

Tế bào Nơron nhân tạo có độ chính xác và tốc độ cao

Nguyễn Quang Anh, Nguyễn Hoàng Dũng*

Trường Đại học Bách Khoa Hà Nội, Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội, Việt Nam

Nhận ngày 16 tháng 12 năm 2016

Chỉnh sửa ngày 18 tháng 01 năm 2017; Chấp nhận đăng ngày 23 tháng 03 năm 2017

Tóm tắt: Bài báo này tập trung trình bày thiết kế tế bào nơron nhân tạo với phương pháp học giám sát có khả năng thích ứng với nhiều thuật toán đòi hỏi độ chính xác và tốc độ cao. Dựa trên thuật toán huấn luyện có giám sát và cấu tạo nơron thực, nhóm nghiên cứu xây dựng một kiến trúc nơron nhân tạo có kiến trúc tương tự đi kèm bộ xử lý số thực. Kiến trúc này dễ dàng tăng tốc độ xử lý bằng cách mở rộng số tầng thực hiện mô phỏng theo cấu trúc đường ống (pipeline). Để đảm bảo tốc độ và độ chính xác cao, nhóm nghiên cứu đã thực hiện tối ưu một số kiến trúc bộ dịch và bộ xử lý số thực song song. Chính vì vậy khi tăng thêm số tầng cho kiến trúc thì tốc độ tăng lên rất nhanh trong khi tài nguyên tăng lên không đáng kể. Kết quả tổng hợp trên chip FPGA Virtex 6 của hãng Xilinx cho thấy kiến trúc nơron của nhóm nghiên cứu đề xuất có thể hoạt động lên đến 5 tầng thực hiện theo cấu trúc pipeline và tốc độ đạt được tối đa là 108Mhz.

Từ khóa: nơron nhân tạo, xử lý số thực, đường ống, bộ dịch.

1. Giới thiệu chung

Mạng nơron nhân tạo (Artificial Neural Network) là một trong những công cụ phi tuyến để mô hình hóa các mối quan hệ phức tạp giữa dữ liệu đầu vào và kết quả đầu ra từ một tập mẫu dữ liệu. Mạng nơron gồm một nhóm các tế bào nơron nhân tạo nối với nhau để xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị tại các lớp nơron. Có ba hướng huấn luyện mạng nơron là học có giám sát, học không giám sát và học bán giám sát. Mỗi hướng huấn luyện đều có những ưu, nhược điểm khác nhau. Nhưng để đạt độ chính xác cao nhất, nhóm nghiên cứu sử dụng mô hình học có giám sát. Với các tham số khởi tạo và cơ chế xấp xỉ hàm tùy ý, sau khi huấn luyện thì mạng có thể xử lý

tương đối tốt các dữ liệu quan sát được và cho ra kết quả chính xác.

Các nghiên cứu [1-3] thường là sử dụng thuật toán và chạy trên máy tính với CPU và GPU tốc độ cao không mang tính gọn nhẹ. Tiêu biểu nhất là nhận dạng trên mã nguồn mở OpenCV. Do đó khi sử dụng theo cách này thì sẽ khó đáp ứng được các ứng dụng đòi hỏi sản phẩm có kích thước nhỏ gọn và sử dụng nguồn pin. Để tập trung vào một số ứng dụng cụ thể nhằm tăng tốc độ và giảm kích thước cũng như công suất tiêu thụ phần cứng, nhóm nghiên cứu đã tiến hành triển khai, thực nghiệm một phần của ứng dụng trên nền tảng phần cứng FPGA. Hầu hết các nghiên cứu trước đây thường chỉ thực hiện được trên số nguyên hoặc là số thực có dấu phẩy cố định vì những ứng dụng đó không cần độ chính xác cao. Ý tưởng thiết kế ra tế bào nơron nhân tạo chuẩn trên nền tảng FPGA để dễ dàng tương thích và sử dụng cho

* Tác giả liên hệ. ĐT: 84-913004120.
Email: dung.nguyenhoang@hust.edu.vn

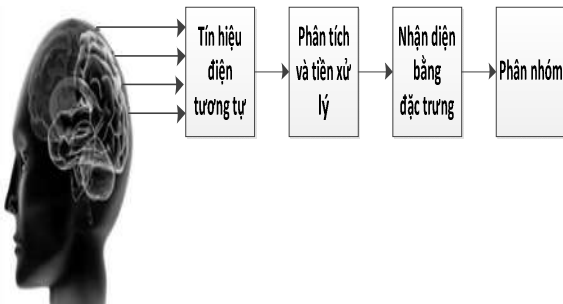
các thuật toán số học đòi hỏi độ chính xác cao mà tài nguyên sử dụng cần phải tiết kiệm cũng như đảm bảo tốc độ xử lý. Thiết kế dựa trên nghiên cứu neuron thực, nghiên cứu kiến trúc neuron nhân tạo song song [4] và kiến trúc neuron nhân tạo nối tiếp [5]. Trong đó, nhóm nghiên cứu sử dụng đã thiết kế bộ xử lý số thực theo chuẩn IEEE 754 [6] và bộ dịch bit mới để đảm bảo độ chính xác cũng như tốc độ thực hiện.

Trong bài báo này nhóm nghiên cứu sẽ trình bày tổng quan về mạng neuron và các nghiên cứu liên quan trong phần II; thiết kế bộ xử lý số thực theo chuẩn IEEE 745 [6] và bộ dịch bit để tăng độ chính xác và tốc độ trong phần III; các kết quả mô phỏng và thảo luận trong phần IV và kết luận ở phần V.

