

Mở đầu

Mạng nơ ron nhân tạo (Artificial Neural Networks – ANN), vốn được dựa trên những nơ ron sinh học trong não bộ con người, đã thúc đẩy sự phát triển của những thuật toán học máy trong vài thập kỷ vừa qua. Những tiến bộ gần đây về sức mạnh tính toán với nền tảng GPU, cùng với sự phát triển của các tập dữ liệu huấn luyện lớn, đã dẫn đến sự ra đời của những kiến trúc mạng ANN sâu với rất nhiều lớp. Mô hình đó được gọi là mạng nơ ron sâu (Deep Neural Networks – DNN). DNN đã được sử dụng trong rất nhiều ứng dụng khác nhau như nhận dạng ảnh, nhận dạng đồ vật, nhận dạng giọng nói, chơi game v.v.

Tuy nhiên, mạng ANN có những sự khác nhau cơ bản khi so với cách thức các nơ ron trong não bộ con người hoạt động, bao gồm các cơ chế truyền thông và quá trình học của các nơ ron. Vì não bộ con người có khả năng tính toán rất hiệu quả, nên một mô hình mạng nơ ron thể hệ thứ ba, gần gũi với cách thức não bộ con người hoạt động hơn, đã được đề xuất. Mô hình mạng đó gọi là mạng nơ ron xung (Spiking Neural Networks – SNN). SNN đã được chứng minh là có năng lực tính toán tương đương với DNN nhưng có hiệu quả sử dụng năng lượng cao hơn.

Tuy nhiên, khi kích thước của SNN và DNN tăng, yêu cầu tính toán của SNN và DNN cũng liên tục tăng lên. Điều này dẫn đến quá trình thực thi của SNN trên các mô hình kiến trúc máy tính Von Neumann truyền thống trở nên kém hiệu quả. Cộng đồng nghiên cứu VLSI đã nghiên cứu và đề xuất nhiều các mô hình phần cứng mới, dựa trên cách thức não bộ con người hoạt động. Lĩnh vực nghiên cứu này được gọi là Neuromorphic Computing. Tuy nhiên, vẫn có những thách thức như sau được đặt ra cho các nhà nghiên cứu:

- Các kiến trúc của mạng SNN hiện nay đều có kích thước rất lớn với rất nhiều trọng số. Khi thực thi trên phần cứng, việc lưu trữ tất cả trọng số trên các bộ nhớ trên chip là bất khả thi. Các trọng số cần phải được lưu trữ trên các bộ nhớ ngoài như DRAM và chuyển đến bộ nhớ trên chip để xử lý. Việc này sẽ làm năng lượng tiêu thụ của hệ thống.
- Phép lọc trong các mạng nơ ron xung tích chập hiện nay chỉ được giới hạn với phép lọc trung bình, vốn có hiệu quả thấp. Một phép lọc cực đại hiệu quả cho SNN là cần thiết.

Luận án của NCS muốn đề xuất các giải pháp hiệu quả để giải quyết hai vấn đề nêu trên. Các đóng góp mới của luận án bao gồm:

- Kiến trúc phần cứng cho mô hình nơ ron LIF và kiến trúc phần cứng cho mạng SNN truyền thẳng với 3 lớp cho ứng dụng nhận dạng chữ số viết tay MNIST.
- Một thuật toán huấn luyện mới cho mạng SNN với các trọng số tam phân (TW-SNN). Để đánh giá hiệu quả tiết kiệm năng lượng của TW-SNN, một kiến trúc phần cứng mới cho TW-SNN cũng được đề xuất.
- Một phương pháp lọc cực đại mới cho mạng SNN tích chập cũng được đề xuất.

