Universidade: presente!

UFRGS PROPESQ

XXXI SIC

Salão UFRGS 2019 – Conhecimento o formação o inovação

21 . 25 . OUTUBRO . CAMPUS DO VALE

Ferramenta para a 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

1. Introdução

A pesquisa em redes neurais nos últimos anos avançou de tal maneira que hoje é possível utilizá-las em um grande número de problemas. Algumas das aplicações mais comuns incluem processamento de linguagem natural, reconhecimento de imagem e algoritmos de aproximação genéricos. No entanto, a resolução de tais problemas requer um grande poder computacional e consequentemente, energia.

Tendo em vista tal problemática, redes neurais são executadas usualmente em sistemas de computação de alto desempenho com um grande número de núcleos de CPU e GPU, reduzindo a quantidade de tempo exigida por aplicações grandes. Apesar do tempo de execução reduzido, a quantidade de energia consumida por essas arquiteturas é bastante alta. Como alternativa para reduzir o consumo de energia, as arquiteturas FPGA (Field Programmable Gate Array) estão ganhando destaque devido ao seu paralelismo intrínseco e baixo consumo de energia.

Uma importante desvantagem, porém, é o fluxo de trabalho mais trabalhoso e a necessidade de conhecimentos específicos da arquitetura.

Como uma alternativa, o trabalho aqui apresentado busca propor a construção de uma ferramenta que automaticamente converta descrições em alto nível de redes neurais em código VHDL sintetizável.

2. Metodologia

Antes da fase de implementação foi conduzido uma pesquisa bibliográfica a fim de conhecer alternativas de implementação para os diferentes módulos necessários para compor as Redes Neurais.

Após definido os módulos a serem implementados e as diferentes alternativas, conforme será comentado na seção 3, foram escolhidas como métricas de avaliação das alternativas:

- Número de FFs, RAMs, ALUTs, DSPs, BRAMs para utilização de área.
- Tempo de execução para desempenho
- Consumo instantâneo e PDP para consumo energético.

Todas estas extraídas a partir da fase de síntese do código gerado para a placa *ARRIA X* e a posterior execução na mesma placa.

3. Implementação e Alternativas de projeto

As redes neurais são um modelo computacional inspirado na maneira como o processamento é realizado no sistema nervoso central dos animais. Sua unidade básica é o neurônio, que é um módulo responsável por transformar o produto escalar de um vetor de entrada por um vetor de peso correspondente e, em seguida, somar um deslocamento e aplicá-lo a uma função de ativação. A função de ativação pode ser qualquer função matemática, mas geralmente são deriváveis e funções “suaves” como sigmoid, tanh e ReLU. Os neurônios são então replicados em paralelo para formar uma camada e sua concatenação é realizada para formar a própria rede neural. Essa concatenação é realizada adotando a saída de cada neurônio de uma camada anterior como entrada da camada atual. Buscando mapear os conceitos do problema para a solução proposta, os seguintes módulos foram considerados:

- Indutor de campo - módulo responsável pelo cálculo do produto escalar do vetor de entrada multiplicado pelo vetor de peso, adicionado ao *bias*.
- Função de Ativação - módulo que implementa uma Função de Ativação, que é aplicada à saída do indutor de campo.
- Neurônio - módulo que une o aplicador de campo à função de ativação, responsável pela lógica de sincronização entre os dois módulos,
- Camada - módulo que reúne os neurônios que a define e tem a responsabilidade de mapear os pesos e entradas à tais neurônios.
- Rede Neural - módulo que concatena as camadas e finaliza a implementação da rede em si.

Em relação às alternativas de projeto para os diferentes módulos, 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 camada.

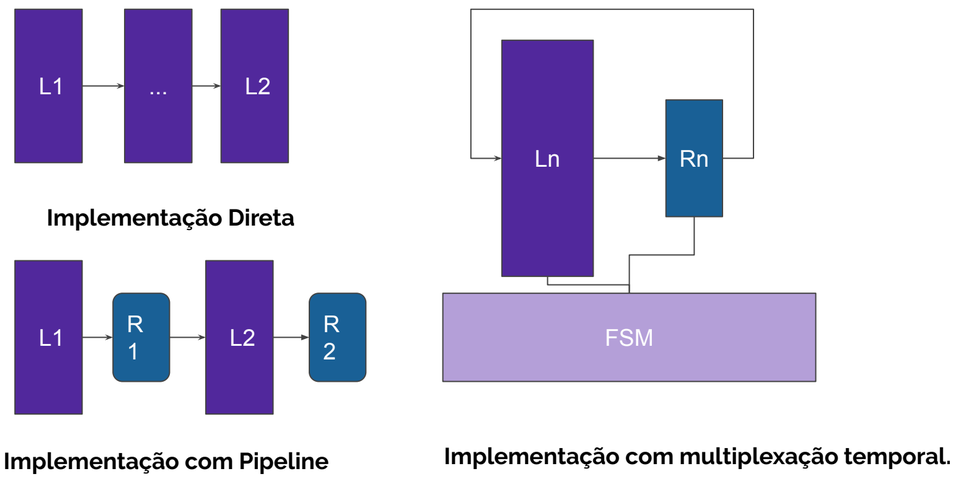

Figura 1: Alternativas consideradas para a arquitetura das camadas.

4. Resultados Preliminares

Implementação completa usando:

- Múltiplas camadas com pipeline
- Função sigmoid usando Look-up tables
- Aplicador de campo usando multiplicadores em conjunto com um somador e *FSM*

Abaixo uma tabela mostrando uso de área para um código gerado pela ferramenta para implementar uma rede com 3 neurônios organizados em 2 camadas e largura de dados de 8 bits.

Area Usage	
Resource Name	Resource Usage
ALM	404
LABs	50
Logic Registers	171
DSPs Blocks	6

Table 1. Arria 10 FPGA resource usage.

Figura 2: Uso de área para uma rede implementando a função XOR.

5. Conclusão

Neste trabalho foi apresentado a proposta e atual estado do projeto de implementação da ferramenta. Devido aos poucos dados recolhidos ainda não é possível tirar conclusões simbólicas. Presente trabalho incluí a implementação considerando o uso do HLS para comparar os dados sobre a ferramenta e outra implementação, a fim de possibilitar conclusões sobre o estado atual da ferramenta e motivar futuras modificações