



(12) **BẢN MÔ TẢ GIẢI PHÁP HỮU ÍCH THUỘC BẰNG ĐỘC QUYỀN
GIẢI PHÁP HỮU ÍCH**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ**

(11)



2-0004515

(51) **G06N 3/00**
2022.01

(13) **Y**

(21) 2-2023-00239

(22) 12/05/2023

(45) 25/11/2025 452

(43) 25/08/2023 425A

(73) 1. Trường Đại học Bách khoa - Đại học Quốc gia Thành phố Hồ Chí Minh (VN)
268 Lý Thường Kiệt, phường 14, quận 10, thành phố Hồ Chí Minh

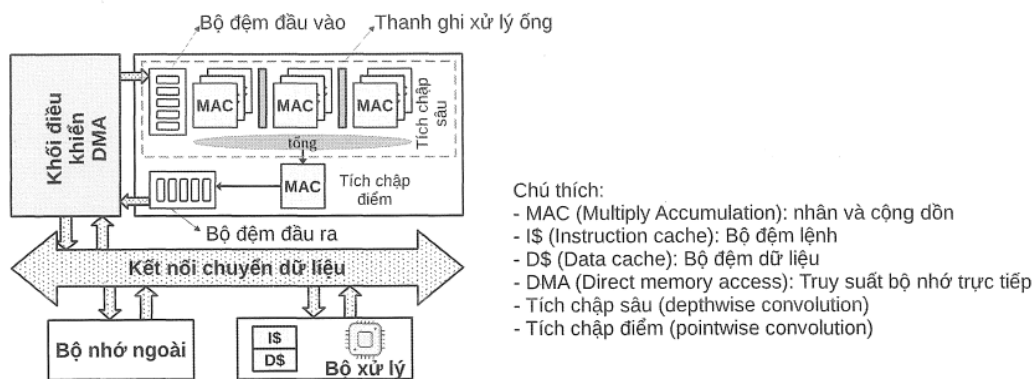
2. Đại học Quốc gia Thành phố Hồ Chí Minh (VN)

Khu phố 6, phường Linh Trung, thành phố Thủ Đức, thành phố Hồ Chí Minh

(72) Phạm Quốc Cường (VN).

(54) **PHƯƠNG PHÁP TĂNG TỐC MẠNG NƠI HỌC SÂU DÙNG KỸ THUẬT
ĐỒNG XỬ LÝ PHẦN CỨNG PHẦN MỀM VỚI CÔNG NGHỆ TÍNH TOÁN TÁI
CẤU HÌNH CHO CÁC HỆ THỐNG ĐIỆN TOÁN BIÊN**

(57) Công trình xây dựng một phương pháp tăng tốc mạng nơ ron học sâu trên nền tảng tính toán tái cấu hình hướng tới các hệ thống điện toán biên với tài nguyên phần cứng và năng lượng hạn chế. Phương pháp cung cấp một lõi phần cứng được hiện thực bằng ngôn ngữ đặc tả phần cứng có khả năng được cấu hình dựa trên kiến trúc và tham số của mạng nơ ron cần được xử lý. Mô hình xử lý ống được áp dụng nhằm tăng hiệu suất tính toán của lõi phần cứng tính toán mạng nơ ron và đạt được hiệu suất xử lý một điểm dữ liệu mỗi chu kỳ. Cách tiếp cận này không làm gia tăng tài nguyên tính toán và năng lượng tiêu thụ. Một kiến trúc tổng quan của toàn bộ hệ thống đồng xử lý phần cứng phần mềm được thiết kế trong công trình với kiến trúc kết nối chuyển dữ liệu dạng đường dây tuần tự và kỹ thuật truy xuất bộ nhớ trực tiếp để thực hiện việc chuẩn bị dữ liệu cho mạng nơ ron và trả kết quả xử lý mạng nơ ron. Trong kiến trúc này, Bộ xử lý đang dụng thực hiện các hàm phần mềm theo yêu cầu của ứng dụng sẽ triển khai trong khi lõi phần cứng chuyên dụng xử lý mạng nơ ron học sâu với cấu hình và tham số tùy thuộc vào ứng dụng.



