

# ANNeF – ARTIFICIAL NEURAL NETWORKS FRAMEWORK: UMA SOLUÇÃO SOFTWARE LIVRE PARA O DESENVOLVIMENTO, ENSINO E PESQUISA DE APLICAÇÕES DE INTELIGÊNCIA ARTIFICIAL MULTIPLATAFORMA

João Ricardo Bittencourt (jrbitt@netu.unisinos.br)

Fernando Osório (osorio@exatas.unisinos.br)

UNISINOS – Universidade do Vale do Rio dos Sinos – Centro de Ciências Exatas e Tecnológicas  
Curso de Informática – Mestrado em Computação Aplicada (PIP/CA).  
Caixa Postal 275, 93022-000. São Leopoldo, RS, Brasil.

## RESUMO

Este trabalho descreve a implementação de um conjunto de classes (framework) destinado a facilitar o desenvolvimento de aplicações multiplataforma que utilizam algoritmos de Redes Neurais Artificiais. Além de destacar a importância do desenvolvimento e da distribuição de softwares livres junto a comunidade científica, onde esperamos contribuir com a criação de uma nova ferramenta aberta de grande utilidade no ensino, pesquisa e aplicação de técnicas de Inteligência Artificial.

## ABSTRACT

*We describe the implementation of a framework that will ease the development of multiplatform applications that use Artificial Neural Networks algorithms. In addition to showing the importance of free software development and distribution to the scientific community, our main contribution is the implementation of an open tool useful for teaching, researching and development of applications in the field of Artificial Intelligence.*

## 1 INTRODUÇÃO

Este trabalho começou com o desenvolvimento do software Neuron Color [4], ferramenta implementada – em um projeto científico da Universidade do Vale do Rio dos Sinos (UNISINOS) com o objetivo de criar filtros inteligentes para o processamento de imagens usando Redes Neurais Artificiais. Durante a execução deste projeto surgiu a necessidade de criar módulos separados para o processamento inteligente das imagens (algoritmos relacionados as Redes Neurais Artificiais) das demais rotinas de software. Isto ocorreu devido as grandes dificuldades em efetuar modificações nos algoritmos existentes, pois não havia uma estruturação formal do software. Com isto buscamos estruturar melhor nossas ferramentas, para que fosse possível criar um novo software mais flexível e fácil de ser expandido.

Iniciamos assim o desenvolvimento do ANNeF (Artificial Neural Networks Framework), que consiste em um conjunto de classes capaz de encapsular as funcionalidades dos algoritmos neurais, permitindo efetuar múltiplas atualizações e modificações, além de permitir a criação de inúmeras aplicações baseadas neste framework. A concepção deste baseou-se fundamentalmente em dois princípios: na criação de um software livre multiplataforma e a busca do desenvolvimento científico na área de Inteligência Artificial.

Gaston Bachelard apud Simões [15] caracteriza o pensamento científico em seis atributos, dos quais destacamos o seguinte – “a verdade científica é estabelecida pelo trabalho **cooperativo** e pela intersubjetividade científica”. Desta forma a distribuição de softwares científicos livremente é uma forma de valorizarmos as descobertas, no campo da ciência da computação, porque estamos compartilhando saberes e permitindo que outros sujeitos interajam com estas produções. O trabalho **cooperativo** permite assim a constante melhoria e difusão dos conhecimentos.

Stallman [10] expôs em seu ensaio “Why Software Should Be Free”, que em muitas universidades americanas, em muitos projetos científicos computacionais somente a discussão teórica é compartilhada com a comunidade, sendo o código-fonte jamais exibido, permanecendo oculto. Nossa postura é a de contribuir para o desenvolvimento de novas ferramentas na área de Inteligência Artificial, evitando este tipo de abordagem que restringe a difusão e a validação integral dos conhecimentos. Também o pensamento de Stallman sobre software livre refere-se “à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software.” Devido estes pensamentos que o ANNeF foi projetado como um framework científico

de livre distribuição para pesquisadores e desenvolvedores.

