

IMPLEMENTAÇÃO E APLICAÇÃO DE UMA REDE NEURAL EM UM FPGA PARA AGRICULTURA

NERCILIO ANTONIO URIAS CAVALCANTE JUNIOR¹, ANTONIO MARCOS
MELO MEDEIROS², BRUNO QUIRINO DE OLIVEIRA³

¹ Graduando em Engenharia Da Computação, Escola Politécnica e de Artes, PUC Goiás, Goiânia - GO,
nercilio.jr@gmail.com

² Eng. Elétrica, Prof. Doutor, Escola Politécnica e de Artes, PUC Goiás, Goiânia - GO.

³ Eng. Elétrica, Prof. Doutor, Escola Politécnica e de Artes, PUC Goiás, Goiânia - GO.

Apresentado no
LII Congresso Brasileiro de Engenharia Agrícola - CONBEA 2023
18 a 21 de outubro de 2023 – Ribeirão Preto - SP, Brasil

RESUMO: A produção agropecuária deixou o mundo rústico para um novo e tecnológico, na qual cada vez mais a agricultura vem utilizando tecnologias de precisão com a ideologia de melhorar a competitividade. São inúmeras as possibilidades de aplicação de inteligência artificial na agricultura. O estudo realizado na pesquisa constata que é possível a implementação de portas lógicas programáveis (FPGA) em conjunto com a inteligência artificial dentro de *hardware* dedicado, reduzindo os custos dos produtores e melhor produtividade. O objetivo do trabalho é apresentar uma análise da aplicação do FPGA e de inteligências artificiais na agricultura, a fim de construir e implementar uma inteligência artificial com FPGA para o controle de sistemas de irrigação e pulverização.

PALAVRAS-CHAVE: inteligência artificial, irrigação, portas lógicas programáveis.

IMPLEMENTATION AND APPLICATION OF A NEURAL NETWORK IN AN FPGA FOR AGRICULTURE

ABSTRACT: Agricultural production left the rustic world for a new and technological one, in which agriculture is increasingly using precision technologies with the ideology of improving competitiveness. There are countless possibilities for applying this technology, the research results show that the FPGA together with artificial intelligence can reduce the costs of producers and improve productivity. The objective of the work was to present a deep analysis of the application of FPGA and artificial intelligence in agriculture with the purpose of facilitating agriculture from a technological point of view, the construction and implementation of an artificial intelligence together with an FPGA for the control of irrigation and spraying systems.

KEYWORDS: agriculture, artificial intelligence, programmable logic gates.

INTRODUÇÃO: Nos últimos 40 anos, o Brasil saiu da condição de importador de alimentos para se tornar um grande provedor para o mundo (GUIMARAES, 2011). Foram conquistados aumentos significativos na produção e na produtividade agropecuárias. O preço da cesta básica, no Brasil, reduziu-se consideravelmente e o país se tornou um dos principais players do agronegócio mundial. Hoje, se produz mais em cada hectare de terra, aspecto importantíssimo para a preservação dos recursos naturais (CORREIA et. Al., 2005). A expectativa é que a população mundial atinja 8,5 bilhões de pessoas em 2030, 16% a mais que em 2016 (LAMAS, 2017). Para as próximas décadas, uma questão primordial relacionada ao planejamento estratégico das organizações públicas e privadas de ciência, tecnologia e inovação (CT&I) é analisar os principais sinais e tendências, antever transformações e

contribuir para o delineamento estratégico da programação de pesquisa, desenvolvimento e inovação (PD&I). Isso é imprescindível para definir o ambiente e o foco de atuação para os próximos anos no intuito de elevar ainda mais o protagonismo da agricultura brasileira. Com a preocupação de estar constantemente conectada a essas transformações e suas implicações em CT&I para a agricultura, a Embrapa tem aprofundado estudos de futuro por meio de uma rede interna de especialistas, vinculados ao Sistema de Inteligência Estratégica. O objetivo do projeto é desenvolver uma inteligência artificial dentro de um FPGA (*field-programmable gate array*) (NASCIMENTO JR & YONEYAMA, 2020), usando a linguagem VHDL e portas lógicas, para o controle de sistemas de irrigação e de pulverização, que serão ativados em épocas específicas do ano.

