

FrameRN: Ambiente 3D para Simulação Redes Neurais Artificiais

Orlando Ohashi*, Roberto Limão*, Marcos Venícios†, Denis Arruda†, Renato Simões† and Andréa Ribeiro*

*Universidade Federal do Pará - PPGEE

Rua Augusto Côrrea, 01

66075-110, Belém, Pará, Brasil

Email: ohashijr@yahoo.com.br, limao@ufpa.br, andrea_gribeiro@yahoo.com.br

†Centro Universitário do Pará (CESUPA)

Av. Governador José Malcher, 1963

66060-230, Belém, Pará, Brasil

Email: denis_arruda@yahoo.com, renatosm@gmail.com, maraujo@cesupa.br

Abstract— This paper presents a neural network simulator that uses virtual reality to made ease the comprehension and the configuration of a neural network, which is kwn as a complex domain. The simulator has characteristic of 3D software and can be used as potential learning tool.

I. INTRODUÇÃO

O simulador de redes neurais em 3D proposto neste artigo é denominado de FrameRN e foi idealizado com o intuito de facilitar o desenvolvimento e a visualização das informações pertinentes as configurações e ao treinamento das redes neurais. Este tema foi escolhido por se tratar de um domínio de aplicação bem fundamentado e que possui um grau de abstração elevado, sendo, dessa forma, de difícil aprendizado. A técnica de realidade virtual pode, através de suas características, propor uma possível solução para este problema e diversos outros.

Como alternativa a uma interface mais amigável, o produto de software aqui proposto foi desenvolvido com o intuito de ser bastante intuitivo para o usuário, onde o mesmo possa, através do manuseio do mouse, configurar a arquitetura de rede desejada, porém, como se trata de um domínio de aplicação muito especializado, é necessária uma noção mínima sobre o mesmo.

Nas sessões subsequentes é explicado como foi desenvolvido o FrameRN, quais foram as decisões tomadas, as dificuldades encontradas e os resultados obtidos. Na sessão II há uma explanação sobre as redes neurais. Na sessão III há uma explicação da arquitetura desenvolvida além de explicar sobre realidade virtual e a linguagem de programação utilizada, na sessão IV é feita uma comparação com outro simulador de redes neurais existente, na sessão V é explicitado o FrameRN com figuras demonstrando os recursos visuais que o mesmo possui e a sessão VI contém as considerações finais e os trabalhos futuros.

II. FUNDAMENTAÇÃO SOBRE REDES NEURAIS

A área de redes neurais se encontra em um bom grau de amadurecimento, pois já vem sendo estudada desde a década

de quarenta [8], possuindo diversas implementações executadas e soluções para problemas nas mais diversas áreas, como: reconhecimento de padrões, classificação dados, controle e automação, previsão e tomada de decisão.

Nas subseções II-A e II-B são detalhadas algumas características das redes abrangidas no FrameRN, as redes neurais perceptrons de múltiplas camadas e a rede neural mapas auto-organizáveis de Kohonen. Estas duas arquiteturas de redes foram desenvolvidas com o intuito de abranger as mais variadas hipóteses de configurações.

A. Perceptron de Múltiplas Camadas

A rede neural perceptron de múltiplas camadas com o algoritmo de aprendizagem de retropropagação de erro (*error back-propagation*), também conhecida como rede *backpropagation*, é das redes neurais a mais popular. Possui implementações em diversas linhas de pesquisa, como controle [2], previsão [6], diagnóstico [1] e classificação [10].

No FrameRN, o modelo da rede perceptrons de múltiplas camadas possui algumas características específicas, como:

- A função de ativação pode ser implementada por neurônio, em vez de ser adotada uma função de ativação global pra rede neural. O usuário tem a possibilidade de escolher a melhor configuração da rede neural no quesito função de ativação, montando neurônios com funções de ativações diferentes se desejado;
- A constante de aprendizagem, constante de momento, intervalo de inicialização dos pesos da sinapse, constante de aprendizagem adaptativa e constante de momento adaptativa são os parâmetros que o usuário pode configurar para cada sinapse da rede neural, em vez de serem configurados para a rede neural como um todo. Oferecendo para o usuário do FrameRN uma gama de configurações da sua rede neural;
- O FrameRN ainda possui algumas configurações no que se refere ao seu treinamento. O usuário pode escolher entre o treinamento tradicional ou a validação cruzada, escolhendo os respectivos parâmetros necessários para cada um dos treinamentos. É possível normalizar os dados

do conjunto de treinamento, tornar a apresentação dos dados do conjunto de treinamento aleatória, entre outras configurações.

