# Mạng nơron có khả năng cấu hình và xử lý tốc độ cao

High Speed and Flexible Configuration Neural Network

***Nguyễn Hoàng Dũng***

*Trường Đại học Bách khoa Hà Nội – Số 1, Đại Cồ Việt, Hai Bà Trưng, Hà Nội*

*Đến Tòa soạn: 02-11-2016; chấp nhận đăng: 5-9-2017*

**Tóm tắt**

*Trong bài báo này nhóm nghiên cứu trình bày thiết kế tế bào nơron nhân tạo có khả năng cấu hình với phương pháp học giám sát và 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 mạng nơron 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 lớp thực hiện mô phỏng cấu trúc đường ống (pipeline). Để đảm bảo tốc độ và độ chính xác cao, nhóm nghiên cứu đã thực hiện tối ưu một phần kiến trúc xử lý và huấn luyện. Kiến trúc của mạng tế bào dễ dàng mở rộng và điều khiển cho nhiều ứng dụng thông qua việc cấu hình các tham số cho mạng trên FPGA. Kết quả nhóm nghiên cứu đạt được rất khả quan khi thực hiện mạng có 30 tế bào với tài nguyên sử dụng là 89379 LUTs và 92761 registers trên nền tảng công nghệ chế tạo 28nm. Tần số hoạt động có thể lên tối đa đến 214 MHz.*

Từ khóa: Nơron nhân tạo, xử lý số thực, đường ống.

**Abstract**

*The research presents the neural cell design with supervised learning method adapting to many algorithms which require high speed and accuracy in this paper. 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 a part of a processing and training architecture. The architecture easily extends and controls many applications by configuring network parameters on the FPGA. The research group's results are very positive when the the network has 30 cells with 89379 LUTs and 92761 registers based on 28nm technology. Operating frequency can be reached to 214 Mhz.*

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

## 1. Giới thiệu

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 bám giám sát. Mỗi hướng huấn luyện đều có những ưu, nhược điểm khác nhau. Tuy nhiên 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 sử dụng thuật toán và chạy trên máy tính với CPU và GPU tốc độ cao nên không thể hiện tính gọn nhẹ, linh hoạt. Thư viện mã nguồn mở OpenCV là một mã nguồn sử dụng ngôn ngữ C/C++ và JAVA nên có cấu trúc lập trình mạng nơron linh hoạt. Mạng được lập trình chạy trên phần mềm nên số lượng đầu vào, ra và số lượng tế bào sử dụng không bị giới hạn như lập trình phần cứng. 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 về mạng nơron thường chỉ hướng tới một ứng dụng cụ thể nên khó có khả năng mở rộng chức năng cũng như sử dụng cho ứng dụng khác. Ý tưởng thiết kế của nhóm nghiên cứu là tạo ra mạng nơron nhân tạo có khả năng cấu hình trên nền tảng FPGA để tương thích cho nhiều ứng dụng có thuật toán hoạt động trên cấu trúc thuật toán deep learning đòi hỏi độ chính xác cao mà tài nguyên sử dụng cần được tiết kiệm cũng như đảm bảo tốc độ xử lý. Thiết kế dựa trên nghiên cứu nơron thực và các cấu trúc cùng ý tưởng thiết kế [4]. Trong đó, nhóm nghiên cứu sử đã thiết kế bộ xử lý số thực theo chuẩn IEEE 754 [5].

* Corresponding author: Tel.: (+84) 913.004.120
Email: dung.nguyenhoang@hust.edu.vn

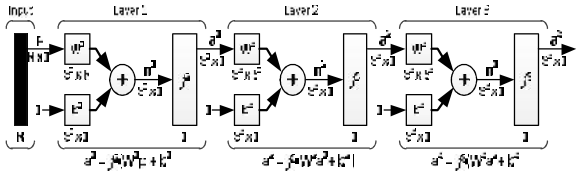
Bài báo này sẽ trình bày tổng quan về mạng nơron và các nghiên cứu liên quan trong phần II; mô tả thiết kế và kiến trúc sẽ được nhóm nghiên cứu trình bày trong phần III; các kết quả mô phỏng, thảo luận trong phần IV và kết luận ở phần V.

## 2. Tổng quan về mạng nơron và các nghiên cứu liên quan

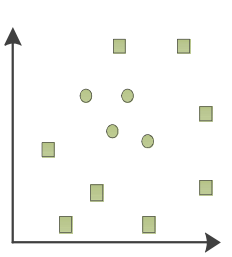
### *2.1. Tổng quan về mạng nơron*

Thông tin từ môi trường được đưa về não bộ thông qua các giác quan và sẽ được bộ não 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 nơron thần kinh (tế bào thần kinh) với chức năng chính truyền dẫn các xung điện. Nơron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là một phần quan trong nhất của não.

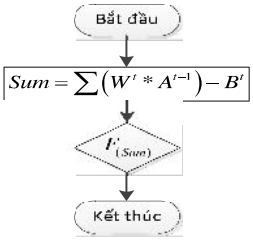
Một nơron gồm có thân nơron (cell body) là nơi xử lý các tín hiệu được đưa vào từ các giác quan. Các dây hình nhánh cây (dendrites) là nơi nhận các xung điện vào trong nơron và các sợi trục (axons) là một dây dài đưa xung điện ra sau quá trình xử lý từ thân của nơron. Giữa các dây hình nhánh cây và các sợi trục có một liên kết với nhau gọi là khớp thần kinh (synapse). Hình 1 mô tả cấu trúc của một hệ thống mạng nơron nối tiếp. Trong đó $P_1$, $P_2$ đến $P_R$ lần lượt là các đầu vào của mạng nơron 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 nơron nhân tạo là a.



