

DOI: 10.33947/2316-7394-v8n1-3896

REDE NEURAL PERCEPTRON APLICADA AO JOGO DA NAVE**PERCEPTRON NEURAL NETWORK APPLIED TO SHIP PLAY**Elisama Santos¹, Matheus S. de Jesus², Bianca Sanches³, Fabio Fonseca Barbosa Gomes⁴, Joserland Souza Santos⁵**RESUMO**

A rede neural propõe o processamento de dados de maneira equivalente ao cérebro humano tendo como princípio a organização de neurônios. O cérebro humano tem a capacidade de aprender e tomar decisões baseadas nas aprendizagens adquiridas ao longo do tempo, a rede neural artificial é formada por algoritmos que simulam o processamento de aprendizagem do cérebro humano projetados para determinada finalidade a fim de aprender a tomar decisões. O conhecimento adquirido através da taxa de aprendizado e pesos sinápticos são usados para memorizar o conhecimento treinado, estas características assemelham-se ao cérebro humano. Neste sentido, será demonstrado a criação do jogo nave com a Rede Neural Perceptron a fim de criar simulações que possibilite desenvolver habilidades frente a tomada de decisão.

PALAVRAS-CHAVE: *Aprendizagem. Redes Neurais. Ajuste de pesos*

ABSTRACT

The neural network proposes the processing of data in an equivalent way to the human brain having as principle the organization of brain neurons. The human brain has the ability to learn and make decisions based on Learnings acquired over time, the artificial neural network is formed by algorithms that simulate the processing of human brain learning designed for certain In order to learn to make decisions. The knowledge acquired through the learning rate and synaptic weights are used to memorize the trained knowledge, these characteristics resemble the human brain. In this sense, it will be demonstrated the creation of the spacecraft game with the Perceptron Neural Network in order to create simulations that enable the development of decision making skills.

KEYWORDS: *Learning. Neural networks. Weight adjustment*

¹ UNINASSAU

² UNINASSAU

³ UNINASSAU

⁴ Centro Universitário UNIRB, Centro Universitário Dom Pedro II e IFBA

⁵ UNINASSAU

1. INTRODUÇÃO

O ser humano é um animal racional, residente no reino animal. Possui inteligência e faz uso da linguagem oral, verbal e corporal, isto o distingue da qualidade do animal irracional (NUNES, 2011). Além disso, o ser humano é capaz de compreender, perceber, analisar e manipular um mundo maior e mais complexo que a sua mente. O cérebro humano possui características desejáveis em qualquer sistema artificial.

Um exemplo disso é a sua capacidade para lidar com informações inconsistentes ou probabilísticas, alta flexibilidade para se adaptar a situações aparentemente pouco definidas, tolerância a falhas, entre outras (BARKOW *et al.*, 1992). Todas estas características mencionadas despertaram o interesse de pesquisadores que, na década de 1980, que intensificaram suas linhas de estudo na área de inteligência artificial com o uso da computação intensiva (CHEVITARESE, 2010).

O ramo da inteligência artificial abrange uma enorme variedade de subcampos, desde áreas de uso geral, como aprendizado e percepção, até tarefas específicas como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia e diagnóstico de doenças (GOMES, 2010). Por possuir uma ampla abordagem está relacionada a diversas áreas científicas como lógica matemática, engenharia, biologia dentre outras. A Inteligência artificial segue quatro linhas de pensamento:

I. Sistemas que pensam como seres humanos: “O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal”. (HAUGELAND, 1985).

II. Sistemas que atuam como seres humanos: “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (KURZWEIL, 1990).

III. Sistemas que pensam racionalmente: “O estudo das faculdades mentais pelo seu uso de modelos computacionais.” (CHARNIAK; MCDERMOTT, 1985).

IV. Sistemas que atuam racionalmente: “A Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (POOLE *et al.*, 1998).

Desta maneira, o principal objetivo deste artigo é aplicar os conceitos de Rede Neural Perceptron, através de um jogo em que uma nave deve fugir de um asteroide que tentará colidir com ela. Será neces-

sário configurar a inteligência artificial da nave espacial para o resultado ser satisfatório.

