КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

УДК 519. 7004. 62

Т.А. САМОЛЮК

ГЛИБОКІ НЕЙРОМЕРЕЖІ ТА ЕЛЕМЕНТИ ЇХ РЕАЛІЗАЦІЇ У ПЛІС FPGA

T. Samolyuk

DEEP NEURAL NETWORK ELEMENTS AND THEIR IMPLEMENTATION IN FPGA XILINX

The description of the deep neural networks, brief theory of building and training are given. Developed their feasibility in the environment of FPGA XILINX.
Key words: neural networks, elements off implementation, deep learning.

Приведено описание глубинных нейронных сетей, краткая теория построения, обучения. Разработаны возможности их реализации в среде ПЛИС XILINX.
Ключевые слова: нейронные сети, элементы реализации., глубокое обучение.

Наведено опис глибинних нейронних мереж, коротка теорія побудови, навчання. Розроблено можливості їх реалізації в середовищі ПЛІС XILINX.
Ключові слова: нейронні мережі, елементи реалізації, глибоке навчання.

Вступ. В даний час теорія і практика навчання нейронних мереж переживають справжню «глибинну революцію», викликану успішним застосуванням методів Deep Learning (глибокого навчання). Нейронні мережі третього покоління, на відміну від класичних, другого покоління мереж 80-90-х років, на основі нової парадигми навчання, дозволили позбутися від ряду проблем, які стримували поширення і успішне застосування традиційних нейронних мереж. Мережі, навчені за допомогою алгоритмів глибокого навчання, не просто перевершили за точністю кращі альтернативні підходи, але й у ряді завдань проявили зачатки розуміння сенсу поданої інформації (наприклад, при розпізнаванні зображень, аналізі текстової інформації і так далі).

Найбільш успішні сучасні промислові методи комп'ютерного зору і розпізнавання мови побудовані на використанні глибоких мереж, а гіганти IT-індустрії скуповують колективи дослідників, що займаються глибокими нейронними мережами.

Загальна частина. Мережі часто реалізуються у вигляді комп'ютерних програм, хоча випускається все більша і більша кількість мікросхем, що реалізують нейронні мережі апаратним шляхом. Головна властивість мереж – здатність до навчання. Глибоке навчання – набір алгоритмів, які намагаються моделювати високорівневі абстракції у даних, використовуючи архітектури, що складаються з безлічі нелінійних трансформацій. Глибока нейронна мережа (DNN – Deep Neu ral Network) це штучна нейронна мережа з

декількома прихованими шарами. Подібно до звичайних нейронних мереж, глибокі нейронні мережі можуть моделювати складні нелінійні відносини між елементами. У процесі навчання глибокої нейронної мережі отримувана модель намагається представити об'єкт у вигляді комбінації простих примітивів (наприклад, у задачі розпізнавання осіб такими примітивами можуть бути частини обличчя: ніс, очі, рот і так далі). Додаткові шари дозволяють будувати абстракції все більш високих рівнів, що і дозволяє будувати моделі для розпізнавання складних об'єктів реального світу.

Як правило, глибинні мережі будуються як мережі прямого поширення. Однак останні дослідження показали, як можна застосувати техніку глибинного навчання для рекурентних нейронних мереж. Згорткові нейронні мережі використовуються в області машинного зору, де цей підхід показав себе як ефективний. Також згорткові нейронні мережі були застосовані для розпізнавання мови.

Навчання глибинних нейронних мереж може бути здійснено за допомогою звичайного алгоритму зворотного поширення помилки. Існує велика кількість модифікацій даного алгоритму. Таким чином, може бути використано кілька правил налаштування ваг. Наприклад, навчання вагових коефіцієнтів $\omega_{ij}(t)$ алгоритмом стохастичного градієнтного спуску:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \frac{\partial C}{\partial \omega_{ij}}, \tag{1}$$

де η – стала для регулювання величини поточного кроку, C – функція втрат. Вибір функції втрат може бути обумовлений класом завдання машинного навчання (з учителем, без учителя, з підкріпленням) і функції активації.

