

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXIII Seminário de Iniciação Científica

## **PROPOSTA DE UMA UNIDADE DE MEMÓRIA PARA ACELERAR A OTIMIZAÇÃO DAS REDES NEURAS RECORRENTES EM AMBIENTES PARCIALMENTE OBSERVÁVEIS<sup>1</sup>**

**Eldair Fabrício Dornelles<sup>2</sup>, Márcia Da Silva<sup>3</sup>, Rogério S. M. Martins<sup>4</sup>, Sandro Sawiki<sup>5</sup>, Rafael Z. Frantz<sup>6</sup>, Fabricia Roos-Frantz<sup>7</sup>.**

<sup>1</sup> Projeto de pesquisa de Iniciação Científica.

<sup>2</sup> Bolsista PROBIC/FAPERGS, Ciência da Computação da UNIJUI, eldair.dornelles@gmail.com.

<sup>3</sup> Bolsista Voluntário, Ciência da Computação da UNIJUI, marcia.dasilva@unijui.edu.br.

<sup>4</sup> Professor Orientador, rogerio.martins@unijui.edu.br.

<sup>5</sup> Professor Orientador, sawicki@unijui.edu.br.

<sup>6</sup> Professor Colaborador, rzfrantz@unijui.edu.br.

<sup>7</sup> Professor Colaborador, frfrantz@unijui.edu.br

### Introdução

Com o uso das estruturas de redes neurais artificiais, é possível evoluir programas que podem proporcionar aos agentes inteligentes a capacidade de atuar em diversos ambientes (SILVA, 2010). Na modelagem da estrutura de uma Rede Neural Artificial (RNA), que tem por objetivo evoluir um programa que irá representar a inteligência de um agente, deve ser amplamente considerado o seu ambiente de atuação. Em determinados ambientes, tais como os parcialmente observáveis, a RNA pode apresentar dificuldades ou ser incapaz de evoluir. Isto ocorre devido ao conceito de memória de curto prazo implementada pelas RNAs (MACHADO, 2011). Para aumentar a eficiência de uma RNA convencional, quando é necessário armazenar informações por longos períodos, pode-se adicionar a essas estruturas alguns blocos de memória. Uma alternativa é adicionar às redes neurais convencionais, blocos de memórias Long Short Term Memory (LSTM). Tendo em vista algumas limitações apresentadas pelas estruturas das redes neurais artificiais convencionais, foi proposto neste trabalho a modelagem de uma unidade de memória para ser incorporada à estrutura de uma RNA, visando acelerar o seu processo de otimização e oferecer menor custo de processamento que os blocos LSTM conforme (LINDEN, 2011), o processo de evolução foi realizado por meio da meta heurística nominada de algoritmo genético, o qual foi responsável pelo ajuste dos pesos sinápticos.

A estrutura proposta foi utilizada para evoluir um programa que tem por objetivo representar a inteligência de um agente com capacidade de atuar em ambientes parcialmente observáveis. Mais precisamente, o agente em questão foi aplicado a uma versão modificada do problema presa predador (LÜTZ, 2011).

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXIII Seminário de Iniciação Científica

## Metodologia

Para o desenvolvimento deste trabalho, buscou-se, primeiramente, conhecimentos teóricos sobre as técnicas a serem utilizadas. Nesta fase foi realizado um estudo sobre agentes inteligentes, ambientes de tarefas, redes neurais recorrentes, estruturas de memórias, e algoritmos genéticos. Em seguida foi modelada e implementada a unidade de memória e o problema presa predador (modificado). Também foram realizadas implementações de um framework de redes neurais e um framework de algoritmo genético. As implementações foram todas realizadas em linguagem de programação C++, utilizando a IDE Eclipse. Por fim realizou-se uma análise de desempenho do modelo computacional com unidades de memória aplicado ao problema citado anteriormente e inferido o quão próximo, ou melhor é o seu desempenho em relação a uma RNA recorrente convencional.

Na versão modificada do problema presa predador foi definido um ambiente de dimensão 10x10. Neste ambiente, existe apenas um predador, uma presa, dois arbustos venenosos e um arbusto não venoso. O objetivo do predador é capturar a presa sem encostar nos arbustos venenosos. Ao iniciar o jogo, os arbustos venenosos são posicionados nas coordenadas [3,3] e [7,5] da matriz que representa o ambiente, a presa é posicionada na coordenada [5,5], o predador é inicializado na posição [0,0] da matriz-ambiente e o arbusto não-venoso é posicionado de forma aleatória em uma posição que esteja vazia. Os movimentos da presa e do predador são intercalados. Os movimentos da presa são todos realizados buscando se posicionar junto ao arbusto não venoso, com o objetivo de se esconder do predador. O predador recebe as informações da localização de todos os componentes do jogo, inclusive a própria posição. A partir do momento que a presa se esconde no arbusto não venoso, a informação referente à sua posição não é mais recebida, desse modo a única informação recebida pelo predador é a própria posição e o posicionamento dos três arbustos, porém sem o conhecimento de suas características. Uma ilustração deste ambiente é apresentada na Figura 1.