2. Tổng quan về mạng neuron và các nghiên cứu liên quan

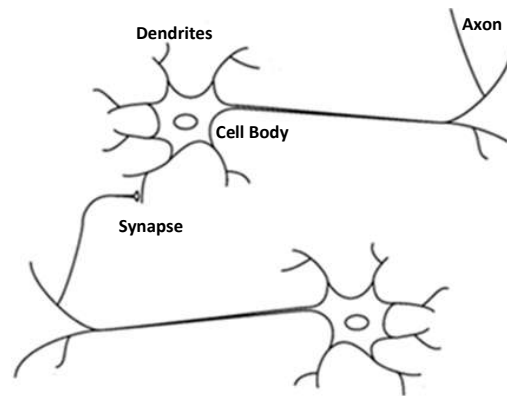
2.1. Tổng quan về mạng neuron

Hình 1 biểu diễn mô hình xử lý thông tin của con người [7]. Thông tin từ môi trường được đưa về não bộ của con người thông qua các giác quan và sẽ được bộ xử lý. Quá trình này được chia ra thành các khối như (1) khối tín hiệu điện tương tự; (2) khối phân tích và tiền xử lý; (3) khối nhận diện bằng đặc trưng và (4) phân chia ra thành các nhóm thông tin khác nhau. Trong não bộ của con người chứa đến hơn 100 tỉ neuron thần kinh (tế bào thần kinh) với chức năng chính truyền dẫn các xung điện. Neuron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là một phần quan trọng nhất của não.



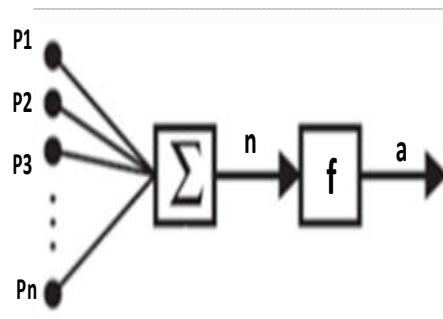
Hình 1. Mô hình xử lý thông tin của con người.

Cấu tạo của một neuron thật trong não người được minh họa trong hình 2. Một neuron gồm có thân neuron (cell body) là nơi xử lý các tín hiệu được đưa vào từ các giác quan. Các dây hình nhánh cây (dendrites) là nơi nhận các xung điện vào trong neuron và các sợi trục (axons) là một dây dài đưa xung điện ra sau quá trình xử lý từ thân của neuron. Giữa các dây hình nhánh cây và các sợi trục có một liên kết với nhau gọi là khớp thần kinh (synapse).



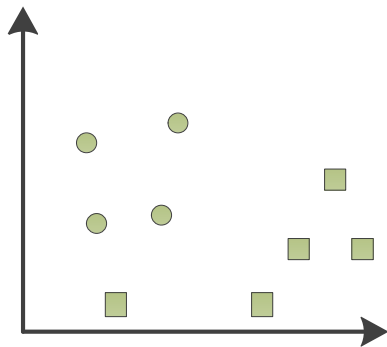
Hình 2. Cấu tạo của một neuron thật trong não người.

Dựa vào cấu tạo của một neuron thật trong não người, nhóm nghiên cứu đưa ra mô hình cấu tạo của neuron nhân tạo trong hình 3 [8]. Trong đó P1, P2 đến Pn lần lượt là các đầu vào của mạng neuron nhân tạo. Tổng của các đầu vào này sau khi nhân với một trọng số nhất định và trừ đi ngưỡng cần so sánh để được sự chính xác cao, sẽ kí hiệu là giá trị n. F là hàm dùng để lọc ngưỡng giá trị n và kết quả đầu ra của mạng neuron nhân tạo là a.

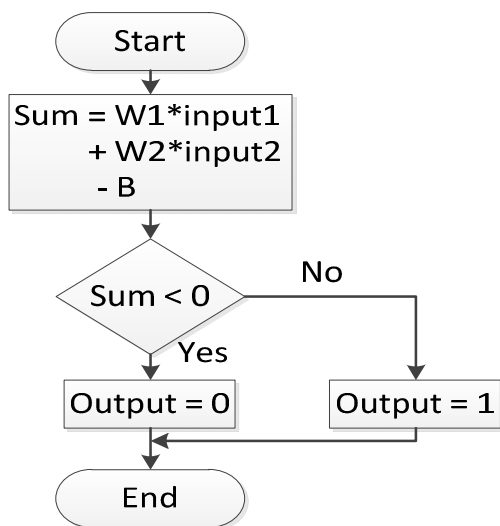


Hình 3. Mô hình neuron nhân tạo.

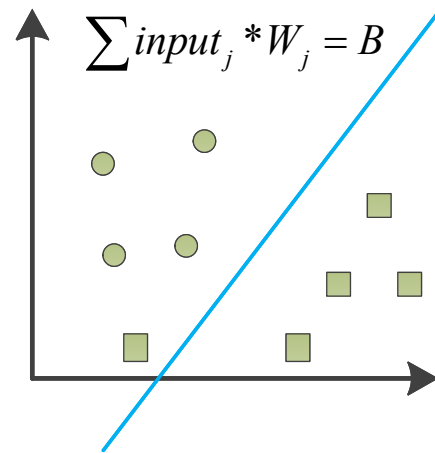
Giả sử có mô hình hai tập dữ liệu kí hiệu hình tròn và vuông cần phân loại như hình 4. Thuật toán xử lý dữ liệu trong mạng nơron nhân tạo khi chưa huấn luyện được mô tả trong hình 5. Các kí hiệu W là trọng số của mạng nơron nhân tạo và B là giá trị ngưỡng xử lý. Ở tại mô hình này có hai tập dữ liệu vào input 1 và input 2. Dữ liệu đầu vào sẽ được nhân với trọng số tương ứng W_1 và W_2 rồi trừ đi ngưỡng B và sau đó mang ra so sánh. Hình 6 biểu diễn kết quả của mô hình đang bị lỗi khi một phần tử hình vuông bị phân loại nhầm sang bên tập dữ liệu các phần tử hình tròn.