B. Mapas Auto-Organizáveis de Kohonen

A rede neural mapas auto-organizáveis de Kohonen, também conhecida como rede de Kohonen, é bastante popular, porém difere da rede perceptrons de múltiplas camadas em algumas características. A principal delas é que nesta rede o tipo de treinamento é não supervisionado, ou seja, os dados do conjunto de treinamento são compostos apenas de entradas, sem saída desejada, o que vai atribuir à rede de Kohonen características distintas, voltando o seu enfoque para extração de características [3] e para classificação de dados [4].

A rede de Kohonen modelada no FrameRN possui algumas características para possibilitar ao usuário uma variedade de parâmetros de configuração desta rede neural:

- Existem quatro funções para a seleção do neurônio vencedor: a norma euclidiana, o produto interno, o quadrado da magnitude do vetor diferença e o valor absoluto;
- Pode ser configurada a seleção da vizinhança de duas maneiras, pela vizinhança retangular ou pela vizinhança gaussiana. Dependendo do tipo de seleção escolhido, parâmetros diferentes devem ser configurados;
- O usuário ainda pode escolher o intervalo de inicialização dos pesos de maneira diferente para cada uma das sinapses;
- Por se tratar de um ambiente de desenvolvimento de três dimensões, é possível que o usuário configure a forma geométrica desejada na camada de saída. Fator este que é bastante interessante, pois, na rede de Kohonen, a distância entre os elementos da camada de saída influencia no treinamento.

III. A PROPOSTA DO FRAMEWORK PARA REDES NEURAIAS

De maneira geral, o *framework* proposto está baseado em técnicas de modelagem de software. Especificamente o uso de padrões de projeto e padrões de arquitetura de software, de forma a dispor de características bastante desejáveis, como modularidade e encapsulamento. Facilitando assim a sua manutenção e extensão.

A. Arquitetura do Software

A arquitetura do *framework* está representada através da Figura 1, onde são apresentados os componentes de software, bem como a interface entre os componentes. Permitindo, deste modo, a independência das funcionalidades internas dos pacotes, a produção dos módulos de maneira isolada e a possibilidade de incorporar novos componentes quando necessário [9].

A Figura 2 apresenta o diagrama de camadas verticais, onde se notam os pacotes com suas respectivas classes. Facilitando deste modo que o projetista visualize as classes de maneira isolada por componente e planeje a implementação dos componentes de forma independente [14].