MATERIAL E MÉTODOS: Para o desenvolvimento da inteligência artificial, primeiro devem ser estabelecidos alguns métodos de busca. Muitos problemas que envolvem IA necessitam de métodos de casamento (*matching*) e busca (*search*) durante o processo de solução. Desta forma, os processos de busca podem ser de encadeamento progressivo (*data driven*) ou retroativo (*goal driven*). No primeiro caso, apresenta-se um conjunto de premissas que irão provar um objetivo (*goal*) como incorreto ou correto (KOVACS, 2006). Já no segundo caso, tem-se um conjunto de objetivos que buscam provar que o conjunto de premissas é correto ou incorreto. Ainda, dependendo de como se executa o controle sobre a busca, pode ser do tipo profundidade primeiro ou largura primeiro, na figura 1 mostra as formas de encadeamento retrativo e progressivo para busca em profundidade.

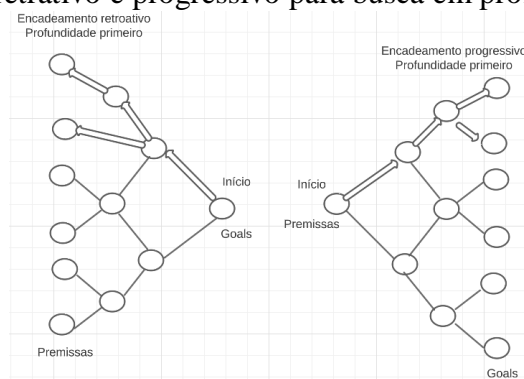


FIGURA 1. Ilustração das formas de encadeamento retroativo e progressivo para o caso de busca em profundidade

RESULTADOS E DISCUSSÃO:

Com diversos artifícios em mão, foi possível estabelecer um modelo teórico da inteligência artificial a ser implementada. Conforme foram apresentados os processos de busca, para esta solução, utilizaremos um encadeamento progressivo. Como explicado anteriormente, um encadeamento progressivo trata-se de um agrupamento de ideias iniciais que irão provar uma intenção. Para o teste da pesquisa, foram utilizadas as seguintes condições iniciais: data, mais especificamente o mês (para comparação com a base de dados); plantação em questão (o que está sendo plantado); tempo na propriedade agrícola (chuvoso, ensolarado etc.); tempo em execução no sistema (quanto tempo os sistemas devem bombear água para a plantação). Baseado nestas premissas, pode-se criar o grafo para a busca em encadeamento progressivo, após a definição das ideias iniciais, foram projetadas as regras de controle, tais regras determinarão quando e quanto o sistema será iniciado. Como exemplo, foi implementado algumas regras genéricas para uma plantação composta por três tipos de plantas diferentes.

1. Regra de intervalo fixo: Irrigar todas as plantas a cada três dias; verificar se a diferença entre a data atual e a data da última irrigação é igual a três dias.

- Regra de intervalo variável por planta: planta 1: irrigar a cada dois dias; planta 2: irrigar a cada quatro dias.