Hình 1

Lĩnh vực kỹ thuật được đề cập

Công trình hướng đến việc xây dựng một phương pháp để tăng tốc các mạng nơ ron dùng trong kỹ thuật học sâu với cách tiếp cận đồng xử lý phần cứng phần mềm cho các công nghệ tính toán tái cấu hình. Phương pháp giúp kết quả thiết kế và hiện thực các mạng nơ ron trên các công nghệ tính toán tái cấu hình đạt được hiệu suất cao và tiêu thụ ít năng lượng. Qua đó, phương pháp hỗ trợ xây dựng các mô đun học sâu trên các thiết bị dùng trong điện toán biên có lượng tài nguyên phần cứng và năng lượng giới hạn. Các mô đun này sẽ giúp cho việc xử lý dữ liệu hiệu quả tại các nút tính toán biên góp phần tăng hiệu suất và giảm năng lượng tiêu thụ của toàn bộ hệ thống điện toán biên sử dụng công nghệ học sâu.

Tình trạng kỹ thuật của giải pháp hữu ích

Hiện tại, các ứng dụng trí tuệ nhân tạo sử dụng kỹ thuật học sâu đang ngày càng được quan tâm và hiện thực cho nhiều lĩnh vực khác nhau từ xử lý dữ liệu quan trắc đến giám sát giao thông và an ninh. Tuy nhiên, các ứng dụng này đòi hỏi hiệu suất tính toán lớn và tài nguyên lưu trữ nhiều khi lượng dữ liệu cần phải xử lý gia tăng. Ví dụ, mạng nơ ron học sâu theo mô hình VGG19 yêu cầu hơn 500MB bộ nhớ để lưu trữ dữ liệu và thực thi lên đến hơn 39 tỉ phép tính số thực dấu chấm động. Điều này dẫn đến các hệ thống xử lý điện toán biên không thể đáp ứng nhu cầu xử lý thời gian thực do các thiết bị dùng trong điện toán biên có tài nguyên phần cứng và khả năng tính toán giới hạn.

Khi định luật Moore và Dennard đạt đến giới hạn và chuẩn bị đi đến hồi kết thúc, các kiến trúc sư hệ thống tính toán đã và đang đề xuất một kỹ thuật phát triển hệ thống mới là kỹ thuật đồng xử lý phần cứng phần mềm để tối ưu quá trình xử lý các ứng dụng nhằm đạt được hiệu suất cao nhất. Kỹ thuật đồng xử lý phần cứng phần mềm có thể tiếp tục gia tăng hiệu suất của các hệ thống tính toán, bao gồm cả các hệ thống nhúng và hệ thống điện toán biên. Trong kỹ thuật đồng xử lý này, phần mềm sẽ xử lý phần ứng dụng không thể song song và không đòi hỏi phải xử lý nhiều phép tính. Trong khi đó, các lõi phần cứng chuyên biệt cho các phần cần xử lý nhiều phép tính sẽ được thiết kế cẩn thận và hiện thực với công nghệ tính toán tái cấu hình nhằm tận dụng khả năng song song của công nghệ này.

Đã biết thông tin về những hệ thống tăng mạng nơ ron học sâu sử dụng phần cứng tái cấu hình, tuy nhiên các hệ thống này hoặc là ứng dụng cho các hệ thống hiệu năng cao có lượng tài nguyên và năng lượng lớn hoặc là chỉ tập trung thiết kế các lõi tính toán phần cứng chuyên biệt thay vì áp dụng kỹ thuật đồng xử lý phần cứng phần mềm.

Hệ thống tăng tốc mạng nơ ron học sâu Aristotle (công bố trong “Software-hardware codesign for efficient neural network acceleration”), hệ thống của tác giả H. Li và cộng sự (công bố trong

“A high performance fpga-based accelerator for large-scale convolutional neural networks”), hệ thống của tác giả X. Lin và cộng sự (công bố trong “Lcp: a layer clusters paralleling mapping method for accelerating inception and residual networks on fpga”) và hệ thống của tác giả Z. Liu và cộng sự (công bố trong “Automatic code generation of convolutional neural networks in fpga implementation”) thiết kế lõi phần cứng cho việc xử lý mạng nơ ron học sâu sử dụng kỹ thuật song song hoá dựa vào lượng tài nguyên dồi dào của công nghệ tính toán tái cấu hình để đạt được hiệu suất tính toán cao. Tuy nhiên, các hệ thống này chỉ tập trung vào thiết kế lõi phần cứng và hướng đến tính toán hiệu năng cao thay vì tập trung vào điện toán biên và tối ưu cả phần cứng và phần mềm.