Chương 1 Tổng quan về mạng nơ ron xung từ góc độ phần mềm và phần cứng

Chương này giới thiệu tổng quan về SNN, từ góc độ đánh giá phần mềm và phần cứng. Các thành phần cơ bản của SNN như cách thức các nơ ron hoạt động và cách mã hóa thông tin trong SNN được trình bày ở đây. Một vài phương pháp huấn luyện phổ biến cho SNN cũng được giới thiệu. Cuối cùng, tổng quan về các kiến trúc phần cứng phổ biến hiện nay cho SNN cũng được tổng hợp. Thông qua các đánh giá từ góc độ phần mềm và phần cứng, luận án mong muốn chỉ ra các phương pháp tiềm năng để tăng cường hiệu suất năng lượng của các mạng SNNs hiện nay trên cả phần mềm và khi triển khai trên các kiến trúc phần cứng riêng biệt.

1.1 Giới thiệu về mạng nơ ron xung

1.1.1 Mô hình nơ ron

Mô hình nơ ron phổ biến nhất là mô hình Leaky Integrate and Fire (LIF), khi các xung đầu vào được cộng dồn ở điện thế thân nơ ron, và một giá trị rò sẽ được trừ vào điện thế này. Nếu điện thế ở thân nơ ron vượt qua một ngưỡng nhất định, nơ ron sẽ gửi một tín hiệu xung đầu ra. Mô hình LIF có ưu điểm là dễ thực thi trên phần cứng. Các mô hình nơ ron khác cũng được quan tâm gần đây là các nơ ron Hodgkin-Huxley hoặc là mô hình nơ ron Izhikevich.

1.1.2 Mã hóa thông tin trong SNN

Có ba phương pháp chính để mã hóa thông tin đầu vào trong SNN, bao gồm mã hóa dựa vào tỷ lệ xuất hiện của xung (rate coding), mã hóa dựa vào thời gian đến có lúc xung đầu tiên (Time To First Spike – TTFS), và dựa vào khoảng thời gian giữa hai xung liên tiếp (Inter-Spike Interval – ISI).

1.2 Các thuật toán huấn luyện cho SNN

Có ba phương pháp chính để huấn luyện cho mạng SNN. Thứ nhất là phương pháp học không giám sát với STDP, thứ hai là phương pháp học có giám sát với biến thể của thuật toán lan truyền ngược, và cuối cùng là chuyển một mạng DNN đã huấn luyện sang mạng SNN.

1.2.1 Học không giám sát với STDP

Phương pháp này có ưu điểm là tương đồng với cách thức học của não bộ con người, với khả năng học trực tuyến, nhưng không đạt độ chính xác cao với các mạng nơ ron sâu với nhiều lớp.

1.2.2 Học giám sát với thuật toán lan truyền ngược

Phương pháp này có thể đạt được độ chính xác cao hơn so với phương pháp học không giám sát, với một khả năng học online nhất định, nhưng không đạt được độ chính xác cao cho một số tập dữ liệu phức tạp.

1.2.3 Chuyển từ mạng DNN sang mạng SNN

Phương pháp này có thể tận dụng được nhiều thuật toán huấn luyện đã có sẵn cho các mạng DNN. Phương pháp huấn luyện này có thể đạt được độ chính xác cao cho một vài tập dữ liệu phức tạp nhưng không phù hợp cho các ứng dụng đòi học trực tuyến.

1.3 Các kiến trúc phần cứng cho SNN

1.3.1 Các kiến trúc phần cứng cỡ lớn cho SNN

The general strategy for implementing large-scale networks is to have many neuromorphic cores which could operate in parallel. Each core has dedicated neuron update logic and synapse memory. The communication is handled with a scalable Network-on-Chip. Notable works include SpiNNaker, TrueNorth, Neurogrid, and Loihi.

Phương pháp tổng quát để thiết kế và thực thi các mạng SNN cỡ lớn là sử dụng rất nhiều lõi hoạt động song song. Mỗi lõi đều có logic để cập nhật các trạng thái hoạt động của nơ ron và bộ nhớ riêng để lưu

trữ trọng số. Mạng trên chip được sử dụng để giải quyết vấn đề truyền thông giữa các lõi. Một số kiến trúc tiêu biểu được trình bày trong chương này như SpiNNaker, TrueNorth, Neurogrid và Loihi.