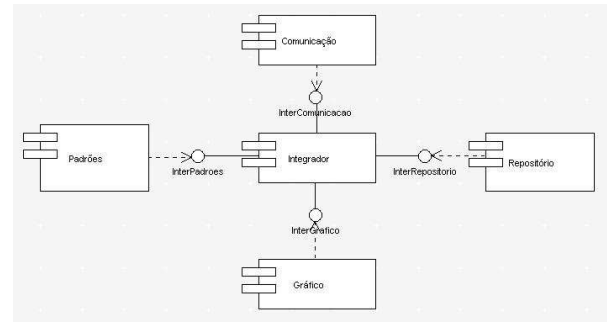


Fig. 1. Diagrama de Componentes

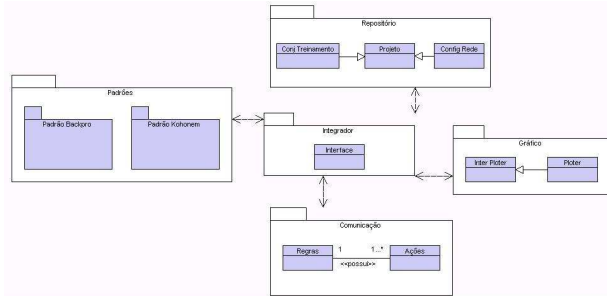


Fig. 2. Diagrama de Camadas Verticais

- Componente I - Padrões: Este componente apresenta as características dos padrões de projetos das redes neurais propostas para o desenvolvimento do framework, que, neste primeiro momento, está sendo trabalhado com os modelos neurais: perceptrons de múltiplas camadas com algoritmo de aprendizagem Back-propagation e mapas auto-organizáveis de Kohonen;
- Componente II - Integrador: Este componente é responsável pelo ambiente ao qual o usuário irá interagir, ou seja, configurar suas redes neurais e treiná-las;
- Componente III - Comunicação: Este componente será responsável pela comunicação entre o framework e o usuário, sendo este responsável por atualizar o usuário com as informações do framework. Baseia-se em um conjunto de regras e ações explicativas, de tal forma que a interação com o usuário seja feita através de textos bem elaborados;
- Componente IV - Gráfico: Este componente é responsável por gerar os possíveis gráficos produzidos pelo treinamento de uma rede neural, como por exemplo: o gráfico de erro em função do número de iterações;
- Componente V - Repositório: Este componente será responsável por armazenar todas as configurações do *framework*, ou seja, o usuário poderá armazenar as configurações e os treinamentos executados por ele.

B. Recursos de Realidade Virtual Utilizados

Realidade Virtual é uma das mais avançadas tecnologias de interação com o usuário através de um dispositivo computacional. Seja este um computador tradicional (monitor, teclado e

mouse) ou um sistema mais robusto: com capacetes e óculos estereoscópicos e luvas tridimensionais [11].

“Realidade Virtual é uma técnica avançada de interface, onde o usuário pode realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais” [11].

Um sistema é considerado imersivo quando possibilita ao usuário a sensação de pertencer ao mundo virtual (ilusão), ou seja, o usuário pensa que esta dentro deste mundo, porém são necessários dispositivos não convencionais, como: capacetes, óculos estereoscópicos e luvas tridimensionais. Os ambientes virtuais não imersivos são os que o usuário interage com o mundo virtual (porém não tem a ilusão de que pertence a este mundo) com dispositivos convencionais, como: mouse, teclado e monitor. O FrameRN foi desenvolvido sem a possibilidade de imersão, sendo assim mais acessível, pois não é necessária a aquisição de nenhum dispositivo em especial.

Várias técnicas também foram desenvolvidas para tentar reproduzir de maneira sintética a realidade em termos de aparência, por exemplo, efeitos de iluminação e de textura de materiais. Devido às características apresentadas pela tecnologia da realidade virtual, esta foi considerada extremamente compatível para as necessidades de construção do FrameRN.

1) *Tecnologia Utilizada - API Java 3D*: A linguagem de programação escolhida para o desenvolvimento do FrameRN foi Java, mais especificamente a API Java 3D, devido as suas diversas características como multi-thread, orientada a objetos, multi-plataforma, etc [5].

A API (*Application Program Interface*) Java 3D é uma hierarquia de classes Java que serve como uma interface para um sofisticado sistema de renderização de gráficos tridimensionais. Esta tecnologia foi criada com o objetivo de criar uma API que fosse independente de plataforma, como o VRML (*Virtual Reality Modeling Language*). Para isso, a Intel, Silicon Graphics, a Apple e a Sun Microsystems, em colaboração, definiram a especificação Java 3D. Em 1998, a Sun colocou a implementação para download [7]. Atualmente consiste de uma API baseada nas bibliotecas gráficas OpenGL e DirectX, e os programas podem ser escritos como aplicação ou applets.

A API Java 3D possui seu mundo virtual baseado na arquitetura de árvore, onde os objetos e comportamentos em relação aos objetos são adicionados na árvore, a qual é denominada grafo de cena. Sendo possível trabalhar em alto nível na construção (orientado a objetos) e manipulação de objetos em três dimensões [15].

As redes neurais possuem, quanto aos desenhos geométricos, uma baixa complexidade. Foram utilizadas apenas as primitivas básicas disponíveis na API, como: cubos, esferas, cilindros e linhas. Devido ao domínio da aplicação, não foi necessária nenhuma figura geométrica especial. Porém, para o controle dos comportamentos das figuras geométricas, foram necessários esforços maiores.