Hệ thống tăng tốc mạng nơ ron của tác giả Y. Yang và cộng sự (công bố trong “Synetgy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas”), hệ thống tăng tốc của tác giả M. Ghasemzadeh và cộng sự (công bố trong “Rebnet: Residual binarized neural network”) và hệ thống tăng tốc mạng nơ ron của tác giả D. Moss (công bố trong “High performance binary neural networks on the xeon+fpga™platform”) tập trung vào việc thiết kế lõi phần cứng cho mạng nơ ron để đạt được thời gian tính toán thấp bằng cách đơn giản hoá các phép tính thành nhị phân và chấp nhận sự suy giảm độ chính xác. Các hệ thống này cũng không tập trung vào đồng thiết kế phần cứng, phần mềm và cũng không hướng tới điện toán biên.

Hệ thống tăng tốc mạng nơ ron của tác giả J. Wang và cộng sự (công bố trong “Efficient hardware architectures for deep convolutional neural network”), hệ thống tăng tốc mạng nơ ron của tác giả L. Lu và cộng sự (công bố trong “Evaluating fast algorithms for convolutional neural networks on fpgas”), hệ thống tăng tốc mạng nơ ron của tác giả Y. Guan và cộng sự (công bố trong “Fp-dnn: an automated framework for mapping deep neural networks onto fpgas with rtl-hls hybrid templates”) tập trung vào việc đệm dữ liệu để giảm thời gian chuyển dữ liệu cho các lõi phần cứng thay vì đồng thiết kế phần cứng phần mềm nhằm tối ưu việc xử lý của toàn bộ hệ thống tính toán. Các hệ thống này cũng không hướng tới điện toán biên.

Bản chất kỹ thuật của

Công trình được thực hiện nhằm giải quyết vấn đề kỹ thuật nêu trên. Mục đích chính của công trình là đề xuất một phương pháp áp dụng kỹ thuật đồng thiết kế phần cứng phần mềm để tăng tốc độ xử lý mạng nơ ron học sâu hướng tới các hệ thống sử dụng trong điện toán biên với tài nguyên hạn chế và năng lượng tiêu thụ thấp. Phương pháp đề xuất xây dựng một kiến trúc lõi phần cứng hiệu quả áp dụng kỹ thuật xử lý ống (pipeline) để tối ưu hoá thông lượng dữ liệu được xử lý. Tận dụng khả năng tái cấu hình của công nghệ tính toán tái cấu hình, kiến trúc phần cứng trong phương pháp này có thay đổi các thông số để phù hợp với kích thước và trọng số của các mạng nơ ron học sâu khác nhau. Phương pháp đề xuất cũng cung cấp một kiến trúc hệ thống tổng thể bao gồm bộ xử lý đa dụng để xử lý các hàm của phần mềm, quản lý và điều khiển các lõi phần cứng và lõi phần cứng chuyên dụng nêu trên.

Để thực hiện mục đích này, nền tảng FPGA và các công cụ hỗ trợ tổng hợp FPGA được sử dụng như một phần của hệ thống được thiết kế. Cụ thể là việc sử dụng công cụ tổng hợp Quartus được cung cấp bởi hãng Intel Altera cho các chip FPGA tương ứng của hãng, công cụ tổng hợp ISE,

Vivado của Xilinx cho các chip tương ứng của Xilinx. Ngoài ra, kiến trúc đồng xử lý phần cứng phần mềm thiết kế với một bộ xử lý đa dụng và lõi phần cứng chuyên dụng được chúng tôi thiết kế riêng cho các nền tảng FPGA dùng trong điện toán biên hướng tới việc sử dụng ít tài nguyên và tiêu thụ ít năng lượng được sử dụng để cấu hình các mô đun điện toán biên.