До двох головних проблем глибоких нейронних мереж відносять ті ж проблеми, що виникають і при навчанні звичайних нейронних мереж: час навчання та перенавчання.

Глибокі структури сильніше схильні до перенавчання, оскільки, маючи більше шарів, що дозволяють моделювати високорівневі абстракції, мережа може "вивчити" рідкісні ситуації. У цьому випадку можуть допомогти різні види регуляризації. Один з можливих методів регуляризації (dropout) припускає випадковим чином виключені вузли під час навчання. У деяких випадках це допомагає менше запам'ятовувати рідкісні залежності в тренувальних даних.

Через простоту реалізації і хорошу збіжність для навчання глибоких нейронних мереж часто використовується метод зворотного поширення помилки і градієнтний спуск. Однак, при навчанні глибоких структур виникає кілька проблем, які особливо важливі при оптимізації функцій у просторі великої розмірності: кількість обчислювальних елементів, початкові умови для ваг мережі, а також описана вище константа регулювання величини кроку.

Крім того, алгоритм стохастичного градієнтного спуску відомий своєю проблемою зникаючого градієнта (vanishing gradient), яка полягає в ослабленні градієнта, а значить і швидкості навчання в міру поглиблення від останніх шарів

мережі до початку мережі. Через це глибокі шари мережі дуже погано навчаються. Проте останнім часом є тенденція замість функції активації вузла мережі виду сигмоїда в глибоких мережах використовувати нелінійність виду ReLU (Rectified Linear Unit), функцію якої можна описати як *max* (0, *x*). Глибока мережа з таким видом функції активації не має проблеми ослаблення градієнта і добре навчається градієнтним спуском. За умов великих розмірностей повний перебір всіх комбінацій значень параметрів непрактичний.

Для прискорення обчислень використовується паралелізм, який закладений у саму суть алгоритму навчання нейромережі при прямому і зворотному проході. Розпаралелювання алгоритму на *T* потоків можливо на рівні:

– фази навчання з одночасним навчанням мережі при різних налаштуваннях її параметрів: числа шарів, нейронів у шарах, початкових установках ваг і алгоритмом управління кроком їх зміни (*T* = 2 – 20);

– пакетного навчання (*T* = 10 – 1000); в цьому випадку навчальна множина розбивається на T підмножин, для кожного обчислюється свій градієнт, отримані градієнти сумуються і, таким чином, виходить сумарний напрямок налаштування ваг;

– конвеєрного навчання шарів нейромережі (*T* = 3 – 30);

– вузлів, тобто нейронів нейромережі (*T* = 100 – 1000 000 і більше);

– ваг нейронів (*T* = 100 – 10 000 і більше);

– біт (байт) орієнтованих обчислювальних, у тому числі стохастичних потоків, з відповідною організацією основних засобів обробки, тобто суматорів, помножувачів і блоків пам'яті (*T* на 1 – 2 порядки більше наведених вище значень);

Останні три рівні забезпечують найбільший коефіцієнт паралелізму і особливо ефективні при використання апаратних засобів для прискорення навчання штучних нейронних мереж, таких як GPU і FPGA.

Більш радикальні способи прискорення навчання включають використання Extreme Learning Machines, «No prop»-нейронних мереж і безвагових нейронних мереж [1].

Як бачимо, з вище сказаного збільшення швидкості роботи нейронних мереж є важливою і актуальною проблемою. Одним із способів, що дозволяють прискорити їх роботу за рахунок використання паралелізму, який властивий самій нейронній мережі, є її реалізація на кристалі.

У літературі по створенню нейронних мереж на кристалах можна простежити тільки кілька ідей для конкретних архітектур, які можуть бути адаптовані для моделювання нейронних мереж, ще менше тих, які мають вбудований алгоритм навчання. Останнім часом можна побачити деякі зрушення в цій галузі.