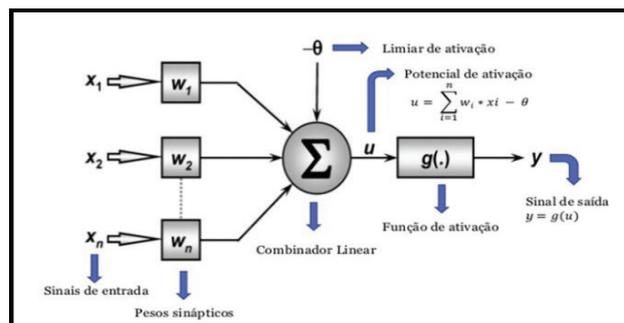
2. REDES NEURAIS ARTIFICIAIS

Segundo CHEVITARESE (2010), uma rede neural é um sistema computacional constituído elementos processadores interligados por unidades conhecidas como neurônios, que trabalham em paralelo para desempenhar uma determinada tarefa. Os modelos de redes neurais artificiais constituem uma técnica estatística não linear capaz de resolver uma gama de problemas de grande complexidade (ALECRIM, 2014). São úteis em situações que não é possível definir explicitamente uma lista de regras. Em geral, isso acontece quando o ambiente gerador dos dados muda constantemente. As principais áreas de atuação são para classificação de padrões e previsão.

Em relação a rede neural da inteligência artificial atual e o cérebro humano, dois fatores indicam a semelhança que as redes neurais artificiais apresentam em relação entre eles: (i) O conhecimento é adquirido em uma rede neural através de um processo de aprendizagem. (ii) O conhecimento é armazenado em conexões (sinapses) existentes entre os elementos da rede, representadas através de pesos (OSÓRIO *et al.*, 2007). As redes neurais artificiais possuem como característica: adaptação por experiência; capacidade de aprendizado; habilidade de generalização; organização de dados; armazenamento distribuído; facilidade de implementação; tolerância a falhas (PALMIERE, 2016).

O interior de um neurônio artificial é formado por sinais de entradas, pesos sinápticos, combinação linear, limiar de ativação, função de ativação e sinal de saída conforme Figura 1.

Figura 1: Estrutura interior de um neurônio artificial de única camada



Fonte: (HAYKIN, 2001)

Segundo LEITE (2018), os sinais da entrada no neurônio são representados pelo vetor $x = [x_1, x_2, \dots, x_N]$. Quando eles chegam ao neurônio, são multiplicados pelos respectivos pesos sinápticos, que são os elementos do vetor $w = [w_1, w_2, \dots, w_N]$, números reais expressando a importância das respectivas entradas para a saída. $\Sigma = ((x_1 * w_1) + x_2 * w_2) + (x_N * w_N)$, o valor obtido é comparado se é menor ou maior que o valor existente em limiar (*threshold*).

Se Σ for maior ou igual a limiar saída da rede igual a 1 (um), se Σ for menor que limiar saída da rede igual a 0 (zero). Gerando o valor u , comumente denominado potencial de ativação. O valor u passa então por uma função matemática de ativação $g(\cdot)$, com a característica de ser não linear responsável por limitar tal valor a certo intervalo, produzindo o valor final de saída y do neurônio.

3. REDE ARTIFICIAL PERCEPTRON DE UMA ÚNICA CAMADA

O modelo Perceptron foi desenvolvido nas décadas de 1950 e 1960 pelo cientista Frank Rosenblatt, inspirado em trabalhos anteriores de Warren McCulloch e Walter Pitts (DATA SCIENCE ACADEMY, 2019). Desenvolvido para lidar com problema de reconhecimento de padrões, segundo Campos (2001) esta é a área de pesquisa que tem por objetivo a classificação de objetos (padrões) em um número de categorias ou classes. Este é um tipo de tarefa que os seres humanos fazem sem nenhum esforço aparente e de forma quase instantânea. Porém, é um dos problemas mais difíceis de serem resolvidos por uma máquina (ALMEIDA, 2000).