Kiến trúc tính toán mạng nơ ron trên phần cứng tái cấu hình được hiện thực bằng ngôn ngữ đặc tả phần cứng Verilog sử dụng các tham số để có thể cấu hình phù hợp cho các mạng nơ ron có kích thước và tham số khác nhau. Tham số và kích thước mạng nơ ron được xác định trong quá trình huấn luyện trên phần mềm sử dụng các tập dữ liệu đã được kiểm định. Quá trình huấn luyện này có thể được thực hiện trên các nền tảng tính toán truyền thống. Mạng nơ ron sau khi đã được huấn luyện và xác định được cấu hình mạng cùng tham số sẽ được đồng xử lý bằng phần cứng và phần mềm trên kiến trúc hệ thống đồng xử lý gồm một bộ xử lý đa dụng (xử lý phần mềm) và lõi tính toán phần cứng xử lý mạng nơ ron học sâu do chúng tôi đề xuất.

Mô tả vắn tắt các hình vẽ

Hình 1 mô tả tổng quan phương pháp đồng xử lý phần cứng và phần mềm tăng tốc các mạng nơ ron học sâu dùng công nghệ tính toán tái cấu hình cho hệ thống điện toán biên.

Hình 2 mô tả kiến trúc lõi phần cứng chuyên dụng khả cấu hình dùng xử lý phép tính tích chập sâu cho mạng nơ ron học sâu áp dụng kỹ thuật xử lý ống.

Hình 3 mô tả chi tiết khối tính toán tích chập trong kiến trúc phần cứng chuyên dụng xử lý mạng nơ ron học sâu.

Hình 4 mô tả quá trình xử lý ống cho lõi phần cứng tính toán mạng nơ ron.

Mô tả chi tiết giải pháp hữu ích

Hình 1 mô tả tổng quan phương pháp đồng xử lý phần cứng phần mềm tăng tốc mạng nơ ron học sâu sử dụng công nghệ tính toán tái cấu hình cho hệ thống điện toán biên. Các bước tiến hành và cách thức thực hiện từng bước được mô tả như sau:

1. Cấu hình hệ thống: Mã nguồn của ứng dụng dựa trên mạng nơ ron học sâu được ghi vào Bộ nhớ ngoài để được xử lý bởi Bộ xử lý. Trong khi đó, kiến trúc mạng nơ ron học sâu sẽ được dùng để cấu hình số lượng MAC và số lượng Thanh ghi xử lý ống để thực hiện phép tính tích chập sâu.

2. Chuẩn bị dữ liệu: Sau khi hệ thống đã được cấu hình ở bước “Cấu hình hệ thống”, Bộ xử lý sẽ kích hoạt Khối điều khiển DMA sẽ tiến hành chép giá trị các tham số của mạng nơ ron học sâu vào các khối MAC đồng thời chuyển dữ liệu cần xử lý của mạng nơ ron học sâu từ bộ nhớ ngoài vào Bộ đệm đầu vào để tiến hành tính toán mạng nơ ron học sâu. Việc chuẩn bị dữ liệu này được thực hiện thông qua Kết nối chuyển dữ liệu. Do đặc thù tài nguyên và năng lượng hạn chế của các hệ thống điện toán biên, Kết nối chuyển dữ liệu là các kết nối dạng các đường dây tuần tự.

3. Tính toán mạng nơ ron: Khi toàn bộ dữ liệu đã sẵn sàng tại Bộ đệm đầu vào, Bộ xử lý sẽ kích hoạt việc tính toán tích chập sâu và tích chập điểm của lõi xử lý phần cứng dùng tăng tốc

mạng nơ ron. Lỗi phần cứng này gồm một chuỗi các khối MAC và các Thanh ghi xử lý ống đã được cấu hình dựa trên kiến trúc của mạng nơ ron cần tính toán ở bước “Cấu hình hệ thống”. Các khối MAC sẽ lần lượt đọc dữ liệu từ Bộ đệm đầu vào và tính toán các phép tích chập sâu và tích chập điểm để ghi kết quả tính toán vào Bộ đệm đầu ra cho đến khi toàn bộ dữ liệu được xử lý xong. Trong bước này, Bộ xử lý sẽ chờ kết quả tính toán hoặc thực hiện các hành vi tính toán khác tùy vào ứng dụng cụ thể.