Tais regras foram convertidas em lógica digital usando portas lógicas disponíveis em VHDL, como portas AND, OR, NOT, entre outras (CULLEN et. al., 2018). Cada regra pode ser representada como uma combinação de portas lógicas para tomar decisões com base nas entradas (COCOLO, 2015). Como exemplo, foi utilizado uma abordagem simplificada, onde apenas uma regra de controle é implementada: a planta precisa de água se pelo menos uma das duas primeiras posições do vetor `current_plant` estiverem definidas como '1'. O intervalo de irrigação é definido como 2 dias. Essa implementação básica pode ser expandida para incorporar mais regras e condições usando portas lógicas adicionais, à medida que a complexidade das regras aumenta, figura 2.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity IrrigationControl is
5  port (
6      clk : in std_logic;
7      current_date : in std_logic_vector(7 downto 0);
8      current_time : in std_logic_vector(7 downto 0);
9      current_plant : in std_logic_vector(3 downto 0);
10     irrigation_signal : out std_logic
11 );
12 end entity IrrigationControl;
13
14 architecture behavioral of IrrigationControl is
15     constant WATERING_INTERVAL : integer := 2; -- Intervalo de dias entre as irrigações
16
17     signal last_watered_date : std_logic_vector(7 downto 0);
18     signal irrigate_plant : std_logic;
19     signal plant_needs_water : std_logic;
20
21 begin
22     process (clk)
23     begin
24         if rising_edge(clk) then
25             -- Verifica se é hora de irrigar a planta
26             if current_date /= last_watered_date and current_date(current_time'high - 1 downto 0) = "000000" then
27                 irrigate_plant <= '1';
28             else
29                 irrigate_plant <= '0';
30             end if;
31
32             -- Determina se a planta precisa de água com base em regras específicas
33             plant_needs_water <= current_plant(0) or current_plant(1); -- Exemplo de regra simples
34
35             -- Atualiza a data da última irrigação
36             if irrigate_plant = '1' then
37                 last_watered_date <= current_date;
38             end if;
39         end process;
40
41     -- Sinal de irrigação ativado se for hora de irrigar e a planta precisa de água
42     irrigation_signal <= '1' when irrigate_plant = '1' and plant_needs_water = '1' else '0';
43
44 end architecture behavioral;
45
46
47

```

FIGURA 2. Exemplo do código de implantação da regra citada.

O bloco de controle denominado "behavioral", e define o comportamento do módulo *IrrigationControl*. Ele contém três sinais internos: `last_watered_date`, `irrigate_plant` e `plant_needs_water`, dentro do bloco de processo, que é sensível ao sinal de clock (clk), o código verifica se é hora de irrigar a planta. Ele compara a data atual (`current_date`) com a data da última irrigação (`last_watered_date`) e verifica se a hora atual (`current_time`) é igual a "000000". Se essas condições forem atendidas, o sinal `irrigate_plant` é definido como '1'; caso contrário, é definido como '0'. Em seguida, o código determina se a planta precisa de água com base em regras específicas. Neste exemplo, ele verifica se o primeiro ou o segundo bit do sinal `current_plant`. Na figura 3 é mostrado os dados da simulação com as variáveis.

Signal Name	Value	Direction	Type
clk	U	Signal In	Signal In
current_date	UUUUUUUU	Signal In	Signal In
(7)	U	Signal In	Signal In