Hình 4. Hai tập dữ liệu cần phân loại riêng giữa kí hiệu tròn và vuông.

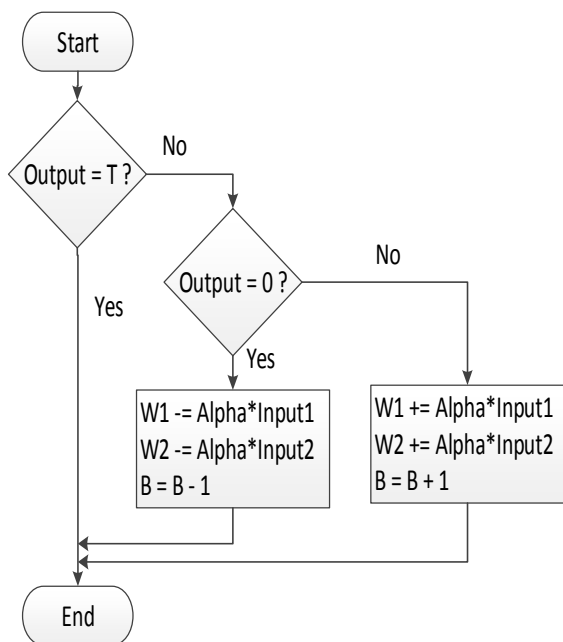


Hình 5. Thuật toán xử lý dữ liệu trong mạng nơron nhân tạo.

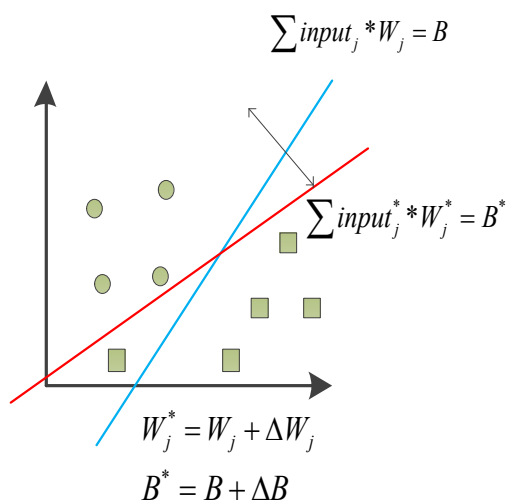


Hình 6. Mô hình phân loại đang có lỗi khi mới khởi tạo trọng số.

Chính vì vậy việc sử dụng thuật toán huấn luyện cho mạng nơron nhân tạo là rất cần thiết để phân loại chính xác các tập dữ liệu đầu vào. Hình 7 và hình 8 lần lượt biểu diễn mô hình thuật toán huấn luyện trong mạng nơron và kết quả phân loại sau khi đã điều chỉnh trọng số từ quá trình học. Sau khi phát hiện có lỗi trong quá trình phân loại như hình 6, trọng số sẽ được cập nhật lại dựa theo kết quả chuẩn và kết quả phân tích ra được. Kết quả quá trình phân loại đã được biểu diễn trong hình 8. Kết quả đưa ra sau khi tính toán như lưu đồ trong hình 5 sẽ được so sánh với một giá trị T để huấn luyện cho mẫu dữ liệu. Trong trường hợp giá trị đầu ra bằng T thì mẫu này đã được học đúng và sẽ không cần thay đổi. Nếu giá trị đầu ra khác với T thì mạng nơron phải được huấn luyện lại cho đến khi ra kết quả đúng. Trong trường hợp giá trị đầu ra bằng 0 và T bằng 1 thì trọng số W_i sẽ được trừ đi một lượng giá trị phụ thuộc đầu vào tương ứng nhân với tốc độ học tập α . Ngược lại nếu giá trị đầu ra bằng 1 và T bằng 0 thì trọng số W_i sẽ được tăng lên một lượng giá trị phụ thuộc đầu vào tương ứng nhân với tốc độ học tập α . Như vậy sau một số lần học lại, đường phân loại hai tập phần tử hình vuông và tròn đã được thay sang đường màu đỏ và kết quả phân loại chính xác.



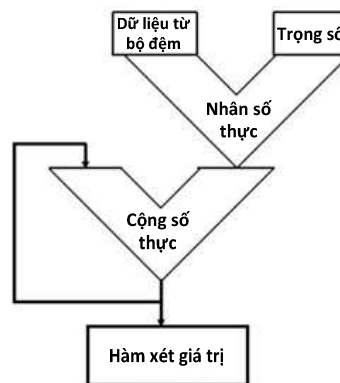
Hình 7. Thuật toán học trong mạng nơron nhân tạo.



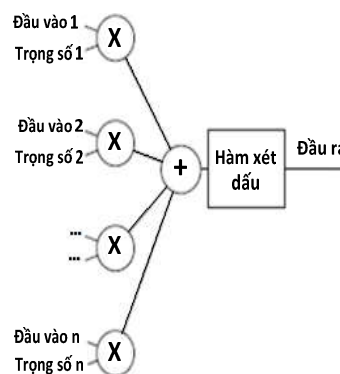
Hình 8. Mô hình phân loại sau khi điều chỉnh trọng số từ quá trình học.

2.2. Các nghiên cứu liên quan

Tế bào nơron nhân tạo thường có hai hướng xử lý chính là đưa dữ liệu liên quan vào theo kiểu nối tiếp hoặc theo kiểu song song. Hình 9 và 10 lần lượt biểu diễn mô hình mạng nơron xử lý nối tiếp [9] và xử lý song song [4].



Hình 9. Mô hình mạng nơron xử lý.



Hình 10. Mô hình mạng nơron xử lý song song.