1.3.2 Các kiến trúc phần cứng công suất thấp cho SNN

Notable works include the ODIN chip by Frenkel et al., the work by Yin et al., Zheng et al., and Chen et al. Most of the works focus on small-scale network design with the MNIST dataset as the benchmark applications.

Các công trình tiêu biểu bao gồm chip ODIN bởi Frenkel et al., các công trình bởi các tác giả Yin et al., Zheng et al. và Chen et al. Đa phần các công trình đều thực thi các mạng nơ ron cỡ nhỏ với ứng dụng chủ yếu là ứng dụng nhận dạng chữ số viết tay MNIST.

Kết luận chương

Trong chương này, NCS đã trình bày tổng quan về lĩnh vực nghiên cứu cho SNN, từ đánh giá ở góc độ phần mềm và phần cứng. Từ các kiến trúc phần cứng cho SNN, có thể thấy hiện nay các công trình nghiên cứu vẫn chưa tập trung vào vấn đề giảm các trọng số dựa vào các kỹ thuật lượng tử hóa. Hơn nữa, các kiến trúc phần cứng công suất thấp cho SNN hiện nay chủ yếu tập trung vào thực thi các mô hình mạng nơ ron truyền thẳng. Với mô hình mạng SNN tích chập, việc nghiên cứu phát triển các phép lọc riêng biệt và hiệu quả trên phần cứng cho SNN cũng là rất cần thiết. Luận án của NCS muốn tập trung giải quyết hai vấn đề nêu trên.

Chương 2 Kiến trúc phần cứng hiệu quả cao cho mạng nơ ron xung sâu

Chương này trình bày kiến trúc phần cứng của một nơ ron IF và kiến trúc phần cứng ở mức hệ thống của mạng SNN cho ứng dụng nhận dạng chữ viết với tập dữ liệu MNIST.

2.1 Tổng quan

Phần này trình bày các thông tin liên quan đến các mô hình nơ ron, các thuật toán chuyển đổi, và cách mã hóa thông tin đầu vào dựa vào tần suất xuất hiện của các xung.

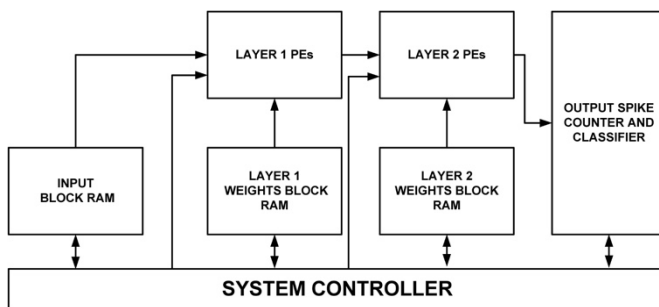
2.2 Kiến trúc phần cứng đề xuất

2.2.1 Kiến trúc của một nơ ron – Đơn vị tính toán cơ bản

Kiến trúc phần cứng đề xuất cho đơn vị tính toán cơ bản là một nơ ron được trình bày ở đây.

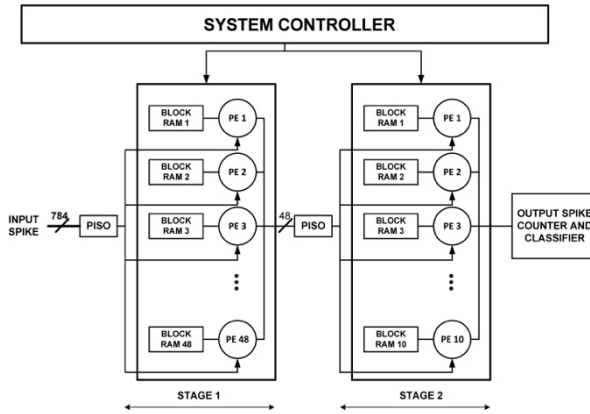
2.2.2 Kiến trúc phần cứng ở mức hệ thống

Sơ đồ khối của phần cứng đề xuất được biểu diễn trong hình 2.7. Kiến trúc chi tiết mức hệ thống của phần cứng đề xuất được biểu diễn



Hình 2.7 Sơ đồ khối của hệ thống.

trong hình 2.8.



