Machine Translated by Google

Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

Artigo de destaque: Arquitetura CNN para sistemas de detecção de intrusão

Energia eficiente CNN baseada em CPU + FPGA Arquitetura para Intrusão Sistemas de Detecção

Lucas A. Maciel
Pontifícia Universidade Católica. de Minas Gerais, Brasil

Matheus A. Souza e Henrique C. Freitas Pontifícia Universidade Católica. de Minas Gerais, Brasil

Resumo—As aplicações de Machine Learning têm sua viabilidade fortemente ligada à capacidade das arquiteturas de computadores em oferecer alto desempenho e eficiência energética. Embora diferentes arquiteturas possam oferecer alto desempenho computacional, elas podem carecer de eficiência energética, o que é crucial para aplicações baseadas no consumidor. Por exemplo, os sistemas de detecção de intrusões podem usar técnicas de aprendizado de máquina para monitorar o tráfego de rede e identificar possíveis atividades maliciosas. Estes sistemas estão constantemente ativos em dispositivos como firewalls, reforçando a necessidade de eficiência energética, por exemplo, em casas inteligentes e veículos autónomos. Field-Programmable Gate Array (FPGA) pode oferecer melhor eficiência energética do que outras arquiteturas. Considerando isso, projetamos e avaliamos uma Rede Neural Convolucional baseada em CPU+FPGA para Sistemas de Detecção de Intrusão. Implantamos nossa estratégia em uma plataforma heterogênea de CPU (Intel Xeon) + FPGA (Arria 10). Em seguida, comparamos a arquitetura proposta com sua respectiva versão de software paralelo para avaliação de potência, energia e desempenho. O conjunto de dados NSL-KDD foi usado para benchmarking de detecção de intrusão. Os resultados de eficiência ene

O RÁPIDO AVANÇO das técnicas de Aprendizado de Máquina (ML) em diversas áreas de pesquisa, como classificação de dados [1] e detecção de intrusão [2], pode produzir soluções viáveis para cenários complexos e heterogêneos baseados em consumidores, como casa inteligente [3] e veículos autônomos [4]. Devido ao aumento do tráfego gerado por dispositivos tradicionais e Internet-of-Things (IoT) [5], são necessárias arquiteturas de ML com eficiência energética. Além disso, aumentar a precisão dos resultados e gerá-los em cenários com restrição de tempo são dois dos principais objetivos originais dos algoritmos de ML.

O grande volume de dados exige atenção em relação à segurança, o que intensifica os investimentos em estratégias de monitoramento. Por exemplo, Sistemas de Detecção de Intrusões (IDS) podem identificar possíveis ataques, informar seus administradores que invasões podem ocorrer ou até mesmo impedir a chegada de pacotes maliciosos [6]. Neste contexto, o IDS também pode empregar métodos de aprendizagem profunda [7] para

Identificador de Objeto Digital 10.1109/MCE.2023.Número Doi
Data de publicação 00 xxxx 0000; data da versão atual 00 xxxx0000

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Machine Translated by Google
Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

promover a avaliação de novos registros de acordo com semelhanças entre os dados já conhecidos.

Um dos algoritmos mais utilizados para classificação, processamento e reconhecimento de imagens de dados na aprendizagem supervisionada é a Rede Neural Convolucional. (CNN) [8], [9], um algoritmo de Deep Learning. Ele processa dados em várias camadas de neurônios conectados onde as camadas ocultas podem representar convoluções, pooling ou processos totalmente conectados.

Um modelo CNN geralmente consiste em dois estágios. O primeiro é feedforward para processos de reconhecimento de dados. Este último é atrasado para atualizar os pesos líquidos durante o treinamento. Além disso, apresenta componentes responsáveis por suas operações, tais como (i) o extrator de recursos, que identifica e seleciona o atributos com as informações essenciais dentro do "mapas de características" dos dados de entrada da rede; e (ii) o classificador, que decide a probabilidade de um determinado entrada pertencente às categorias avaliadas.