De acordo com CHEVITARESE (2010), uma rede neural aprende a partir de dados, por isso a escolha das variáveis de entrada é de grande importância. Isso porque embora a rede neural tenha condições de modelar problemas difíceis de especificar, é preciso que existam dados e observações suficientes e representativas para o conhecimento ser extraído e para que o aprendizado aconteça com sucesso. O conhecimento é passado para a rede por um algoritmo de treinamento e o aprendizado é transformado e armazenado em densidades de conexões que são os pesos. O aprendizado é o resultado das muitas apresentações de um determinado conjunto de exemplos de treinamento.

O treinamento da rede pode se dar de duas maneiras: batelada ou incremental.

1. Batelada ou por ciclos: a atualização dos pesos acontece somente após a apresentação de todos os padrões. Cada padrão é avaliado na mesma configuração de pesos (CHEVITARESE, 2010).

2. Padrão a Padrão ou incremental: o algoritmo faz a atualização dos pesos após a apresentação de cada novo padrão. Por isso mesmo, a frequência das atualizações em um mesmo período tende a ser maior que no caso anterior (CHEVITARESE, 2010).

3.1 MODELO DE APRENDIZADO PERCEPTRON

Segundo ROMANI (2017), para a criação do algoritmo de treinamento da Rede Perceptron é necessário:

1. Obter o conjunto de amostras de entradas e saídas.
2. Iniciar os pesos com valores aleatórios pequenos.
3. Iniciar a contagem das épocas, ou seja, dos ciclos de treinamento.
4. Durante o ciclo de treinamento corrigir os pesos a partir do cálculo da variação destes e atualização de seus valores pelas equações dadas.
5. Calcular o Erro encontrado e identificar se for menor ou igual que o erro aceitável.
6. Se o erro é maior do que o aceitável, reiniciar o processo a partir do passo 3. Senão, simular a rede treinada para os dados de entrada.

3.2 TIPOS DE APRENDIZADO NAS REDES NEURAIS ARTIFICIAIS

Existem uma série de etapas que a rede neural precisa seguir, são elas:

Nenhuma saída antecipada ou resposta desejada é fornecida.

Usualmente utilizado para a classificação dos padrões de entrada.

Em geral, apenas um único neurônio na camada de saída dispara uma resposta.

Tal resposta classifica o padrão de entrada (ALVES *et al.*, 2012).

No modelo de treinamento supervisionado funciona da seguinte forma: (1) Um conjunto de resultados esperados é fornecido. (2) A diferença entre a saída atual e a saída desejada consiste no erro. (3) O cálculo do erro permite que a matriz de pesos seja ajustada. (4) A rede é aperfeiçoada a cada série de

treinamentos entrada (ALVES *et al.*, 2016).

Já na Aprendizagem híbrida funciona desta maneira: um método que combina duas ou mais técnicas de aprendizagem, para ajuste dos pesos sinápticos de uma rede neural artificial, com o objetivo de unir as vantagens e superar limitações individuais de cada técnica. Este método permite construir sistemas mais robustos, resolver problemas mais complexos, dentre outras vantagens (ALVES *et al.*, 2016).

4 ALGORITMO DE APRENDIZAGEM APLICADO AO JOGO DA NAVE

O algoritmo desenvolvido tem por objetivo ensinar a nave a desviar do asteroide mediante treinamento supervisionado. Cada vez que houver colisão, o peso da nave e do asteroide serão ajustados, porém quando a nave aprender a desviar do asteroide, não haverá mais ajuste de pesos.