Ở hình 9 dễ dàng nhận thấy với một tập dữ liệu đầu, chỉ có một luồng đầu vào duy nhất kết hợp với trọng số tương ứng đi vào bộ nhân. Sau đó kết quả được cộng tích lũy lại cho tới khi tính toán hết các dữ liệu đầu vào. Ngược lại, ở hình 10 có rất nhiều dữ liệu đầu vào khác nhau nhân với trọng số tương ứng sau đó đi vào một bộ cộng song song để tạo ra tín hiệu xử lý cho nơron nhân tạo. Như vậy nhóm nghiên cứu thấy rằng kiến trúc xử lý nối tiếp có cấu trúc và trạng thái điều khiển rất phức tạp. Kiến trúc này chỉ phù hợp để tính toán riêng lẻ, khó mở rộng số tầng tính toán khi hoạt động thành mạng nơron phức tạp. Trong khi đó với kiến trúc xử lý song song, tín hiệu được tính toán với thông lượng lớn do thực hiện các phép tính song song, đặc biệt là bộ cộng tổng hợp tín hiệu. Nếu thực hiện các phép tính cộng trừ song song sẽ giảm được rất nhiều thời gian đợi.

Bảng 1. Ưu nhược điểm của mạng nơron xử lý song song và nối tiếp

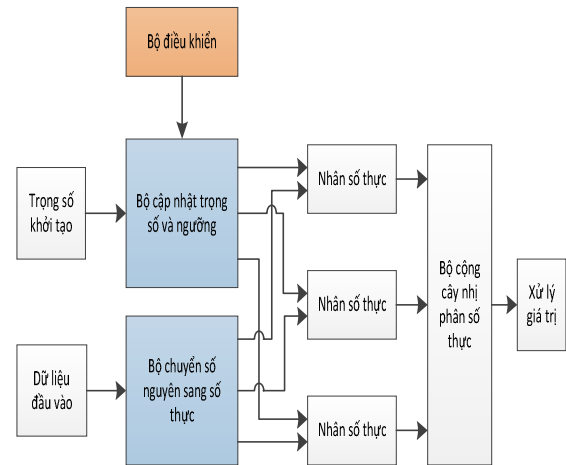
Mô hình mạng nơron	Ưu điểm	Nhược điểm
Xử lý song song	<ul style="list-style-type: none"> - Xử lý dữ liệu nhanh, đảm bảo tính toán thời gian thực. - Tăng tỉ lệ dữ liệu đầu vào hiệu dụng và tiết kiệm tài nguyên bộ nhớ. - Dễ đưa kỹ thuật pipeline để tăng tốc độ tính toán. - Tiết kiệm được khá nhiều tài nguyên công logic phân cứng. 	<ul style="list-style-type: none"> - Số bộ nhân và bộ cộng tăng lên tương ứng với số đường dữ liệu vào. - Nếu có số lượng đường dữ liệu vào lớn sẽ ảnh hưởng đến yêu cầu về kích thước của vi mạch thực hiện.
Xử lý nối tiếp		<ul style="list-style-type: none"> - Khó xây dựng hàng đợi và đồng bộ dữ liệu đầu vào khi thực hiện trên nhiều lớp nơron. - Số thanh ghi và flipflop tăng lên để lưu giữ giá trị khi tiết kiệm công logic. - Bộ điều khiển trở nên phức tạp và khó kiểm soát. - Thời gian sẽ chậm khi dữ liệu vào phải tính toán nối tiếp. - Chu kỳ một vòng tính toán lớn.

Dựa trên các nghiên cứu đó, nhóm nghiên cứu xây dựng một bảng so sánh các ưu nhược điểm của mô hình mạng nơron xử lý nối tiếp và xử lý song song như trình bày trong bảng 1. Với những ưu nhược điểm của từng mô hình, nhóm nghiên cứu nhận thấy mô hình xử lý dữ liệu song song có nhiều ưu điểm hơn hẳn so với mô hình nối tiếp. Tuy nhiên để hạn chế các nhược điểm của mô hình này, nhóm nghiên cứu đề xuất và trình bày một số cải tiến cho mô hình xử lý dữ liệu song song trong phần tiếp theo của bài báo này.

3. Kiến trúc mạng nơron song song

3.1. Cải tiến mô hình mạng nơron xử lý dữ liệu song song

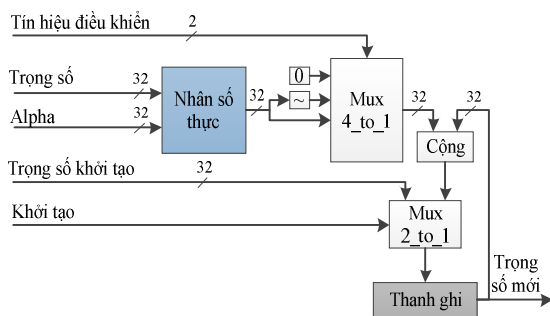
Trong thiết kế kiến trúc mạng nơron của Christodoulou [10] sử dụng bộ cộng nối tiếp cho (N-1) đầu vào sau khi nhân với trọng số. Phương pháp này thiếu hợp lý vì đã không tối ưu được nền tảng phần cứng và làm tăng thời gian trễ từ bộ nhân cho tới bộ lọc giá trị. So với cấu tạo của một mạng nơron thật sự, nhóm nghiên cứu đã đưa thêm các đường tín hiệu điều khiển và đường dữ liệu vào để thuận tiện hơn trong quá trình xử lý. Hình 11 minh họa sơ đồ khối của một mạng nơron nhân tạo có kiến trúc xử lý song song được nhóm nghiên cứu đề xuất.



Hình 11. Đề xuất mô hình mạng nơron xử lý song song.