Кристал серії FPGA SPARTAN 3 має архітектуру, що дозволяє створити нейронну мережу для розпізнавання образів. Програмне середовище для роботи з кристалами серії FPGA пакет WEBPACK XILINX є безкоштовним, що створює перевагу в його застосуванні [2].

Робота в середовищі програмування Xilinx ISE Webpack відбувається в наступному порядку:

1) створення принципової схеми проектованого пристрою у схемотехнічному редакторі Xilinx Ise Design Suite 13.2 (FPGA Editor) або описі даного пристрою на мові VHDL або Verilog;

2) попереднє функціональне (Behavioral Simulation) або тимчасове моделювання для виявлення помилок і перевірки працездатності створюваного проекту або окремих його частин;

3) прив'язка висновків проекту до входів-виходів кристала, вибір вихідних рівнів, критичних контурів (Constraints Editor) і т. д.;

4) запуск автоматизованого розміщення проекту в кристалі і аналіз звітів, які генеруються для виявлення попереджень і помилок (Implement Disign), а за відсутності таких і не критичних перехід до наступного етапу;

5) верифікація проекту, тобто остаточне тимчасове моделювання (Post-Fit Simulation) після розміщення проекту в кристалі при всіх реальних затримках поширення сигналів всередині мікросхеми ПЛІС;

6) конфігурування кристала ПЛІС за допомогою бітового потоку (iMPACT 10.1i). Для того, щоб конфігурувати ПЛІС необхідно мати завантажувальний JTAG-кабель. Завантаження бітового потоку здійснюється через спеціально виділені конфігураційні виводи з використанням різних способів і режимів завантаження ПЛІС. Після вдалого завантаження проекту перевіряється та налагоджується проект надалі.

У разі необхідності та подальшого розвитку чи проектування проекту на ПЛІС всі пройдені етапи повторюються до повного завершення проекту в цілому.

Як і лінійні методи класифікації і регресії, за своєю суттю, нейронні мережі видають відповідь у вигляді:

$$y(x,\omega) = f\left(\sum_{j=1}^{N} \omega_j \phi_j(x)\right), \qquad (2)$$

де f – нелінійна функція активації, ω – вектор ваг, ϕ – нелінійні базисні функції [3].

Для лінійних базисних функцій реалізація розрахунків можлива з паралельною обробкою всіх нейронів чергового шару і послідовним накопиченням зважених сум ваг для кожного з них. Таке рішення передбачає застосування DSP IP-ядер, що входять до складу останніх серій FPGA Xilinx. Це забезпечує гнучке управління областями аргументів карт ознак, але є досить витратним.

Варіант з паралельним обчисленням входу функції активації (послідовно для кожного нейрона чергового шару мережі) ефективно реалізується в більш доступних ПЛІС типу Spartan3. На структурній схемі на рисунку цей варіант показаний пірамідальним суматором ADD зважених за допомогою помножувачів MUL значень функцій активації всіх нейронів попереднього шару. Значення функцій активації і вагові коефіцієнти зберігаються в двопортовій блочній пам'яті RAM.