**Hình 1.** Mô hình mạng nơron nhân tạo [4]

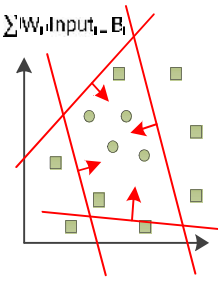


**Hình 2.** Hai tập dữ liệu cần phân loại giữa kí hiệu tròn và vuông




**Hình 3.** Thuật toán xử lý dữ liệu trong mỗi tế bào nơron nhân tạo

Giả sử có mô hình hai tập dữ liệu kí hiệu hình tròn và vuông cần phân loại như hình 2 trong mạng nơron có m lớp. Thuật toán xử lý dữ liệu trong mạng nơron nhân tạo khi chưa huấn luyện được mô tả trong hình 3. Các kí hiệu W là trọng số của mạng nơron nhân tạo và B là giá trị ngưỡng xử lý. Ở mô hình này có tập dữ liệu vào $A^0$. Dữ liệu đầu vào sẽ được nhân với trọng số tương ứng $W_i$ rồi trừ đi ngưỡng B và sau đó đưa vào xử lý ở các lớp nơ ron tiếp theo với thuật toán $a^3 = f^3(W^3f^2(W^2f^1(W^1p + b^1) + b^2) + b^3)$. Hình 4 biểu diễn kết quả của mô hình đang bị lỗi khi các phần tử hình vuông bị phân loại nhầm sang bên tập dữ liệu các phần tử hình tròn.




**Hình 4.** Mô hình phân loại đang có lỗi khi mới khởi tạo trọng số

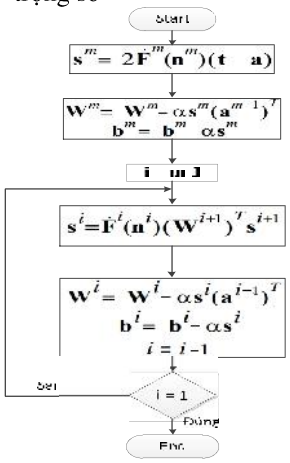
$$s^m = \frac{\partial \hat{F}}{\partial n^m} = \left( \frac{\partial \hat{F}}{\partial n_1^m}, \frac{\partial \hat{F}}{\partial n_2^m}, \ldots, \frac{\partial}{\partial r} \right.$$

$$W^m := W^m - \alpha\, s^m (a^{m-1})^T$$

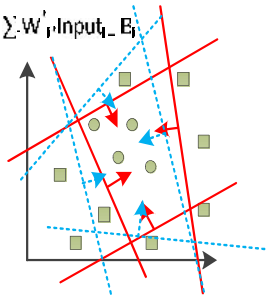
$$b^m := b^m - \alpha\, s^m$$

$$s^i = \dot{F}(n^i) s^{i+1} (W^{i+1})^T$$

**Hình 5.** Thuật toán lan truyền ngược




**Hình 6.** Thuật toán học trong tế bào mạng nơron nhân tạo




**Hình 7.** Mô hình phân loại sau khi điều chỉnh trọng số từ quá trình học

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. Thuật toán lan truyền ngược được mô tả như hình 5. Phương thức thực hiện thuật toán huấn luyện lan truyền ngược trong mạng nơron được thể hiện rõ trong hình 6. Sau khi phát hiện ra lỗi trong quá trình phân loại như hình 4, trọng số của mạng sẽ được cập nhật lại dựa theo kết quả chuẩn và kết quả phân tích ra được. Kết quả phân loại sau khi đã điều chỉnh trọng số từ quá trình học nhiều lần cho ra sự phân loại chính xác giống trong hình 7.

Quá trình học cụ thể được mô tả như sau: kết quả đưa ra sau khi tính toán như lưu đồ trong hình 3 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 Wi sẽ được trừ đi một lượng giá trị phụ thuộc đầu vào tương ứng nhân với tốc độ học tập alpha. 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 alpha. 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.

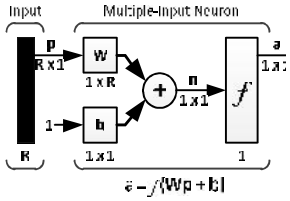
### *2.2 Các nghiên cứu liên quan*

Hiện nay, có rất nhiều nghiên cứu liên quan đến trí tuệ nhân tạo được triển khai trên các nền tảng khác nhau như máy tính, hệ thống nhúng, FPGA và thiết kế ASIC. Mỗi nền tảng thực hiện đều có các ưu và nhược điểm khác nhau. Nhược điểm khi triển khai trên FPGA là khả năng mở rộng hệ thống và kích thước mạng cùng với tốc độ của mạng nơron phụ thuộc vào tài nguyên phần cứng và công nghệ sản xuất. Bù lại, ưu điểm rất lớn của FPGA là cấu trúc phần cứng linh hoạt, có khả năng thực hiện nhiều thiết kế số song song cũng như nối tiếp. Công nghệ hiện nay của chip FPGA có nhiều tài nguyên phần cứng mà giá thành vẫn vừa phải. Phân tích [6] đã chỉ rõ các ưu điểm khi thực hiện trên mạng nơron về các yếu tố giá thành, tốc độ và khả năng xử lý trên các nền tảng máy tính, FPGA và ASIC.