4. Trả kết quả tính toán: Khi kết quả tính toán của lỗi phần cứng chuyên dụng cho mạng nơ ron học sâu đã hoàn toàn sẵn sàng ở Bộ đệm đầu ra, Bộ xử lý kích hoạt Khối điều khiển DMA thực hiện việc chuyển kết quả này về lại Bộ nhớ ngoài để Bộ xử lý tiếp tục xử lý ở các bước tiếp theo. Các bước tính toán tiếp theo phụ thuộc vào ứng dụng cụ thể và được xử lý bằng phần mềm đã cài đặt trên Bộ xử lý trong bước “Cấu hình hệ thống”.

5. Hoàn tất ứng dụng: Ở bước cuối cùng này, Bộ xử lý đảm nhiệm việc xử lý các phần còn lại của ứng dụng dựa trên kết quả tính toán mạng nơ ron học sâu của lỗi phần cứng chuyên dụng. Các bước tính toán này phụ thuộc vào ứng dụng cụ thể ví dụ như hiển thị thông tin cho người sử dụng, phát cảnh báo đến người quản lý hệ thống, hoặc phân loại đối tượng dựa vào kết quả tính toán. Do các hành vi tính toán phụ thuộc vào ứng dụng cụ thể nên không thể thực hiện các bước này bằng phần cứng chuyên dụng mà phải được thực hiện bằng Bộ xử lý đa dụng theo dạng xử lý phần mềm.

Hình 2 mô tả kiến trúc lỗi phần cứng dùng tính toán phép tích chập sâu cho các mạng nơ ron học sâu. Kiến trúc này áp dụng phương pháp xử lý ống (pipeline) để tăng tốc độ tính toán nhưng không làm gia tăng tài nguyên phần cứng như các phương pháp tính toán song song do kiến trúc này hướng đến điện toán biên. Ngoài ra, kiến trúc này tính toán trực tiếp trên dữ liệu gốc nhằm giữ nguyên được độ chính xác của mạng nơ ron thay vì biến đổi dữ liệu để giảm thời gian xử lý. Lỗi tính toán phần cứng cho phép tích chập sâu bao gồm một Bộ đệm dữ liệu vào chứa ma trận dữ liệu có kích thước $F \times F$. Kích thước này có thể cấu hình được nhờ đặc tính của công nghệ tái cấu hình. Giá trị kích thước F phụ thuộc vào lượng tài nguyên hiện hữu của nền tảng tính toán được sử dụng để triển khai phương pháp. Ở mỗi chu kỳ tính toán, các hàng dữ liệu sẽ được dịch sang phải cho đến khi hoàn toàn xử lý hết một hàng. Sau khi các hàng đầu tiên được xử lý xong thì sẽ thực hiện việc dịch lên, tức là hàng đầu tiên sẽ được bỏ đi và các hàng phía dưới sẽ được đưa lên thay thế. Quá trình này lặp lại cho đến khi xử lý xong toàn bộ dữ liệu. Tại mỗi chu kỳ, các khối MAC trên một cột sẽ thực hiện phép nhân và cộng dồn một cách song song và chuyển kết quả vào Thanh ghi xử lý ống để cột tiếp theo có thể sử dụng dữ liệu này tiếp tục tính toán ở chu kỳ tiếp theo. Đây chính là phương pháp xử lý ống (pipeline). Khi các khối MAC trong cột cuối cùng hoàn thành tính toán thì kết quả của các khối này được cộng lại với nhau và cộng với giá trị thiên vị bias đã được cấu hình trước đó để hình thành kết quả tính toán cho một điểm dữ liệu (một phần tử của ma trận kết quả $F' [i, j]$).