Os valores iniciais são determinados pela Figura 2, onde o valor da posição nave e posição do asteroide são representados através de vetores. Se a nave está na posição 0 (zero) e asteroide na posição 0 (zero) a nave deverá se mover para a posição 1 (um), se nave está na posição 0 (zero) e asteroide na posição 1 (um) a nave deverá permanecer na posição 0 (zero), se nave está na posição 1 (um) e asteroide na posição 0 (zero) a nave deverá permanecer para a posição 1 (um), se a nave está na posição 1 (um) e asteroide na posição 1 (um) a nave deverá se mover para a posição 0 (zero), conforme pode ser visto na figura 2.

Figura 2: Valores iniciais

```
int posicaoNave[4] = {0,0,1,1};
int posicaoAste[4] = {0,1,0,1};
int saidaEsperada[4] = {1,0,1,0};
```

Fonte: (Próprio Autor, 2019)

Os valores de pesos são definidos como W_a referenciando o valor do peso da nave e W_b referenciando o valor do peso do asteroide. O valor da regulação do peso é definido por duas variáveis $regulaW_a$ irá regular o novo peso da nave, $regulaW_b$ irá regular o novo peso do asteroide. Os valores de taxa de aprendizagem e limiar foram definidos como valores reais fixos para posterior cálculo de saída do neurônio será pro-

duzido após a chamada da função de ativação seguida pela função de transferência. O valor de W_a (valor peso nave) e W_b (valor peso asteroide) são definidos conforme sugestão para inicialização dos pesos utilizando a porta NOT segundo Gatto (2018) que sugere valores entre 0.0 e 0.5. Conforme Figura 3.

Figura 3: Valores iniciais definidos

```
float Wa = 0.3;
float Wb = 0.1;
float taxaAprendizagem = 0.1;
float regulaWa, regulaWb;
float limiar = 0.0
```

Fonte: (Próprio Autor, 2019)

Para alcançar a posição desejada do vetor, é utilizado um contador, pois os vetores criados possuem 4 (quatro) posições iniciando em 0 (zero) e finalizando em 3 (três). A função `redeNeural` (Figura 4) irá realizar o somatório verificando se o valor armazenado em somatório é maior ou menor que limiar. Existirá 1 (uma) saída para cada possibilidade. O erro será calculado ($saidaEsperada - saidaAtual$) e de acordo com a posição da nave um valor será atribuído sendo 1 (um) ou 0 (zero). A atribuição está diretamente relacionada ao valor obtido na variável somatório.

Figura 4: Função para execução de rede neural

```
void redeNeural(){
    while(i<=3){
        somatorio = (posicaoNave[i]*Wa) + (posicaoAste[i]*Wb);
        if(somatorio < limiar){
            saidaAtual[i] = 0;
        }else{
            saidaAtual[i] = 1;
        }
        erro[i] = saidaEsperada[i] - saidaAtual[i];
        i++;
    }
    if(pynave == 0 && pyaste == 0){
        pynave = saidaAtual[0];
    }else if(pynave == 0 && pyaste == 1){
        pynave = saidaAtual[1];
    }else if(pynave == 1 && pyaste == 0){
        pynave = saidaAtual[2];
    }else if(pynave == 1 && pyaste == 1){
        pynave = saidaAtual[3];
    }
}
```

Fonte: (Próprio Autor, 2019)

A função `regulaPeso` (Figura 5) terá por parâmetro a variável do tipo inteira, denominada posição, que representará a posição do vetor que deseja obter. Identificando a posição, a regulação do peso será realizada tanto para W_a (peso nave) quanto para W_b (peso asteroide). O novo peso será armazenado nas variáveis $regulaW_a$ (regula peso nave) e $regulaW_b$ (regula peso asteroide).

