Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) Aplicada em *Software* de *Interface* gráfica

Leonardo do Espirito Santo[1,2], Rodrigo Silvério da Silveira[1,2], João Inácio da Silva Filho[1], Cláudio Luís Magalhaes Fernandes[1]

[1]UNISANTA- Universidade Santa Cecília – Programa de Pós Graduação em Engenharia Mecânica – PPGEMec - Rua Oswaldo Cruz, 288 , Boqueirão Santos-SP CEP- 11045-000

[2]ESCOLA SENAI "HESSEL HORÁCIO CHERKASSKY' – Serviço Nacional de Aprendizagem Industrial – Praça da Bíblia, 1 -, CEP 11510-030 Centro, Cubatão - SP
Email: leonardosantoe@gmail.com

Resumo

As Células Neurais Artificiais Paraconsistentes (CNAPs) são baseadas na lógica Paraconsistente Anotada com anotação de dois valores (LPA2v) e foram construídas para oferecer características semelhantes as de um neurônio biológico. Estas células quando agrupadas permitem que sejam criadas Redes Neurais Artificiais Paraconsistentes (RNAP) de modo a apresentar as funcionalidades de um cérebro humano. Este trabalho apresenta a modelagem de uma célula neural artificial Paraconsistente de Aprendizagem (CNAPap) em um software de *interface* gráfica. Esta *interface* permite que seja configurado um Padrão e um Fator de aprendizagem, no qual a cada reinserção do padrão é gerado um novo resultado de aprendizado. Todos esses dados são monitorados por meio de um gráfico *on-line* que pode ser ajustado quando necessário.

Palavras-Chave: Célula Neural Artificial, Lógica Paraconsistente, Redes Neurais Artificiais, Simulação.

Paraconsistent Artificial Neural Cell Applied Learning in Graphical Interface Software

Abstract

Paraconsistent Artificial Neural Cells (PANCels) are based on Paraconsistent Annotated logic with two-valued annotation (PAL2v) and were constructed to offer characteristics similar to those of a biological neuron. These clustered cells allow the creation of Paraconsistent Artificial Neural Networks (PANNet) in order to present the functionalities of a human brain. This work presents the modeling of a Paraconsistent Learning Neural Cell (PANCelap) in a graphical interface software. This interface allows you to configure a Learning Pattern and a Learning Factor, in which each reinsertion of the pattern generates a new learning result. All of this data is monitored through an online chart that can be adjusted as needed.

Keywords: Artificial Neural Cell, Paraconsistent Logic, Artificial Neural Networks, Simulation.

Introdução

Os estudos com Redes Neurais Artificiais tem se apresentado nos últimos tempos como um grande campo pesquisas, que buscam reproduzir de forma artificial o comportamento do

neurônios biológicos [1] através de modelagem matemática. Diante deste contexto a lógica Paraconsistente que faz parte das lógicas não-clássicas, tem sido aplicada com sucesso nas Redes Neurais Artificiais Paraconsistentes (RNAP), conforme foi observado em publicações deste segmento. Neste trabalho é apresentado uma modelagem de Célula Neural Artificial de Aprendizagem (CNAPap) aplicada em um software de interface gráfica com o intuito de propiciar estudos com simulações fundamentadas no CNAPap, outra possibilidade é permitir pesquisas com o algoritmo do CNAPap agregado com aplicações de áreas diversas que buscam possuir inteligência artificial.

Célula Neural Artificial Paraconsistente de Aprendizagem - CNAPap

A Lógica Paraconsistente Anotada com Anotação de dois Valores (LPA2v) tem como característica analisar duas evidências de entrada, denominadas como crença (μ_1) e descrença (μ_2), uma é considerada favorável e outra desfavorável [3]. Dessa forma o algoritmo da LPA2v denominado Para-analisador as interpreta e apresenta como saída um estado lógico correspondente. Com base no algoritmo Para-Analisador foi construída as Células Neurais Artificiais Paraconsistentes de forma a permitir maior simplicidade e eficiência em projetos com redes neurais Artificiais [1]. Ao combinar essas células, é possível estruturar uma rede neural capaz de armazenar dados e manipulá-los.