Foi também adicionada iluminação, aparência, entre outras características para melhorar o efeito visual da interface. Não foi utilizada nenhuma fórmula para a escolha das cores e quantidade de luz, foi mais uma medida artística.

Durante o treinamento da rede neural, é possível visualizar a modificação das cores das sinapses, assim como a construção sincronizada do gráfico do erro médio MSE (no caso da rede neural perceptrons de múltiplas camadas). Para implementar tais características, foi necessário utilizar algumas *thread's* para simular o processamento paralelo da rede neural, da renderização das cores das sinapses e do gráfico.

IV. FERRAMENTA SEMELHANTE

O nome MATLAB se origina de Matrix Laboratory, trata-se de um programa, desenvolvido pela MathWorks, destinado a realizar cálculos de matrizes. Seus comandos são semelhantes às escritas de expressões algébricas [12].

Este software possui um toolbox, conjunto de executáveis que podem ser chamados pela linha de comando do MATLAB, que permite a construção de diversos tipos de redes neurais artificiais.

A interface do MATLAB, além de ser totalmente em inglês, pode gerar dificuldades para usuários sem experiência com redes neurais, como, por exemplo, a definição de função de ativação da rede neural.

Este software pode ser considerado bastante estável, está no mercado a vários anos e é utilizado em vários países.

Um usuário com experiência em redes neurais pode usufruir praticamente todos os recursos necessários para modelagem e treinamento de uma rede neural artificial no MATLAB.

Quanto a portabilidade, o MATLAB possui versões para Windows, Linux, Macintosh e Solaris.

Este software possui um documentação bastante extensa disponível no website do fabricante e na função de ajuda do software.

A proposta do FrameRN é oferecer uma interface mais intuitiva para usuários que ainda estejam aprendendo redes neurais, fornecendo assim mais um recurso facilitador no aprendizado, pois fornece uma melhor noção visual e espacial da arquitetura de uma rede neural.

Os recursos e a documentação para usuários finais do FrameRN ainda são poucos, devido ao fato de ter sido finalizado a pouco tempo.

O FrameRN ainda não pode ser considerado estável, pois algumas melhorias, principalmente na interface gráfica, ainda estão sendo feitas.

Quanto a portabilidade, o FrameRN pode ser utilizado em todas as plataformas que possuam uma máquina virtual Java e API Java 3D disponível, como Window e Linux.

V. SIMULAÇÃO

Foram realizados dois exemplos, o primeiro para demonstrar a rede perceptrons de múltiplas camadas sendo treinada para solucionar uma função matemática na seção V-A e um exemplo da rede mapas auto-organizáveis de kohonen para treinar um exemplo da classificação de animais na seção V-B.

A. *Treinamento FrameRN*

Na Figura 3, pode ser visualizada a arquitetura de uma rede neural configurada para a solução da função matemática representada na Equação 1.

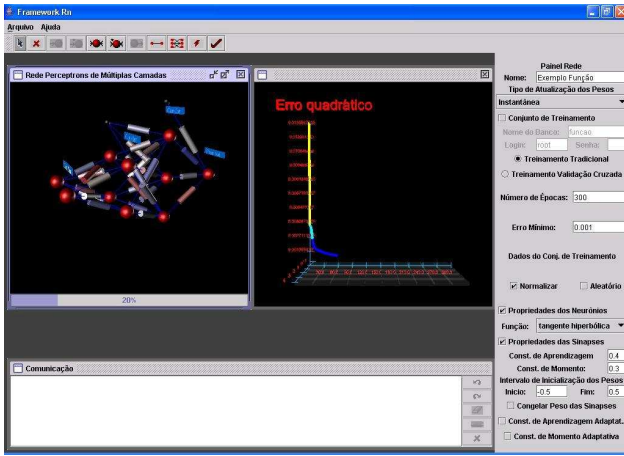


Fig. 3. Rede Neural Perceptron de Múltiplas Camadas

$$y = 1 + \sqrt{x_1} + \frac{1}{x_2} \quad (1)$$

Nesta função, x_1 e x_2 podem variar de 1 a 6. O conjunto de treinamento utilizado para este teste contém 100 padrões [13].