Figura 5: Função para execução da regulação de peso

```

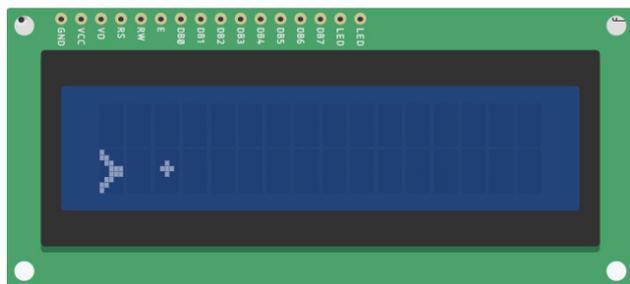
void regulaPeso(int posicao){
  while(i<=3){
    if(i == posicao){
      regulaWa= (taxaAprendizagem*erro[i]*posicaoNave[i])+ Wa;
      regulaWb= (taxaAprendizagem*erro[i]*posicaoAste[i])+ Wb;
    }
    i++;
  }
  Wa = regulaWa;
  Wb = regulaWb;
}

```

Fonte: (Próprio Autor, 2019)

5. SIMULAÇÃO DO JOGO E RESULTADOS

O jogo foi instalado em um microcontrolador com um módulo que possui uma pequena tela, também foi feita uma simulação do jogo pelos autores do artigo. O *game* tem como principal objetivo é que a nave escape do asteroide que está indo em sua direção. A Rede Neural irá “aprender” durante as simulações quais são as possíveis rotas de fuga. Ao iniciar o jogo, a nave ainda não identifica se o asteroide está vindo da posição 0 (zero) ou posição (um), conforme visto na figura 6. Quando ocorre da nave colidir com o asteroide, o usuário é sinalizado com uma mensagem de “perdeu”, conforme é apresentado na figura 7.

Figura 6: Início do jogo

Fonte: (Próprio Autor, 2019)

Figura 7: Fim de jogo

Fonte: (Próprio Autor, 2019)

5.1 RESULTADO DA SIMULAÇÃO

Sempre que a nave colidir haverá execução da rede neural bem como ajuste de pesos. Quando os pesos são regulados a nave em determinado momento conseguirá identificar a posição do asteroide podendo desviar ou manter seu posicionamento. Neste momento, a aprendizagem aconteceu de tal forma que não haverá mais regulação de peso e a nave passará a desviar do asteroide, conforme é apresentado na Figura 8.

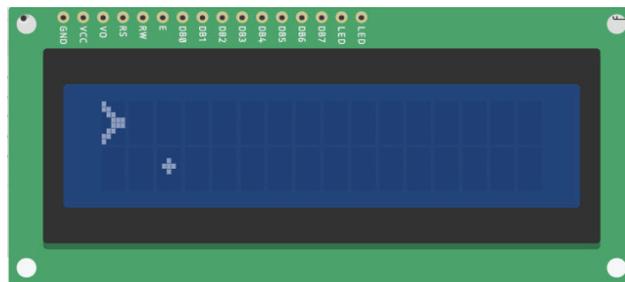


Figura 8: Nave desvia de asteroide

CONSIDERAÇÕES FINAIS

Neste artigo, o objeto do jogo criado com a utilização do algoritmo desenvolvido com a Rede Neural Perceptron mostrou-se bastante promissor. Visto que, através das simulações a nave “aprende” e armazena aquela informação para que consiga fazer uma jogada diferente na próxima partida do jogo. Portanto, sempre que a nave colidir com o asteroide haverá ajuste de pesos, a rede aprenderá mediante treinamento supervisionado, conseqüentemente a rede desenvolverá a habilidade de tomada de decisão.

As Redes Neurais mostram que a Inteligência Artificial está se desenvolvendo cada vez mais, para trabalhos futuros, pretende-se desenvolver uma simulação mais complexa, com uma quantidade de naves e asteroides maior, para avaliar o uso da Rede Neura Perceptron neste caso.