O objetivo deste artigo é elucidar sobre o desenvolvimento de uma aplicação científica livre, aplicada as técnicas de Inteligência Artificial, baseada na livre distribuição de software e do conhecimento científico. Para isto este artigo está organizado da seguinte forma: seção 2 é uma breve conceituação sobre Redes Neurais Artificiais; a seção 3 trata da metodologia usada para projetar o ANNeF, destacando as vantagens de usar frameworks e uma documentação estruturada em projetos de softwares livres; finalmente a seção 4 destaca as características, vantagens, aplicabilidade e estruturação do ANNeF.

## 2 REDES NEURAS ARTIFICIAIS

Através de um modelo abstrato e simplificado dos neurônios humanos podemos desenvolver um simulador que seja capaz de classificar, generalizar e aprender funções desconhecidas. Um dos modelos de aprendizado neural mais utilizados na atualidade é o modelo denominado Backpropagation [1,2].

Para que ocorra o aprendizado é necessária uma base de exemplos, conjunto de dados que descrevam os casos que serão apresentados para a Rede Neural Artificial. O aprendizado do modelo Backpropagation é conhecido como um aprendizado do tipo supervisionado, onde a rede aprende a responder de modo similar as respostas apresentadas junto com os casos, sendo estes um conjunto de dados de entrada mais as saídas desejadas.

Usando um processo iterativo (cada uma destas iterações é chamada de época) apresentamos a rede neural os exemplos contidos na base para que ocorra a adaptação dos pesos (simula o reforço e a inibição das conexões sinápticas existentes entre os neurônios reais). Desta adaptação de pesos que surge o aprendizado, e por consequência surge também a adaptação do comportamento da rede, que deverá aprender a responder aos estímulos de entrada de acordo com os exemplos que lhe foram apresentados.

Atualmente encontramos vários softwares de simulação de redes neurais, mas muitos deles são de uso comercial (**Matlab**, **NeuroDimension/NeuroSolution**, entre outros), sem acesso ao código fonte, ou mesmo restritos a certas plataformas específicas como é o caso do **SNNS** e do **Nevprop**. A característica fundamental, o principal diferencial de nossa proposta é a criação de um software livre em uma linguagem portátil (Java), permitindo assim uma maior disseminação do uso de nossas ferramentas.

## 3 METODOLOGIA

Nas subseções seguintes serão detalhadas algumas decisões tomadas em relação ao projeto de desenvolvimento do ANNeF.

### 3.1 Abordagem Baseada em Framework

Segundo Johnson & Foote (apud Fayad, Shimidt e Jonhson) [17] framework é uma aplicação reusável e semi-completa que pode ser especializada para produzir aplicações personalizadas. Ao contrário das técnicas usuais de orientação a objetos que organiza as classes em bibliotecas, frameworks são voltados para unidades particulares de negócios e **domínios de aplicações**. O maior benefício desta abordagem está no momento de criar sub-classes, porque além de herdar as características inerentes as classes-pai, herda-se o comportamento e o relacionamento destas.

Os principais benefícios em desenvolver um framework são a modularidade, reusabilidade, extensibilidade e inversão de controle. A modularidade é garantida através da definição de interfaces e o encapsulamento da implementação de métodos, permitindo um melhoramento na qualidade. Quanto a reusabilidade é possível criar novas aplicações através de componentes genéricos, garantindo maior produtividade de programação, garantia de qualidade, performance e confiabilidade do software.

Quanto a extensibilidade é a capacidade de estender funcionalidades a partir de comportamentos, denominados *hot spots*, criando desta forma novas características. Finalmente a inversão de controle é a capacidade do framework a responder eventos externos, mantendo o controle da execução da aplicação.

A escolha deste tipo de técnica de engenharia de software é bastante aplicável quando relacionamos com a elaboração de softwares científicos livres, porque os desenvolvedores da aplicação conhecem o domínio da aplicação, no caso Redes Neurais Artificiais. Além disso, quando se desenvolve um software de forma cooperativa devemos permitir a incorporação de novos requisitos, a reusabilidade de código e a programação baseada em métodos de interfaces bem definidos, que garantam uma integração modular.

Para elaborar a modelagem foi usado o **ArgoUML** [13], uma ferramenta software livre que auxilia na criação de diagramas formalizados na Unified Modeling Language (UML) [8].