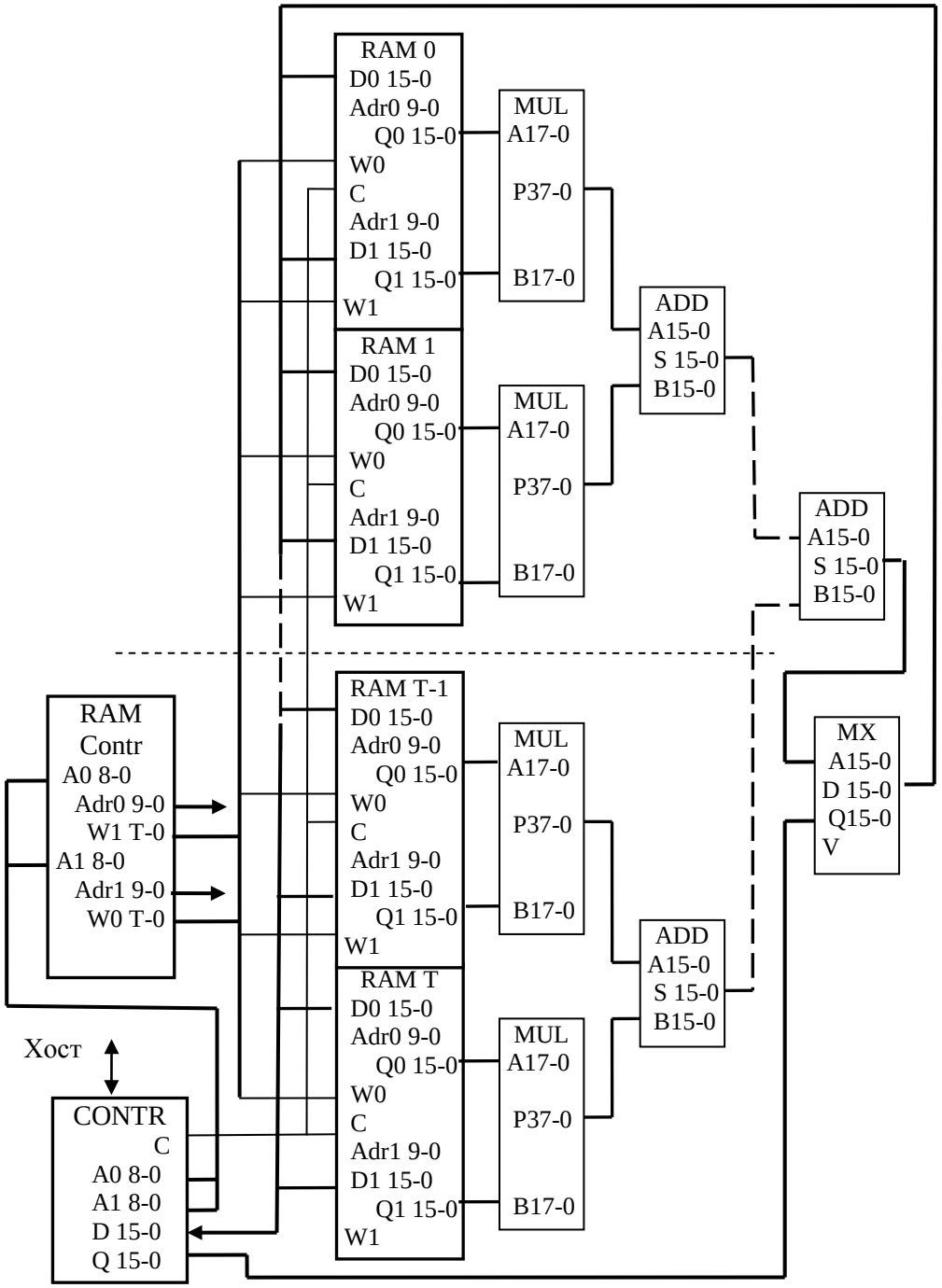

РИСУНОК. Структурна схема нейромережі на FPGA з паралельним обчисленням ваг

Для ефективної роботи з змінними форматами рецептивних полів і проміжних шарів глибинних нейронних мереж необхідний оптимальний вибір параметра *T*, рівний або кратний найбільш часто використовуваному розміру формату. Тоді формати більші за вибраний будуть оброблятися за кілька проходів, а менші – із застосуванням маскування зайвих входів нульовими значеннями на виходах відповідних блоків пам'яті.