REFERÊNCIAS

- ALMEIDA, Maria Aparecida. **O Perceptron de Rosenblatt**. Introdução ao Estudo de Redes Neurais e Artificiais, UFSC, Florianópolis, 2000.
- ALECRIM, Emerson. **Redes Neurais Artificiais**. 2004. Disponível em: <<https://www.infowester.com/redesneurais.php>>. Acesso em: 26 maio 2019.
- ALVES, Franciele A. S.; FERREIRA, Evelyne L.; GOUVÊA JUNIOR, Maury M. **Rede Neural Artificial com Aprendizagem Híbrida para Problemas de Classificação de Padrões**. 2016. Disponível em: <<http://eati.info/eati/2016/assets/anais/Longos/77.pdf>>. Acesso em: 18 maio 2019.
- ALVES, L.; ALEXANDRE, W.; DE ARAÚJO, S.; LIBRANTIZ, H.; FELIPE, A; **Reconhecimento de padrões de texturas em imagens digitais usando uma rede neural artificial híbrida**, Exacta, São Paulo, 2006. p. 325-332.
- BARKOW, J. H.; COSMIDES, L.; TOOBY, J. (Eds.). **The adapted mind: evolutionary psychology and the generation of culture**. Oxford University Press, 1992.
- CAMPOS, Teofilo Emidio de. **Reconhecimento de Padrões**. Universidade de São Paulo, São Paulo. 2001.
- CHARNIAK, Eugene; MCDERMOTT, Drew. **A Bayesian Model of Plan Recognition**. Massachusetts: Addison-Wesley, 1985.
- CHEVITARESE, D. S. ANNCOM – **Biblioteca de Redes Neurais Artificiais para Alto Desempenho Utilizando Placas de Vídeo**. PUC, Rio de Janeiro, 2010.
- COSMIDES, L.; TOOBY, J. **Cognitive Adaptations for Social Exchange, The Adapted Mind: Evolutionary psychology -- and the generation culture**, Oxford University Press, Nova York, 1992.
- DATA SCIENCE ACADEMY. **Deep Learning Book, 2019**. Capítulo 6 – O Perceptron – Parte 1. 2019. Disponível em: <<http://deeplearningbook.com.br/o-perceptron-parte-1/>>. Acesso em: 12 jul. 2019.
- GOMES, Dennis dos Santos. **Inteligência Artificial: Conceitos e Aplicações**. 2010. Disponível em: <<http://www.olharcientifico.kinghost.net/index.php/olhar/article/view/49/37>>. Acesso em: 18 maio 2019.
- GATTO, Elaine Cecília. **Apostila de Perceptron e Multilayer Perceptron**, São Carlos, 2018.
- GSIGMA. **Introdução a redes neurais e artificiais**. UFSC, Florianópolis, 2002.
- HAUGELAND, John. **Artificial Intelligence: The Very Idea**. Massachusetts: The MIT Press, 1985.
- HAYKIN, S. **Redes neurais: princípios e prática**. Porto Alegre: Bookman, 2001.
- KURZWEIL, Ray. **The Age of Spiritual Machines**. Massachusetts: The MIT Press, 1990.
- LEITE, Tiago M. **Redes Neurais, Perceptron Multicamadas e o Algoritmo Backpropagation**. 2018. Disponível em: <<https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation-eaf89778f5b8>>. Acesso em: 12 jul. 2019.
- NUNES, Benedito. O animal e o primitivo: os outros de nossa cultura. **Novos Cadernos NAEA**, Belém, 2011, ISSN 1516-6481.
- OSÓRIO, F.; PESSIN, G.; FERREIRA, S.; NONNEN-MACHER, V. **Inteligência Artificial para Jogos: Agentes especiais com permissão para matar... e raciocinar!**. In: **VI Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital**, São Leopoldo, 2007.
- PALMIERE, Sérgio Eduardo. **Inteligência Artificial**. 2016. Disponível em: <<https://www.embarcados.com.br/serie/inteligencia-artificial/>>. Acesso em: 12 jul. 2019.
- POOLE, D.; MACKWORTH, A. K.; GOEBEL, R. **Computational Intelligence: A Logical Approach**. Oxford: Oxford University, 1998.
- ROMANI, Lucas. **Aplicação de Redes Neurais Artificiais na Sugestão de Investimentos**. Universidade de Brasília, Brasília, 2017.