As aplicações CNN realizam operações matemáticas que exigem computação de alto desempenho com eficiência energética. Assim, arquiteturas heterogêneas eficientes, por exemplo, Unidade Central de Processamento (CPU) + Matriz de Portas Programáveis em Campo (FPGA), podem atingir valores mais baixos. tempo de execução e energia. Neste artigo, nosso objetivo é projetar e avaliar uma rede neural convolucional (CNN) para uma CPU híbrida (Intel Xeon) + FPGA (Intel Plataforma Arria 10), usando Open Computing Language (OpenCL) em comparação com uma versão OpenMP multithread baseada em CPU. Este processamento híbrido se encaixa bem com firewalls de próxima geração, que empregam inspeções de pacotes. Esses firewalls incluem dispositivos como o FPGA além da CPU tradicional em suas caixas, que podem receber rapidamente atualizações para novas técnicas [10], [11]. Além disso, eles podem implantar algoritmos de aprendizado de máquina para detectar ataques de dia zero, que os sistemas de detecção padrão não conseguem identificar [12]. Além disso, usando OpenCL como uma síntese de alto nível (HLS) permite uma estratégia mais eficiente e portátil processo de desenvolvimento de software para aceleração de hardware, fornecendo uma interface de programação padrão para sistemas computacionais heterogêneos.

Em resumo, nossas contribuições são:

- Uma estratégia de eficiência energética para implantar uma arquitetura CNN para detecção de intrusão de rede projetada para um sistema Hybrid Multi-Chip Package.
- Uma síntese de alto nível mais geral e fácil de usar código, baseado em OpenCL, com arquitetura CNN que pode ser reutilizado para sistemas FPGA[1] .

Este artigo está organizado da seguinte forma: A seção TRABALHOS RELACIONADOS apresenta os trabalhos relacionados. A seguir, Seção PROPOSTA CNN ARQUITETURA mostra nossa arquitetura de aprendizado de máquina proposta. Em seguida, descrevemos a metodologia de avaliação na Seção METODOLOGIA DE AVALIAÇÃO. Por fim, mostramos o resultados obtidos na Seção RESULTADOS e nossos resultados conclusões e trabalhos futuros na Seção CONCLUSÃO.

TRABALHO RELATADO

Redes Neurais Artificiais podem ser aplicadas em uma amplo conjunto de aplicações [13], [14], [15]. Seu uso vão desde fins de segurança na Internet das Coisas (IoT) [16], [17] para veículos autônomos [4], [18]. Independentemente da aplicação ou cenário, esta seção concentra-se em trabalhos relacionados que descrevem CPU ou FPGA-projetos baseados em CNN para detecção de intrusão de rede.

Devido ao aumento do tráfego da Internet, alguns re-trabalhos de pesquisa focam em FPGA como acelerador para melhorar o desempenho da Detecção de Intrusão de Rede (NID). Jeune et al. [19] projetou uma CNN em tempo real sem reduzir o desempenho com base no uso do fluxo baldes para coletar recursos de tráfego. Os autores também usaram um design baseado em FPGA para avaliar o desempenho com alta velocidade de rede em trabalhos futuros [20].

Apesar de não usar a abordagem baseada em FPGA, Ding e Zhai [22] propuseram um sistema de detecção de intrusão com uma rede neural convolucional usando um conjunto de dados NSL-KDD. Eles avaliaram a proposta da CNN em comparação com outros algoritmos de aprendizado de máquina, como como Random Forest, Support Vector Machine (SVM), Rede de Crenças Profundas (DBN) e Longo Curto Prazo Memória (LSTM). Os resultados apontaram a melhoria na precisão da detecção de intrusões.

Da mesma forma, dois trabalhos de pesquisa [21], [23] propuseram e avaliou CNNs para sistemas de detecção de intrusão com conjuntos de dados KDD99 e NSL-KDD, respectivamente. O primeiro alcançou precisão de até 99,23% em comparação com outros algoritmos, como vetor de suporte máquina e rede de crenças profundas. O segundo trabalho alcançou uma precisão melhor, por exemplo, 97%, do que outras técnicas, como Rede Neural Recorrente (RNN), Memória de longo e curto prazo (LSTM) e Unidades Recorrentes Fechadas (GRU).

O trabalho relacionado ainda deixa lacunas para pesquisas sobre desenvolvendo Redes Neurais Convolucionais para FPGA.

[1]https://github.com/cart-pucminas/machine-learning

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Machine Translated by Google
Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

