Logotipo

Descrição gerada automaticamente com confiança baixa

**Universidade do Vale do Itajaí**

**Computação  
Trabalho de Conclusão de Curso (TCC)**

|  |
| --- |
| PROPOSTA DE TRABALHO |

|  |  |
| --- | --- |
| **Nome do acadêmico:** | João Heitor Zabel da Rocha |
| **Código de Pessoa:** | 7719086 |
| **E-mail de Contato:** | joaoheitorzabel1808@gmail.com |
| **Telefone(s) de Contato:** | (47) 98852-0136 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome do Professor Orientador:** | | Felipe Viel | |
| **E-mail de Contato:** | viel@univali.br | | |
| **Telefone(s) de contato:** | +55 47 9606-0737 | | |
| **Dia da semana e horário de Atendimento ao acadêmico:** | | |  |
|  | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Curso:** | **( )** | | **Ciência da Computação** | | | | | **( X )** | **Engenharia de Computação** | | |
| **Modalidade do Trabalho:** | | | | **( X ) Monografia** | | | **( ) Produto** | | | | **( ) Artigo** |
| **Modalidade da Orientação:** | | | | | **(X) Presencial** | | | | | **( X) Remota** | |
| **Área do Trabalho:** | | Sistemas embarcados | | | | | | | | | |
|  | | | | | | |  | | | | |
| *Considerando verídicas as Informações fornecidas neste formulário,*  *encaminhamos a Proposta de Trabalho para avaliação.* | | | | | | | | | | | |
| **Assinatura do acadêmico:** | | | | | |  | | | | | |
| **Assinatura do Professor Orientador:** | | | | | |  | | | | | |
| *Itajaí, 19/outubro/2023* | | | | | | | | | | | |



Desenvolvimento de UMA BIBLIOTECA DE COMPONENTES PARA CONSTRUÇÃO de redes neurais Convolucionais em FPGA

João Heitor Zabel da Rocha

Outubro / 2023

Orientador: Felipe Viel, Mestre

Área de Trabalho: Inteligência Artificial

1. Introdução

O aprendizado de máquina é definido por Tom Mitchel (1997) como a capacidade de um programa de aperfeiçoar a sua habilidade de exercer uma tarefa a partir de experiências. Essa definição encapsula a essência de que, a partir dos dados de execuções e resultados anteriores, é possível a criação de curvas de aprendizado e parametrizações que reflitam os conhecimentos obtidos durante o treinamento. Esses conhecimentos, quando consolidados, podem ser aplicados em processos para uma tomada de decisão ou até na antecipação de um resultado.

O aprendizado profundo, é considerado como aprendizado de máquina, só que apesar de também ser treinado, possui a capacidade de geração de conhecimentos de forma intuitiva e não supervisionada. Sua estrutura analisa os dados a partir de várias camadas de neurônios que além de interconectados, também interagem entre si. Com cada conexão tendo um peso definido, os neurônios passam as informações para os seus vizinhos em camadas superiores, transformando os dados em informações cada vez menos abstratas e mais específicas. (THEOBALD, 2017).

Algumas das aplicações dos fundamentos da aprendizagem profunda são direcionadas a atividades de visão computacional. Caracterizada pela presença de camadas convolucionais, responsáveis por obter as características relevantes de uma imagem, as redes neurais convolucionais (do inglês Convolutional Neural Network – CNN) utilizam de fundamentos do aprendizado profundo para a aprendizagem e aplicação em reconhecimento de escritas, falas e objetos (VOULODIMOS, 2017).

À medida que as arquiteturas de redes neurais convolucionais evoluíram, houveram mudanças significativas nas suas formas de desenvolvimento e implementação. Apesar de comumente implementadas em hardware com a utilização de unidades de processamento gráfico (GPU), a baixa eficiência energética apresentada propiciou a busca de alternativas. Componentes como o Field Programming Gateway Array (FPGA) se tornaram viáveis e muito mais eficientes (ZHANG, 2015, p.1).

