РАЗРАБОТКА АППАРАТНОЙ НЕЙРОСЕТИ ДЛЯ РЕАЛИЗАЦИИ КОМПЬЮТЕРНОГО ЗРЕНИЯ НА ПЛИС

Зоев И.В., Рыжова С. Е., БересневА.П.
Научный руководитель: МальчуковА.Н.
Томский политехнический университет
zoev.ivan@yandex.ru

Введение

Наиболее популярной реализацией компьютерного зрения являются свёрточные нейронные сети, которые показывают большой процент распознавания объектов на изображении.

Однако в силу больших объемов вычислений данный алгоритм плохо подходит для работы с обработкой данных в реальном времени. На сегодняшний день существуют множество алгоритмов программной оптимизации СНС. Но, несмотря на это вычислительные системы СНС остаются громоздкими и не энергоэффективными.

Аппаратная реализация СНС

Одним из способов решения вышеуказанных проблем является оптимизация вычислений происходящий в нейросети на аппаратном уровне.

Для решения данной задачи необходимо сначала решить следующие проблемы:

1. Выбор представления чисел и реализация операций над ними.
2. Реализация хранения и считывания данных промежуточных результатов.
3. Реализация основных функций СНС
4. Обеспечение взаимодействия всех функциональных узлов СНС.

Хранение данных

Если проблема выбора представления данных вместе с реализацией вычислителей уже были представлены статье [1], то проблема хранения данных остается не освещённой. Поскольку основные реализации являются программными, то им не представляет труда работать с матрицами данных. Однако, в аппаратном уровне доступ к памяти является одномерным. Для правильного чтения данных необходимо это учесть при создании контроллера памяти, который будет производить подобие двухмерного доступа.

Так же с целью экономии блоков памяти и невозможностью распараллеливания вычислений при операции свертки с множеством входов необходимо реализовать переключение множества входных слоев в этом контроллере.

Блок, несущий в себе данные функции – *memory_controller* представлен на рис. 1.

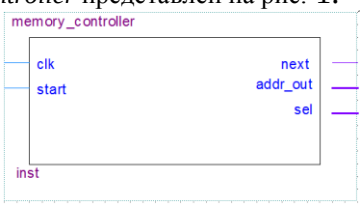

Рис. 1 Модуль memory_controller

Функциональная схема алгоритма представлена на рис.2

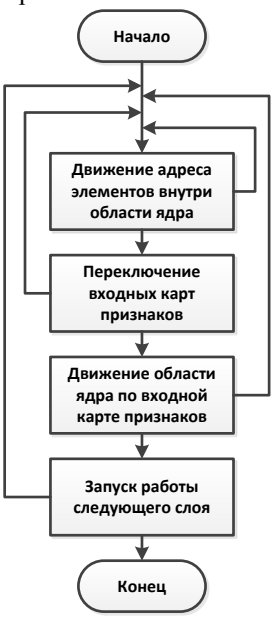

Рис. 2 Алгоритм работы блока контроллера памяти

Реализация основных функций СНС

Основные функции СНС делятся на:

- Функции свёрточного слоя
- Функции слоя субдискретизации
- Функция полносвязного слоя
- Функция активации

Свёрточный слой представляется в виде блока изображенным на рис. 3.

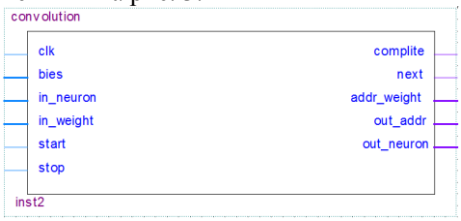

Рис. 3 Модуль свёрточного слоя

Данный блок состоит из модулей:

- *St_machine_weight*, подает соответствующие значения весов на умножитель
- *St_machine_neuron*, подает соответствующие значение входных нейронов на умножитель
- *Neuron*, выполняет непосредственную операцию свёртки. Блок состоит в свою очередь состоит из:
  - *Mult*, реализует умножение
  - *Reg*, записывает результат умножения

- *Sum*, производит сложение.
- *St_machine_sum*, реализует автомат управления суммы с накоплением и запись результата

Основной алгоритм работы блока заключается в перемножении значений входного слоя с ядром свёртки. *St_machine_weight*и *St_machine_neuron* выполняют функцию подачи значений весов и нейронов на блок умножителя. В начале работы умножителя блоки *Sum* и *St_machine_sum* производят операцию сложение обнуленного блока *Reg* и значения смешения нейрона. Значения выхода умножителя записывается в регистр. Последующая работа связки *Reg*,*Sum* и *St_machine_sum* работают как сложение с накоплением. По завершению операции свертки блок *St_machine_sum* производит обнуление накопленной суммы, а результат сверки записывается в память выходного слоя.