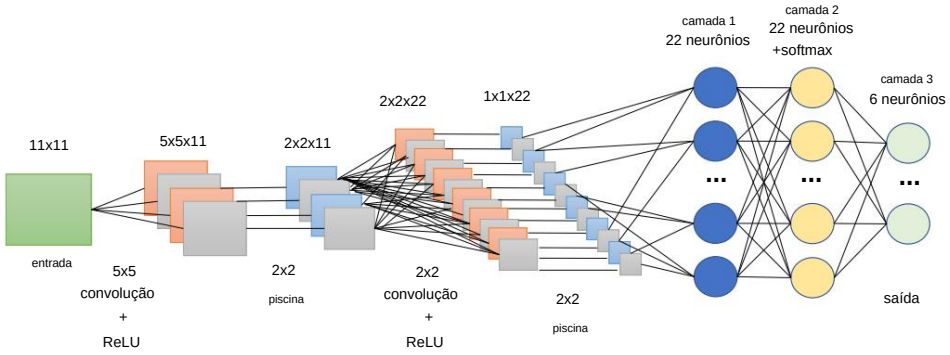

Figura 1. Diagrama de arquitetura CNN

Alguns trabalhos anteriores concentram-se exclusivamente no uso de uma CPU ou FPGA para implementar CNNs. Em contrapartida, outros pretendem alcançar alta precisão comparando os resultados obtidos com diferentes algoritmos de aprendizado de máquina. No entanto, essas abordagens não priorizam o desempenho e eficiência energética.

Nosso trabalho se diferencia dos demais por propor uma CNN arquitetura em um pacote multi-chip híbrido CPU + FPGA com alta largura de banda para explorar o desempenho com eficiência energética. Seguindo os conceitos de particionamento de tarefas, projetamos fluxos de execução para aproveitar o poder de unidades de execução altamente paralelas no FPGA para operações de convolução. Este pacote híbrido multi-chip é um sistema padrão em firewalls de próxima geração, permitindo flexibilidade e a implantação de soluções inovadoras técnicas [10], [11]. Além disso, implementações da CNN usando OpenCL para multichip baseado em FPGA pacotes e detecção de intrusão ainda são pouco explorados. Assim, nosso principal avanço para o trabalho relacionado é uma Arquitetura CNN baseada em CPU + FPGA com baixo consumo de energia para detecção de intrusão na rede.

PROPOSTA DE ARQUITETURA CNN

Nossa arquitetura de hardware foi projetada para suportar aritmética de ponto flutuante para manter a consistência com a implementação da CNN baseada em software, que também usou aritmética de ponto flutuante. Por esta razão, este o papel não explora a economia no consumo de energia da quantização devido à perda potencial de precisão.

Embora os FPGAs ofereçam alta eficiência energética, eles são limitados por memória finita e blocos lógicos. Além disso, implementar uma CNN inteira em um FPGA pode ser desafiador e demorado, exigindo ampla otimização e ajuste para alcançar alto desempenho eficiência energética. Portanto, decidimos implementar apenas a operação de convolução no FPGA. Esta operação é computacionalmente intensiva e se beneficia ao máximo da aceleração FPGA. Por descarregando esta operação para o FPGA, conseguimos ganhos significativos de desempenho e eficiência energética melhorias sem sobrecarregar o FPGA com cálculos adicionais. Assim, conseguimos uma eficiência e implementação escalável da CNN que alavancou o Pontos fortes da CPU e FPGA.

Nossa arquitetura CNN baseada em CPU+FPGA foi desenvolvida assinado usando OpenCL como uma síntese de alto nível (HLS) estratégia, que fornece uma interface de programação padrão para sistemas de computação heterogêneos. Esse abordagem nos permite desenvolver código portátil facilmente implantado em diferentes plataformas de hardware, incluindo sistemas híbridos CPU+FPGA. Além disso, usamos o Biblioteca de aprendizagem profunda Tiny-DNN-DNN [25] para construir nossa rede neural personalizada. Com o C++ e OpenCL se aproxima, o host (CPU) inicia o FPGA e seus objetos e unidades de memória e sincroniza as operações executadas em paralelo, usando itens de trabalho no nível do kernel. Esta abordagem de implementação nos permitiu adotar uma lógica centrada em torno de unidades de execução paralela para as operações de convolução no FPGA. Usando OpenCL, essas unidades de execução pode ser projetado de forma eficiente, como por meio do uso de itens de trabalho indexados e organizados em um intervalo N-Dimensional (NDRange). Este NDRange permite a execução de instâncias do kernel para cada trabalho item, facilitando o processamento paralelo.

A arquitetura CPU+FPGA (Figura 2) é baseada nos seguintes blocos interligados: (i) o Xeon é responsável por conter as interfaces de conexão

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Machine Translated by Google
Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

com o FPGA, com a memória DRAM e armazenando a lógica de programação da CNN; (ii) o Código CNN inclui os módulos responsáveis por executar as funções da rede neural e controlar as operações realizadas em cada fluxo do algoritmo, sejam etapas na CPU ou acionamento do dispositivo; (iii) o módulo FPGA-FIU possui o bitstream responsável por fazer a interface entre as interconexões e o bitstream fornecido pela UCP; (iv) o FPGA-

A interface CCI-P expõe canais de comunicação para CPU; (v) e o módulo FPGA-AFU representa a etapa de convolução do algoritmo CNN que será executado no FPGA.