Kiến trúc này chỉ cần nạp trọng số vào một lần lúc khởi động và sau đó cập nhật cùng với giá trị lưu trữ song song bên trong mạng nơron. Một bộ cập nhật trọng số gồm 2 thành phần chính là các bộ nhân và các bộ cộng số thực. Như vậy, mỗi một đường dữ liệu đi vào thiết kế chỉ cần cần thêm 2 bộ cộng số thực và 2 bộ nhân số thực. Nghiên cứu của Pierre Auger Collaboration [11] đã chỉ xử lý được cho duy nhất số nguyên nên kiến trúc đó sẽ không dùng được vào những ứng dụng lớn. Trong khi đó thiết kế của Jeannette Chin [12] sử dụng 5 bit cho phần thập phân hay thiết kế của Valeri

Mladenov [4] sử dụng 10 bit cho phần thập phân cũng chỉ có thể xử lý tới sai số cỡ phần nghìn. Ngoài ra, thiết kế Alessandro [13] sử dụng một nhân vi điều khiển NIOS trong chip *FPGA* chỉ để đọc dữ liệu từ RAM và xử lý số thực là hoàn toàn lãng phí tài nguyên trong khi trên thực tế chỉ cần xây dựng một bộ xử lý số thực để nâng cao độ chính xác cho các ứng dụng. Chính vì vậy, nhóm nghiên cứu thiết kế bộ nhân và cộng bằng cách cải tiến thêm số module toán tử trên nền tảng kiến trúc bộ cộng và nhân số thực của Prof. Al-Khalili [14] theo chuẩn IEEE 754. Với kiến trúc không xử lý số thực, tần số hoạt động tối đa của mạng nơron dễ dàng đạt được đến 200MHz. Tuy nhiên, đối với kiến trúc xử lý số thực, để xử lý được cần nhiều công đoạn nhân, dịch và đếm nối tiếp nên tần số hoạt động tối đa sẽ bị giảm đi nhiều.



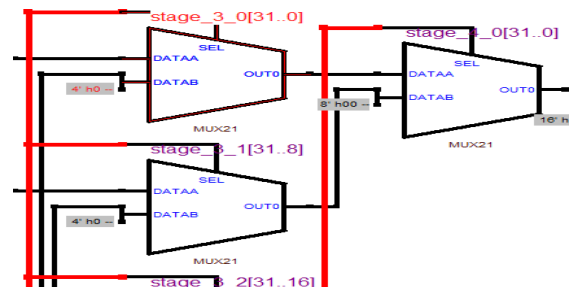
Hình 12. Cấu trúc bộ cập nhật trọng số.

Cấu trúc bộ cập nhật trọng số được biểu diễn trong hình 12. Hình 13 mô tả một phần cấu trúc của bộ dịch bit. Với bộ nhân sử dụng khối dịch bit theo từng xung đồng hồ, thời gian để thực hiện tính toán phép nhân số thực sẽ tăng chu kỳ thực hiện lên rất nhiều. Đối với phép dịch theo xung đồng hồ, trường hợp tốt nhất là không mất xung nào và xấu nhất là mất số xung bằng đúng kích thước đầu vào của bộ dịch. Tiêu biểu cho trường hợp này là thiết kế của Hung-Ming Tsai [15]. Với kiến trúc dịch bit và xử lý nối tiếp, mỗi lớp nơron của Réjean Fontaine [16] cần sử dụng tới 46 xung đồng hồ để tính toán. Nếu muốn sử dụng kiến trúc này để thực hiện ứng dụng thời gian thực thì tần số hoạt động tối đa của nơron phải rất lớn. Vì vậy, nhóm nghiên cứu đề xuất kiến trúc bộ dịch bit

song song 32 bit trên mạch tổ hợp dựa vào phương pháp chia đôi và lựa chọn bit. Ý tưởng của phương pháp này là nếu có một số 32 bit cần dịch đi một số lượng bit $0 \leq n \leq 32$. Biến đổi giá trị n ra dạng số nhị phân có 5 bit từ [4:0] với bit 0 là bit có trọng số thấp nhất (Least Significant Bit – LSB) và bit 4 là bit có trọng số cao nhất (Most Significant Bit – MSB). Như vậy, giá trị $n = b_4 * 16 + b_3 * 8 + b_2 * 4 + b_1 * 2 + b_0 * 1$. Với bộ dịch số có n bit thì số tầng dịch sẽ là $\log_2 n$. Thuật toán được mô tả như sau:

- Xử lý bit LSB để xác định số 32 bit cần dịch số bit chẵn hay số bit lẻ.
- Xử lý bit thứ 1 để xác định số 32 bit cần dịch đi thêm 2-bit hay không?
- Xử lý bit thứ 2 để xác định số 32 bit cần dịch đi thêm 4-bit hay không?
- Xử lý bit thứ 3 để xác định số 32 bit cần dịch đi thêm 8-bit hay không?
- Xử lý bit MSB để xác định số 32 bit cần dịch đi thêm 16-bit hay không?

Thực hiện tương tự như bộ dịch bit song song, bộ đếm sẽ đếm số bit 0 từ trái sang phải cho tới khi gặp bit 1.

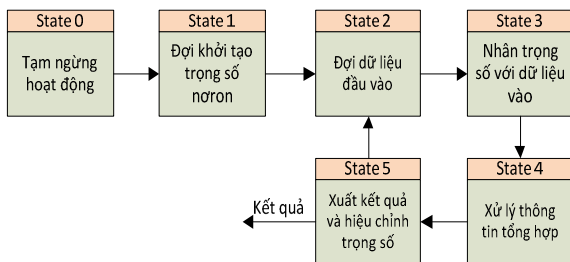


Hình 13. Một phần cấu trúc của bộ dịch bit.