Слой субдискретизации представляется в виде блока представленного на рис. 4.

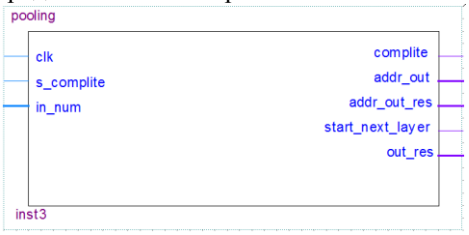

Рис. 4 Модуль слоя субдискретизации

В свою очередь данный блок также содержит Данный блок состоит из блоков:

- *St_machine_pooling*, выполняет управление и подачу значений на блок сравнения
- *Comp*, компаратор двух чисел

Модуль слоя субдискретизации выполняет функцию сокращения размерности входного слоя на основе операции выбора максимального элемента[3]. *St_machine_pooling*, последовательно подает на входы компаратора считанный из памяти элемент и найденный максимальный. Благодаря такому алгоритму работы, количество тактов требуемых для поиска максимального элемента, равно количеству элементов поиска.

Функция полносвязного слоя является частным случаем свёрточного слоя. Поэтому отдельной реализации для данного типа слоев в данной работе нет.

Функций активации для нейронный сетей существует достаточно много, однако для аппаратной реализации при возможности выбора лучше выбирать простую для реализации. А наиболее простой, является ReLU[4] представляющую собой кусочно-линейная функцию.

Исходя из этого, был создан блок func_active который содержит в себе функцию вида:

$$f(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases}$$

Данную функцию легко построить для любого представления чисел с помощью мультиплексора.

На рис. 5 представлен блок *func_active*, который принимает значение и возвращает в соответствии с его функцией.

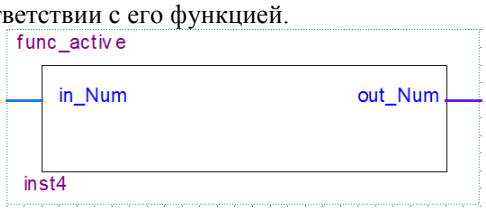

Рис. 5 Модуль функции активации

Синхронизация функциональных блоков

Описанные модули созданы на основе блочно-ориентированного подхода, который используется для реализаций на ПЛИС [4].

Синхронизация блоков происходит за счет автоматов управления (*St_machine_weight, St_machine_neuron, St_machine_sum, St_machine_pooling*), блоков контроллера памяти (*memory_controller*) и сигналов управление (start, stop, reset).

Заключение

В данной статье рассматриваются основные моменты аппаратной реализации свёрточной нейросети. На основе описанных блоков можно составлять различные архитектуры сетей, на основе которой возможно дальнейшее исследование в области аппаратных нейросетей. В частности, на основе данной работы возможно увеличение скорости работы и повышения энергоэффективности в сравнении с программными аналогами.

Список использованных источников

1. Зоев И.В. Разработка вычислителя для плавающей точки для нейронных сетей / науч. рук. А.Н. Мальчуков // Информационные технологии в науке, управлении, социальной сфере и медицине: сборник научных трудов III Международной научной конференции, Томск 23-26 мая 2016 г. — Томск: Изд-во ТПУ, 2016. – Ч. 1. – С. 162-164.

2. Giusti A. et al. Fast image scanning with deep max-pooling convolutional neural networks //arXiv preprint arXiv:1302.1700. – 2013.

3. LeCun Y., Bengio Y., Hinton G. Deep learning //Nature. – 2015. – Т. 521. – №. 7553. – С. 436-444.

4. Еремин В. В. , Мальчуков А. Н. О применении блочно-ориентированного подхода к разработке устройств на ПЛИС [Электронный ресурс] // Вестник науки Сибири. Серия: Информационные технологии и системы управления. - 2011 - №. 1 - С.379-381. Режим доступа: http://sjs.tpu.ru/journal/issue/view/2/showToc/sect/4