(6)	U	Signal In	Signal In
(5)	U	Signal In	Signal In
(4)	U	Signal In	Signal In
(3)	U	Signal In	Signal In
(2)	U	Signal In	Signal In
(1)	U	Signal In	Signal In
(0)	U	Signal In	Signal In
current_time	UUUUUUUU	Signal In	Signal In
current_plant	UUUU	Signal In	Signal In
irrigation_signal	U	Signal Out	Signal Out
last_watered_date	UUUUUUUU	Signal Internal	Signal Internal
irrigate_plant	U	Signal Internal	Signal Internal
plant_needs_water	U	Signal Internal	Signal Internal
WATERING_INTERVAL	2	Constant Internal	Constant Internal

FIGURA 3. Bloco de variáveis no simulador

Após isso, o programa foi gravado em um FPGA Altera, modelo EPM7032SLC44-10 e conectado à uma base de dados que contém o nome das plantas, datas para pulverização e irrigação. Na figura 3, o hardware desenvolvido durante a fase de implementação da IA no FPGA.

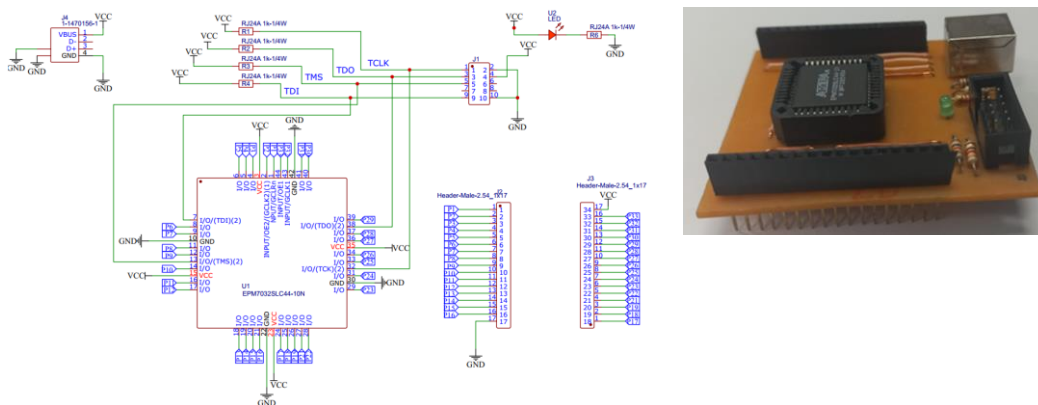


FIGURA 3. Circuito e placa desenvolvida para a implementação da IA no FPGA.

CONCLUSÕES: Em conclusão, este artigo científico explorou a implementação e aplicação de uma rede neural em um FPGA para a área da agricultura. Através desse estudo, foi possível observar os benefícios dessa abordagem, como a capacidade de processamento em tempo real e a eficiência energética proporcionada pela utilização de um FPGA. A rede neural implementada mostrou-se capaz de realizar tarefas complexas, como o reconhecimento de padrões e a tomada de decisões, que são essenciais para otimizar a produção agrícola. Além disso, a flexibilidade do FPGA permitiu a adaptação e atualização da rede neural de acordo com as necessidades específicas do ambiente agrícola. Com isso, conclui-se que a implementação de redes neurais em FPGAs apresenta um potencial significativo para melhorar a eficiência e a precisão dos sistemas agrícolas, contribuindo para o avanço da agricultura de precisão e sustentável.

AGRADECIMENTOS: Os autores agradecem a PUC Goiás pelo uso dos laboratórios para a montagem e testes do FPGA

REFERÊNCIAS:

- CORREIA, G. R.; Rocha, H. R. O.; Rissimo, S. D. Automação de sistemas de irrigação com monitoramento via aplicativo web, **Revista Engenharia na Agricultura – REVENG** Engenharia na agricultura, viçosa - MG, V.24 N.4, JULHO / AGOSTO 2005, 314-325p.
- GUIMARAES, V. G. **Automação e Monitoramento Remoto de Sistemas de Irrigação Visando Agricultura Familiar**, [Distrito Federal]. xiii, 81p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2011). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia, 2011.
- LAMAS, F. M. **A tecnologia na agricultura. Embrapa Agropecuária Oeste**. EMBRAPA, Brasília, DF, 2017. Disponível em: < <https://www.embrapa.br/busca-de-noticias/-/noticia/30015917/artigo-a-tecnologia-na-agricultura> > Acesso em: 08 junho 2023.
- COCOLO, C. **Implementação em FPGA de uma rede neural de Hopfield**, 2015. 52 f. Trabalho de Conclusão de Curso – Departamento de Engenharia Elétrica e de Computação. Universidade de São Paulo – Escola de Engenharia de São Carlos.
- CULLEN, J.; GERBETH, A.; DOROJEVETS, M. FPGA-based Satisfiability Filters for Deep Packet Inspection. 2018, **IEEE Long Island Systems, Applications and Technology Conference (LISAT)**. Farmingdale.: 2018. p. 1-4
- KOVACS, Zsolt Laszlo. **Redes neurais artificiais: fundamentos e aplicações - um texto básico**. 4. ed. São Paulo: Livraria da Física, 2006.
- NASCIMENTO JR., C.; YONEYAMA, T. **Inteligência artificial em controle e automação**. São Paulo: Blucher, 2000.