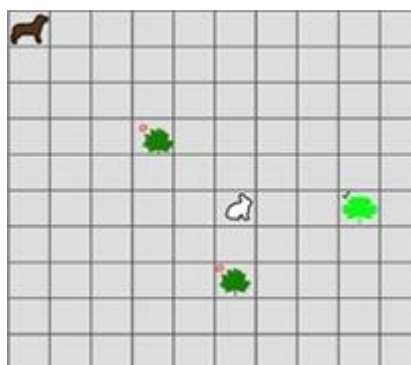


Figura 1 – Ilustração do modelo presa-predador(modificado).

As percepções são passadas ao agente em forma de lista. Esta lista contém as informações referente a posição [x,y] de cada componente do ambiente. Para interagir com ambiente o predador possui 4

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXIII Seminário de Iniciação Científica

ações possíveis, são elas: se movimentar para frente, para trás, para esquerda e para a direita. O predador é pontuado positivamente pela captura ou pela proximidade que o mesmo se encontra em relação à presa no momento em que o jogo é encerrado. Sendo que ao capturar a presa recebe 100 pontos, e a cada passo distante da presa, o predador perde 5 pontos do total de 100. Desta forma, se o jogo encerrar e o predador ficar a 2 passos da presa sua pontuação será 90. Caso o predador encoste em um arbusto venoso a pontuação obtida é dividida por 2. A unidade de memória modelada neste trabalho é composta por dois módulos. O primeiro define o valor a ser armazenado e o segundo define se o novo valor vai ser armazenado ou se vai ser descartado, caso este novo valor seja descartado a unidade mantém o valor anterior na memória. O valor a ser retornado pela unidade de memória é sempre o valor que estiver armazenado. A Figura 2 ilustra a estrutura da unidade de memória modelada neste estudo.

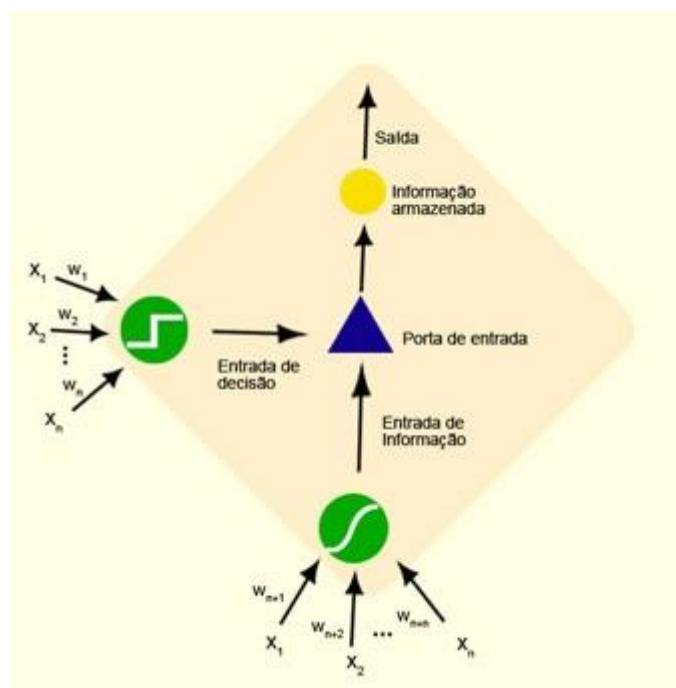


Figura 2 – estrutura da unidade de memória proposta neste trabalho.

Na estrutura definida para a rede neural, o número de neurônios e de unidade de memória não foi fixado, podendo assim variar com vistas a encontrar uma combinação que proporcione uma convergência mais rápida aos resultados desejados. Os pesos sinápticos utilizados pela rede neural recorrente são fornecidos pelo algoritmo genético, o qual gera uma população de indivíduos, onde cada indivíduo é formado por um vetor que contém uma combinação de pesos sinápticos (números reais), de acordo com o tamanho da rede neural. O processo de otimização ocorre da seguinte maneira: para cada indivíduo da população do algoritmo genético são definidos os pesos na rede neural, configurando assim a inteligência do agente a ser testado. Para cada um dos agentes é

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXIII Seminário de Iniciação Científica

inicializado o jogo presa predador de modo que o jogo é finalizado se o agente encostar em um arbusto venoso, encontrar a presa ou atingir o número limite de iterações. Para cada agente são realizadas 100 execuções com o jogo, visando obter uma melhor precisão do desempenho, a cada execução é inicializado um novo jogo. Ao final das 100 execuções é retornado para o indivíduo do algoritmo genético a pontuação média obtida pelo agente. Essa pontuação é utilizada como base para definir a eficiência deste indivíduo, sendo esse valor atribuído à sua aptidão. No passo seguinte o Algoritmo Genético se encarrega da otimização dos pesos, seguindo os seus critérios de evolução.