Hình 3 mô tả chi tiết Khối nhân và cộng dồn (MAC). Các khối cộng dồn gồm hai khối phần cứng thực hiện phép nhân và phép cộng trong đó giá trị $K[i, j]$ chính là một phần tử tham số của mạng nơ ron học sâu và được ghi vào trong khối MAC ở bước “Cấu hình hệ thống”. Giá trị $K[i, j]$ này

được nhân với giá trị dữ liệu đầu vào $F[x,y]$ có được từ Bộ đệm dữ liệu vào và cộng với kết quả tính toán của Khối MAC ở cột trước đó.

Hình 4 mô tả minh họa quá trình xử lý ống trong bốn chu kỳ đầu tiên với mạng nơ ron học sâu gồm bộ trọng số có ba hàng và ba cột (chín khối MAC được tổ chức theo mô hình ma trận 3×3). Ở Chu kỳ đầu tiên (Chu kỳ 1), các khối MAC trong Cột 1 sẽ thực hiện phép tính nhân và cộng dồn cho các phần tử dữ liệu đầu vào ở vị trí thứ nhất của ba hàng đầu tiên ($F[1,1]$, $F[2,1]$ và $F[3,1]$) được xử lý nhân và cộng dồn bởi ba khối MAC trong Cột 1 trong khi ba khối MAC trong Cột 2 và Cột 3 trong trạng thái chờ vì chưa nhận được dữ liệu. Ở Chu kỳ 2, kết quả xử lý của ba khối MAC trong Cột 1 ở Chu kỳ 1 được chuyển sang cho các khối MAC trong Cột 2 để cộng dồn với kết quả phép nhân các phần tử dữ liệu ở vị trí thứ hai của ba hàng đầu tiên ($F[1,2]$, $F[2,2]$ và $F[3,2]$). Cũng trong Chu kỳ này, các khối MAC trong Cột 1 sẽ tiếp tục xử lý phần tử thứ hai của ba hàng đầu dữ liệu đầu vào trong khi các khối MAC trong Cột 3 vẫn tiếp tục chờ dữ liệu từ các khối MAC trong Cột 2. Tại Chu kỳ 3, tất cả các khối MAC trong các cột sẽ đồng loạt xử lý phần tử thứ ba của ba hàng đầu tiên ($F[1,3]$, $F[2,3]$ và $F[3,3]$). Các khối MAC trong Cột 2 cộng dồn phép nhân với kết quả của các khối MAC trong Cột 1 sinh ra ở Chu kỳ 2 trong khi các khối MAC trong Cột 3 cộng dồn phép nhân với kết quả của các khối MAC trong Cột 2 sinh ra ở Chu kỳ 2. Cuối Chu kỳ 3, sau khi tất cả các khối MAC trong ba cột đã hoàn thành, kết quả xử lý của điểm dữ liệu đầu tiên được hình thành ($F'[1,1]$). Từ đó, ở mỗi chu kỳ tiếp theo, kết quả của điểm ảnh tiếp theo sẽ được hình thành, $F'[1,2]$ trong Chu kỳ 4, $F'[1,3]$ trong Chu kỳ 5.

YÊU CẦU BẢO HỘ

1. Phương pháp tăng tốc mạng nơ ron học sâu trên nền tảng công nghệ tính toán tái cấu hình bằng cách sử dụng kỹ thuật đồng xử lý phần cứng phần mềm, trong đó lõi phần cứng được thiết kế chuyên biệt có khả năng được cấu hình theo kiến trúc và tham số của mạng nơ ron cần xử lý, phương pháp bao gồm các bước:

cấu hình hệ thống: kiến trúc và tham số mạng nơ ron được dùng để cấu hình số lượng khối MAC và giá trị tham số K trong phép nhân và cộng dồn của lõi phần cứng tính toán mạng nơ ron;