Контролер CONTR забезпечує необхідну послідовність адрес і дозвільних рівнів на входах запису відповідних блоків пам'яті. Найбільш економічним рішенням для формування зазначених послідовностей є використання додаткових блоків керуючої пам'яті для незалежного зберігання та видачі адрес і дозвільних рівнів запису. У функції власне контролера тоді входить формування істотно більш простих циклічних підпослідовностей адрес для читання і запису великих областей керуючої пам'яті, а також ініціалізація всієї пам'яті нейромережі і вивантаження результатів обчислень.

Слід зазначити, що рівномірне використання з одного боку розподілених по комірках LUT ресурсів для побудови пірамідального суматора, а з іншого – виділених блоків пам'яті і помножувачів, забезпечує досить високий коефіцієнт використання площі кристала ПЛІС.

Функціональні можливості ПЛІС Spartan3 різних модифікацій, використовуваної в інструментальному модулі Spartan-3 Starter Board, характеризуються наступними показниками:

– наявність двох видів внутрішньої оперативної пам'яті: розподіленої Distributed RAM, яка реалізується на базі 4-входових таблиць перетворення (LookUp Table, LUT) конфігурованих логічних блоків, і вбудованої блокової пам'яті Block RAM, яка може бути організована як двопортовий ОЗП;

– достатній для реалізації нейромережі середнього розміру обсяг внутрішньої розподіленої пам'яті Distributed RAM і вбудованої блочної пам'яті Block SelectRAM;

– застосування чотирьох цифрових блоків управління синхронізацією (DCM), що виконують функції множення, ділення і зсуву фаз тактових частот, і забезпечують розширені можливості керування тактовими сигналами не тільки всередині кристала, а й на рівні друкованої плати проектованої системи;

– висока продуктивність, що допускає реалізацію проектів з системними частотами до 326 МГц;

– використання глобальної мережі тактових сигналів надає можливість розподілу сигналів синхронізації всередині кристалів з малими розходженнями фронтів;

– можливість реалізації швидких внутрішніх інтерфейсів до зовнішніх високопродуктивних елементів пам'яті (ОЗП або ПЗП);

– застосування спеціальної логіки прискореного перенесення для виконання високошвидкісних арифметичних операцій;

– наявність вбудованих апаратних помножувачів, призначених для обчислення добутку двох 18-розрядних операндів;

– наявність ланцюжків каскадування забезпечує можливість реалізації функцій з великою кількістю вхідних змінних;

– підтримка передачі даних з подвоєною швидкістю Double Data Rate (DDR), що відкриває широкі можливості для реалізації високошвидкісних пристроїв цифрової обробки сигналів;

– використання технології Select I/O дозволяє підтримувати 17 однополюсних і 6 диференціальних цифрових сигнальних стандартів введення-виведення, зокрема, LVTTL, LVCMOS12, GTL, SSTL2 (II), HSTL (III), PCI 3.3, AGP, CTT;

– повна підтримка протоколу периферійного сканування відповідно до стандартів IEEE Std 1149.1 (JTAG) і IEEE Std 1532;

– підтримка 5 режимів конфігурування ПЛІС (Master Serial, Slave Serial, Master Parallel, Slave Parallel, JTAG).

Висновки. Таким чином, розробка і побудова нейронних мереж у ПЛІС FPGA дозволяє значно прискорити процеси їх функціонування і навчання за рахунок використання можливості паралельної обробки від сотень до декількох тисяч обчислювальних потоків. Запропоновані варіанти структури нейронної мережі відрізняються високим коефіцієнтом використання кристала і можливістю застосування для реалізації глибинних нейронних мереж.

1. *Geoffrey E. Hinton.* Learning multiple layers of representation // TRENDS in Cognitive Sciences. – 2007. – Vol. 11, N 10. – P. 428 – 434.
2. *Зотов В.* Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. – М.: Горячая линия – Телеком, 2003. – 624 с.
3. *Заенцев И.В.* Нейронные сети: основные модели. – Воронеж, 1999. – 74 с.

Одержано 05.10.2015