Além do baixo consumo de energia, Zhang (2015) cita que características como a reprogramabilidade, baixa latência e o rápido desenvolvimento, fazem do FPGA ser ideal para atividades computacionalmente intensivas, como as redes neurais convolucionais. Porém, com o crescimento da complexidade a dificuldade de design também aumenta em circuitos deste tipo. Abordagens em mais alto nível, como o descrito por Aarrestad (2021) reiteram a importância da criação de ferramentas para a viabilização de um desenvolvimento rápido e confiável de CNNs em FPGA.

Com isso, uma biblioteca de componentes facilitaria o design e a implementação. Ao criar uma biblioteca de componentes específicos para o processamento de redes neurais convolucionais em FPGAs, os desenvolvedores teriam acesso a blocos de construção pré-definidos e otimizados para tarefas comuns. Além de também contribuir em pesquisas e projetos de fomento estadual e federal na área de aceleradores em hardware, como o Laboratório de Sistemas Embarcados e Distribuídos (LEDS), onde suas pesquisas e projetos tem como foco, em especial, a aceleração em hardware de modelos e arquiteturas de aprendizado de máquina profundo, incluindo as redes neurais convolucionais.

* 1. OBJETIVOS
     1. Objetivo Geral

Desenvolver componentes necessários para a construção de redes neurais convolucionais em VHDL ou Verilog.

* + 1. Objetivos Específicos

1. Analisar técnicas de desenvolvimento de redes neurais em FPGA.
2. Descrever arquiteturas de referência de redes neurais convolucionais
3. Disponibilizar uma interface de configuração e geração de modelos.
4. Disponibilizar uma arquitetura funcional como prova de conceito.
5. Avaliar a geração correta de arquiteturas de CNN para FPGA alvo por meio de métricas de acurácia e consumo de recursos lógicos do FPGA.
   1. Plano de Trabalho
6. Análise de diferentes arquiteturas de Redes neurais Convolucionais: auxiliará na elaboração de como deverão ser feita correta implantação de uma CNN em FPGA.
   1. Pesquisa: busca das arquiteturas de redes neurais convolucionais mais utilizadas e o seus funcionamentos
   2. Análise do comportamento das camadas: identificar quais os comportamentos esperados em cada camada das redes neurais convolucionais e seus requisitos de funcionamento.
   3. Elaboração dos blocos lógicos necessários: a partir dos comportamentos esperados, será possível dividir em pequenos blocos lógicos de funcionamento simples.
7. Desenvolvimento dos blocos lógicos: desenvolvimento em linguagem de descrição de Hardware dos blocos lógicos necessários.
   1. Desenvolvimento: desenvolvimento dos blocos lógicos em linguagem de descrição de hardware
   2. Validação dos blocos desenvolvidos: Verificar o correto funcionamento dos blocos desenvolvidos e, caso necessário, fazer os ajustes.
8. Desenvolvimento em alto nível: desenvolvimento da manipulação dos blocos lógicos desenvolvidos e alocação dos arquivos corretos, para a geração completa da CNN.
   1. Desenvolvimento: criar funções em linguagem de alto nível que recebam parâmetros e construam os arquivos necessários para a implantação de CNNs em FPGA.
   2. Validação: Verificar o correto funcionamento dos arquivos gerados.
9. Desenvolvimento de interface: criação de uma interface que facilite ao usuário a entrada dos parâmetros
10. Verificação final: testar a correta geração dos arquivos necessários para a implantação de uma CNN a partir da entrada dos parâmetros pela interface criada.
11. Análise de Resultados: analisar os resultados obtidos
12. Monografia: a partir dos resultados obtidos, redigir a monografia e a partir dela, demonstrar os resultados obtidos.
    * 1. Cronograma

