira configuração.

**Tabela 11.** Configuração de rede 2 para o experimento de predição

<b>Entradas</b>	Comprimento da alça intestinal
	Peso corpóreo
<b>Saída</b>	Valor predito da quantidade de neurônios presente no plexo entérico

As duas últimas configurações possuem as mesmas características da configuração anterior (ver Tabela 11), ou seja, duas entradas (comprimento da alça intestinal e peso corporal) e uma saída (quantidade de neurônios no plexo entérico). A diferença é que as redes foram utilizadas para trabalhar especificamente com cada grupo (nutrido ou desnutrido). Portanto, para uma das configurações foram utilizados apenas os dados do grupo nutrido no treinamento e para a outra apenas os dados do grupo desnutrido.

## 4.3 Simulações

As simulações foram divididas nas seguintes etapas:

- Pré-processamento dos dados relacionados a cada configuração do experimento correspondente e manipulação dos arquivos de treinamento para aplicar a técnica de *crossvalidation*;
- Criação das redes de acordo com as configurações pré-estabelecidas;
- Treinamento das redes;
- Escolha da melhor rede;
- Simulações utilizando a melhor rede em cada experimento.

### 4.3.1 Pré-processamento dos dados de treinamento

Nesta etapa, utilizamos uma ferramenta que compõe os vários módulos do projeto InteliColheita do DSC-Poli e serve para pré-processar dados gerando arquivos no mesmo padrão utilizado pelo simulador de redes neurais da Universidade de Stuttgart, chamado Stuttgart Neural Network Simulator (SNNS) [36].

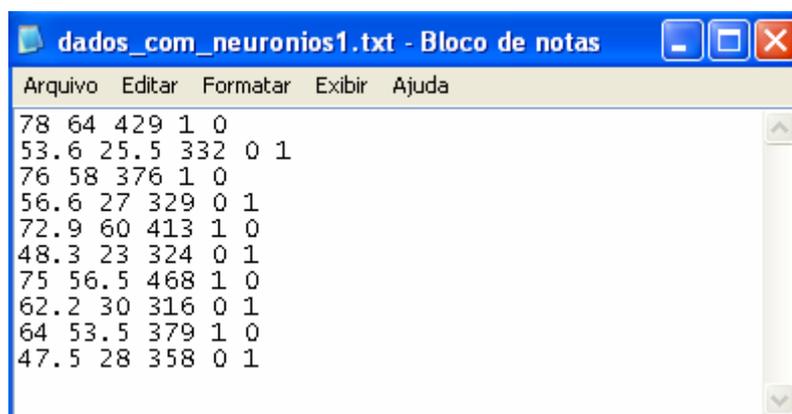
Para utilizar o programa de pré-processamento basta criar um arquivo de texto com todos os padrões que se deseja pré-processar, o arquivo precisa obedecer a uma formatação que é especificada no próprio pré-processador. Depois, este arquivo deve ser importado para o programa, os parâmetros do pré-processamento devem ser informados e então o arquivo é pré-processado. O resultado deste procedimento é um outro arquivo de texto com os valores pré-processados na mesma ordem em que apareciam no arquivo original e três arquivos de extensão .pat para serem usados no treinamento da rede neural.

Os dados utilizados em todos os experimentos foram pré-processados juntos, sendo que cada arquivo era montado conforme as características da rede a qual ele correspondia (de acordo com as subseções 4.2.1 e 4.2.2). Um detalhe importante diz respeito ao uso da técnica de *crossvalidation*, que fazia necessária a geração de vários arquivos de treinamento para um mesmo conjunto de dados; porém, sempre os padrões de treinamento, validação e testes da rede neural se alternavam fazendo com que todos os arquivos criados ficassem diferentes entre si.

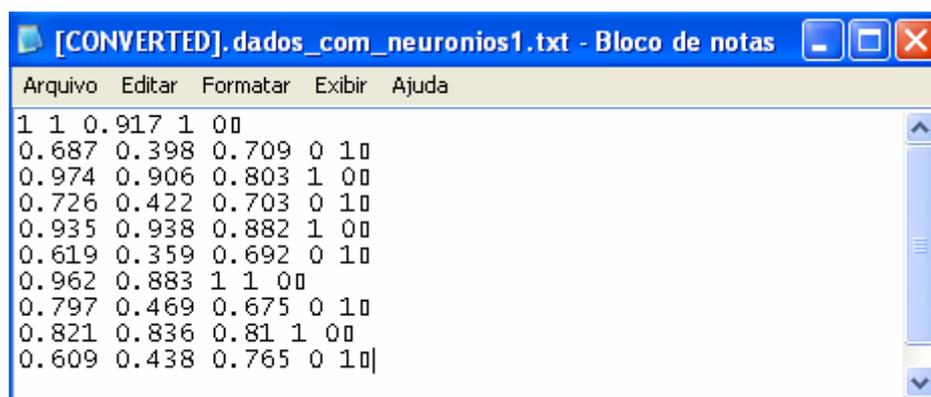
Para o experimento de classificação foram criados dois modelos de arquivo de padrões para treinamento, cada um atendendo uma das duas configurações de redes neurais que foram

definidas para este problema (ver Tabela 8). Portanto, em um dos modelos havia três entradas (comprimento da alça intestinal, peso corpóreo e quantidade de neurônios no plexo entérico) e duas saídas correspondentes aos indicadores de ser um padrão referente a nutridos ou desnutridos. No outro modelo de arquivo, correspondente a segunda configuração para este experimento (ver Tabela 9), havia apenas duas entradas (comprimento da alça intestinal e peso corpóreo) e as mesmas duas saídas do modelo anterior.