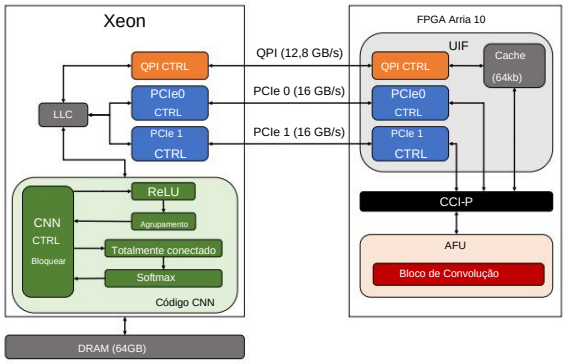

Figura 2. Arquitetura CPU+FPGA CNN

Nosso projeto recebe uma CNN previamente treinada e um conjunto de dados de registros de testes como entrada para avaliação e simulação da detecção de intrusões na rede. Como saída, a classe identificada para cada registro e sua precisão estão disponíveis.

A Figura 1 mostra uma arquitetura CNN, recebendo entrada de 121 recursos processada por duas camadas de convolução, duas camadas de pooling e três camadas totalmente conectadas. A última camada é responsável por classificar os registros de intrusão na rede em seis tipos de classes. O tamanho dos kernels da camada de convolução é [5*5] e [2*2], e o tamanho das camadas de pooling é [2*2]. Além disso, a arquitetura utiliza camadas totalmente conectadas com 22 neurônios nas duas primeiras camadas e seis neurônios na última. A CNN utiliza a função de ativação da Unidade Linear Rectificada (ReLU) entre cada bloco de camada, exceto no último, onde aplica a função Softmax [26].

O tamanho das camadas de rede foi definido inicialmente considerando o número de tipos de ataques (22) e atributos (144 ou 11x11) que seriam avaliados.
Com base nessas informações, foram feitas alterações nos hiperparâmetros da CNN para avaliar qual deles teria uma acurácia mais aceitável.

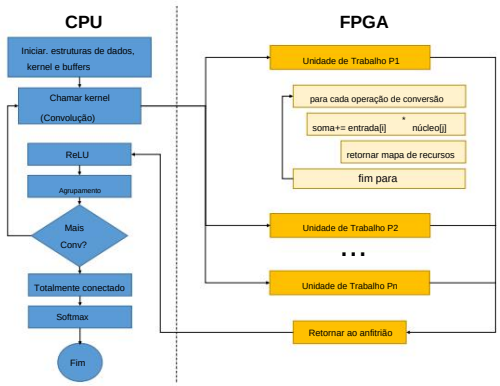

Figura 3. Fluxo da arquitetura CNN

A operação CNN (Figura 3) inicia instanciando memórias FPGA e carregando estruturas de dados para armazenar os registros que irá processar. Após, o modelo previamente treinado é carregado.

A CNN processa itens em camadas, executando apenas aqueles relacionados à convolução no FPGA. A razão
O objetivo é alcançar alto desempenho, uma vez que as operações relacionadas à multiplicação de matrizes são altamente paralelizáveis e demandam maior poder computacional. Além disso, também aplicamos a estratégia NDRange, permitindo ao FPGA utilizar diversas unidades computacionais com alto poder de paralelismo em cada neurônio disponível.

A arquitetura envia os registros processados de volta ao host. Em seguida, o host executa as outras camadas de pooling, ReLU e totalmente conectado, usando estruturas de loop paralelo, melhorando o poder de processamento.
Por fim, o sistema informa ao usuário a classificação final do registro avaliado e essas etapas são realizadas de forma iterativa para cada registro existente no conjunto de dados. A precisão da rede neural é verificada apenas ao final de todo o processo, comparando o rótulo original do registro com o classificado pela rede.

METODOLOGIA DE AVALIAÇÃO

Comparamos nossa arquitetura CPU + FPGA baseada em OpenCL com a versão de software baseada em OpenMP somente CPU com 24 threads, usando os mesmos dados de entrada. Usamos um processador Intel Xeon E5-2620 com 2,4 GHz.
O número de threads escolhidos está relacionado ao melhor resultados Alcançados. Nossa proposta de arquitetura heterogênea roda no pacote multi-chip híbrido Intel, com CPU Intel Xeon de 2,4 GHz + FPGA Arria 10 GX1150,
com frequência de até 400MHz e 65,6MB de memória interna, permitindo alta transferência de dados.