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quadro 1. Cronograma de execução do TCC II   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Atividade** | **Janeiro**  **2024** | **Fevereiro**  **2024** | **Março**  **2024** | **Abril**  **2024** | **Maio**  **2024** | **Junho**  **2024** | | 1.a Pesquisa |  | X X |  |  |  |  | | 1.b Análise do comportamento das camadas |  | X |  |  |  |  | | 1.c Elaboração dos Blocos lógicos necessários |  | X | XX |  |  |  | | 2.a Desenvolvimento dos blocos lógicos |  |  | X X | X X |  |  | | 2.b Validação dos blocos lógicos desenvolvidos |  |  |  | X X |  |  | | 3.a Desenvolvimento em alto nível |  |  |  |  | X X X X | X X | |
| Quadro 2. Cronograma de execução do TCC III   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Atividade** | **Julho**  **2024** | **Agosto**  **2024** | **Setembro**  **2024** | **Outubro**  **2024** | **Novembro**  **2024** | **Dezembro**  **2024** | | 3.b Validação do desenvolvimento em alto nível | XXXX |  |  |  |  |  | | 4 Desenvolvimento de Interface |  | XXXX | XX |  |  |  | | 5 Verificação Final |  |  | XX |  |  |  | | 6 Análise de resultados |  |  |  | XX |  |  | | 7.a Monografia |  |  |  | XX | XX |  | |
| * 1. Análise de riscos  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Quadro 3. Análise de riscos   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Risco** | **Probabilidade** | **Impacto** | **Gatilho** | **Plano de contingência** | | 1. Dispositivo alvo não comporte arquitetura de validação | Baixa | Médio | Dispositivo alvo não possui recursos necessários para a síntese da arquitetura de validação física. | Gerar a arquitetura para um dispositivo com mais recursos e validar por meio de simulação. | | | |

1. Fundamentação teórica

REDES NEURAIS CONVOLUCIONAIS

INTELIGÊNCIA ARTIFICIAL

DEEP LEARNING

Descrever camadas totalmente conectadas e funções de ativação sigmoide

REDES NEURAIS CONVOLUCIONAIS

Foram a partir da inspiração de estudos sobre o funcionamento do córtex visual de felinos que as redes neurais convolucionais (Convolutional Neural Network - CNN) foram primeiramente idealizadas (GÉRON, 2021). Os estudos demonstraram que os neurônios desta parte do cérebro apresentam pequenos campos receptivos e, portanto, significando que cada um deles reage a uma pequena e suscinta característica presente no campo visual. Além disso, foi observada a existência de neurônios com o campo receptivo maior que recebiam como entrada as saídas dos de nível inferior, gerando uma hierarquia que combinava as diferentes observações.

Foi com base nestes preceitos biológicos que a rede neural para a identificação de padrões Neocognitron de Fukushima (1980) foi criada. Onde uma organização de hierárquica de duas camadas de células, uma de simples e uma de complexas, permitiu a identificação de caracteres em qualquer espaço da imagem. Além de obter a capacidade de autoaprendizagem, se organizando conforme os repetidos estímulos à camada de entrada fossem acontecendo.

Um grande marco no desenvolvimento de redes neurais convolucionais foi a arquitetura LeNet 5 (LECUN; BOTTOU; BENGIO, 1998) que sintetizou a possibilidade de classificar padrões multidimensionais, neste caso específico, de caracterizar caracteres manuscritos com pouco pré-processamento. Além de utilizar das já descritas camadas totalmente conectadas e funções de ativação sigmoide, introduziu dois elementos que serão descritos em seguida: as camadas convolucional e de pooling.

CAMADAS CONVOLUCIONAIS

As camadas convolucionais são as que diferem esse tipo de rede neural de outras que agem sobre imagens, nelas os neurônios na primeira camada não estão interligados a cada pixel na imagem de entrada, mas somente aos pixels do seu campo receptivo (GÉRON, 2021), ou seja, de somente uma área presente na imagem. Já os neurônios da segunda camada estão conectados somente a uma parte da camada anterior (Conforme observável na figura 1).