Na Figura 29, é possível observarmos um arquivo de dados com os valores originais utilizados para o experimento de classificação, de acordo com o modelo 1 citado no parágrafo anterior. Na Figura 30, temos o resultado do pré-processamento realizado no arquivo da Figura 28.



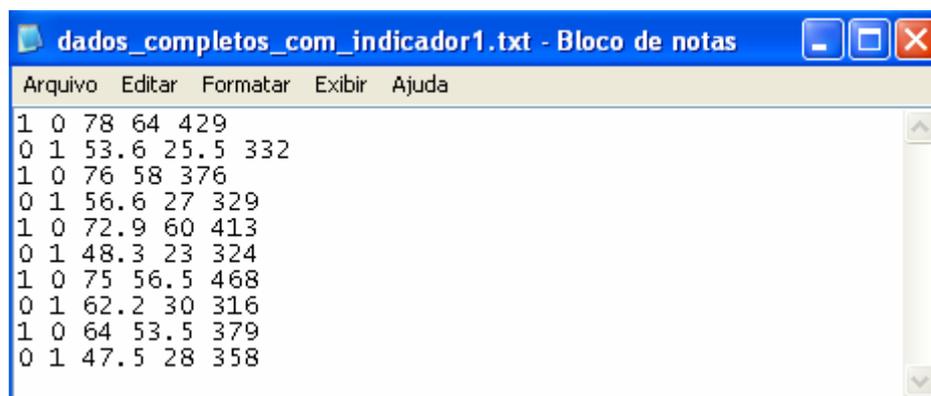
**Figura 29.** Exemplo de arquivo de dados para pré-processamento do experimento de classificação



**Figura 30.** Exemplo de arquivo de dados pré-processado do experimento de classificação

Para o experimento de predição foram criados dois modelos de arquivo de padrões para treinamento, cada um atendendo uma das duas configurações de redes neurais que foram definidas para este problema (ver Tabela 10). Portanto, em um dos modelos havia quatro entradas (indicador nutrido, indicador desnutrido, comprimento da alça intestinal e peso corpóreo) e uma saída correspondente ao valor predito da quantidade de neurônios no plexo entérico dos animais. No outro modelo de arquivo, correspondente a segunda configuração para este experimento (ver Tabela 11), havia apenas duas entradas (comprimento da alça intestinal e peso corpóreo) e a mesma saída do modelo anterior.

Na Figura 31, é possível observarmos um arquivo de dados com os valores originais utilizados para o experimento de predição, de acordo com o modelo 1 citado no parágrafo anterior. Na Figura 32, temos o resultado do pré-processamento realizado no arquivo da Figura 31.



**Figura 31.** Exemplo de arquivo de dados para pré-processamento do experimento de predição



**Figura 32.** Exemplo de arquivo de dados pré-processado do experimento de predição

No pré-processamento de todos os dados destes experimentos (classificação e predição) aplicamos uma normalização entre 0 e 1.

### 4.3.2 Criação das redes neurais

As configurações das redes quanto aos neurônios das camadas de entrada e saída já haviam sido definidas quando determinamos os experimentos que seriam realizados (ver subseções 4.2.1 e 4.2.2). Na etapa de criação das redes, acrescentamos a estas configurações diferentes valores de neurônios na camada escondida, para posteriormente, identificarmos a melhor rede para cada experimento. Apenas para as redes 1 e 4 da tabela 12, não variamos a quantidade de neurônios na camada escondida, por motivos técnicos não realizamos testes com 4 e 3 neurônios na camada escondida das redes 1 e 4 respectivamente. Uma observação importante é que todas as redes criadas para os dois experimentos possuem apenas uma camada escondida.

A partir da etapa de criação das redes, todo trabalho envolve o uso direto do InteliLab. As redes criadas para cada experimento são descritas na Tabela 8:

**Tabela 12.** Configuração das redes neurais criadas para cada experimento

Redes	Experimentos	Entradas	Saídas	Neurônios na Camada Escondida
1	Classificação (Configuração 1 – ver Tabela 8)	Comprimento da alça intestinal	Indicador nutrido	2 neurônios
		Peso corpóreo	Indicador desnutrido	

		Nº de neurônios no plexo entérico		
2	Classificação (Configuração 2 – ver Tabela 9)	Comprimento da alça intestinal	Indicador nutrido	2 neurônios
		Peso corpóreo	Indicador desnutrido	
3	Classificação (Configuração 2 – ver Tabela 9)	Comprimento da alça intestinal	Indicador nutrido	3 neurônios
		Peso corpóreo	Indicador desnutrido	
4	Predição (Configuração 1 – ver Tabela 10)	Indicador nutrido	Nº de neurônios no plexo entérico	5 neurônios
		Indicador desnutrido		
		Comprimento da alça intestinal		
		Peso corpóreo		
5	Predição (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	1 neurônios
		Peso corpóreo		
6	Predição (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	3 neurônios
		Peso corpóreo		
7	Predição - apenas nutridos (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	1 neurônios
		Peso corpóreo		
8	Predição - apenas nutridos (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	3 neurônios
		Peso corpóreo		
9	Predição - apenas desnutridos (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	1 neurônios
		Peso corpóreo		
10	Predição - apenas desnutridos (Configuração 2 – ver Tabela 11)	Comprimento da alça intestinal	Nº de neurônios no plexo entérico	3 neurônios
		Peso corpóreo		