A Célula Neural Artificial Paraconsistente básica (CNAPb), foi construída com base no algoritmo Para-Analisador, que por sua vez deu origem a criação da Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) Figura 1. O CNAPap pode aprender valores entre 0 e 1 que devem ser definidos no padrão de aprendizado μ_1, já a entrada μ_2 possui um valor inicial de 0,5 e é realimentada pela saída $\mu_1 r$. O Fator de Aprendizagem (F_A) tem a função de influenciar o quão rápido será o aprendizado da célula [2].

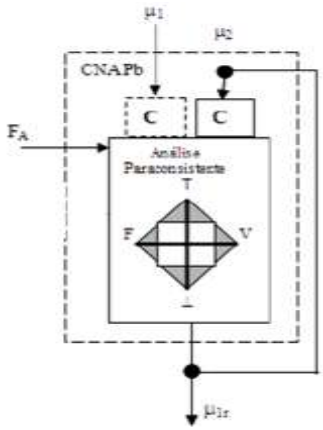

Figura 1 - Símbolo da célula CNAPap [1]

Objetivos: Neste trabalho, foi construído uma Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap) em um *software* de *interface* gráfica com o objetivo de apresentar uma simulação de modo a permitir o seu monitoramento e também aplicações simuladas que sejam baseadas em Redes Neurais Artificiais Paraconsistentes RNAPs.

Materiais e Métodos

O *software* utilizado neste trabalho foi o *BluePlant* do fabricante *Altus*, que é aplicado principalmente em sistemas de supervisão e controle de automação, mas que neste estudo teve como função simular uma Célula Neural Artificial Paraconsistente de Aprendizagem (CNAPap). A versão utilizada foi a *Express*, a qual possibilita utilizar as linguagens de programação C# e VB.net, sendo que esta última, foi a escolhida para a escrita do algoritmo.

A tela de simulação foi construída inicialmente inserindo todos objetos necessários para demonstração do treinamento da Célula Neural Artificial Paraconsistente, utilizando para isso a própria biblioteca do *BluePlant*, conforme demonstrado na Figura 2.



Figura 2– Tela de Simulação do CNAPap.

A tela é composta pelo padrão de entrada (μ1), pelo fator de aprendizagem (F_A), o grau de evidência (μE) e um gerador de gráfico. Para o início da simulação é necessário inserir o padrão de aprendizado da célula, sendo que neste estudo foi adotado o valor 1. Deve ser inserido também o Fator de aprendizagem F_A no intervalo de 0 a 1, no qual 0 determina que a célula aprenderá mais rapidamente e 1 de forma mais lenta, ou seja, neste último caso é

necessário ter mais passos para finalizar o aprendizado. O botão "início" configura os valores iniciais do CNAPap, atribuindo 0,5 para μE e 1 para F_A. O botão "Atualizar" faz a reinserção do valor 1, incrementando assim, o passo seguinte.

Resultados

Os testes foram realizados na tela de *interface* do *software* simulador, onde foi inserido inicialmente os valores de μ1 e F_A. Após pressionar o botão "Atualizar" o padrão automaticamente foi reinserido, incrementando um novo passo, o que gerou também um novo resultado de μE. Com um F_A igual a 1, foram necessários 10 passos de iteração até μE atingir o valor 1, o que corresponde ao padrão de aprendizagem definido inicialmente. Neste instante a mensagem "Aprendeu" é apresentada na tela, confirmando o aprendizado da célula.

A seguir é apresentado no quadro 1 todos os passos realizados com os valores de F_A, μ1 e os respectivos valores de grau de evidência resultante (μE).