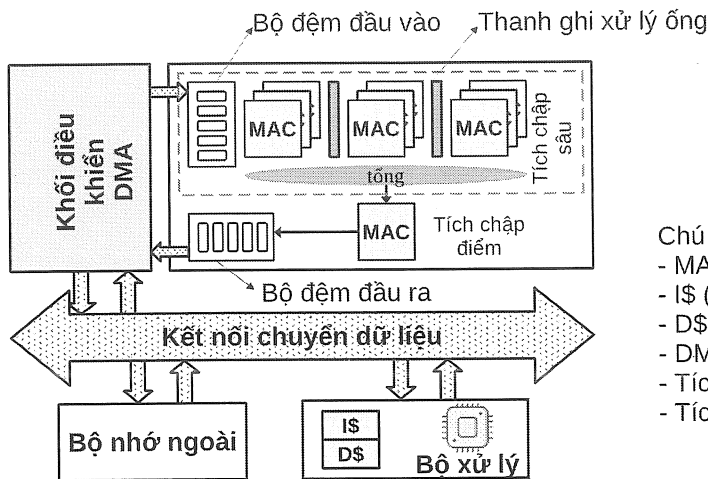
chuẩn bị dữ liệu: kỹ thuật truy xuất bộ nhớ trực tiếp được sử dụng để chuyển dữ liệu đầu vào cần xử lý của mạng nơ ron từ bộ nhớ ngoài đến Bộ đệm dữ liệu vào thông qua kết nối dạng đường dây tuần tự để các khối MAC trong lõi phần cứng xử lý;

tính toán mạng nơ ron: các khối MAC được tổ chức theo dạng cột và được phân cách bởi các thanh ghi xử lý ống lần lượt đọc dữ liệu từ Bộ đệm dữ liệu vào và thực hiện các phép nhân và cộng dồn theo mô hình xử lý ống để tạo ra kết quả đầu ra ở mỗi chu kỳ xử lý và ghi vào Bộ đệm đầu ra;

tra kết quả tính toán: kết quả xử lý mạng nơ ron trong Bộ đệm đầu ra được chuyển ra bộ nhớ ngoài bằng phương pháp truy xuất bộ nhớ trực tiếp thông qua kết nối dạng đường dây tuần tự để tiếp tục được xử lý bằng phần mềm bởi Bộ xử lý đa dạng;

hoàn tất ứng dụng: phần mềm triển khai trên Bộ xử lý đa dụng sẽ tiếp tục xử lý kết quả đầu ra của mạng nơ ron đã được tính toán bởi lõi phần cứng chuyên dụng nhằm hoàn tất toàn bộ ứng dụng theo yêu cầu.

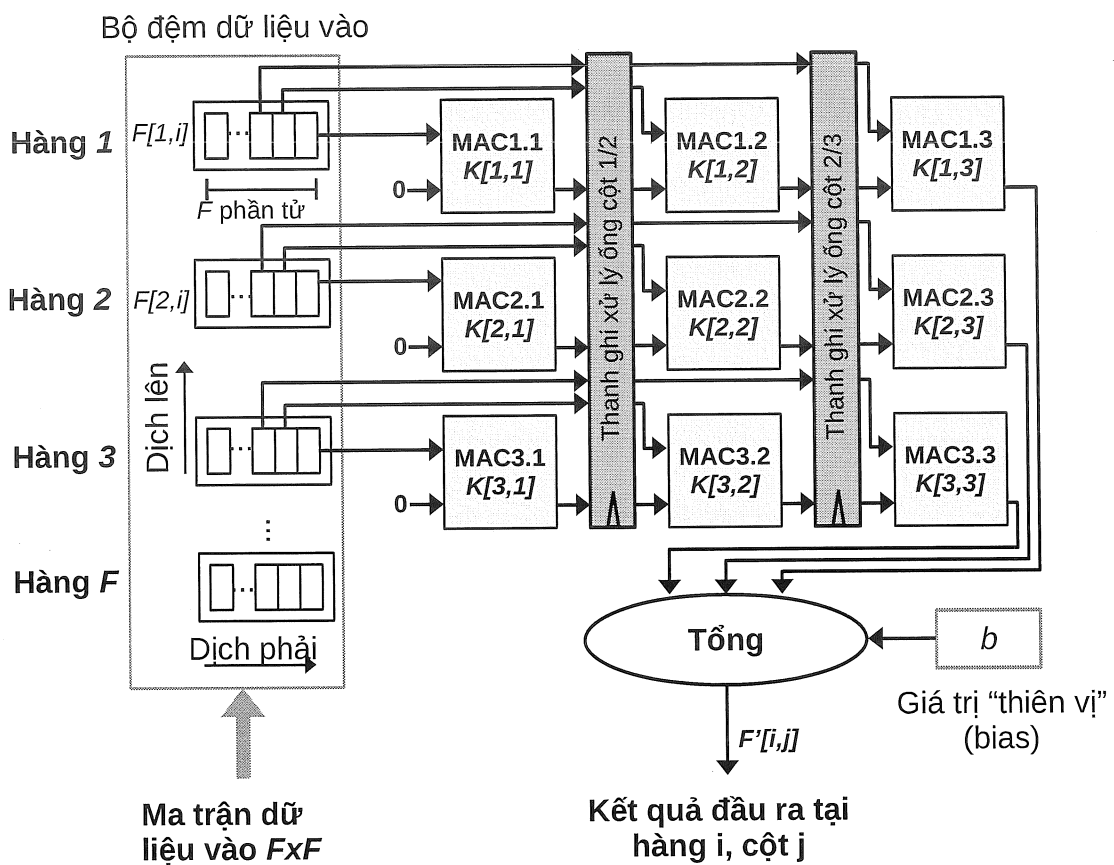
2. Phương pháp theo điểm 1, trong đó tích hợp kiến trúc lõi phần cứng chuyên biệt với các khối MAC có số lượng có thể cấu hình và được tổ chức theo các cột và phân tách bởi các thanh ghi xử lý ống.
3. Phương pháp theo điểm 1, trong đó tích hợp khả năng cấu hình lõi phần cứng hướng tới hệ thống điện toán biên với tài nguyên và năng lượng hạn chế nhưng vẫn tăng tốc độ xử lý mạng nơ ron nhờ vào mô hình xử lý ống.