Na primeira coluna da Tabela 8, fica o número da rede criada, na segunda coluna é informado o experimento (classificação ou predição) e a configuração da rede no experimento (conforme seções 4.2.1 e 4.22). Na terceira coluna da Tabela 8 ficam registradas as entradas de cada rede, na quarta coluna estão representadas as saídas, e na quinta coluna a quantidade de neurônios na camada escondida daquela rede.

### 4.3.3 Treinamento das redes criadas

Criados os arquivos de treinamento e as redes neurais, definimos as taxas de aprendizado e os ciclos que utilizaríamos para treinar as redes, tudo feito com base nas boas práticas de

treinamento de redes neurais voltada para os dados disponíveis nos experimentos e às características das redes criadas.

Cada rede foi treinada com taxa de aprendizado igual a 0,1 e 0,01 com número de ciclos iguais a 10 e 100, todos os percentuais de erro (MSE) nos treinamentos das redes foram registrados em uma tabela para posterior escolha da melhor rede.

Como utilizamos *crossvalidation*, cada rede precisava ser treinada com todos os arquivos criados a partir da alternância dos padrões nos conjuntos de treinamento, validação e testes. Desta forma, uma mesma rede chegava a ser treinada até 10 vezes para obtenção do MSE.

#### 4.3.4 Escolha da melhor rede

A melhor rede para cada configuração dos dois experimentos foi escolhida com base nos menores percentuais de erro obtidos na fase de treinamento. Obviamente, uma posterior análise era feita durante as simulações para verificar a capacidade de generalização das redes e seu desempenho com novos padrões.

Abaixo estão listadas as configurações de rede e treinamento que obtiveram os melhores resultados em cada experimento:

- Classificação:

- Configuração 1: 3 entradas (comprimento do intestino delgado, peso corporal e quantidade de neurônios no plexo entérico) e 2 saídas (indicador de nutrido e indicador de desnutrido). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 2 neurônios na camada escondida, treinada com taxa de aprendizado de 0,1 e 100 ciclos que apresentou MSE igual a 0,09053;
- Configuração 2: 2 entradas (comprimento do intestino delgado e peso corporal) e 2 saídas (indicador de nutrido e indicador de desnutrido). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 2 neurônios na camada escondida, treinada com taxa de aprendizado de 0,1 e 100 ciclos que apresentou MSE igual a 0,00115168;

- Predição:

- Configuração 1: 4 entradas (indicador de nutrido, indicador de desnutrido, comprimento do intestino delgado e peso corporal) e 1 saída (quantidade de neurônios no plexo entérico). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 5 neurônios na camada escondida, treinada com taxa de aprendizado de 0,1 e 100 ciclos que apresentou MSE igual a 0,001361;
- Configuração 2: 2 entradas (comprimento do intestino delgado e peso corporal) e 1 saída (quantidade de neurônios no plexo entérico). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 1 neurônio na camada escondida, treinada com taxa de aprendizado de 0,01 e 100 ciclos que apresentou MSE igual a 0,00000945;

- Predição (apenas nutridos):

- Configuração 1: 2 entradas (comprimento do intestino delgado e peso corporal) e 1 saída (quantidade de neurônios no plexo entérico). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 3 neurônios na camada escondida, treinada com taxa de aprendizado de 0,1 e 100 ciclos que apresentou MSE igual a 0,0076;

- Predição (apenas desnutridos):

- **Configuração 1:** 2 entradas (comprimento do intestino delgado e peso corporal) e 1 saída (quantidade de neurônios no plexo entérico). A rede que obteve o menor erro MSE durante os treinamentos foi a rede com 3 neurônios na camada escondida, treinada com taxa de aprendizado de 0,1 e 100 ciclos que apresentou MSE igual a 0,00000951;

### 4.3.5 Simulações

Até esta etapa, o InteliLab já foi utilizado para criar redes, carregar e alterar configurações de redes existentes e realizar treinamentos. Mas é a tarefa desempenhada no tópico atual que concentra a principal funcionalidade do programa, pois é nela que provavelmente o eventual usuário iria obter informações para usar em suas pesquisas ou na elaboração de suas hipóteses.

As simulações neste trabalho serviram como um meio de avaliar os aspectos inteligentes e funcionais do programa, ou seja, verificar se o objetivo de implementar uma ferramenta que implementa computação inteligente através do uso de redes neurais foi cumprida e quão satisfatoriamente isto foi desempenhado.

Dividimos as simulações em dois grupos para cada melhor rede identificada: um deles que utiliza apenas padrões desconhecidos, que são aqueles que ficaram de fora dos treinamentos, e outro com padrões que foram utilizados nos treinamentos. Para cada grupo, em cada uma das redes, foram feitas duas simulações: uma, utilizando um padrão correspondente a um animal conhecidamente nutrido e, outra, com um desnutrido, exceto para as redes de predição treinadas especificamente com animais nutridos ou desnutridos.

