Evento	Salão UFRGS 2019: SIC - XXXI SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2019
Local	Campus do Vale - UFRGS
Título	Ferramenta para geração de código para FPGAs a partir de descrições de alto nível de Redes Neurais
Autor	MATHEUS WOEFFEL CAMARGO
Orientador	PHILIPPE OLIVIER ALEXANDRE NAVAUX

Ferramenta para geração de código para FPGAs
a partir de descrições de alto nível de Redes Neurais

Matheus Woeffel Camargo, Philippe O. A. Navaux
Instituto de Informática - Universidade Federal do Rio Grande do Sul - Porto Alegre, Brasil

As pesquisas em Redes Neurais têm avançado de forma que possibilitem sua aplicação em inúmeros contextos tais como: processamento de linguagens naturais, reconhecimento de imagens e aproximação de aplicações genéricas. Dado o grande poder computacional e consumo energético demandado por essas aplicações, a arquitetura FPGA tem ganhado destaque em aplicações que utilizam redes neurais devido ao paralelismo desta arquitetura e o baixo consumo energético que a mesma proporciona.

Um grande inconveniente dessa arquitetura é a dificuldade em programá-la utilizando linguagens de descrição de *hardware*, como *Verilog e VHDL*. Nesse sentido, o presente trabalho tem como objetivo desenvolver uma ferramenta que converta automaticamente uma descrição de alto nível de uma Rede Neural, usando para esta descrição o *framework Caffe*[1],para código VHDL sintetizável. Ao considerarmos a arquitetura de uma rede neural genérica pode-se notar a presença de uma unidade modular básica responsável pelo processamento de um neurônio, que é então replicado formando uma camada. A implementação completa-se então com o encadeamento e integração das camadas..

Para a implementação de um neurônio foi desenvolvido código *VHDL* genérico dedicado a realizar a parte de computação linear do mesmo, e um *software* em *Python* dedicado à geração de código *VHDL* responsável pela parte de aplicação da função de ativação sobre a saída do primeiro módulo. A subsequente integração dos neurônios em camadas e a integração e encadeamento destas foi então realizada a partir do software escrito em *Python*. Ao passo que o objetivo é encontrar um projeto que melhor adeque aspectos de economia de energia, desempenho e área, foi escolhido implementar as alternativas a seguir.

Para a computação linear dos neurônios foram consideradas como alternativas o uso de multiplicadores em paralelo em conjunto com um acumulador e uma máquina de estados ou o uso de multiplicadores em paralelo com somadores em árvore. Considerando a aplicação da função de ativação foram encontradas como alternativas o uso de *Look-up tables* que implementem a função e aproximação por sigmoid-allipi. Já para a arquitetura das camadas foram encontradas como alternativas a integração das *layers* em cascata com e sem *pipeline*, bem como multiplexação temporal de uma única *layer*[2].

Para avaliar as consequências em termos de área consumida das alternativas anteriormente especificadas, os números de FFs, DSPs, RAMs, BRAMs, ALUTs serão recolhidos a partir de relatórios gerados pela ferramenta Quartus Pro após o processo de síntese. Para avaliar as consequências em termos de desempenho e energia, será realizada a execução da fase de classificação das redes neurais usando a Placa Arria X[3] a fim de extrair tempo de execução e consumo energético instantâneo do sistema. Isso permitirá aperfeiçoar a ferramenta e uma futura comparação com outras alternativas de implementação, como o compilador de alto nível Intel HLS[4].

[1] Framework Caffe https://caffe.berkeleyvision.org/
[2] Temporal Layer Multiplexing https://ieeexplore.ieee.org/document/4182384
[3] HLS
https://www.intel.com.br/content/www/br/pt/software/programmable/quartus-prime/hls-compiler.html
[4]Arria X
https://www.intel.com/content/www/us/en/products/programmable/fpga/arria-10.html