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Machine Translated by Google
Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

A Rede Neural Convolucional usou dois mesmos arquivos de registros NSL-KDD [24] para treinamento e teste. Além disso, desenvolvemos um script em linguagem C para ler um arquivo CSV com 41 colunas, onde 40 são atributos e 1 é uma gravadora. Seus valores foram normalizados entre 0-1, usando o min-max técnica [22]. A estratégia converte as colunas com valores categóricos (tipo de protocolo, serviço, sinalizador) em novas colunas com valores binários. Por exemplo, O tipo de protocolo possui três tipos de atributos: TCP, UDP e ICMP, que formam três colunas [1,0,0], [0,1,0] e [0,0,1]. Assim, ao final do processo, teremos 122 colunas, ou seja, 121 atributos e um rótulo. Por fim, seis classes agrupam 22 tipos de ataques de acordo com suas características (Dos, U2r, R2l, P robe, Normal e Desconhecido).

Inicialmente, treinamos a CNN utilizando um software versão com 125.000 registros de entrada e 64 iterações/épocas para gerar um modelo com 90,59% de precisão para as plataformas CPU-Only e CPU+FPGA.

Com o modelo treinado, testamos a CNN, variando o número de registros de acordo com Fibonacci sequência vezes mil (1000, 2000, 3000, 5000, 8.000, 13.000 e 21.000) devido aos 22.543 registros limite no conjunto de dados NSL-KDD.

Para comparar as abordagens software e CPU+FPGA, avaliamos o tempo de execução e a energia consumo. O Intel PowerPlay EPE retorna o Consumo de energia da abordagem FPGA. Enquanto isso, o A ferramenta PowerTOP [27] faz o mesmo para o processador Xeon. Calculamos a métrica de eficiência energética como a proporção entre o número total de operações de cada carga de trabalho e a energia consumida. Em outras palavras, o métrica é chamada MFLOPS (Millions of Floating Point Operações por segundo) por watt, ou seja, MFLOPS/W.

RESULTADOS

Esta seção destaca nossos principais resultados relacionados a Ocupação de FPGA e eficiência energética. O pacote multichip consiste em um dispositivo com duas unidades: o Unidade de Interface FPGA (FIU) e Unidade de Função Acelerada (AFU). A Tabela 1 mostra a ocupação do dispositivo como o número de elementos lógicos, Adaptive Lookup Tabelas (ALUTs), registradores, blocos de memória e digitais Blocos de processamento de sinal (DSP). Arria 10 continua sendo um grande número de recursos disponíveis para outros potenciais grãos.

Para uma análise abrangente da CNN, as Figuras 4, 5 e 6 mostram tempo de execução, consumo de energia e

Tabela 1. Ocupação do FPGA

Recursos	UIF+CCI	AFU
Elementos Lógicos	563 (49%)	58 (5%)
ALUTs	98256 (23%)	4272 (1%)
Registros	461376 (27%)	68352 (4%)
Blocos de memória	15.466 (23%)	4.035 (6%)
Blocos DSP	182 (12%)	15 (1%)

resultados de eficiência energética para plataformas CPU+FPGA e somente CPU. Assim, executamos 24 threads, com um variação de registros (Rec) entre 1000 e 21000.
O tempo de execução para processar vários registros de intrusão de rede pela abordagem CPU+FPGA é de até 77% inferior à versão somente CPU. Além disso, há há uma diferença crescente nos resultados à medida que o número de registros aumenta, o que mostra melhor escalabilidade da plataforma heterogênea ao trabalhar com grandes cargas de trabalho devido ao FPGA como acelerador. O consumo de energia mostra que a CPU + FPGA plataforma consome até 78% menos que a plataforma somente CPU versão do software.

Em termos de eficiência energética, medida em milhões de operações de ponto flutuante por segundo (MFLOPS) por watt, a plataforma heterogênea pode funcionar até 4,5× mais MFLOPS/watt do que o software somente CPU versão. O tempo de execução impacta os resultados energéticos em ambos os cenários, mesmo considerando o consumo médio do Xeon em 85W e CPU+FPGA em 81,45W.