Um dos problemas dos frameworks é a dificuldade que os novos usuários tem de aprendê-lo. Para amenizar este processo de aprendizagem é fundamental uma documentação estruturada [5,6] que descreva os

pontos que podem ser modificados e a forma de utilizar a biblioteca padrão. Para isto foi criado o Guia do Programador [9], um documento baseado na GNU Free Documentation License (GFDL) [14], que reúne todos os procedimentos de uso e personalizações.

### 3.2 Linguagem de Programação – Java

A escolha da linguagem Java[12] foi feita de acordo com uma tendência do uso desta linguagem pela vantagem de ser multiplataforma, permitindo deste modo a construção de aplicações inteligentes portáteis. Desta forma o ANNeF, pode ser executado sob qualquer plataforma, seja um sistema operacional ou um outro aparelho eletrônico, por exemplo, um aparelho celular. Além disto, destaca-se pela integração com a Internet, permitindo a construção de applets e servlets.

## 4 ANNeF – ARTIFICIAL NEURAL NETWORKS FRAMEWORK

Basicamente o ANNeF constitui-se de um pacote de classes (package) incluído em um pacote de maior nível hierárquico, denominado **ai** (Artificial Intelligence), cujo objetivo consiste simplesmente em reunir uma série de soluções de software livre aplicadas a Inteligência Artificial.

O ANNeF tem como domínio qualquer aplicação que necessite a implementação de algum tipo de Rede Neural Artificial (RNA). As implementações dos inúmeros modelos de RNAs serão desenvolvidas de forma gradual, sendo o algoritmo de Backpropagation [1] a primeira implementação.

As principais funcionalidades do framework são: fornecer a implementação de modelos clássicos de RNAs, tais como Backpropagation e Mapas de Kohonen [2]; permitir a adaptação destes modelos criando variações, tais como o Quickprop (variação da implementação clássica do Backpropagation); inclusão de novos modelos de RNAs sem alterar o funcionamento dos demais; definir novas funções de ativação dos neurônios; definição de importadores de arquivos sinápticos; e oferecer um visualizador gráfico de topologias de RNAs.

Além deste benefício podemos citar os seguintes: totalmente independente de plataforma, pode ser executado sob qualquer sistema operacional ou dispositivo eletrônico que possua uma Java Virtual Machine (JVM); utiliza o paradigma de programação orientada a objetos, facilitando a compreensão do problema através de abstrações e inter-relacionamentos de classes; reutilização de código, reduzindo o tempo de desenvolvimento; e a facilidade de construir novas aplicações de RNAs, devido a uma API formalizada.

Quanto as possibilidades de aplicação deste framework são inúmeras [7], diversas áreas atualmente estão utilizando Redes Neurais Artificiais para tratar seus problemas. Dentro da diversidade de áreas possíveis de uso podemos enumerar as seguintes: reconhecimento de caracteres e/ou palavras; classificação de padrões; identificação de padrões em imagens; processamento e reconhecimento de sons; processamento de imagens [4]; aproximação de funções matemáticas; robótica autônoma; sistemas especialistas; agentes autônomos; interfaces de interações em ambientes virtuais [3]; entretenimento; e segurança, por exemplo, reconhecimento de impressões digitais.

### 4.1 A Estruturação do Framework

O ANNeF consiste de um núcleo, com as principais abstrações e características implementadas e uma biblioteca com modelos neurais.

As classes pertencentes ao núcleo do framework permitem que o desenvolvedor execute as seguintes personalizações:

- i) criação de novas funções de ativação;
- ii) criação de importadores de pesos sinápticos;
- iii) criação de novos modelos neurais;
- iv) geração de bases de exemplos.

Quanto a biblioteca, esta primeira versão do ANNeF oferece para o desenvolvedor as seguintes facilidades:

- i) implementação das funções sigmóide e linear;
- ii) importação de arquivos de pesos sinápticos, gerados pelo Nevprop [17];
- iii) implementação do algoritmo de aprendizado supervisionado Backpropagation;
- iv) implementação do BackpropSimulator, uma ferramenta completa para executar a simulação de aprendizado do modelo neural citado acima.