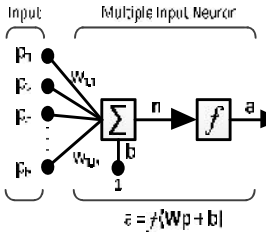
Trong so sánh [2] mạng nơron được sinh ra trên FPGA từ ngôn ngữ bậc cao như C/C++ hay đơn giản nhất là matlab cho nền tảng FPGA đã có hiệu năng cao hơn thực hiện trên CPU nhiều lần. Tuy nhiên, cấu trúc phần cứng này tạo ra chỉ có thể tạo ra một hoặc một số ứng dụng cụ thể mà khó có khả năng phát triển mở rộng. Độ chính xác của thiết kế chỉ đạt ở mức tương đối và khó phát hiện ra các lỗi tiềm ẩn. Một số thiết kế bậc cao hoạt động đúng nhưng không thể sinh ra ngôn ngữ phần cứng vì không tạo ra được cấu trúc thuật toán tương xứng. Với các thiết kế không qua phần mềm để sinh mã (code) của các nhóm nghiên cứu khác, một số vấn đề lớn vẫn còn phát sinh như [7] gặp phải vấn đề tốc độ xử lý khi triển khai cấu trúc LSTM- kiến trúc có khả năng xử lý được nhiều dữ liệu đầu vào nhưng thời gian tính toán cho mỗi đầu vào này là rất lớn vì độ phức tạp lên tới **O ($n^2$).** Ngoài ra, nhóm nghiên cứu [7] còn triển khai cấu trúc tính toán song song trên mỗi tế bào nơron với hàm phi tuyến $\tan sig_{(n)} = \frac{e^n - e^{-n}}{e^n + e^{-n}}$ hay hàm $\log sig_{(n)} = \frac{1}{1+e^{-n}}$ của nghiên cứu [8] dẫn tới tài nguyên khi triển khai là rất lớn. Với một số kiến trúc xử lý song song của [9], một tế bào có 5 đầu vào cần tới 5 bộ nhân và 5 bộ cộng. Đặc biệt, các bộ cộng không được thiết kế song song mà đi nối tiếp nhau làm hiệu suất của tế bào giảm đi rất nhiều. Thời gian tính toán tương đương với tế bào có 5 đầu vào được xử lý nối tiếp. Ngoài ra, hàm phi tuyến tansig của nhóm nghiên cứu [9] còn chiếm tới 2/3 tài nguyên phần cứng của một tế bào trong khi dữ liệu được xử lý dưới dạng số nguyên sẽ không đạt được hiệu quả cao. Để giải quyết các vấn đề về khả năng mở rộng cùng tốc độ xử lý, nhóm nghiên cứu đề xuất một thiết kế mạng nơron có khả năng cấu hình với kích thước tầm trung và có độ chính xác cao cho các ứng dụng trên hệ thống FPGA. Đánh giá tổng quan về 2 cấu trúc mạng được nhóm nghiên cứu trình bày trong phần 2.3.

### *2.3. Đánh giá về hai loại cấu trúc*

Tế bào nơron nhân tạo thường có hai hướng đưa tín hiệu vào xử lý theo kiểu nối tiếp hoặc theo kiểu song song. Hình 8 và hình 9 lần lượt biểu diễn mô hình mạng nơron xử lý nối tiếp và xử lý song song [4].




**Hình 8.** Mô hình tế bào nơron xử lí nối tiếp

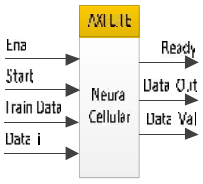



**Hình 9.** Mô hình tế bào nơron xử lý song song

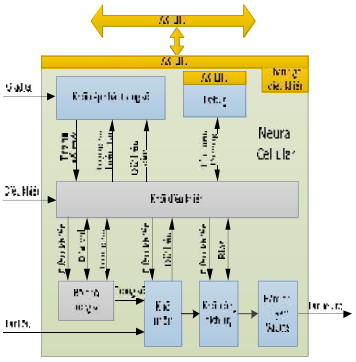
Ở hình 8 dễ dàng nhận thấy với một tập dữ liệu đầu, chỉ có một luồng đầu vào duy nhất kết hợp với trọng số tương ứng đi vào bộ nhân. Sau đó kết quả được cộng tích lũy lại cho tới khi tính toán hết các dữ liệu đầu vào. Ngược lại, ở hình 9 có rất nhiều dữ liệu đầu vào khác nhau nhân với trọng số tương ứng sau đó đi vào một bộ cộng song song để tiến vào xử lý trong nơron nhân tạo. Bản chất của 2 cấu trúc đều thực hiện các chức năng đã được nói trong phần 2.1, điểm khác biệt lớn nhất là việc triển khai thực hiện các quá trình tính toán của cấu trúc nối tiếp có dạng vòng lặp ra xử lý song song các công đoạn tính toán. Kiến trúc song song chỉ phù hợp để tính toán trên các ứng dụng có ít đầu vào, khó mở rộng số tầng tính toán khi hoạt động thành mạng nơron có kích thước lớn. Dựa trên các nghiên cứu, nhóm nghiên cứu xây dựng một bảng so sánh các ưu nhược điểm của từng mô hình tế bào nơron xử lý nối tiếp và song song như trình bày trong bảng 1.

**Bảng 1.** So sánh ưu nhược điểm của mô hình mạng nơron xử lý nối tiếp và song song

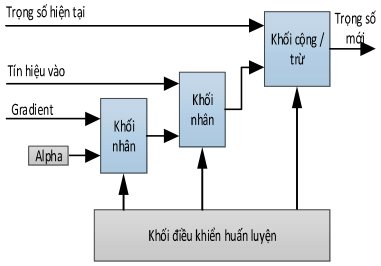
| Mô hình mạng nơron | Ưu điểm | Nhược điểm |
|---|---|---|
| Xử lý nối tiếp | - Tiết kiệm được nhiều tài nguyên cổng logic phần cứng cho các phép toán.<br>- Dễ điều khiển và sử dụng.<br>- Có thể triển khai mạng kích thước lớn dưới phần cứng.<br>- Ứng dụng được trên các thuật toán cần xử lí nhiều đầu vào. | - Cần xây dựng hàng đợi và đồng bộ dữ liệu đầu vào khi thực hiện trên nhiều lớp nơron.<br>- Số thanh ghi và flipflop tăng lên để lưu giữ giá trị khi tiết kiệm cổng logic.<br>- Thiết kế bộ điều khiển phức tạp và có thể lặp dữ liệu đầu vào gây ra sai số.<br>- Thời gian sẽ chậm khi dữ liệu vào phải tính toán nối tiếp.<br>- Chu kỳ một vòng tính toán lớn. |
| Xử lý song song | - Xử lý dữ liệu nhanh, đảm bảo tính toán thời gian thực.<br>- Tăng tỉ lệ dữ liệu đầu vào hiệu dụng và tiết kiệm tài nguyên bộ nhớ.<br>- Sử dụng kỹ thuật pipeline để tăng tốc độ tính toán. | - Số bộ nhân và bộ cộng tăng lên tương ứng với số đường dữ liệu vào.<br>- Nếu có số lượng đường dữ liệu vào lớn sẽ ảnh hưởng đến yêu cầu về kích thước của vi mạch thực hiện. |