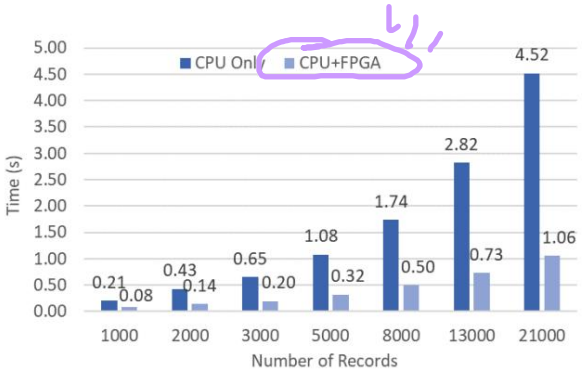

Figura 4. Tempo de execução (segundos)

Nossos resultados mostram que podemos implementar CNN arquiteturas em pacotes multi-chip heterogêneos CPU+FPGA para sistemas de detecção de intrusão em redes. Eles podem atender cenários com variados tipos e quantidades de dados para identificar prontamente anomalias com eficiência energética.

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Machine Translated by Google
Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

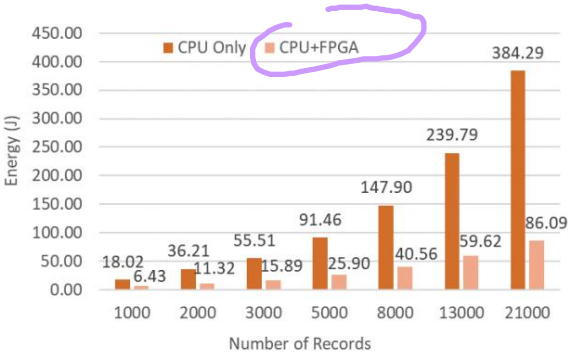

Figura 5. Consumo de energia (Joules)

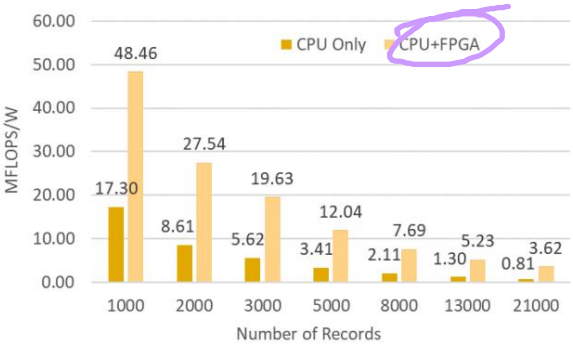

Figura 6. Milhões de operações de ponto flutuante por segundo por watt (MFLOPS/W)

CONCLUSÃO

Este trabalho apresentou uma arquitetura CNN heterogênea para detecção de intrusões utilizando as linguagens OpenCL e C/C++ na plataforma Intel HARPv2 (Intel Xeon + Arria 10). Implementamos uma Rede Neural Convolucional (CNN) usando a biblioteca C++ OpenSource chamada Tiny-DNN. Esta versão CNN usando abordagens paralelas obteve melhor desempenho em plataforma heterogênea CPU + FPGA do que a versão somente CPU de até 77% em tempo de execução, até 78% em consumo de energia e até 4,5× em MFLOPS/watt. Para trabalhos futuros, pode-se adotar estratégias de compartilhamento de memória e distribuição de carga para explorar um nível mais alto de heterogeneidade para obter mais eficiência e escalabilidade. Além disso, pretendemos comparar esta atual solução CPU+FPGA de síntese de alto nível baseada em OpenCL com outras arquiteturas heterogêneas, por exemplo, CPU+GPU, com foco em flexibilidade, portabilidade, desempenho e eficiência. Finalmente, em relação ao potencial trade-off entre desempenho de processamento e precisão, nosso planejamento inclui uma exploração do espaço de design de FPGA para ser eficiente em tempo de execução, consumo de energia e precisão para diferentes algoritmos de redes neurais.

AGRADECIMENTOS

Este estudo foi parcialmente financiado pela CAPES (Código 001), CNPq, FAPEMIG e Intel HARP.