A rede configurada na Figura 3 possui uma camada de entrada com dois neurônios. A primeira camada intermediária possui quatro neurônios e um bias, a segunda camada intermediária possui três neurônios e um bias e a camada de saída possui apenas um neurônio. O tipo de atualização da rede é instantâneo, o tipo de treinamento é o tradicional com (300) épocas e com o erro mínimo MSE (0,001). Os dados do conjunto de treinamento foram normalizados, todos os neurônios da rede estavam configurados com a função de ativação “tangente hiperbólica”, todas as sinapses da rede foram configuradas com os mesmos atributos: constante de aprendizagem (0,4), constante de momento (0,3) e foram inicializadas entre o intervalo (-0,5 e 0,5).

Com base na Figura 3, é possível perceber na barra de progresso do treinamento (20%) que a rede neural convergiu pelo erro mínimo estabelecido. Sendo assim, depois da rede ter sido treinada, a mesma foi testada para alguns padrões e o resultado foi positivo (consultar Tabela I). Apesar de, na configuração do treinamento, os dados terem sido normalizados, tal fato fica transparente para o usuário, pois os testes podem ser efetuados com os valores de entrada não normalizados.

X_1	X_2	Y	Saída da rede
1,5	5,5	4,7603	4,9203
1,5	4,0	6,1243	6,0286
5,5	3,0	13,5316	13,6418

TABLE I
TESTE DA FUNÇÃO MATEMÁTICA

B. FrameRN: Rede de Mapas Auto-Organizáveis de Kohonen

Na Figura 4, pode ser visualizada a rede neural configurada para solução do seguinte problema: classificar animais de

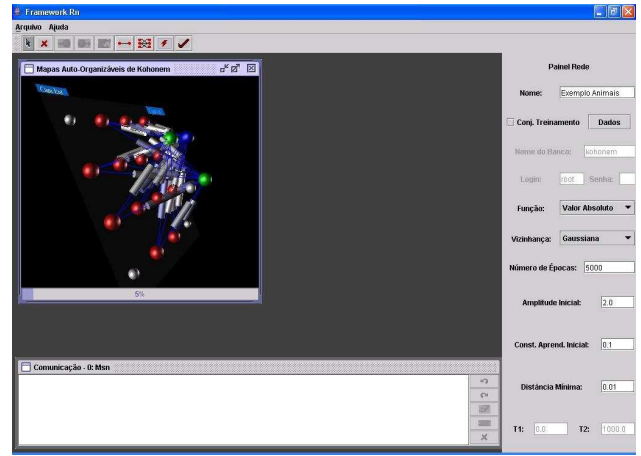


Fig. 4. Rede Neural Mapas Auto-Organizáveis de Kohonen

acordo com suas características. O conjunto de treinamento é composto por 16 padrões, ou seja, 16 animais diferentes, sendo que em cada padrão existem 13 atributos distintos, como: se possui 4 patas, se possui pena, se voa, entre outros [8]. A rede neural terá a função de encontrar padrões neste conjunto de treinamento e agrupar os animais por características semelhantes.

A rede neural modelada para a solução do problema proposto possui uma camada de entrada com 4 linhas e 4 colunas, sendo que, dos 16 possíveis neurônios, apenas 13 foram habilitados (devido à quantidade de entrada do conjunto de treinamento) e uma camada de saída com 2 linhas e 2 colunas, sendo que apenas 3 dos neurônios foram habilitados. Fato este motivado pela tentativa de se encontrar 3 padrões no conjunto de treinamento que está sendo treinado. Foi implementada a conectividade total entre os elementos. A função utilizada para a seleção do neurônio vencedor é a do “Valor Absoluto” e o tipo de vizinhança utilizada no treinamento foi a “Gaussiana”. Foram utilizadas (5000) épocas durante o treinamento, a amplitude inicial de (2,0), a constante de aprendizagem inicial foi de (0,1), a distância mínima foi de (0,01) e o intervalo de inicialização dos pesos das sinapses foi (-0,5 e 0,5).