## 5 CONSIDERAÇÕES FINAIS

Conclui-se o presente trabalho constatando que a construção de softwares livres é de suma importância para o progresso da ciência em geral, obviamente afetando o contexto social existente. Desenvolvendo soluções computacionais e distribuindo estes trabalhos livremente para a comunidade, seja esta acadêmica ou não, estaremos possibilitando a melhoria desta solução através de um trabalho cooperativo e estaremos permitindo que novos softwares sejam gerados.

No que se refere ao ANNeF, este software ainda está em sua versão inicial onde se espera que ele ainda possa sofrer muitas melhorias, que possivelmente serão mais facilmente atingidas com o trabalho cooperativo. A versão atual da biblioteca do framework contempla somente o algoritmo de Backpropagation, onde possivelmente a implementação de novos modelos neurais será realizada.

O ANNeF já possui uma versão preliminar disponível que pode ser obtida na Internet no seguinte endereço: <http://www.inf.unisinos.br/~jrbitt/annef/>.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] RUMELHART, D. ; HINTON, G & WILLIAMS, R. **Learning Internal Representations by Error Propagation**. In: **Parallel Distributed Processing: Explorations in the Microstructure of Cognition** – Vol. 1. Cambridge: MIT Press, 1986.
- [2] FREEMAN, James A. & SKAPURA, David M. **Neural Networks: Algorithms, Applications, and Programming Techniques**. Addison-Wesley Publishing, Reading, 1992. 401p.
- [3] MUSSE, Soraia R. ; OSÓRIO, Fernando S.; GARAT, Fabien et al. Interaction with Virtual Human Crowds using Artificial Neural Networks to Recognize Hands Posture. In: **Proceedings of III Workshop on Virtual Reality**. Gramado: SBC, 2000, p.107-118.
- [4] BITTENCOURT, João Ricardo; OSÓRIO, Fernando S. Adaptive Filters for Image Processing based on Artificial Neural Network. In: **Proceedings of XIII Brazilian Symposium on Computer Graphics and Image Processing**. Gramado: SBC, 2000, p.336.
- [5] FROEHLICH, Garry. HOOVER, James H et al. Designing Object-Oriented Frameworks. In: ZAMIR, Saba (org). **Handbook of Object Technology**. Nova Iorque: CRC Press, 1999.
- [6] FROEHLICH, Garry. HOOVER, James H et al. Using Object-Oriented Frameworks . In: ZAMIR, Saba (org). **Handbook of Object Technology**. Nova Iorque: CRC Press, 1999.
- [7] OSÓRIO, Fernando S. **Redes Neurais Artificiais: Do Aprendizado ao Aprendizado Artificial**. I Fórum de Inteligência Artificial, Canoas, agosto. 1999. Internet: <http://www.inf.unisinos.br/~osorio/IForumIA/fia99.pdf>
- [8] UMG. **Unified Modeling Language Specification**. March, 1999. Internet: <http://www.omg.org>
- [9] **ANNeF Guia do Programador** – <http://www.inf.unisinos.br/~jrbitt/annef/index.html>
- [10] STALLMAN, Richard. **Why Software Should Be Free**. 1992. Internet: <http://www.gnu.org/philosophy/shouldbefree.html>
- [11] **What is Free Software?** – <http://www.gnu.org/philosophy/free-sw.html>
- [12] Sun Microsystems – <http://java.sun.com>
- [13] **ArgoUML 0.8.1** – <http://www.argouml.org>
- [14] **GNU Free Documentation License** – <http://www.gnu.org/copyleft/fdl.html>
- [15] SIMÕES, Reinério Luiz M. **A Imaginação Material segundo Gaston Bachelard**. Rio de Janeiro, 1999. 95p. Dissertação (Mestrado), Instituto de Filosofia e Ciências Humanas, Universidade do Estado do Rio de Janeiro, 1999. Internet: <http://www.geocities.com/Athens/Acropolis/9568/filosofia.html>
- [16] GOODMAN, Phil; ROSEN, David; PLUMMER, Allen. **NevProp (Users Manual)**. University of Nevada Center for Biomedical Modeling Research.
- [17] FAYAD, M.E; SCHMIDT, D.C.; JOHNSON, R.E. **Building Application Frameworks**. New York: Willey, 1999, 664p.