CNN EM HARDWARE DEDICADO: COMPARAÇÃO DE IMPLEMENTAÇÕES EM PONTO FLUTUANTE DE 32 BITS E INTEIROS DE 8 BITS

VANESSA ALDRIGHI[1]; DENIS MAASS[1]; RUHAN CONCEIÇÃO[1]; MARCELO PORTO[1]; LUCIANO AGOSTINI[1]

[1]Universidade Federal de Pelotas (UFPel) – Video Technology Research Group (ViTech)
{vanessa.a, dlmaass, radconceicao, porto, agostini} @inf.ufpel.edu.br

1. INTRODUÇÃO

Redes Neurais Convolucionais (CNNs) são fundamentais na área de inteligência artificial e visão computacional, sendo capazes de realizar diversas tarefas como detecção de objetos (REN *et al.*, 2017), reconhecimento facial (SCHROFF *et al.*, 2015) e classificação de imagens (KRIZHEVSKY *et al.*, 2012). Contudo, as CNNs enfrentam um desafio significativo: seu elevado custo computacional. As operações de convolução e de multiplicação de matrizes nas camadas totalmente conectadas exigem muitas operações de multiplicação-acumulação (MAC) para processar uma única imagem, resultando em alta latência e consumo de energia, o que dificulta a implementação de CNNs em sistemas com recursos restritos.

Tradicionalmente, o treinamento e a inferência de CNNs são realizados com aritmética de ponto flutuante de 32 bits (padrão IEEE 754, ou *float32*). Embora o float32 ofereça alta precisão, sua implementação em hardware tem um custo elevado em área e energia, como aponta o estudo HOROWITZ (2014), além de introduzir problemas práticos de inconsistência entre plataformas, devido a pequenas diferenças no arredondamento de operações entre dispositivos (CONCEIÇÃO *et al.*, 2025).

Para mitigar esses problemas, a quantização de modelos se tornou uma técnica essencial. A quantização consiste em converter os pesos e as ativações da rede de float32 para formatos de inteiros com menor largura de bits, como 8 bits (int8), por exemplo. Estudos como o de WU *et al.* (2016) demonstram que em alguns casos é possível realizar essa conversão com uma perda de acurácia muito pequena (frequentemente inferior a 1%), enquanto se obtém uma redução significativa no consumo de memória e no custo computacional.

Com o objetivo de avaliar o impacto da quantização em hardware, este trabalho apresenta o projeto, a implementação e a análise comparativa de dois blocos fundamentais de uma CNN: uma camada convolucional e uma camada totalmente conectada. Ambas as arquiteturas foram descritas em VHDL e sintetizadas para um Circuito Integrado de Aplicação Específica (Application-Specific Integrated Circuit – ASIC) usando uma tecnologia de 40 nm, com implementações em *float32* e *int8*. O objetivo é quantificar de forma precisa os ganhos em área e potência obtidos com a quantização, validando sua eficácia para o desenvolvimento de hardware de baixo consumo.

2. METODOLOGIA

2.1. Arquitetura de Redes Neurais Convolucionais

As CNNs processam dados visuais em camadas sequenciais. A principal, a camada convolucional, aplica filtros à imagem de entrada para gerar mapas de características que detectam padrões. Nesta operação (Equação 1), a saída $yk_{i,j}$ é calculada a partir da soma ponderada entre os canais de entrada x^c e os pesos do filtro $w^{k,c}$, e seguida pela ativação, que pode ser uma Unidade Linear Retificada (*Rectified Linear Unit* – ReLU). Na sequência, uma camada de *pooling* reduz a dimensão desses mapas, diminuindo o custo computacional (LECUN *et al.*, 1998).