Hình 2.8 Kiến trúc mức hệ thống của phần cứng đề xuất

2.3 Kết quả đánh giá

2.3.1 Kết quả mô phỏng phần mềm

Mô hình SNN được chọn là mô hình SNN truyền thẳng gồm 2 lớp, với kích thước $784 \times 48 \times 10$, cho ứng dụng MNIST. Mạng được huấn luyện với MATLAB (với độ chính xác 32b dấu phẩy động), và được lượng tử hóa dưới dạng 10-b dấu phẩy tĩnh. Kiến trúc hệ thống được miêu tả với ngôn ngữ VHDL ở mức dịch chuyển thành ghi (RTL). Kết quả mô phỏng cho thấy mạng có thể đạt độ chính xác 94.6%. Quá trình lượng tử hóa giảm một lượng nhỏ độ chính xác (0.2%) và kết quả mô phỏng phần cứng đã khớp với kết quả mô phỏng phần mềm.

2.3.2 Kết quả thực thi phần cứng

Hệ thống đã được thực thi với thư viện 45nm NANGATE. Kết quả thực thi phần cứng cho một nơ ron IF cho thấy kiến trúc đề xuất có thể hoạt động ở một mức tần số tương đương nhưng giảm được 4,2 lần về chi phí diện tích phần cứng.

Kết quả thực thi phần cứng ở mức hệ thống cho thấy hệ thống tiêu tốn ít chi phí diện tích phần cứng với diện tích phần lõi là 15

μm^2 (19.2k 2-input NAND tương đương), tần số hoạt động tối đa là 250MHz và thông lượng tối đa là 325k ảnh/giây.

Kết luận chương

Trong chương này, NCS đã đề xuất một kiến trúc phần cứng nhỏ gọn cho mạng SNN, với ứng dụng nhận dạng chữ số viết tay. Kết quả mô phỏng cho thấy mặc dù sử dụng độ chính xác dấu phẩy tĩnh với một số bit trọng số có hạn, kiến trúc phần cứng vẫn có thể đạt độ chính xác với các hệ thống sử dụng độ chính xác dấu phẩy động. Một số hướng nghiên cứu phát triển tiếp theo bao gồm việc phát triển kiến trúc phần cứng có thể mở rộng. Ngoài ra một số thuật toán huấn luyện với lượng tử hóa cho SNN cũng sẽ được nghiên cứu.

Chương 3 Mạng nơ ron xung tam phân

Chương 3 trình bày một thuật toán huấn luyện cho mạng SNNs với các trọng số được biểu diễn dưới dạng tam phân (TW-SNN). Thuật toán đã được ứng dụng cho các tập dữ liệu MNIST và CIFAR, với mô hình mạng nơ ron truyền thẳng và mô hình mạng tích chập. Một kiến trúc phần cứng hiệu quả cho TW-SNN cũng được đề xuất.

3.1 Các công trình liên quan

Phần này trình bày các công trình liên quan đến các thuật toán huấn luyện cho SNNs và các phương pháp lượng tử hóa đang được sử dụng cho các mô hình mạng nơ ron.

3.2 Thuật toán huấn luyện đề xuất

3.2.1 Mô hình nơ ron

NCS đã áp dụng một mô hình nơ ron LIF, dựa trên mô hình nơ ron trong ANN truyền thống với một hàm kích hoạt đặc biệt.

3.2.2 Định nghĩa về bước thời gian và độ trễ ứng dụng trong SNNs

Để mô phỏng các hoạt động của mạng SNN, thông thường thời gian mô phỏng sẽ được rời rạc hóa và biểu diễn dưới một số lượng nhất định bước thời gian T . Độ trễ ứng dụng trong được định nghĩa là số lượng bước mô phỏng cần để mạng có thể đạt được một độ chính xác biết trước.