**Hình 10.** Lược đồ các tín hiệu chính của tế bào



**Hình 11.** Đề xuất mô hình nơron nối tiếp lai hóa song song




**Hình 12.** Cấu trúc khối cập nhật trọng số cho tế bào nơron

Với những ưu nhược điểm của từng mô hình, nhóm nghiên cứu nhận thấy mô hình xử lý dữ liệu nối tiếp có khả năng thực hiện một mạng nơron có kích thước lớn hơn rất nhiều so với kiến trúc song song. Tuy nhiên để hạn chế nhược điểm của các mô hình này, nhóm nghiên cứu đề xuất và trình bày một số cải tiến cho mô hình xử lý dữ liệu nối tiếp trong phần tiếp theo của bài báo này.

## 3. Kiến trúc mạng nơron

### *3.1. Cải tiến mô hình tế bào nơron xử lý dữ liệu*

Trong tế bào nơron nhân tạo mới, nhóm nghiên cứu kết hợp sử dụng hai ưu điểm của hai loại tế bào nơron xử lý nối tiếp và xử lý song song. Với các ưu nhược điểm của hai loại tế bào được nhóm nghiên cứu trình bày ở trên, chúng tôi lựa chọn việc thiết kế một tế bào và mạng có khả năng xử lý được nhiều tín hiệu đầu vào. Đây là ưu điểm lớn nhất của tế bào xử lý nối tiếp, các tín hiệu được xử lí vào một cách tuần tự nên chỉ cần một bộ nhớ lưu trữ các trọng số. Vì đặc điểm các dữ liệu vào nối tiếp, chúng ta phải xây dựng một khối hàng đợi để lưu trữ và đồng bộ các tín hiệu vào. Nếu chỉ sử dụng một tế bào nối tiếp, hiệu quả của hệ thống sẽ giảm đi rất nhiều dựa trên tỉ lệ tài nguyên so với số lượng kết quả tính toán được. Nhưng nếu triển khai trên một mạng có nhiều tế bào, cơ chế này thực hiện lại rất hiệu quả vì giảm được nhiều tài nguyên phần cứng cho việc thực hiện các phép tính trong mỗi tế bào. Các tín hiệu đầu vào và ra của tế bào được thể hiện như hình 10.

Để đảm bảo cho việc xử lý được nhanh chóng, nhóm nghiên cứu đã thiết kế và sử dụng một bộ đệm tín hiệu đầu vào song song, đầu ra nối tiếp cùng với một khối đệm dữ liệu vào ra nối tiếp để đảm bảo quá trình lấy dữ liệu xử lý cho mạng luôn được ổn định.

Để tăng tính chính xác cho tế bào và cả hệ thống, nhóm nghiên cứu đã sử dụng các phép tính toán số học trên số thực 32-bit của 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, quá trình xử lý dữ liệu cần thực hiện nhiều công đoạn nhân, dịch phức tạp và đếm nối tiếp nên tần số hoạt động tối đa sẽ bị giảm đi nhiều. Hơn nữa, dữ liệu đi vào liên tiếp nhau dễ sinh ra hiện tượng lặp dữ liệu đầu vào khi chạy thực tế trên FPGA. Vì vậy, nhóm nghiên cứu đề xuất một khối nhân bán song song để tránh hiện tượng này. Kiến trúc bên trong của khối gồm có một bộ điều khiển bắt tín hiệu và điều khiển các bộ nhân khác nhau để tránh lặp dữ liệu. Kết quả tính toán được cộng tích lũy rồi đẩy sang hàm phi tuyến Satlin như sau:

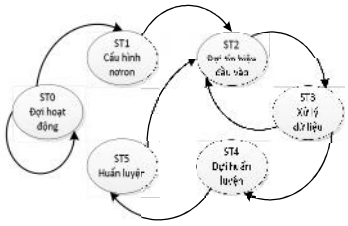
$$Satlin(x) = \begin{cases} -1, x \leq -1 \\ x, -1 \leq x \leq \\ 1, x \geq 1 \end{cases}$$

Hàm phi tuyến Satlin có các xử lý đơn giản nhưng vẫn thu được hiệu quả cao và có thể thực hiện dễ dàng trên mạch phần cứng số. Cấu trúc tế bào nơron của nhóm nghiên cứu được mô tả như hình 11. Trong nghiên cứu [10], mỗi tế bào được đưa vào một hàm phi tuyến logsig dẫn tới tế bào cần rất nhiều tài nguyên. Ngoài ra, để chạy được tần số lớn, tế bào còn cần thêm nhiều thanh ghi đệm dữ liệu. Việc tăng số lớp trong mạng lên gần như không khả thi. Điều này được thể hiện rất rõ trong nghiên cứu [11], các hàm logsig và tansig, radbas có cấu trúc xử lý số thực rất phức tạp bao gồm bộ dịch cordic tạo ra số E cùng với các phép toán cộng và chia số thực. Trong khi đó, hàm Satlin thuần túy cấu tạo từ các phép so sánh nhị phân của số mũ, dấu và biên độ trong việc biểu diễn hai số thực. Với nghiên cứu [12], tế bào được sinh ra mã (code) từ matlab có thể phát sinh ra nhiều lỗi và mạng không thể tự do cấu hình cho nhiều ứng dụng khác nhau.