Na fase de evolução da rede neural, buscou-se, primeiramente, encontrar as quantidades de neurônios que representassem as estruturas com melhor convergência e desempenho. Nesta fase, após alguns testes alterando a quantidade de neurônios pode-se perceber que a RNA recorrente convencional com 6 neurônios foi a mais eficiente. Portanto, foram realizados alguns testes adicionando memórias em estruturas com 5 e 6 neurônios. Em seguida deu-se início a fase de ajuste dos parâmetros do algoritmo genético, os quais por fim, foram definidos com os seguintes valores: 500 indivíduos, 50 gerações, tipo de seleção torneio, cruzamento simples, taxa de mutação 0,05% e taxa de elitismo em 2%.

## Resultados e Discussões

Após a realização dos treinamentos e definidos as melhores estruturas de RNA, foram realizados os testes e coleta dos resultados. Na fase de teste foram realizados 100.000 execuções para cada estrutura, sendo que a cada nova execução gerava-se um novo jogo, de modo que o arbusto não venoso era posicionado de forma aleatória. Na Tabela 1, são apresentados em porcentagem os resultados dos testes utilizando as melhores estruturas encontradas neste trabalho.

Estrutura da Rede Neural	Capturou a presa	Encostou em um arbusto venoso	Atingiu o limite de iterações
5 Neurônios	86.45%	8.78%	4.77%
6 Neurônios	89.10%	8.76%	2.14%
7 Neurônios	59.63%	5.49%	34.88%
<b>5 Neurônios e 1 Unidade de Memória</b>			
5 Neurônios e 1 Unidade de Memória	89.50%	10.50%	0.00%
<b>5 Neurônios e 2 Unidades de Memórias</b>			
5 Neurônios e 2 Unidades de Memórias	61.74%	11.37%	26.89%
<b>6 Neurônios e 1 Unidade de Memória</b>			
6 Neurônios e 1 Unidade de Memória	90.65%	9.35%	0.00%
<b>6 Neurônios e 2 Unidades de Memórias</b>			
6 Neurônios e 2 Unidades de Memórias	67.76%	10.78%	21.46%

Tabela 1 – Resultados obtidos para cada estrutura após 100.000 execuções.

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXIII Seminário de Iniciação Científica

Através de uma análise da Tabela 1, pode-se perceber que entre as RNA recorrentes convencionais, a que obteve melhor desempenho foi a rede com 6 neurônios, e que ao adicionar a esta rede uma unidade de memória, foi possível elevar em 1,55% o percentual de captura da presa e em nenhum momento foi atingido o limite de iterações para este ambiente, o qual havia sido fixado em 100 iterações.

### Conclusões

Na fase de testes pode-se perceber que ao adicionar a unidade de memória à estrutura de uma RNA recorrente convencional a mesma melhorou o desempenho no ambiente ao qual foi aplicado. Observou-se, também, que o número de neurônios convencionais definidos para compor a estrutura da RNA, afeta diretamente o seu desempenho. Do mesmo modo, a quantidade de unidades de memória, deve ser ajustada de acordo com o problema que estiver sendo abordado e a dimensão da RNA.

### Palavras-Chave

Algoritmos Genéticos, Agentes Inteligentes, Presa Predador.

### Agradecimentos

A UNIJUI que por meio do programa institucional de bolsas de iniciação científica, apoiado pelo FAPERGS – (PROBIC/FAPERGS), proporcionaram o aprendizado de técnicas e métodos de pesquisa, estimulando o desenvolvimento do pensar científico. Também aos professores orientadores e aos colegas bolsistas que incentivaram e auxiliaram neste caminho de aprendizado.

### Referências Bibliográficas

DA SILVA, Ivan Nunes; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. Redes Neurais Artificiais para engenharia e ciências aplicadas curso prático. Artliber, 2010.

LINDEN, Ricardo. Algoritmos Genéticos. 3. ed. Rio de Janeiro: Ciência Moderna, 2011.

MACHADO, S. C. S. Reconhecimento Off-line de Escrita Cursiva Usando Redes Neurais Recorrentes Multidimensionais, 2011.

LÜTZ, A. F. Competição e Coexistência em Populações Biológicas. Tese (Doutorado)-Universidade Federal do Rio Grande do Sul, 2011.