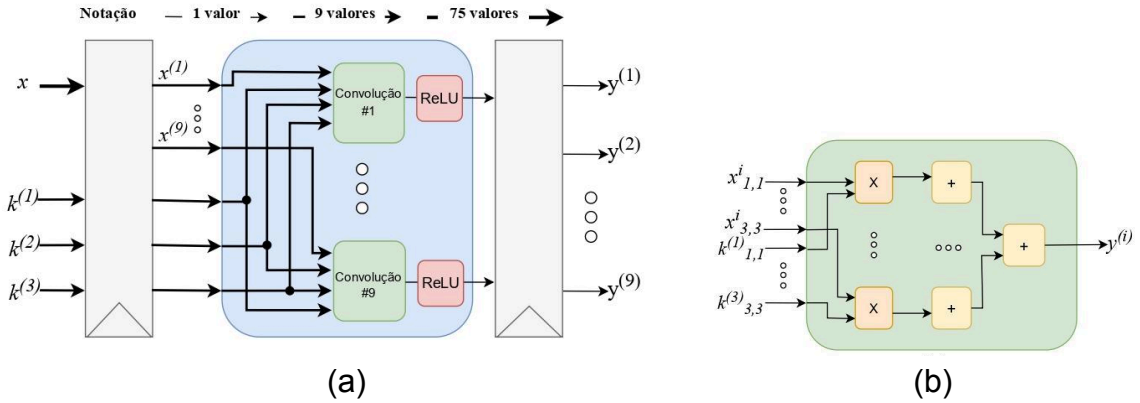
$$yk_{i,j} = max(0, \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \sum_{n=o}^{N-1} w_{m,n}^{k,c} . x_{i+m,j+n}^{c}) \quad (1)$$

Finalmente, a camada totalmente conectada utiliza as características extraídas para a classificação ou regressão, onde a saída y é obtida multiplicando o vetor de entrada x pela matriz de pesos W, também com ativação ReLU (Equação 2), associando os padrões aprendidos aos resultados finais.

$$y = max(0, W.x) \quad (2)$$

2.2. Projeto de Hardware e Metodologia de Avaliação

Para a análise comparativa, dois módulos de hardware foram projetados em VHDL, representando as camadas de maior custo computacional de uma CNN. A Figura 1 (a) representa a implementação de uma camada convolucional, que foi projetado para aplicar três filtros de dimensões 3x3 sobre uma janela de uma imagem de entrada de 5x5 pixels. A estrutura interna do bloco de convolução está apresentada na Figura 1 (b) e consiste em vinte e sete multiplicadores que operam em paralelo, seguidos por uma árvore de somadores para acumular os produtos, gerando um único pixel de saída.

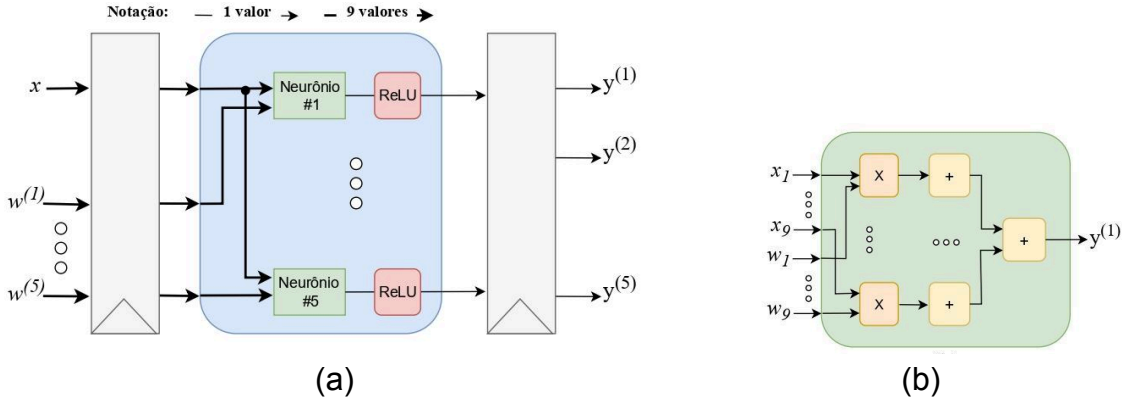


(a) (b)

Figura 1 – (a) Diagrama de blocos de alto nível da camada convolucional e (b) diagrama interno do Bloco de Convolução.