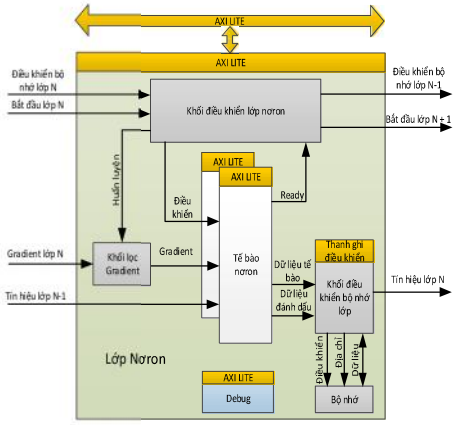
Trong nghiên cứu [13], ứng dụng điều khiển xe tự dò đường bằng nơron nhân tạo có cấu trúc chưa được tối ưu. Để điều khiển một tế bào hoạt động cần tới một kit Arduino để truyền và nhận các tín hiệu điều khiển cùng dữ liệu. Thiết kế này khó có khả năng xử lý thời gian thực do tần số hoạt động của Arduino chỉ tới 30MHz. Ngoài ra thiết kế này còn cần tới nhiều loại giao tiếp đồng bộ phức tạp cho việc xử lý và huấn luyện. Vì vậy, mỗi tế bào của nhóm nghiên cứu được thiết kế thêm một khối huấn luyện cục bộ để thay đổi trọng số dựa trên khối huấn luyện của mạng và các dữ liệu được đưa vào. Khối huấn luyện tổng quan của hệ thống sử dụng thuật toán lan truyền ngược. Cấu trúc của khối huấn luyện được trình bày trong hình 12.

Ngoài ra, để phát triển mở rộng, mỗi tế bào được cung cấp thêm một giao diện truyền dữ liệu AXI LITE – chuẩn giao tiếp của hãng Xillin. Tế bào có khả năng giao tiếp trực tiếp với vi xử lý trung tâm MicroBlaze hoặc ARM Cortex-M0. Đây là một chuẩn giao tiếp rất hiệu quả giữa vi xử lý và các khối ngoại vi trên nền tảng system onchip và trên FPGA hiện nay. Mỗi tế bào được cấp cho 5 thanh ghi trong đó 2 thanh ghi điều khiển và 3 thanh ghi dữ liệu giao tiếp AXI LITE. Thanh ghi điều khiển chứa tín hiệu điều khiển đọc ghi và địa chỉ lưu trữ các trọng số cũng như số lượng tế bào được sử dụng trong lớp nơron. Bộ nhớ trọng số chứa tối đa 32 thanh ghi 32 bits tương ứng với 32 tín hiệu đầu vào cần xử lí.

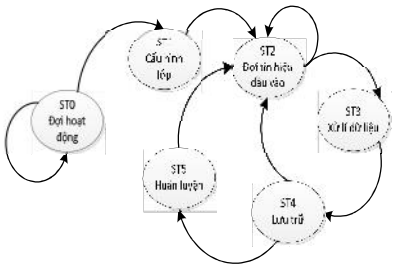
Sơ đồ máy trạng thái của tế bào được mô tả như hình 13. Khi bắt đầu hoạt động, vi xử lý cấu hình số lượng trọng số và các giá trị này vào địa chỉ tương ứng trong bộ nhớ. Tế bào đợi dữ liệu được truyền tới và tính toán. Nếu đang ở trong trạng thái huấn luyện, nó sẽ đợi các lớp khác được tính toán xong và khối huấn luyện toàn cục sẽ điều khiển các tế bào cập nhật lại trọng số. Ngược lại, tế bào sẽ trở về trạng thái đợi dữ liệu đầu vào. Mỗi tế bào sẽ cần từ 10 xung đồng hồ (cho một đầu vào) tới 300 xung cho tế bào xử lí tối đa 32 đầu vào.




**Hình 13.** Máy trạng thái của một tế bào




**Hình 14.** Mô hình nơron trong một lớp

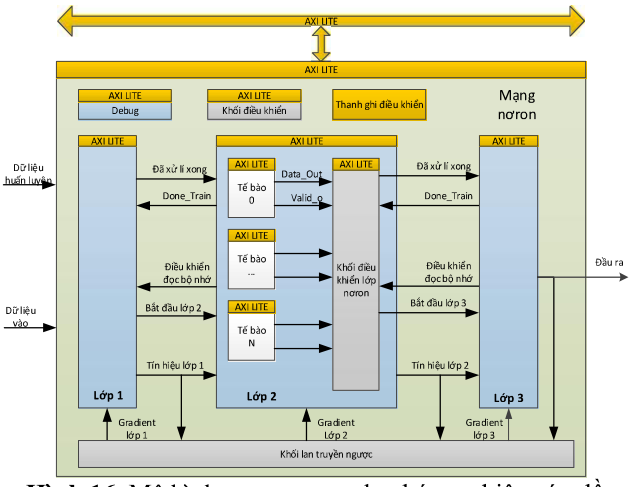



**Hình 15.** Trạng thái hoạt động của lớp nơron

### 3.2. Cấu trúc lớp nơron

Cấu trúc của lớp tế bào được nhóm nghiên cứu trình bày ở hình 14. Nhóm nghiên cứu đề xuất số lượng tế bào cho mỗi lớp là 10 nơron. Trong đó, khối điều khiển kiểm soát toàn bộ các hoạt động trong lớp thông qua các tín hiệu cấu hình từ vi xử lý với giao tiếp AXI LITE. Người sử dụng có thể dễ dàng cấu hình số lượng tế bào trong lớp và cấu hình vị trí dữ liệu sẽ được lưu trữ trong bộ nhớ lớp. Trong đó, từ lớp nơron đầu tiên trở đi, số lượng nơron lớp trước là số lượng đầu vào tối đa của lớp nơron phía sau. Sau khi tính toán, lớp sẽ xuất ra một tín hiệu thông báo là nó đã tính toán xong cho lớp tiếp theo nhận dữ liệu của nó. Khối điều khiển huấn luyện có nhiệm vụ phân tách tín hiệu cập nhật lớp ra tín hiệu huấn luyện cho tế bào đang hoạt động trong lớp.