3.2.3 Phân tích về yêu cầu lưu trữ phần cứng và yêu cầu về tiêu thụ năng lượng khi truy cập bộ nhớ trong SNNs

Phần này trình bày các phân tích cụ thể về yêu cầu lưu trữ phần cứng và yêu cầu về tiêu thụ năng lượng khi truy cập bộ nhớ trong SNNs, cho cả mô hình mạng truyền thẳng và mô hình tích chập.

3.2.4 Huấn luyện mạng SNN với trọng số tam phân

Thuật toán huấn luyện đề xuất được trình bày trong thuật toán 2.

Thuật toán 2: Thuật toán huấn luyện cho mạng TW-SNN

Dữ liệu đầu vào: Số epoch để huấn luyện N , ngưỡng lượng tử hóa β , các trọng số được khởi tạo ngẫu nhiên

For $i \leftarrow 1$ **to** N **do**

1. Lượng tử hóa các trọng số
 - Lưu các trọng số với dấu phẩy động 32-bit
 - Tính hệ số tỷ lệ cho trọng số α
 - Tính mức ngưỡng Δ_{th}
 - Tính các trọng số tam phân w_L^{tern}
2. Chạy mạng với các trọng số tam phân
 - Chạy mạng với hàm kích hoạt được nêu trong công thức (2)
 - Tính giá trị mất mát (loss value) khi so với các mục tiêu (targets)
3. Cập nhật các trọng số với thuật toán lan truyền ngược
 - Phục hồi các trọng số với độ chính xác dấu phẩy động
 - Tính gradient
 - Cập nhật các trọng số với độ chính xác dấu phẩy động theo thuật toán lan truyền ngược

Return Các tham số sau huấn luyện của mạng

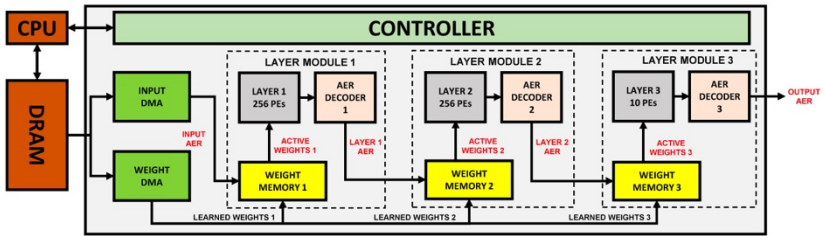
3.3 Kiến trúc phần cứng đề xuất

3.3.1 Kiến trúc phần xử lý lỗi

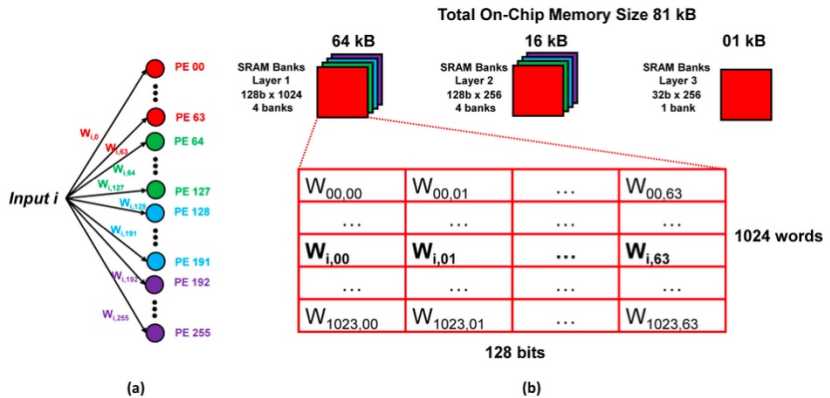
Phần lỗi của kiến trúc đề xuất là một mô đun xử lý lỗi, bao gồm các PEs, bộ giải mã AER và bộ nhớ SRAM để lưu các trọng số.