## 4.4 Resultados

Nesta seção detalharemos cada simulação realizada, com tabelas informando os valores de entrada utilizados e os resultados obtidos, além do percentual de erro em algumas simulações do experimento de predição.

### Experimento 1: “Classificação”

Para este experimento, realizamos oito simulações utilizando diferentes configurações de rede e, também, variando os dados de entrada. Cada uma das simulações será detalhada a seguir.

- 1ª e 2ª Simulações: conforme pode ser observado na Tabela 13, estas simulações utilizaram dados provenientes de padrões não utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 1, descrita na subseção 4.3.4, e foi aplicada para um padrão conhecidamente nutrido e para outro conhecidamente desnutrido. Nas duas situações avaliadas, a classificação da rede induziu ao valor correto, pois o resultado percentual sugeriu que o animal pertencesse ao grupo que de fato ele fazia parte.

**Tabela 13.** Resultado da classificação utilizando a configuração 1 para novos padrões

Novos padrões				
	Nutrido		Desnutrido	
Entradas	Comprimento da alça intestinal	72,2	Comprimento da alça intestinal	55
	Peso corpóreo	54	Peso corpóreo	24
	Quantidade de neurônios	390	Quantidade de neurônios	330
Saídas	Resultado esperado	100% - 0%	Resultado esperado	0%-100%

	Resultado obtido	68,95% - 29,68%	Resultado obtido	18,88% - 81,48%
--	------------------	-----------------	------------------	-----------------

- 3ª e 4ª Simulações: conforme pode ser observado na Tabela 14, estas simulações utilizaram dados provenientes de padrões utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 1, descrita na subseção 4.3.4, e foi aplicada para um padrão conhecidamente nutrido e para outro conhecidamente desnutrido. Nas duas situações avaliadas, a classificação da rede induziu ao valor correto, pois o resultado percentual sugeriu que o animal pertencesse ao grupo que de fato ele fazia parte.

**Tabela 14.** Resultado da classificação utilizando a configuração 1 para padrões conhecidos

Padrões conhecidos				
	Nutrido		Desnutrido	
<b>Entradas</b>	Comprimento da alça intestinal	75	Comprimento da alça intestinal	48,3
	Peso corpóreo	56,5	Peso corpóreo	23
	Quantidade de neurônios	468	Quantidade de neurônios	324
<b>Saídas</b>	Resultado Esperado	100%-0%	Resultado Esperado	0%-100%
	Resultado Obtido	66,16%-30,60%	Resultado Obtido	19,28%-83,05%

- 5ª e 6ª Simulações: conforme pode ser observado na Tabela 15, estas simulações utilizaram dados provenientes de padrões não utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 2, descrita na subseção 4.3.4, e foi aplicada para um padrão conhecidamente nutrido e para outro conhecidamente desnutrido. Nas duas situações avaliadas, a classificação da rede induziu ao valor correto, pois o resultado percentual sugeriu que o animal pertencesse ao grupo que de fato ele fazia parte.

**Tabela 15.** Resultado da classificação utilizando a configuração 2 para novos padrões

Novos padrões				
	Nutrido		Desnutrido	
<b>Entradas</b>	Comprimento da alça intestinal	72,2	Comprimento da alça intestinal	55
	Peso corpóreo	54	Peso corpóreo	19
<b>Saídas</b>	Resultado Esperado	100%-0%	Resultado Esperado	0%-100%
	Resultado Obtido	65,50%-37,31%	Resultado Obtido	19,21%-73,93%

- 7ª e 8ª Simulações: conforme pode ser observado na Tabela 16, estas simulações utilizaram dados provenientes de padrões utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 2, descrita na subseção 4.3.4, e foi aplicada para um padrão conhecidamente nutrido e para outro conhecidamente desnutrido. Nas duas situações avaliadas, a classificação da rede induziu ao valor correto, pois o resultado percentual sugeriu que o animal pertencesse ao grupo que de fato ele fazia parte.

**Tabela 16.** Resultado da classificação utilizando a configuração 2 para padrões conhecidos

Padrões conhecidos				
	Nutrido		Desnutrido	
Entradas	Comprimento da alça intestinal	75	Comprimento da alça intestinal	48,3
	Peso corpóreo	56,5	Peso corpóreo	23
Saídas	Resultado Esperado	100%-0%	Resultado Esperado	0%-100%
	Resultado Obtido	68,19%-35,11%	Resultado Obtido	25,75%-68,58%

Experimento 2: “Predição”

Para este experimento realizamos 12 simulações, utilizando diferentes configurações de rede e também variando os dados de entrada. Cada uma das simulações será detalhada a seguir.

- 1ª e 2ª Simulações: conforme pode ser observado na Tabela 17, estas simulações utilizaram dados provenientes de padrões não utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 1 do experimento predição, descrita na subseção 4.3.4, e foi aplicada na predição da quantidade de neurônios de um padrão nutrido e outro desnutrido. Não foi possível avaliar o erro entre o valor real e o valor predito, porque a quantidade de neurônios para estes padrões é desconhecida.