Trong nghiên cứu [14], kết quả đầu ra của mỗi tế bào phải đi vào khối trễ nhiều lần để đi vào một bộ nhớ chung không những làm hao phí tài nguyên bộ nhớ mà còn tăng thời gian cần để xử lý và huấn luyện dữ liệu trong một lớp. Nhóm nghiên cứu thiết kế một khối lưu trữ trong mỗi lớp nhằm tránh việc phải đợi lưu trữ dữ liệu sau mỗi tế bào quá lâu. Với M tế bào trong lớp, khối điều khiển chỉ cần M+2 xung đồng hồ để lưu trữ các giá trị này. Dựa theo thiết kế lớp, nhóm nghiên cứu đề xuất một máy trạng thái điều khiển hoạt động cho lớp tế bào như hình 15.




**Hình 16.** Mô hình mạng nơron do nhóm nghiên cứu đề xuất

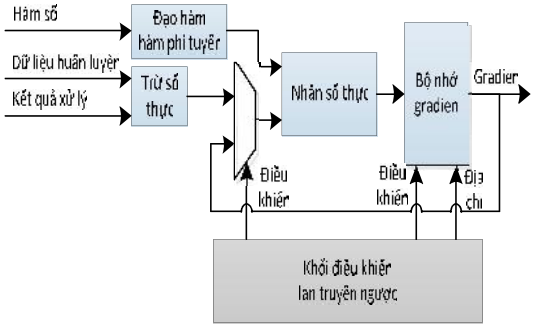
Khi bắt đầu được sử dụng, lớp sẽ được cầu hình số lượng tế bào hoạt động và vị trí đầu ra tương ứng trong bộ nhớ. Sau đó, từng tế bào được cấu hình các trọng số và các thông số khác cho tới tế bào cuối cùng của lớp. Dữ liệu đầu vào của mỗi lớp có thể là tín hiệu đầu vào hoặc kết quả tính toán của lớp nơron trước. Sau khi đọc dữ liệu vào các tế bào bắt đầu tính toán và kết quả tính ra được lưu trữ vào bộ nhớ của lớp. Lớp cuối cùng xử lí dữ liệu xong, khối huấn luyện toàn cục sẽ tính toán và đưa ra chi thị điều khiển cho các lớp cập nhật trọng số. Sau đó, lớp quay về trạng thái nhận tín hiệu vào và bắt đầu chu kì hoạt động mới. Lớp tế bào có 5 thanh ghi trong đó 1 thanh ghi điều khiển hoạt động và 4 thanh ghi cấu hình vị trí dữ liệu của các tế bào bên trong lớp. Lớp có thể giao tiếp trực tiếp với vi xử lý thông qua bus giao tiếp AXI LITE.

### *3.3 Cấu trúc mạng nơron*

Nghiên cứu [15] thực hiện mạng nơron theo thiết kế học sâu (deep learning) sử dụng bảng băm để phân vùng địa chỉ trên bộ nhớ ngoài. Việc làm này có thể xử lý được rất nhiều đầu vào nhưng thời gian tính toán và huấn luyện cho một mẫu dữ liệu là rất dài do thiết kế không tối ưu được các phép toán xử lý không có ý nghĩa. Bộ điều khiển cũng trở nên phức tạp và giảm tần số hoạt động của hệ thống do có nhiều tín hiệu cần điều khiển. Trong nghiên cứu [16], để đơn giản hóa việc điều khiển, nhóm nghiên cứu [16] đã kết hợp sử dụng CPU với FPGA thông qua khe cắm PCI-Express của mainboard máy tính. Cấu trúc này sẽ dễ thực hiện nhưng không có khả năng hoạt động đọc lập. Nhóm nghiên cứu đề xuất thực hiện hệ thống mạng có cấu trúc như hình 16. Hệ thống gồm có 3 lớp nơron có khả năng cấu hình số lượng tế bào mỗi lớp và khối thuật toán lan truyền ngược để huấn luyện cho mạng.

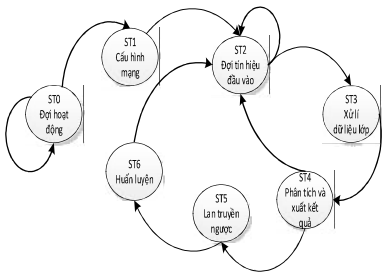
**Bảng 2.** Một số kết quả trọng số và ngưỡng thu được sau khi huấn luyện mạng

| Giá trị thực nghiệm | Giá trị chuẩn | Sai số | Tỉ lệ sai số (%) |
|---|---|---|---|
| 1.5816988 | 1.58620 | 0.0045012 | 0.283772538 |
| 1.0003099 | 1.00000 | -0.0003099 | -0.03099 |
| 19.8939999 | 20.00000 | 0.1060001 | 0.5300005 |
| 0.3457219 | 0.34480 | -0.0009219 | -0.26737239 |
| 0.9968999 | 1.00000 | 0.0031001 | 0.31001 |
| 76.2068481 | 76.20690 | 0.0000519 | 6.81041E-05 |




**Hình 17.** Mô hình khối lan truyền ngược cho mạng nơ ron

Với thuật toán lan truyền ngược đã mô tả trong phần 2.1, nhóm nghiên cứu đề xuất thiết kế khối thuật toán lan truyền ngược như hình 17. Khối lan truyền ngược dễ dàng được điều khiển và cấu hình thông qua giao tiếp AXI LITE với nhân vi xử lý ARM hoặc MicroBlaze onchip. Nhóm nghiên cứu đã tối ưu các phép toán nhân ma trận để ra được Gradien cho lớp bằng cách loại các phép nhân không có ý nghĩa so với thuật toán nguyên bản được thực hiện trên phần mềm máy tính. Quá trình hoạt động của mạng được nhóm nghiên cứu mô hình hóa như hình 18.