3.3.2 Kiến trúc phần cứng cho mạng SNN với 3 lớp

Để có thể so sánh hiệu quả năng lượng của thuật toán đề xuất với các kiến trúc phần cứng công suất thấp khác, NCS đã thực thi một kiến trúc phần cứng cho một mạng TW-SNN với 3 lớp (với mô hình FC256-256-10), cho ứng dụng MNIST. Sơ đồ khối của hệ thống được thể hiện trên hình 3.6



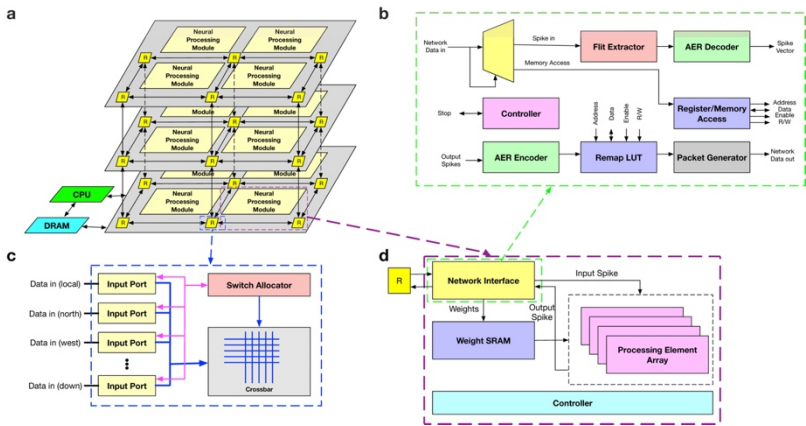
Hình 3.6 Sơ đồ khối của kiến trúc phần cứng cho mạng TW-SNN. Các trọng số tam phân được lưu trữ trên chip với bộ nhớ SRAM. Hình 3.7 biểu diễn cách các trọng số được lưu trữ.



Hình 3.7 Cách thức lưu trữ các trọng số

3.3.3 Kiến trúc phần cứng có thể mở rộng cho TW-SNN

Kiến trúc phần cứng với mạng TW-SNN truyền thẳng, có 3 lớp được trình bày trước đây có ưu điểm là tiêu tốn ít năng lượng và đạt chi phí diện tích phần cứng thấp, tuy nhiên chỉ thích hợp với những mạng SNN nhỏ và không thể mở rộng cho các mô hình mạng lớn hơn. Hình 3.8 trình bày một kiến trúc phần cứng có thể mở rộng, dựa trên một mạng trên chip 3D.



Hình 3.8 Kiến trúc phần cứng có thể mở rộng

3.4 Kết quả mô phỏng

3.4.1 Kết quả cho mô hình mạng TW-SNN truyền thẳng

Các mạng SNNs đã được huấn luyện với PyTorch, với 4 cấu hình mạng khác nhau và với 3 tập dữ liệu khác nhau. Kết quả mô phỏng cho thấy TW-SNN có thể đạt độ chính xác tương đương với các mô hình mạng SNN và DNN sử dụng dấu phẩy động. TW-SNN có thể giảm yêu cầu về bộ nhớ lưu trữ trọng số từ 2.5-3 lần khi so sánh với các công trình khác.

3.4.2 Kết quả cho TW-SNN với mô hình mạng tích chập

Để đánh giá hiệu quả của TW-SNN với các mô hình mạng SNN tích chập, NCS đã huấn luyện một mô hình mạng sâu dựa trên mô hình mạng VGG và kiểm thử trên tập dữ liệu CIFAR-10. Kết quả cho thấy TW-SNN có thể đạt độ chính xác tương đương với phương pháp chuyển đổi trực tiếp từ DNN và các mạng DNN truyền thống. Khi so sánh với các mạng SNN khác, TW-SNN có giảm độ chính xác (0.74% - 1.84%), nhưng có độ trễ ứng dụng thấp hơn và có thể giảm đến 16 lần về yêu cầu bộ nhớ lưu trữ.

3.5 Kết quả thực thi phần cứng

3.5.1 Kết quả cho mô hình TW-SNN truyền thẳng ba lớp

Kiến trúc phần cứng cho mô hình TW-SNN truyền thẳng ba lớp được mô tả bằng ngôn ngữ VHDL và thực thi với thư viện công nghệ TSMC 65nm. Hình 3.15 biểu diễn chip layout. Tổng diện tích thiết kế là 0,96 mm², với 0,24 mm² diện tích cho phần logic và 0,72 mm² cho bộ nhớ.