REFERÊNCIAS

1. MA Souza et al., “Clustering K-Means paralelo com eficiência energética para um pacote Intel Hybrid Multi-Chip”, em 30th Int. Simpósio sobre Comp. Arquitetura e Computação de Alto Desempenho, pp. 372-379, 2018, doi: 10.1109/CAHPC.2018.8645850.
2. LA Maciel, MA Souza e HC Freitas, “K-Means/K-Modes reconfiguráveis baseados em FPGA Arquitetura para detecção de intrusão em rede”, no IEEE Trad. Sistema de Circuitos II Exp. Resumos, vol. 67, não. 1459-1463, 2020, doi: 10.1109/TCSII.2019.2939826.
3. MO Farooq, I. Wheelock e D. Pesch, “IoT-Connect: Uma Estrutura de Interoperabilidade para Protocolos de Comunicação Doméstica Inteligente”, em IEEE Consumer Elect. Revista, vol. 9, não. 1, pp. 22-29, 1º de janeiro de 2020, doi: 10.1109/MCE.2019.2941393.
4. I. Ahmed, G. Jeon e A. Ahmad, “Profundo Sistema de detecção de intrusão baseado em aprendizagem para Internet de veículos”, em IEEE Consumer Elect. Revista, vol. 12, não. 1, pp. 117-123, 1º de janeiro de 2023, doi: 10.1109/MCE.2021.3139170.
5. M. Krishna S e T. Perumal, “Fazendo Edifícios Mais Inteligente e Eficiente em Energia – Usando a Internet de Plataforma de Coisas”, em IEEE Consumer Elect. Revista, vol. 10, não. 3, pp. 34-41, 1º de maio de 2021, doi: 10.1109/MCE.2021.3053182.
6. A. Halimaa A. e K. Sundarakantham, “Máquina Sistema de detecção de intrusão baseado em aprendizagem”, em 3rd Int. Conf. sobre Tendências em Elect. e Informática, pp. 2019, doi: 10.1109/ICOEI.2019.8862784.
7. Y. Li, “Pesquisa sobre Aplicação de Rede Neural Convolucional na Detecção de Intrusão”, em 7º Int. Fórum em Eleger. Eng. e Automação, 2020, pp. 10.1109/IFEEA51475.2020.00153.
8. E. Chen et al., “Aplicação de Rede Neural Convolucional Melhorada na Classificação de Imagens”, Int. Conf. sobre Aprendizado de Máquina, Big Data e Business Intelligence, pp. 109-113, 2019, doi: 10.1109/MLBDBI48998.2019.00027.
9. J. Li et al., “Um acelerador de rede neural convolucional reconfigurável com eficiência energética baseado em FPGA para aplicações de reconhecimento de objetos”, em IEEE Trans. Sistema de Circuitos II Exp. Resumos, vol. 68, não. 3143-3147, 2021, doi: 10.1109/TCSII.2021.3095283.
10. MS Brunella et al., “HXDP: pacote de software eficiente

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.

Este artigo foi aceito para publicação na IEEE Consumer Electronics Magazine. Esta é a versão do autor que não foi totalmente editada e
Machine Translated by Google O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/MCE.2023.3283730

processamento em NICs FPGA”, em Commun. ACM, vol. 65, não. 8, pp. 92–100, agosto de 2022, doi: 10.1145/3543668