**Tabela 17.** Resultado da predição utilizando a configuração 1 para novos padrões

Novos padrões				
	Nutrido		Desnutrido	
Entradas	Indicador nutrido	1	Indicador nutrido	0
	Indicador desnutrido	0	Indicador desnutrido	1
	Comprimento da alça intestinal	71,4	Comprimento da alça intestinal	46
	Peso corpóreo	57	Peso corpóreo	17
Saídas	Média	413	Média	332
	Resultado obtido	403	Resultado obtido	317

- 3ª e 4ª Simulações: conforme pode ser observado na Tabela 18, estas simulações utilizaram dados provenientes de padrões utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 1 do experimento predição, descrita na subseção 4.3.4, e foi aplicada na predição da quantidade de neurônios de um padrão nutrido e outro desnutrido. Ainda na Tabela 18, está registrado o erro identificado em cada simulação.

**Tabela 18.** Resultado da predição utilizando a configuração 1 para padrões conhecidos

Padrões conhecidos				
	Nutrido		Desnutrido	
Entradas	Indicador nutrido	1	Indicador nutrido	0
	Indicador desnutrido	0	Indicador desnutrido	1
	Comprimento da alça intestinal	64	Comprimento da alça intestinal	56,6
	Peso corpóreo	53,5	Peso corpóreo	27
Saídas	Resultado		Resultado	

	esperado	379	esperado	329
	Resultado obtido	399	Resultado obtido	337
	Erro observado	5,27%	Erro observado	2,42%

- 5ª e 6ª Simulações: conforme pode ser observado na Tabela 19, estas simulações utilizaram dados provenientes de padrões não utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 2 do experimento predição, descrita na subseção 4.3.4, e foi aplicada na predição da quantidade de neurônios de um padrão nutrido e outro desnutrido. Não foi possível avaliar o erro entre o valor real e o valor predito, porque a quantidade de neurônios para estes padrões é desconhecida.

**Tabela 19.** Resultado da predição utilizando a configuração 2 para novos padrões

Novos padrões				
	Nutrido		Desnutrido	
<b>Entradas</b>	Comprimento da alça intestinal	72,2	Comprimento da alça intestinal	55
	Peso corpóreo	54	Peso corpóreo	19
<b>Saídas</b>	Média	413	Média	332
	Resultado obtido	468	Resultado obtido	318

- 7ª e 8ª Simulações: conforme pode ser observado na Tabela 20, estas simulações utilizaram dados provenientes de padrões utilizados no treinamento da rede selecionada. Nestas simulações, a rede escolhida segue a configuração 2 do experimento predição, descrita na subseção 4.3.4, e foi aplicada na predição da quantidade de neurônios de um padrão nutrido e outro desnutrido. Ainda na Tabela 20, está registrado o erro identificado em cada simulação.

**Tabela 20.** Resultado da predição utilizando a configuração 2 para padrões conhecidos

Padrões conhecidos				
	Nutrido		Desnutrido	
<b>Entradas</b>	Comprimento da alça intestinal	78	Comprimento da alça intestinal	62,2
	Peso corpóreo	64	Peso corpóreo	30
<b>Saídas</b>	Resultado esperado	429	Resultado esperado	316
	Resultado obtido	401	Resultado obtido	344
	Erro observado	6,53%	Erro observado	8,86%

- 9ª e 10ª Simulações: conforme pode ser observado na Tabela 21, nestas simulações a rede escolhida segue a configuração 1 do experimento predição específica de nutrientes, descrita na subseção 4.3.4. Ela foi aplicada na predição da quantidade de neurônios de padrões nutridos novos (não utilizados no treinamento) e conhecidos.

**Tabela 21.** Resultado da predição utilizando a configuração 1 para a rede específica de animais nutridos

	Novos padrões		Padrões conhecidos	
<b>Entradas</b>	Comprimento da alça intestinal	68,7	Comprimento da alça intestinal	76
	Peso corpóreo	48,5	Peso corpóreo	58

<b>Saídas</b>	Resultado esperado	413	Resultado esperado	376
	Resultado obtido	406	Resultado obtido	410
	Erro observado	-	Erro observado	9,03%

- 11ª e 12ª Simulações: conforme pode ser observado na Tabela 22, nestas simulações a rede escolhida segue a configuração 1 do experimento predição específica de desnutridos, descrita na subseção 4.3.4. Ela foi aplicada na predição da quantidade de neurônios de padrões desnutridos novos (não utilizados no treinamento) e conhecidos.

**Tabela 22.** Resultado da predição utilizando a configuração 1 para a rede específica de animais desnutridos

	<b>Novos padrões</b>		<b>Padrões conhecidos</b>	
<b>Entradas</b>	Comprimento da alça intestinal	45,9	Comprimento da alça intestinal	47,5
	Peso corpóreo	21	Peso corpóreo	28
<b>Saídas</b>	Resultado esperado	332	Resultado esperado	358
	Resultado obtido	323	Resultado obtido	325
	Erro observado	-	Erro observado	9,22%

## Capítulo 5

# Conclusões e Trabalhos Futuros

Este trabalho se propôs a demonstrar mais uma vez a viabilidade de utilizar técnicas de Inteligência Artificial, como Redes Neurais, em problemas de pesquisas científicas de áreas como a Biologia. Com isso exploramos a flexibilidade de uso das Redes Neurais Artificiais em uma ferramenta para usuários finais, criando um maior leque de possibilidades em pesquisas sobre um problema específico de Biologia.