3.2. Trạng thái hoạt động của mạng nơron xử lý dữ liệu song song

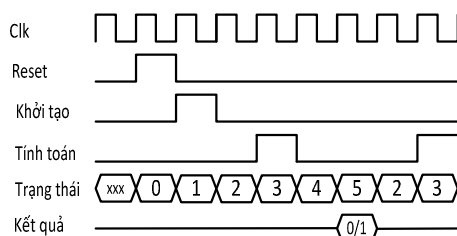
Trạng thái hoạt động của mạng nơron có kiến trúc xử lý song song do nhóm nghiên cứu đề xuất được biểu diễn lần lượt trong hình 14. Khi bắt đầu khởi động tế bào nơron trong mạng nơron thì trạng thái bên trong của tế bào sẽ không xác định được. Để đảm bảo tính chính xác nên hệ thống cần phải thiết lập lại để đưa về trạng thái 0 – trạng thái tạm ngừng hoạt động. Sau đó tế bào nơron sẽ tiến vào trạng thái 1 –

trạng thái đợi khởi tạo trọng số. Trạng thái 2 sẽ là trạng thái đợi dữ liệu đầu vào để tính toán. Trạng thái 3 sẽ thực hiện nhân dữ liệu đầu vào này với trọng số đã được khởi tạo ở trạng thái 1. Kết quả của trạng thái 3 sẽ được chuyển sang trạng thái 4 để xử lý thông tin tổng hợp. Ở trạng thái 5 cuối cùng, tế bào nơron sẽ xuất kết quả và tiến hành hiệu chỉnh trọng số nếu cần thiết để kết quả lần sau chính xác hơn và sau đó sẽ lặp lại về trạng thái 2 đợi dữ liệu cho đến khi kết thúc hoàn toàn.



Hình 14. Trạng thái hoạt động của mạng nơron nhân tạo xử lý song song do nhóm nghiên cứu đề xuất.

Hình 15 minh họa giản đồ thời gian hoạt động của mạng nơron nhân tạo xử lý song song do nhóm nghiên cứu đề xuất. Tín hiệu clock được sử dụng để điều khiển hoạt động của hệ thống. Tín hiệu reset sẽ đưa tế bào nơron về trạng thái tạm ngừng hoạt động trước khi có tín hiệu khởi tạo được đưa vào để khởi tạo trọng số tương ứng. Sau khi dữ liệu được đưa vào ở trạng thái 2 thì tín hiệu tính toán sẽ thiết lập ở mức 1 để tính toán phép nhân trọng số với dữ liệu đầu vào tại trạng thái 3, xử lý thông tin tổng hợp ở trạng thái 4 và kết thúc một chu trình làm việc ở trạng thái 5 trước khi quay về trạng thái 2 như tín hiệu trạng thái đã biểu diễn ở trong hình 15. Kết quả sẽ được tính toán và cập nhật ở trạng thái cuối cùng đó.

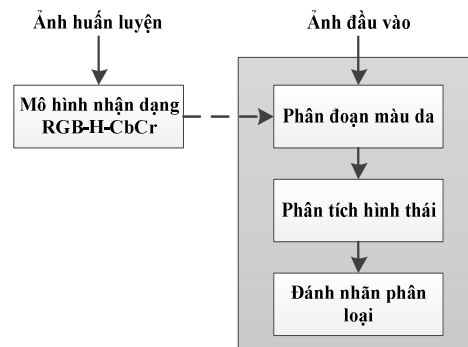


Hình 15. Giản đồ thời gian hoạt động của mạng nơron nhân tạo xử lý song song do nhóm nghiên cứu đề xuất.

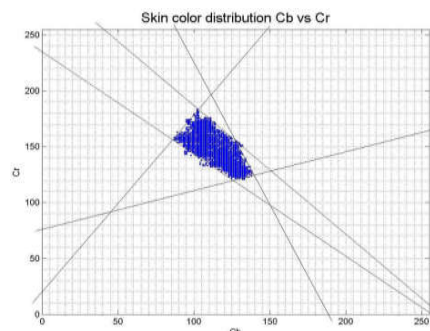
4. Kết quả và thảo luận

4.1. Kiểm thử mô hình đề xuất trên nền tảng phần cứng

Dựa vào thiết kế xử lý theo kiến trúc song song, nhóm nghiên cứu tiến hành thử nghiệm vào ứng dụng nhận diện khuôn mặt của John See [17]. John See đã thử nghiệm thành công việc nhận dạng với hơn 600 khuôn mặt dựa trên tập 100 bức ảnh mẫu theo mô hình tại hình 16 với công cụ sử dụng là phần mềm máy tính. Từ thực nghiệm cụ thể, John See đã tìm ra những biểu thức để phân loại màu da theo đồ thị trong hình 17. Nhóm nghiên cứu sẽ tiến hành mô phỏng bằng cách tạo một tập dữ liệu huấn luyện trên phần mềm máy tính và tiến hành kiểm chứng dữ liệu bằng cách so sánh với dữ liệu của John See. Từ tập dữ liệu và tín hiệu phân loại, nhóm nghiên cứu sử dụng thư viện bộ nhớ chuẩn của Xilinx để tạo ra các bộ nhớ RAM chứa các đường dữ liệu mô phỏng thay cho việc kết nối trực tiếp tới dữ liệu.



Hình 16. Mô hình nhận dạng khuôn mặt dựa vào màu da của John See [17].



Hình 17. Đồ thị biểu diễn vùng màu da theo hai giá trị màu Cb và Cr [17].