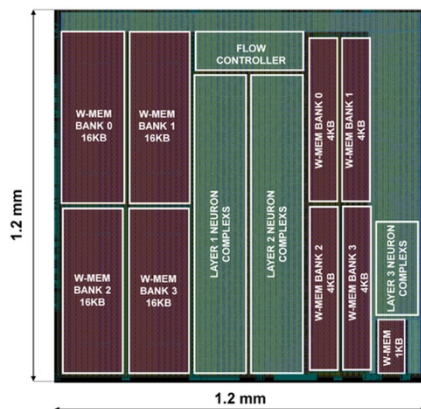


Figure 3.15 Chip layout cho thiết kế TW-SNN truyền thẳng 3 lớp.

Hình 3.17 thể hiện sự so sánh giữa độ chính xác, năng lượng tiêu thụ và thông lượng giữa thiết kế đề xuất và các thiết kế khác. Thiết kế

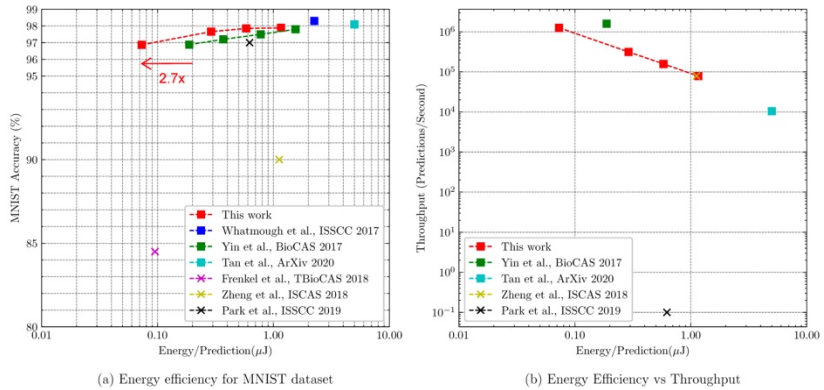


Figure 3.17 So sánh hiệu suất năng lượng với các công trình khác

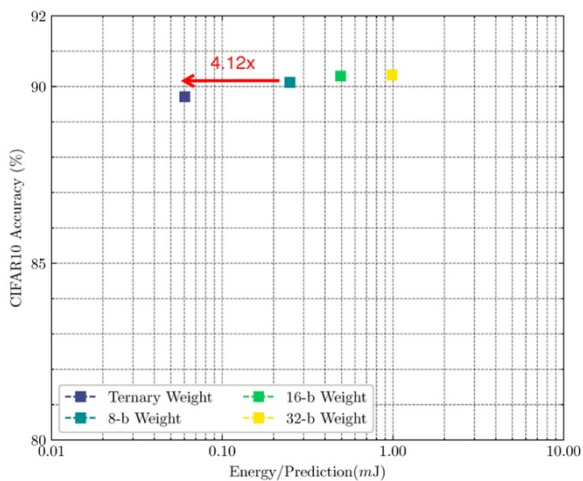
đề xuất giúp giảm 2,7 lần năng lượng tiêu thụ khi so ở cùng độ chính xác (97%-98%) với các công trình khác.

3.5.2 Kết quả của một lỗi cho thiết kế có thể mở rộng của TW-SNN

Kết quả tổng hợp phản ứng cho một lỗi trong thiết kế có thể mở rộng cho TW-SNN được trình bày ở đây.

3.5.3 Ước lượng về năng lượng tiêu thụ của TW-SNN với mô hình mạng tích chập

Phần này trình bày ước lượng về năng lượng tiêu thụ của TW-SNN khi sử dụng với mô hình mạng tích chập. Kết quả được biểu diễn trong hình 3.19.