### 5.1 Contribuições

Muitos trabalhos aplicam neurocomputação em problemas de Biologia, seja no diagnóstico de doenças, na área de genoma ou na predição de características. Porém poucos destes trabalhos estão organizados como uma ferramenta voltada para o usuário final, neste caso o pesquisador-cientista, e tampouco oferecem flexibilidade de uso em novos problemas. Considerando esses aspectos, a maior contribuição que este trabalho oferece é um produto em que um pesquisador, sem conhecimento elevado em RNA, poderá utilizar como complemento às práticas laboratoriais.

A diversidade de simulações que podem ser atingidas variam a princípio, conforme a necessidade do pesquisador e a disponibilidade dos dados estudados na pesquisa. Por exemplo, a pesquisadora que avaliou o InteliLab rapidamente teve condições de prospectar outras simulações com base nas que lhe foram apresentadas. Isso demonstra a simplicidade que existe para utilizar a ferramenta, mesmo para um usuário leigo no uso de RNA.

A título de ilustração do uso da ferramenta, apresentamos ao longo deste trabalho dois experimentos que se mostraram viáveis segundo a análise da pesquisadora responsável pelos dados que utilizamos.

Ao utilizar o motor neural de geração de dados preditivos, advindo do projeto InteliColheita, em um novo domínio de problema, reforçamos a capacidade preditiva das redes MLP e sua adequação aos mais variados tipos de problema, mesmo quando existem poucos dados disponíveis para treinamento.

Conseguimos também abrir caminho para que mais trabalhos sejam desenvolvidos em parceria com o Departamento de Anatomia da UFPE, que pode se tornar um fornecedor de dados e problemas para serem utilizados em novos projetos de IA.

## 5.2 Discussão

Acredito que com a utilização desta ferramenta o pesquisador usuário passa a ter a possibilidade de não apenas comprovar suas conjecturas de forma mais econômica, mas também passa a formular-testar mais rapidamente novas hipóteses em sua pesquisa por meio da manipulação dos dados de entrada. Por exemplo, determinar que fator ou combinação de fatores têm maior influência na redução da quantidade de neurônios no plexo entérico. A ferramenta oferece esta liberdade que é difícil de ser obtida com os métodos utilizados atualmente.

As simulações propiciadas pela ferramenta também podem livrar parcialmente o pesquisador de precisar esperar que os animais cumpram todo um ciclo de suas vidas até que os resultados do experimento possam ser estudados. Além disso, a tarefa mais árdua na pesquisa em Biologia, que é realizar o experimento com sacrifício de animais também pode ser atenuada através do uso da ferramenta. Com isso, podemos afirmar que a otimização do tempo de trabalho e o direcionamento do esforço do pesquisador para atividades mais intelectuais do que manuais são um grande benefício decorrente da ferramenta produzida neste trabalho.

Com a redução quantitativa de cobaias, que pode ser obtida pelo uso da ferramenta, a diminuição do custo da pesquisa também é um resultado provável. Principalmente porque menos material será gasto na manutenção dos criadouros e na alimentação dos animais. Os custos com sacrifícios também serão menores porque estas atividades serão realizadas com menor frequência.

Obviamente para se obter os resultados e impactos desejados através da utilização da ferramenta, deve ser dada uma importância especial à fase de treinamento das redes neurais artificiais utilizadas nas simulações definidas para a pesquisa. Este cuidado deve ser dado para se obter a precisão necessária para validar os resultados das simulações, permitindo assim utilizá-las em substituição parcial às práticas de laboratório. Reiteramos o fato de que as simulações, por mais precisas que sejam, não são suficientes e portanto não vão eliminar a necessidade de experimentos animais.

Outra possibilidade real é o programa proposto aqui permitir a criação e o treinamento de novas redes, o que o torna flexível no domínio do problema. Esta característica possibilita o seu uso em outras pesquisas, uma característica muito positiva para difundir o uso de IA no ambiente científico em outras áreas.

Estamos confiantes de que também contribuimos para a nobre causa de tentar preservar vidas animais em pesquisas científicas, encaramos isto como sendo um benefício tão importante quanto os demais, por seu caráter ético.

## 5.3 Trabalhos Futuros

Como a ferramenta desenvolvida é voltada para o pesquisador-cientista, recursos adicionais que facilitem o trabalho do usuário em outros aspectos de sua pesquisa seriam bem aceitos. O desenvolvimento de recursos para auxiliar o pesquisador no uso do programa no que se refere à RNA, por exemplo sugerindo configurações de rede baseadas nos dados de entrada e saída, e ainda realizando automaticamente treinamentos com diferentes parametrizações e sugerindo topologias de redes que sejam adequadas para um determinado problema.

A ferramenta poderia ser estendida com recursos para identificar a entrada ou combinação de entradas mais significativas em relação ao resultado de uma simulação. Sendo assim o usuário teria um caminho de investigação sugerido pelo programa, um diferencial ainda maior para a ferramenta e um atrativo muito interessante para os pesquisadores.

A ferramenta ainda não contempla o pré-processamento dos arquivos de dados, este fato força o usuário a utilizar outro meio para pré-processamento dos arquivos de dados, inclusive neste trabalho foi utilizado o pré-processador desenvolvido no projeto InteliColheita. Portanto implementar uma interface simples de pré-processamento dos arquivos de dados também possibilitaria uma melhor experiência de uso do programa para o usuário leigo em RNA.