Để mô phỏng quá trình huấn luyện trong bài báo trên nền tảng phần cứng FPGA, nhóm nghiên cứu tạo tập dữ liệu huấn luyện cho mạng nơron nhân tạo dựa trên biểu thức:

$$C_r \leq 1.5862 * C_b + 20 \quad [17]$$

Nhóm nghiên cứu tạo một mẫu dữ liệu có 2000 phần tử, với 2 giá trị Cr, Cb. Sau đó nhóm nghiên cứu đánh giá và đưa ra tập T là giá trị đích huấn luyện với thuật toán lựa chọn như sau:

$$Cr = \text{rand}() \% 256; Cr = \text{rand}() \% 256;$$

$$T = C_r \leq 1.5862 * C_b + 20 < 0 ? 0 : 1;$$

Kết quả mô phỏng trên máy tính bằng công cụ phần mềm là ngôn ngữ C với tốc độ học 0.0003 được biểu diễn trong hình 18. Nhóm nghiên cứu muốn thử nghiệm kiến trúc mạng nơron có kiến trúc xử lý song song trên nền tảng phần cứng FPGA Vertex 6 (40nm) của hãng Xilinx. Nhóm nghiên cứu mô phỏng mạng nơron nhân tạo trên công cụ Isim của Xilinx ISE với trọng số đầu tiên đều được khởi tạo là 1. Sau quá trình học tập, mạng nơron nhân tạo cho ra kết quả như trong hình 19:

$$\text{Alpha} = 0x399d4952 = 0.0003d;$$

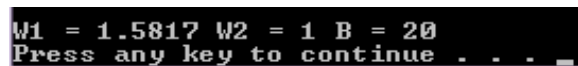
$$W2 = 0x3f800000 = 1d$$

$$W1 = 0x3fca751b = 1.5816988d;$$

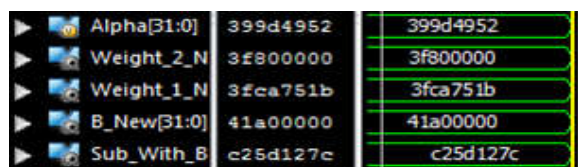
$$\text{Bias} = 41a00000 = 20d$$

Như vậy dễ dàng nhận thấy được sai số khi thực hiện trên nền tảng phần cứng và công cụ phần mềm là:

$$\frac{1.5862 - 1.5816988}{1.5862} = 0.284\%$$



Hình 18. Kết quả mô phỏng với tốc độ học 0.0003 bằng ngôn ngữ C.



Hình 19. Kết quả mô phỏng với tốc độ học 0.0003 bằng Isim của Xilinx ISE.

4.2. Kết quả khi nạp lên phần cứng FPGA của hãng Xilinx

So sánh kết quả của nhóm nghiên cứu khi thực hiện trên nền tảng phần cứng của hãng Xilinx và các kết quả đã được đề cập trong các nghiên cứu [18, 19] được đề cập trong bảng 2. Qua bảng so sánh nhận thấy công nghệ FPGA mà nhóm nghiên cứu sử dụng là Virtex 6 (40nm), số dữ liệu đầu vào là 5 với tính toán số thực lên đến 32 bit trong khi số tài nguyên phần cứng (LUT/LE và thanh ghi) lại ít hơn rất nhiều nếu so sánh với kết quả của các nghiên cứu trước.

Bảng 2. Kết quả so sánh khi thực hiện trên nền tảng phần cứng

Đặc điểm	[18]	[19]	Nhóm nghiên cứu
Công nghệ	Virtex 5 (65nm)	Cyclone II (90nm)	Virtex 6 (40nm)
Số dữ liệu đầu vào	5	1	5
Tính toán số thực	16 bit	32 bit	32 bit
LUT/LE	8984	8737	2254
Thanh ghi	7591	2867	341
Bộ nhân	18 DSP	42 bộ nhân 9 bit	10 DSP

Bảng 3 biểu diễn kết quả so sánh khi thực hiện tổng hợp mạng nơron nhân tạo xử lý song song với số tầng pipeline khác nhau. Qua bảng 3 nhóm nghiên cứu nhận thấy với số tầng tăng lên thì tốc độ xử lý sẽ nhanh hơn rất nhiều (tăng lên đến 2 lần nếu so sánh giữa việc sử dụng 2 tầng và sử dụng 5 tầng pipeline) trong khi tài nguyên phần cứng tăng lên không đáng kể.

Bảng 3. Kết quả so sánh khi thực hiện trên các tầng khác nhau

Số tầng pipeline	2	3	4	5
LUT/LE	2112	2136	2179	2254
Thanh ghi	219	251	309	341
Tần số (MHz)	52.5	71	95.3	108

Dựa vào các phân tích thông qua bảng 2 và 3 ở trên, có thể thấy kiến trúc của nhóm nghiên cứu rất đơn giản và dễ thực hiện trong khi hoạt động của mạng nơron nhân tạo hoàn toàn kiểm soát được bộ điều khiển. Bằng cách tự xây dựng các module cần thiết, nhóm nghiên cứu tạo đã tạo ra được mạng nơron có nhiều điểm tối ưu về tài nguyên phần cứng cũng như thời gian tính toán. Trong trường hợp nếu tăng thêm số tầng pipeline thì thiết kế chỉ cần tăng thêm số lượng trạng thái tương ứng để đồng bộ dữ liệu trong khi không cần phải thay đổi về kiến trúc tổng thể.

5. Kết luận và hướng phát triển

Trong bài báo này nhóm nghiên cứu đã trình bày tổng quan về mạng nơron và cách thiết kế bộ xử lý số thực theo chuẩn IEEE 745. Thông qua đó nhóm nghiên cứu cũng đã đề xuất mô hình cải tiến mạng nơron nhân tạo xử lý song song và kiểm thử trên nền tảng phần cứng FPGA của hãng Xilinx. Bằng cách tự xây dựng các module cần thiết, nhóm nghiên cứu tạo đã tạo ra được mạng nơron có nhiều điểm tối ưu về tài nguyên phần cứng cũng như thời gian tính toán. Trong trường hợp nếu tăng thêm số tầng pipeline thì thiết kế chỉ cần tăng thêm số lượng trạng thái tương ứng để đồng bộ dữ liệu. Trong thời gian tiếp theo, nhóm nghiên cứu sẽ đề xuất mô hình mạng nơron có khả năng triển khai nhiều thuật toán trên các thiết bị sử dụng nguồn pin và đòi hỏi độ chính xác cực cao hoặc kết hợp lại giữa các ưu điểm của kiến trúc mạng nơron xử lý song song và nối tiếp.