Quadro1 - Valores obtidos no ensaio com o simulador

Passo	**F_A**	**Padrão μ1**	**Evidência μE**
0	**1**	**0**	**0**
1	**1**	**1**	**0,5**
2	**1**	**1**	**0,75**
3	**1**	**1**	**0,875**
4	**1**	**1**	**0,938**
5	**1**	**1**	**0,969**
6	**1**	**1**	**0,984**
7	**1**	**1**	**0,992**
8	**1**	**1**	**0,996**
9	**1**	**1**	**0,998**
10	**1**	**1**	**1**

A seguir, na figura 3 é apresentado o gráfico correspondente aos valores do quadro 1.

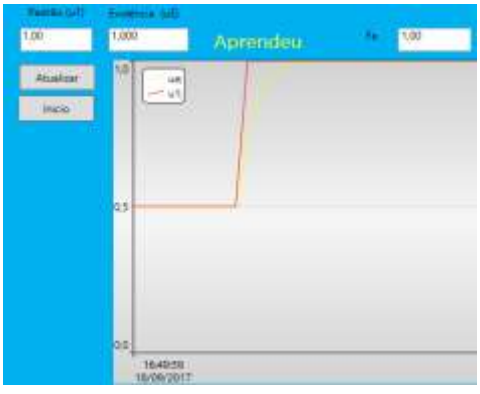

Figura 3 – Gráfico *on-line* do processo de aprendizagem.

Discussão

Os resultados apresentados demonstraram o pleno funcionamento da célula de aprendizagem (CNAPap) de modo que ao comparar os resultados apresentados no quadros 1 e no gráfico, com resultados de outros trabalhos tais como em [3] e [4], é possível afirmar que o CNAPap construída em VB.net possui características semelhantes as já apresentadas por outros autores.

É possível afirmar que o CNAPap construído no simulador de *interface* gráfica abre possibilidades para que com seu algoritmo, sejam construídas outras telas com características aplicadas em controle diversos, como na área de automação e processos industriais.

Verifica-se também que o controle simulado com o algoritmo do CNAPap proporciona alguns benefícios, tais como o como um melhor monitoramento do algoritmo e possibilidade de realizar ensaios simulados antes de uma real aplicação, o que poderá proporcionar maior agilidade e confiabilidade em determinadas aplicações.

Conclusões

Neste trabalho se aplicou o algoritmo da Célula Neural Artificial de Aprendizagem em um *software* de *interface* gráfica a fim de apresentar uma tela que pudesse proporcionar ao usuário, a possibilidade de monitorar e interagir com a célula CNAPap. O trabalho também permitiu a visualização *on-line* do aprendizado da célula CNAPap diante dos padrões pré definidos. Diante do exposto, pode-se afirmar que estes resultados contribuem para que outras pesquisas sejam realizadas utilizando como base a simulação descrita neste estudo. Esta configuração pode ser utilizada em aplicações diversas que necessitem de um modelamento com características do cérebro humano.

Referências

1. Da Silva Filho, J. I., “Introdução às Células Neurais Artificiais Paraconsistentes”. Revista Seleção Documental, N.8 Ano2, 12/2007.
2. Mario, M.C. Modelo de análise de variáveis craniométricas através de Redes Neurais Artificiais Paraconsistentes, Tese de Doutorado – USP, São Paulo, 2006.
3. Fernandes, C.L.M., Mario C. M., Da Silva Filho, J. I. “Study for inclusion of Paraconsistente Annotaded logic in specific Standards for use in Programable Controllers”, Science and Technology, ISSN: 2317-1316, Vol 1 2, p.p 49-53, 2012.
4. Fernandes, C. L. M. “Lógica Paraconsistente aplicada em Sistemas de Automação e Controle”. Dissertação de Mestrado. Universidade Santa Cecília Unisanta, Santos/SP, 2012.
5. Fausett, L. Fundamentals of Neural Networks, Architectures, Algoritms and Aplications Editora Prentice-Hall, Englwood, Cliffs – 1994