## Bibliografia

- [1] CARDOSO, C. V. P., *Princípios Éticos na Experimentação Animal*, Colégio Brasileiro de Experimentação Animal/Cobea, Disponível em: <<http://www.cobea.org.br/etica.htm#3>>, Acesso em: 07 mar. 2006.
- [2] BAPTISTA NETTO - *Manual de dissecação*, O Livro Médico Ltd., Rio de Janeiro, 1983.
- [3] BALLONE, G. J., *Portal Farmácia*, Fluoxetina, Disponível em: <<http://www.portalfarmacia.com.br/farmacia/principal/conteudo.asp?id=63>>, Acesso em: 07 mar. 2007.
- [4] ARAUJO, E. J. DE ALMEIDA; SANTANA, D. DE M. G.; MOLINARI, S. L. M. H. DE MIRANDA NETO, *Efeito de carência de proteínas e vitaminas do complexo B sobre os aspectos morfoquantitativos do plexo mioentérico do colo descendente de ratos adultos*, Arq. Neuro-Psiquiatr., jun. 2003, vol. 61, nº 2 A, pp. 226-233.
- [5] MARINHO, S. M. O. C.; MENEZES DA SILVA, A. E. T.; AFONSO, L. B.; NEVES, P. R. A.; VIANNA, M. T.; MORAES, S. R. A.; MANHAES DE CASTRO, R., *Desnutrição Neonatal Reduz Neurônios do Plexo Mioentérico do Cólon Descendente em Ratos Adultos*, Neurobiologia 68, 2004, pp. 138 – 143.
- [6] RUSSEL, S. J., *Artificial Intelligence: A Modern Approach*, 2ª ed., Prentice Hall, 2002, pp. 3 – 30.
- [7] BRAGA, A., *Redes Neurais Artificiais: Teoria e Aplicações*, 1ª ed., LTC, 2002.
- [8] BENDER, EDWARD A., *Mathematical Methods in Artificial Intelligence*, IEEE Computer Society Press, Los Alamitos, CA, pp. 26, 1996.
- [9] BARRETO, J. M., *Redes Neurais – Fundamentos e Aplicações*, 2º Simpósio Brasileiro de Automação Inteligente, CEFET-PR, 1995.
- [10] HAUNGELAND, J., Ed., *Artificial Intelligence: The Very Idea*, MIT Press, Cambridge, 1985.
- [11] KURZWEIL, R., *The Age of Intelligent Machines*, MIT Press, Cambridge, 1990.
- [12] RICH, E. AND KNIGHT, K., *Artificial Intelligence*, McGraw-Hill, New York, 1996.
- [13] KASABOV, NIKOLA K., *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, pp. 1, 1998.
- [14] NEWELL, A. AND SIMON, H.A., *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [15] KONAR A., *Uncertainty Management in Expert Systems using Fuzzy Petri Nets*, Ph.D. thesis, Jadavpur University, Calcutta, 1994.
- [16] KONAR, A., *Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain*, CRC Press, 2000, pp. 3.
- [17] FERBER, J., *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley Professional, 1999.
- [18] WILLIAMSON, J., *Bayesian Nets and Causality: Philosophical and Computational Foundations*, Oxford University Press, 2005.

- [19] HOLSHEIMER, M. AND SIEBES, A., *Data Mining: the search for knowledge in databases*, 1994.
- [20] MURRAY, T., *Authoring Intelligent Tutoring Systems: In the analysis of the state of the art*, International Journal of Artificial Intelligence in Education, pp. 98-129, 1999.
- [21] NAVEGA, S. C., *Inteligência Artificial: Presente, Passado e Futuro*, Publicado nos Anais do INFOIMAGEM 2001, Cenadem, 2001.
- [22] SHANNON, C. E. AND MCCARTY, J., *Automata Studies*, Princeton University Press, New Jersey, 1956.
- [23] POERSCH, J. M., *Simulações Conexionistas: A Inteligência Artificial Moderna*. Disponível em: <http://www3.unisul.br/paginas/ensino/pos/linguagem/0402/09.htm>. Acesso em novembro de 2006.
- [24] MCCULLOCH, W. S., *Embodiments of Mind*, MIT press, Cambridge, Massachusetts, 1965.
- [25] DAYHOFF, J. E., *Neural Networks Architectures*, New York: Van Nostrand Reinhold, 1992.
- [26] ALMEIDA, M. A. F., *Introdução ao Estudo de Redes Neurais Artificiais*, UFSC, Florianópolis, 1999.
- [27] BARRETO, J. M., *Inteligência Artificial no Limiar do século XXI*, 2ª Ed., Duplic Edições, Florianópolis, 1999.
- [28] AZEVEDO, F. M., *Notas de Aulas da disciplina "Redes Neurais por Software"*, Curso de Mestrado em Engenharia Biomédica - GPEB-DEE, UFSC, 1995.
- [29] HASSOUN, M. H., *Fundamentals of Artificial Neural Networks*, MIT Press, 1995.
- [30] ROSENBLATT, F. *Principles of neurodynamics*, Spartan Books, Washington-DC, USA, 1961.
- [31] RUMELHART, D. E.; HINTON, G. E. e WILLIAMS, R. J. *Learning internal representations by error propagation*. In: D.E. RUMELHART and J.L. MC-CLELLAND (Eds.): *Parallel Distributed Processing Vol 1*. MIT Press, Cambridge, 1986, pp. 318-362.
- [32] HAYKIN, S., *Redes Neurais: Princípios e prática 2.ed.*, Bookman, Porto Alegre, 2001.
- [33] BUARQUE DE LIMA NETO, F., *InteliColheita – Tecnologia neural para apóio a decisão em colheita de cana-de-açúcar*, Intelicolheita, Projeto de Pesquisa financiado pelo CNPq-Brasil (PDPG-TI 506595/2004-8).
- [34] MONTEIRO, F.M.F, LAHLOU, S., ALBUQUERQUE, J.A., CABRAL, A.M.S. *Influence of a multideficient diet from northeastern Brazil on resting blood pressure and baroreflex sensitivity in conscious, freely moving rats*. Brazilian Journal of Medical and Biological Research, 2001, pp. 271-280.
- [35] TEODÓSIO, N.R., LAGO, E.S., ROMANI, S.A.M., GUEDES, R.C.A. *A regional basic diet from Northeast Brazil as a dietary model of experimental malnutrition*. Archivos Latinoam. de Nutrición, 1990, pp. 533-547.
- [36] *Stuttgart Neural Network Simulator - SNNS*, Disponível em: <http://www-ra.informatik.uni-tuebingen.de/SNNS/>. Acesso em novembro de 2006.