Chú thích:

- MAC (Multiply Accumulation): nhân và cộng dồn
- I\$ (Instruction cache): Bộ đệm lệnh
- D\$ (Data cache): Bộ đệm dữ liệu
- DMA (Direct memory access): Truy suất bộ nhớ trực tiếp
- Tích chập sâu (depthwise convolution)
- Tích chập điểm (pointwise convolution)

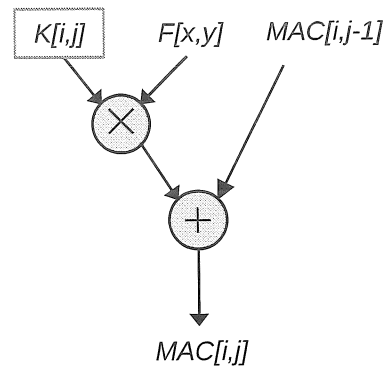
Hình 1



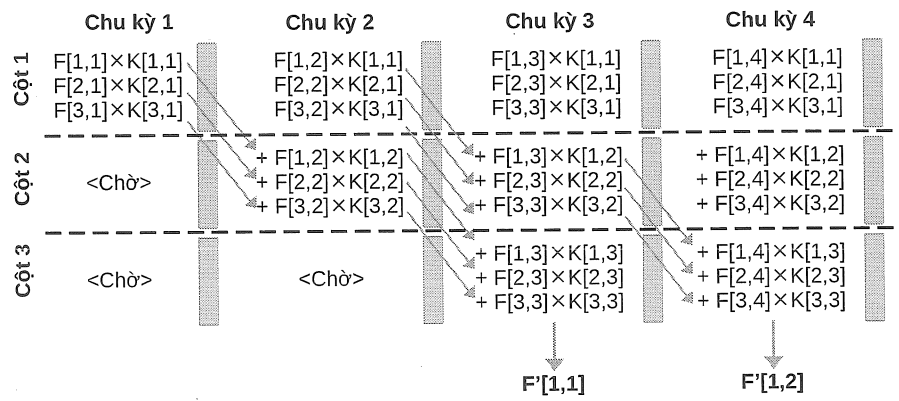
Chú thích:

- $K[i,j]$: giá trị tham số hàng i cột j
- MAC (Multiply Accumulation): nhân và cộng dồn

Hình 2



Hình 3



Hình 4