**Hình 18.** Trạng thái hoạt động của mạng nơron

## 4. Kết quả

Nhóm nghiên cứu tiến hành thử nghiệm ứng dụng nhận diện da người dựa trên màu sắc của John See [17] đã được nhận diện trên 600 khuôn mặt. Để mô phỏng quá trình huấn luyện thuật toán nhận diện màu da, nhóm nghiên cứu tạo ra tập dữ liệu huấn luyện có 2000 mẫu với giá trị Cb, Cr cùng với giá trị tham chiếu cho mạng nơ ron dựa trên biểu thức như sau:

$$Cr \le 1.5862 * Cb + 20$$
$$Cr \ge 0.3448 * Cb + 76.2069$$
$$Cr \ge -4.5652 * Cb + 234.5652$$
$$Cr \le -1.15 * Cb + 301.75$$
$$Cr \le -2.2857 * Cb + 432.85$$

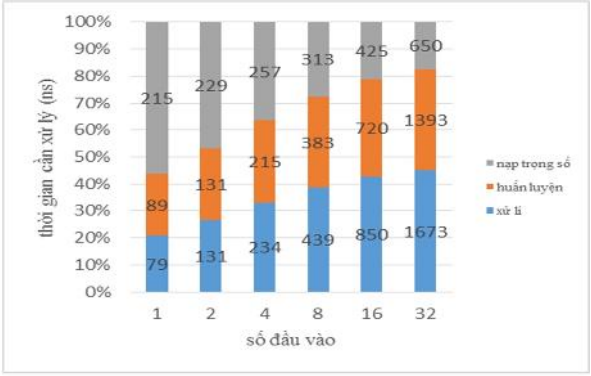
Với các thông số cấu hình tốc độ học cho mạng alpha= 0x399D4952 = 0.003, khi tiến hành mô phỏng và chạy thực tế, nhóm nghiên cứu thu được một tập các giá trị trọng số và ngưỡng của tế bào nhận diện màu da như bảng 2.

Như vậy, có thể thấy được sai số lớn nhất xuất hiện vào khoảng 0.5% chủ yếu là do trong quá trình tính toán, tác giả đã làm tròn kết quả tính toán giữa mỗi phép tính nên có sự sai khác trong kết quả

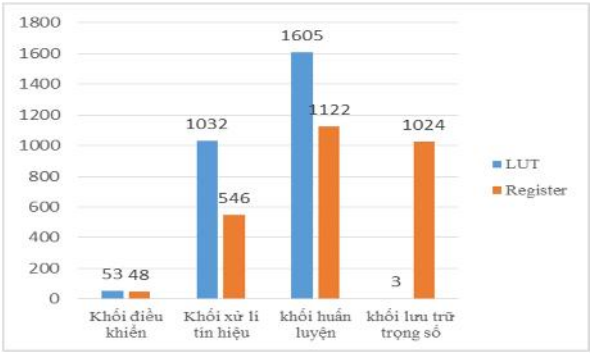
Sau khi thiết kế và tiến hành triển khai trên nền tảng phần cứng FPGA trên kit Zedboard ZynQ7000-XC7Z020-CLG484 và board có sử dụng chip ZynQ7000-XC7Z035ffg900-1, nhóm nghiên cứu đã tổng hợp lại biểu đồ phân tích thời gian cần thiết để tế bào nơron hoạt động tương ứng với số lượng đầu vào lớn nhất cấu hình trong lớp tế bào như biểu diễn trong hình 19. Trong đó, tế bào chỉ cần nạp trọng số một lần khi bắt đầu khởi tạo, người sử dụng chỉ mất thời gian chính cho quá trình huấn luyện tế bào và xử lý dữ liệu. Như vậy, có thể dễ dàng thấy được thời gian cần thiết thực sự cho 1 lượt tính toán có huấn luyện, tế bào mất tối đa 3066 ns tương ứng với 1 giây, lớp tế bào có thể xử lý khoảng 320.000 khối dữ liệu với 32 đầu vào.

Để tạo ra mỗi tế bào, nhóm nghiên cứu cần sử dụng 2693 LUTs và 2740 Registers với sự phân bổ tài nguyên sử dụng cho các khối như biểu diễn trong hình 20. Bảng 3 so sánh kết quả triển khai trên nền tảng FPGA của nhóm nghiên cứu với các kết quả của Ravikant [9] và Jorge [18].

Qua bảng 3 có thể nhận thấy hiệu suất tế bào nơron mới của nhóm nghiên cứu có hiệu quả cao hơn trong việc sử dụng tài nguyên phần cứng. Khi triển khai thành các khối lớn hơn là các lớp và mạng, nhóm nghiên cứu thu được các kết quả ghép vào một lớp có 10 tế bào cùng với khối điều khiển lớp. Nhóm nghiên cứu sử dụng 28793 LUTs và 29587 thanh ghi (register) bao gồm cả khối lưu trữ tín hiệu đầu ra của mỗi tế bào noron. Khi ghép lại thành một mạng nơron có 3 lớp, nhóm nghiên cứu sử dụng 92761 thanh ghi và 89379




**Hình 19.** Tương quan giữa thời gian xử lý mạng nơron với số đầu vào của mạng




**Hình 20.** Sự phân bổ tài nguyên cho các khối thiết kế

**Bảng 3.** So sánh kết quả của nhóm nghiên cứu với các nhóm khác

| **Đặc điểm** | Ravikant **[9]** | Jorge **[18]** | Nhóm nghiên cứu |
|---|---|---|---|
| Công nghệ | Virtex 5 (65nm) | Cyclone II (90nm) | zynq(28nm) |
| Số dữ liệu đầu vào | 5 | 1 | 32 |
| Kiểu xử lý | Song song | Nối tiếp | Nối tiếp |
| Tính toán số thực | 16 bit | 32 bit | **32 bit** |
| LUT/LE | 8984 | 8737 | **2693** |
| Thanh ghi | 7591 | 2867 | **2740** |
| Bộ nhân | 18 DSP | 42 bộ nhân 9 bit | 0 |