## Anexo A

# Ferramentas e Tecnologias Utilizadas

Apresentaremos aqui as duas principais ferramentas utilizadas no desenvolvimento deste trabalho. Faremos um resumo do propósito de cada ferramenta e indicaremos o *website* onde é possível encontrar mais informação e também fazer o *download* de qualquer uma delas.

### - NetBeans IDE

É um ambiente de desenvolvimento gratuito para Java, possui módulos que podem ser agregados à ferramenta básica através de *download*. Possui excelentes recursos para o desenvolvimento do ambiente gráfico de programas em Java, além disso tem *code-completion* com *help* integrados, que facilitam o trabalho do programador exibindo métodos, classes e API compatíveis ao que estiver sendo digitado.

Até a conclusão desta monografia o NetBeans estava em sua versão 5.5 e encontrava-se disponível no site <http://www.netbeans.org/>.

### - JUDE

Consiste em uma ferramenta para modelagem UML. Encontra-se disponível em três versões: *Professional*, *Community* e *Server*. A versão utilizada neste trabalho foi a *Community*, que é gratuita e permite fazer vários tipos de diagramas, como: Diagrama de Classe, Diagrama de Caso de Uso, Diagrama de Atividade, Diagrama de Sequência, Diagrama de Colaboração e Diagrama de Componentes.

Até a conclusão desta monografia o JUDE *Community* estava em sua versão 3.1.1 e encontrava-se disponível no site <http://jude.change-vision.com/jude-web/index.html>.

# Apêndice A

## Questionário de Validação do Programa InteliLab

### Seção 1 – Identificação e Perfil

Os dados a seguir não são obrigatórios.

**Nome:** Silvia Regina Arruda de Moraes    **Data:** 20/11/2006

**Nível:**

- Aluno de Iniciação Científica
- Mestrando
- Doutorando
- Orientador
- Outros: Doutora

### Seção 2 – Perguntas sobre o Programa InteliLab

#### **Corretude**

*Avaliação sobre a qualidade da informação fornecida pelo InteliLab.*

- Péssimo
- Ruim
- Regular
- Bom
- Ótimo

#### **Usabilidade**

*Avaliação sobre a facilidade de uso do programa.*

- Péssimo
- Ruim
- Regular
- Bom
- Ótimo

Observações: Apesar de ser uma ferramenta nova, na rotina de trabalho, após algumas explicações fica fácil utilizá-la.

### **Utilidade**

*Avaliação sobre a relevância e importância das informações possíveis de se conseguir com o InteliLab em pesquisas científicas.*

- Péssimo
- Ruim
- Regular
- Bom
- Ótimo

Observações: A ferramenta abre um campo vastíssimo para o desenvolvimento de outros recursos que poderão auxiliar e otimizar em muito as atividades de pesquisa na área de ciências biológicas.

### **Desempenho**

*Avaliação sobre o tempo de resposta do InteliLab.*

- Péssimo
- Ruim
- Regular
- Bom
- Ótimo

Observações: O tempo de resposta se mostrou rápido para processar os dados.

## Apêndice B

# Declaração de Aplicabilidade do Programa InteliLab

Declaro para fins de comprovação da aplicabilidade do InteliLab, que os recursos oferecidos por esta ferramenta possibilitam a exploração de outros aspectos do problema estudado em laboratório. Sua capacidade de predição de valores, quando bem calibrada, poderá constituir, também, numa utilização mais racional de animais de laboratório em diferentes pesquisas na área de biologia e da saúde.

Considero também que a variedade de modelagens possíveis para um mesmo problema, através da manipulação das características dos animais, possibilita criar novas hipóteses sobre os problemas estudados.

Recife, 20 de novembro de 2006.



---

Silvia Regina Arruda de Moraes