Por sua vez, a Figura 2 (a) descreve a implementação de uma camada totalmente conectada. A arquitetura é composta por cinco neurônios, cada um recebendo um

vetor de nove entradas. Como detalhado na Figura 2 (b), cada neurônio possui nove multiplicadores e uma árvore de somadores para calcular a soma ponderada das entradas. Os cinco neurônios operam em paralelo para maximizar o desempenho.



(a) (b)

Figura 2 – Diagrama de blocos de alto nível da camada totalmente conectada e (b) diagrama interno do Bloco de Neurônio.

Ambas as arquiteturas foram descritas em VHDL e implementadas em duas versões: *float32* e *int8*. A avaliação ocorreu via síntese lógica para um ASIC TSMC 40 nm com a ferramenta Cadence RTL Compiler, tendo como alvo a frequência de 100 MHz. As métricas de área (em contagem de portas NAND2 equivalentes) e dissipação de potência (em mW) foram extraídas dos relatórios de síntese da ferramenta.

3. RESULTADOS E DISCUSSÃO

A síntese dos módulos em ASIC permitiu quantificar e comparar os custos de hardware das duas abordagens. A Tabela 1 apresenta os resultados de área (em contagem de portas equivalentes) e dissipação de potência (em mW) para cada módulo e versão.

Módulo	Ponto Flutuante (32-bit)		Inteiro (8-bit)		Redução (%)	
	Potência (mW)	Área (Gates x10³)	Potência (mW)	Área (Gates x10³)	Potência	Área
Camada Convolucional	293,94	1254,30	20,44	100,67	93,04	91,97
Camada Totalmente Conectada	695,23	1197,94	4,23	20,11	99,39	98,31
Total	**989,17**	**2449,24**	**24,67**	**120,78**	**97,50**	**95,06**

Tabela 1 – Resultados de Síntese Comparativos com frequência de 100 MHz. Fonte: Autoria própria.

Os resultados da Tabela 1 confirmam que a quantização para *int8* promove uma redução drástica nos recursos de hardware. A economia total, superior a

95% em área e 97% em potência, provém da menor complexidade dos circuitos de inteiros, que dispensam as custosas unidades de processamento de ponto flutuante (alinhamento de expoente, multiplicação de mantissa e normalização). Essa simplicidade estrutural valida a abordagem para o desenvolvimento de hardware energeticamente eficiente, viabilizando a implementação de CNNs em dispositivos de borda (*edge*) e sistemas embarcados com restrições severas de energia.

4. CONCLUSÕES

Este trabalho demonstrou que a quantização para inteiros de 8 bits em hardware ASIC dedicado para camadas de CNNs resulta em ganhos de eficiência superiores a 95% em área e 97% em potência, quando comparada à implementação tradicional em ponto flutuante. Trabalhos futuros incluem a integração dos módulos em uma rede funcional para testes de acurácia e a otimização da arquitetura com técnicas de paralelismo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CONCEIÇÃO, R.; PORTO, M.; PENG, W. H.; AGOSTINI, L. Cross-Platform Neural Video Coding: A Case Study. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), Londres, 2025. Anais... Londres: IEEE, 2025. p. 1-5.

HOROWITZ, M. Computing's energy problem (and what we can do about it). In: IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC), San Francisco, 2014. Digest of Technical Papers... San Francisco: IEEE, 2014. p. 10-14.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), Lake Tahoe, 2012. Anais... Lake Tahoe: NeurIPS, 2012. v. 25, p. 1097-1105.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, New York, v. 86, n. 11, p. 2278-2324, 1998.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, New York, v. 39, n. 6, p. 1137-1149, 2017.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. FaceNet: A unified embedding for face recognition and clustering. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), Boston, 2015. Proceedings... Boston: IEEE, 2015. p. 815-823.

WU, J.; LENG, C.; WANG, Y.; HU, Q.; CHENG, J. Quantized Convolutional Neural Networks for Mobile Devices. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), Las Vegas, 2016. Proceedings... Las Vegas: IEEE, 2016. p. 4820-4828.