11. J. Lazaro et al., “Firewall incorporado para barramento on-chip transações”, no Comp. & Eleito. Eng., Vol. 98, 2022, doi: 10.1016/j.compeleceng.2022.107707
12. Y. Guo, “Uma revisão do dia zero baseado em aprendizado de máquina detecção de ataques: desafios e direções futuras”, em Comp. Comun., Vol. 198, 2023, doi: 10.1016/j.comcom.2022.11.001
13. D. Wang, K. Xu e D. Jiang, “PipeCNN: Um Acelerador FPGA de código aberto baseado em OpenCL para redes neurais de convolução”, 2017 Int. Conf. em campo Programa. Tecnologia, pp. 279-282, 2017, doi: 10.1109/FPT.2017.8280160.
14. MR Vemparala, A. Frickenstein e W. Stechele, “Um projeto eficiente de acelerador fpga para cnns otimizados usando opencl”, em Int. Conf. em Arquitetura da Comp. Sistemas, Springer, Cham, 2019, doi: 10.1007/978-3-030-18656-2 18.
15. J. Jiang et al., “Um CPU-FPGA heterogêneo Sistema de aceleração para rede de detecção de texto de cena”, em IEEE Trans. Sistema de Circuitos II Exp. Resumos, vol. 69, não. 6, pp.
16. L. Ioannou e SA Fahmy, “Intrusão de Rede Detecção usando redes neurais em SoCs FPGA”, 2019 29th Int. Conf. no Campo Prog. Lógica e Aplicações, Barcelona, 2019, pp. 232-238, doi: 10.1109/FPL.2019.00043.
17. D.-M. Ngo et al., “Estrutura de aceleração de hardware FPGA para sistema de detecção de intrusão baseado em anomalias em IoT”, 2021 31st Int. Conf. no Campo Prog. Lógica e Aplicações, Dresden, 2021, pp. 69-75, doi: 10.1109/FPL53798.2021.00020.
18. L. Zhang, X. Yan e D. Ma, “Acelerando o sistema de detecção de intrusão de rede no veículo usando rede neural binarizada”, Int. J. Adv. & Curr. Prac. em Mobilidade, 4(6):2037-2050, 2022, doi: 10.4271/2022-01-0156.
19. L. Le Jeune, T. Goedeme, N. Mentens. "Em direção a Detecção de intrusão de rede baseada em aprendizagem profunda em tempo real em FPGA", em Applied Cryptography and Network Security Workshops, vol 12809. Springer, Cham, 2021, doi: 10.1007/978-3-030-81645-2_9.
20. L. Le Jeune et al., “SoK - Detecção de Intrusão de Rede em FPGA”, em Segurança, Privacidade e Criptografia Aplicada Eng., vol 13162. Springer, Cham, 2022, doi: 10.1007/978-3-030-95085-9 13.
21. RU Khan et al., “Um modelo de rede neural convolucional aprimorado para detecção de intrusão em redes”, em Conferência de Segurança Cibernética e Ciberforense, pp. 74-77, 2019, doi: 10.1109/CCC.2019.000-6.
22. Y. Ding e Y. Zhai. “Sistema de Detecção de Intrusão para Conjunto de dados NSL-KDD usando redes neurais convolucionais”, em ACM Int. Conf. em Comp. Ciência e Inteligência Artificial, Nova York, pp.81–85, 2018, doi: 10.1145/3297156.3297230.
23. S. Al-Emadi, A. Al-Mohannadi e F. Al-Senaid, “Usando Técnicas de Aprendizado Profundo para Detecção de Intrusão em Rede”, em IEEE Int. Conf. em Informática, IoT e Habilitando Tecnologia, pp. 171-176, 2020, doi:10.1109/ICIoT48696.2020.9089524.
24. LCHong. (2015) Conjunto de dados Nsl-kdd. [On-line]. Disponível: https://github.com/defcom17/NSL KDD.
25. E. Riba. (2016) Tiny-Dnn. [On-line]. Disponível: https://github.com/tiny-dnn/tiny-dnn.
26. M. Abadi et al., “TensorFlow: um sistema para larga escala aprendizado de máquina”, na USENIX Conf. em operação Projeto e implementação de sistemas, pp. 265–283, 2020, doi: 10.5555/3026877.3026899.
27. Informações. (2007) PowerTOP. [On-line]. Disponível: https://01.org/powertop.

Lucas Andrade Maciel é engenheiro de TI líder da Localiza&Co e professor adjunto da Pontifícia Universidade Católica de Minas Gerais (PUC Minas), Belo Horizonte, Brasil. Ele recebeu seu B.Eng. em Engenharia da Computação (2017) e M.Sc. Doutor em Informática (2020) pela PUC Minas. Seus interesses de pesquisa são Arquitetura de Computadores, Computação de Alto Desempenho e Sistemas de Detecção de Intrusão. Contate-o em lucasmaciel@pucminas.br.

Matheus Alcântara Souza é professor assistente Pontifícia Universidade Católica de Minas Gerais (PUC Minas), Belo Horizonte, Brasil. Ele recebeu sua tecnologia. (2007) em Desenvolvimento de Sistemas pela Fabrai, M.Sc. (2015) e doutorado. (2021) em Informática pela PUC Minas. Seus interesses de pesquisa são Arquitetura de Computadores e Computação de Alto Desempenho. Entre em contato pelo e-mail matheusalcantara@pucminas.br.

Henrique Cota de Freitas é professor associado da Pontifícia Universidade Católica de Minas Gerais (PUC Minas), Belo Horizonte, Brasil. Ele recebeu seu bacharelado em Ciência da Computação (2000) e M.Sc. Possui graduação em Engenharia Elétrica (2003) pela PUC Minas e doutorado. Doutor em Ciência da Computação (2009) pela Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brasil. Seus interesses de pesquisa são Arquitetura de Computadores e Computação de Alto Desempenho. Contate-o em hcfreitas@ieee.org.

Uso licenciado autorizado limitado a: PUC MG. Baixado em 06 de março de 2024 às 21:34:15 UTC do IEEE Xplore. Restrições aplicadas.
© 2023 IEEE. O uso pessoal é permitido, mas a republicação/redistribuição requer permissão do IEEE. Consulte https://www.ieee.org/publications/rights/index.html para obter mais informações.