Tài liệu tham khảo

[1] Sicheng Li, Chunpeng Wu, Helen, Boxun Li, Yu Wang, Qinru Qiu - "FPGA Acceleration of Recurrent Neural Network based Language Model".
 [2] Lei Liu, Jianlu Luo, Xiaoyan Deng, Sikun Li – "FPGA-based Acceleration of Deep Neural Networks Using High Level Method" - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.

[3] Eriko Nurvitadhi, Jaewoong Sim, David Sheffield, Asit Mishra, Srivatsan Krishnan, Debbie Marr - "Accelerating Recurrent Neural Networks in Analytics Servers: Comparison of FPGA, CPU, GPU, and ASIC".
 [4] Philippe Dondon, Julien Carvalho, Rémi Gardere, Paul Lahalle, Georgi Tsenov and Valeri Mladenov - "Implementation of a Feed-forward Artificial Neural Network in VHDL on FPGA" - 978-1-4799-5888-7/14/\$31.00 ©2014 IEEE.
 [5] Yufei Ma, Naveen Suda, Yu Cao, Jae-sun Seo, Sarma Vrudhula - "Scalable and Modularized RTL Compilation of Convolutional Neural Networks onto FPGA".
 [6] "IEEE Standard for Floating-Point Arithmetic" - September 03,2015 at 19:44:10 UTC from IEEE Xplore.
 [7] Peng Li, Ming Liu, Xu Zhang and Hongda Chen - "Efficient Online Feature Extraction algorithm for Spile Sorting in A Multichannel FPGA-Based Neural Recording System" - 978-1-4799-2346-5/14/\$31.00 ©2014 IEEE.
 [8] SAMI EL MOUKHLIS, ABDESSAMAD ELRHARRAS, ABDELLATIF HAMDOUN - "FPGA Implementation of Artificial Neural Networks" - IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 2, No 1, March 2014.
 [9] Suhap Sahin, Yasar Becerikli, and Suleyman Yazici - "Neural Network Implementation in Hardware Using FPGAs" - ICONIP 2006, Part III, LNCS 4234, pp. 1105 – 1112, 2006. © Springer-Verlag Berlin Heidelberg 2006.
 [10] E.Al Zuraiqi, M.Joler, C.G.Christodoulou - "Neural Networks FPGA Controller for Reconfigurable Antennas" - 978-1-4244-4968-2/10/\$25.00 ©2010 IEEE.
 [11] Zbigniew Szadkowski, Krzysztof Pytel, Pierre Auger Collaboration - "Artificial Neural Network as a FPGA Trigger for a Detection of Very Inclined Air Showers"- IEEE TRANSACTIONS ON NUCLEAR SCIENCE - 0018-9499 © 2015 IEEE.
 [12] Alin Tisan , Jeannette Chin - "An End User Platform for FPGA-based design and Rapid Prototyping of FeedForward Artificial Neural Networks with on-chip Back Propagation learning"- 10.1109/TII.2016.2555936, IEEE.
 [13] Gabriele-Maria LOZITO, Antonino LAUDANI, Francesco RIGANTI-FULGINEI, Alessandro SALVINI - "FPGA Implementations of Feed Forward Neural Network by using Floating Point Hardware Accelerators" - c 2014

- ADVANCES IN ELECTRICAL AND ELECTRONIC ENGINEERING.
- [14] Asim J. Al-Khalili of Concordia University – Distinguished Emeritus Professor, P. Eng - “FLOATING POINT ADDERS AND MULTIPLIERS”.
- [15] Cheng-Jian Lina, Hung-Ming Tsai - “FPGA implementation of a wavelet neural network with particle swarm optimization learning” - Mathematical and Computer Modelling 47 (2008) 982–996.
- [16] Charles Geoffroy, Jean-Baptiste Michaud, Marc-André Tétrault, Julien Clerk-Lamallice, Charles-Antoine Brunet, Roger Lecomte, Réjean Fontaine - “Real Time Artificial Neural Network FPGA Implementation for Triple Coincidences Recovery in PET”- 0018-9499 © 2015 IEEE -
- IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 62, NO. 3, JUNE 2015.
- [17] Nusirwan Anwar bin Abdul Rahman, Kit Chong Wei, John See - “RGB-H-CbCr Skin Colour Model for Human Face Detection”- Faculty of Information Technology, Multimedia University.
- [18] Ravikant G. Biradar, Abhishek Chatterjee, Prabhakar Mishra, Koshy George - “FPGA Implementation of a Multilayer Artificial Neural Network using System-on-Chip Design Methodology”-978-1-4799-7171-8/15/\$31.00 ©2015 IEEE.
- [19] Jorge C. Romero-Aragon, Edgar N. Sanchez, Alma Y Alanis - “FPGA Neural Identifier for Insulin-Glucose Dynamics”- World Automation Congress ©2014 TSI Press.

High Accuracy and Speed of an Artificial Neural Cell

Nguyen Quang Anh, Nguyen Hoang Dung

*Hanoi University of Science and Technology,
No. 1, Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam*

Abstracts: This paper focusses on the neural cell design with supervised learning method adapting to many algorithms which require high speed and accuracy. Based on the supervised learning method and real neural structure, we built an artificial neural architecture which can process real numbers. This architecture easily increases the speed by expanding the floor numbers modeled on pipeline structure. To ensure high speed and accuracy we try to optimize some blocks of the shifters and parallel processor real number architectures. Therefore, when increasing the floor numbers of the pipeline architecture the processing frequency increases rapidly while resources are not significantly increased. The synthesis results implementing on Xilinx Virtex 6 FPGA show that our artificial neural architecture can operate up to 5 floors of the pipeline structure and maximum speed is reached 108 Mhz.

Keywords: Artificial Neural Network, Floating Point Processing, Pipeline, Shifter.