LUTs cùng với khối với tần số hoạt động tối đa lên tới 214 MHz. Có thể nhận thấy việc mở rộng lớp và mở rộng mạng của nhóm nghiên cứu cần tài nguyên khá tuyến tính. Chỉ cần có chip FPGA đủ lớn, thiết kế của mạng dễ dàng được tăng kích thước. Việc điều khiển mạng dễ dàng thông qua một lõi vi xử lý nhân ARM có thể tạo ra các ứng dụng linh hoạt. Với nền tảng FPGA mới, chúng ta có thể dễ dàng tích hợp mạng nơron này như một phần cứng đi kèm với các kiến trúc vi xử lý Intel hay AMD hay chip nhúng ARM hiện nay

giống như khối xử lý DMA hay nén JPEG, VGA. Hoạt động sử dụng, điều khiển hoàn toàn có thể thông qua hệ điều hành nhân Linux trên chip nhúng ARM/FPGA.

## 5. Kết luận và hướng phát triển

Trong bài báo này, nhóm nghiên cứu đã trình bày bày kiến trúc về tế bào, lớp và mạng nơ ron trên nên tảng FPGA. Với độ chính xác cao khi xử lý dữ liệu và một số cải tiến tế bào, nhóm nghiên cứu đã tạo ra được mạng nơron có nhiều ưu điểm về tài nguyên phần cứng cũng như tốc độ xử lý dữ liệu và khả năng thích ứng với nhiều thuật toán. Trong thời gian tiếp theo, nhóm nghiên cứu sẽ tích hợp vào hệ thống thêm các khối giao tiếp mạng Ethernet, USB và UART cho thiết kế để mở rộng khả năng tạo ra các ứng dụng lớn hơn. Bên cạnh đó nhóm nghiên cứu cũng sẽ thiết kế giao diện trên máy tính để có thể cấu hình và giao tiếp với hệ thống thông qua mạng ethernet một cách trực quan hơn. Về mặt kiến trúc, nhóm nghiên cứu sẽ tiến hành tích hợp thêm các ưu điểm của 2 cấu trúc xử lí song song và nối tiếp và tiến hành đo kiểm đánh giá chất lượng của mạng đề xuất.

## 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" - 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines - Vancouver, BC, Canada

[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 - Krakow, Poland

[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" - 2016 26th International Conference on Field Programmable Logic and Applications (FPL) - Lausanne, Switzerland

[4] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesús - Book "Neural network design" - 2nd edtion, 2002.

[5] "IEEE Standard for Floating-Point Arithmetic" - September 03,2015 at 19:44:10 UTC from IEEE Xplore

[6] Mr Prashant D. DeotaleProfLalit Dole – "Design of FPGA Based General Purpose Neural Network" - International Conference on Information Communication and Embedded Systems (ICICES2014) - Chennai, India

[7] Yijin Guan, Zhihang Yuan, Guangyu Sun1, Jason Cong2– "FPGA-based Accelerator for Long Short-Term Memory Recurrent Neural Networks" - Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific - Chiba, Japan

[8] Tuan Linh Dang, Yukinobu Hoshino – "An-FPGA based classification system by using a neural network and an improved particle swarm optimization algorithm" - 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS) - Sapporo, Japan

[9] Ravikant G. Biradar, Abhishek Chatterjee, Prabhakar Mishra, Koshy George - "FPGA Implementation of a Multilayer Artificial Neural Network using System-on-Chip Design Methodology" – 2015 International Conference on Cognitive Computing and Information Processing(CCIP) - Noida, India

[10] Ismail Koyuncu - "Design and Implementation of High Speed Artificial Neural Network Based Sprott 94 S System on FPGA" - International Journal of Intelligent Systems and Applications in Engineering - © Advanced Technology & Science 2013

[11] Sahin, Koyuncu - "Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA" – JOURNAL ELEKTRONIKA IR ELELTROTECHNIKA 2012.

[12] Djalal Eddine KHODJA, Aissa KHELDOUN, Larbi REFOUFI - "Sigmoid Function Approximation for ANN Implementation in FPGA Devices" - Recent Researches in Circuits, Systems, Electronics, Control & Signal Processing 2010.

[13] Etienne Dumesnil, Philippe-Olivier Beaulieu and Mounir Boukadoum – "Robotic Implementation of Classical and Operant Conditioning as a Single STDP Learning Process" - 2016 International Joint Conference on Neural Networks (IJCNN)- Vancouver, BC, Canada

[14] Raghid Morcel, Mazen Ezzeddine, and Haitham Akkary – "FPGA-based Accelerator for Deep Convolutional Neural Networks for the SPARK Environment" - 2016 IEEE International Conference on Smart Cloud (SmartCloud) - New York, NY, USA

[15] Jingyang Zhu, Zhiliang Qian, Chi-Ying Tsui - "A Memory-Efficient Accelerator for Compressing Deep Neural Networks with Blocked Hashing Techniques" - 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC) - Chiba, Japan

[16] Wenlai Zhao, Haohuan Fu, Wayne Luk, Teng Yu, Shaojun Wang, Bo Feng, Yuchun Ma, Guangwen Yang – "F-CNN: An FPGA-based Framework for Training Convolutional Neural Networks" - 2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP) - London, UK

[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] Jorge C. Romero-Aragon, Edgar N. Sanchez, Alma Y Alanis - "FPGA Neural Identifer for Insulin-Glucose Dynamics"- World Automation Congress (WAC), 2014 - Waikoloa, HI, USA.