De acordo a Figura 4, a rede neural modifica-se visualmente durante o seu treinamento. A barra de progresso indica a porcentagem do treinamento com base na época em questão e os neurônios da camada de saída mudam de cor. A cor azul indica o neurônio vencedor e a cor verde indica a vizinhança em torno deste neurônio.

A rede neural treinada conseguiu classificar o conjunto de treinamento proposto em três padrões diferentes, um em cada neurônio de saída. Os padrões dividem-se da seguinte forma:

- Grupo 1: Pombo, Galinha, Pato, Ganso, Coruja, Falcão e Águia;
- Grupo 2: Raposa, Cão, Lobo, Gato, Tigre e Leão;
- Grupo 3: Cavalo, Zebra e Vaca.

Desta forma, cada grupo excita um neurônio respectivo na camada de saída, logo, quando os dados de um respectivo padrão forem testados na rede, o neurônio disparado ficará na

tonalidade de cor azul.

VI. CONSIDERAÇÕES FINAIS

De um modo geral, acredita-se que este trabalho apresenta uma preocupação relevante no que se refere a facilitar o desenvolvimento e a compreensão de um domínio de aplicação como as redes neurais, que, apesar de bem fundamentado, ainda apresenta dificuldades de compreensão. Utilizando a técnica de realidade virtual, esta mostrou ser mais adequada para o tipo de problema em questão.

O FrameRN possui um diferencial em termos de interação com o usuário, pois o mesmo utiliza técnicas de realidade virtual e devido a este recurso de interface em três dimensões. As informações pertinentes às redes neurais e ao seu treinamento podem ser facilmente visualizadas. Podendo ser caracterizada como uma potencial ferramenta para o ensino de redes neurais. Como propostas futuras, podemos acrescentar novas arquiteturas de redes neurais. Assim como fazer uma análise ergonômica no FrameRN.

REFERENCES

- [1] J. Seixas A. Santos, B. Pereira. Árvore de classificação e redes neurais artificiais: Uma aplicação à predição de tuberculose pulmonar. *IV CBRN*, pages 427–432, 2003.
- [2] E. Hermeley A. Silveira. Controle de robôs móveis via rede de wavelet. *IV CBRN*, pages 67–72, 2003.
- [3] R. Santos E. Flek, C. Pedreira. Agrupamento de dados sísmicos através do algoritmo de kohonen. *IV CBRN*, pages 445–450, 2003.
- [4] M. Roseinberg F. Moreira. Reconhecimento e classificação de padrões de imagens de núcleos de linfócitos do sangue periférico humano. *IV CBRN*, pages 7–12, 2003.
- [5] C. Horstmann G. Cornell. *Core Java 2: Fundamentos*. Makron Books do Brasil, 2001.
- [6] S. Vergilio G. Souza, A. Ramirez. Utilizando redes neurais para estimar o tamanho de um software. *IV CBRN*, pages 73–78, 2003.
- [7] M. Deering H. Sowizral, K. Rushforth. *The Java 3D API Specification*. Addison Wesley, 1997.
- [8] S. Haykin. *Neural Networks Principles and Practice*. Bookman, 2002.
- [9] J. Daniels J. Cheesman. *UML Components*. Addison Wesley, 2001.
- [10] E. Teatini K. Coco, A. Almeida. Detecção de doenças cardíacas - através de lifting wavelet e redes neurais artificiais. *IV CBRN*, pages 235–238, 2003.
- [11] C. Kiner. Sistemas de realidade virtual. Technical report, Faculdade de Marília, 2002.
- [12] Mathworks. *Matlab*. Mathworks, Inc., Natick, MA, 1999.
- [13] O. Ohashi. Framework 3d para simulação de redes neurais. Master's thesis, Universidade Federal do Pará - Programa de Pós-Graduação em Engenharia Elétrica, 2005.
- [14] J. Robbins. *Modeling Software Architectures in the Unified Modeling Language*. Acmisem, 2002.
- [15] D. Shelman. *Java 3D Programming*. Hanning, 2002.