Essa arquitetura permite que pequenas características sejam observadas nas camadas mais baixas e que essas sejam observadas de um nível maior de abstração nas camadas superiores, proporcionando o mapeamento das características presentes nas imagens e conseguindo identificar padrões com base nas suas observações.

Diagrama

Descrição gerada automaticamente com confiança média

Além disso, neurônios responsáveis por reagir a características específicas como a detecção de linhas horizontais também foram mapeados.

foram observados outros dois pontos correlacionados ao funcionamento de uma CNN, foram mapeados neurônios que reagiam apenas à característica de linhas horizontais, e outros para linhas não horizontais. Comprovando que existem

pequenos campos receptivos locais dos neurônios desta parte do cérebro tinham como função, a reação apenas a pontos isolados e localizados do campo visual.

As

a partir de observaçõessemicondutores baseados em blocos lógicos configuráveis que permitem a programação de suas conexões internas (Referência Esteban). Apesar de ser um componente completamente manufaturado, permitem a implementação de designs programáveis através de seus blocos lógicos, as Lookup table, e seus blocos de entrada e saída.

APLICAÇÕES

FPGA

Os Field-Programmable gate arrays (FPGAs) são dispositivos semicondutores baseados em blocos lógicos configuráveis que permitem a programação de suas conexões internas (Referência Esteban). Apesar de ser um componente completamente manufaturado, permitem a implementação de designs programáveis através de seus blocos lógicos, as Lookup table, e seus blocos de entrada e saída.

Os blocos lógicos de

Explicar como O FPGA FUNCIONA, de forma sucinta

Os blocos de entrada e saída são os responsáveis por serem as interfaces de entrada e saída dos dados do FPGA para dispositivos externos, sua composição básica é de buffers I/O bidirecionais e registradores dos inputs, outputs, sinal de habilitação de saída e da taxa de transferência bidirecional de dados.

VHDL

REFERÊNCIAS

GÉRON, Aurélien. Mãos à obra: aprendizado de máquina com Scikit-Learn, Keras & TensorFlow: conceitos, ferramentas e técnicas para a construção de sistemas inteligentes. 2. ed. São Paulo: Alta Books, 2021.

MITCHELL, Tom M. Machine Learning. 1. ed. Portland: McGraw-Hill Education, 1997.

THEOBALD, Oliver. Machine Learning for Absolute Beginners: A Plain English Introduction. 3. ed. Scatterplot Press, 2017.

DESCHAMPS, Jean-Pierre; SUTTER, Gustavo D.; CANTÓ, Enrique. Guide to FPGA Implementation of Arithmetic Functions. 1. ed. Madrid: Springer Science, 2012.

BAILEY, Donald G. Design for Embedded Image Processing on FPGAs. 1. ed. Singapore: John Wiley & Sons, 2011.

KHAWAJA, S. G.; TARIQ, R.; AKRAM, M. U.; HUSSAIN, F. (2022). Reconfigurable Architecture for Real-time Decoding of Canonical Huffman Codes. In: 2nd International Conference on Digital Futures and Transformative Technologies. DOI: 10.1109.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, v. 36, n. 4, p. 193–202, abr. 1980.

NI SIYU; HUANG CHAO; GENGSHENG CHEN. A layer-based structured design of CNN on FPGA. In: 2017 IEEE 12th International Conference on ASIC (ASICON). Disponível em: https://ieeexplore.ieee.org/document/8252656 Acesso em: 28 nov. 2023.

ZHANG, C. et al. Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks. Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA ’15, 2015.

VOULODIMOS, A. et al. Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, v. 2018, p. 1–13, 2018.

AARRESTAD, T. et al. Fast convolutional neural networks on FPGAs with hls4ml. Machine Learning: Science and Technology, v. 2, n. 4, p. 045015, 16 jul. 2021.

LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learnin g applied to document recognition[J/OL]. Proceedings of the IEEE, 1998: 2278-2324. http://dx.doi.org/10.1109/5.726791. DOI:10.1109/5. 726791.