Energy efficiency for CIFAR10 dataset

Figure 3.19 Energy Efficiency vs Classification Accuracy for CIFAR10 dataset

Từ kết quả có thể thấy, năng lượng tiêu thụ trung bình cho một kết quả dự đoán với CIFAR10 là $6.01e-01$ mJ. Đồng thời, TW-SNN có thể giúp tiết kiệm từ 4,12-15,35 lần về mặt hiệu quả sử dụng năng lượng, khi so sánh với các mạng SNN sử dụng 8-b hoặc 32-b để biểu diễn trọng số.

3.6 Thảo luận

Phần này trình bày một số vấn đề còn tồn tại và hướng giải quyết.

Kết luận chương

Trong chương này, NCS đã đề xuất TW-SNN, là một thuật toán huấn luyện hiệu quả cho mạng SNNs, với các trọng số của mạng được biểu diễn dưới dạng tam phân. Kết quả mô phỏng phần mềm cho thấy TW-SNN có thể đạt độ chính xác lên đến 97% với tập dữ liệu MNIST và đạt độ chính xác lên đến 89.71% với tập dữ liệu CIFAR10. NCS cũng đề xuất kiến trúc phần cứng để thực thi mạng SNN được huấn luyện với TW-SNN. Kiến trúc phần cứng với mạng TW-SNN truyền thẳng với 3 lớp đạt độ chính xác 97% với tập dữ liệu MNIST và tiêu thụ 74 nJ cho một dự đoán. NCS cũng đề xuất một thiết kế có thể mở rộng để hỗ trợ TW-SNN cho các mạng SNNs có kiến trúc phức tạp hơn.

Chương 4 Một phương pháp lọc cực đại hiệu quả và kiến trúc phần cứng cho phép lọc cực đại ứng dụng cho mạng nơ ron xung tích chập

Chương này trình bày một phương pháp lọc cực đại mới cho mạng CSNN, với ưu điểm là không làm tăng yêu cầu tính toán và yêu cầu về lưu trữ bộ nhớ. Ngoài ra, một kiến trúc phần cứng cho bộ lọc cực đại mới cũng được đề xuất.

4.1 Động lực thực hiện và đóng góp mới

Động lực thực hiện và các đóng góp mới của chương được trình bày ở đây

4.2 Tổng quan và những công trình liên quan

4.2.1 Tổng quan về phép lọc trong mạng DNNs

Trong những mạng DNNs truyền thống, các lớp lọc thường được đặt sau những lớp tích chập để làm giảm chiều của các đặc trưng đầu ra, và có thể giúp cho các mạng DNNs không bị ảnh hưởng bởi những chuyển vị nhỏ trong dữ liệu đầu vào. Các thông số chính của các lớp lọc bao gồm kích thước của cửa sổ lọc N_p và độ trượt S của cửa sổ lọc. Có 2 loại lớp lọc chính được sử dụng trong DNNs, bao gồm lọc cực đại (lọc ra giá trị lớn nhất trong cửa sổ lọc) và lọc trung bình (lọc ra giá trị trung bình của tất cả giá trị trong cửa sổ lọc).

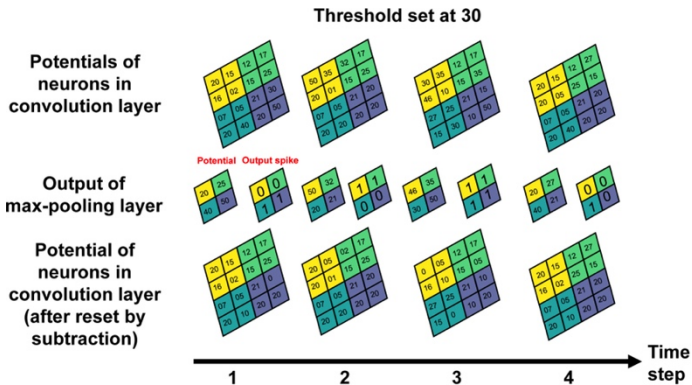
4.2.2 Phép lọc trong SNN

Trong các mạng CSNN, việc xây dựng một bộ lọc cực đại hiệu quả là một việc không dễ dàng, khi mà dữ liệu đầu ra của các lớp tích chập trong mạng CSNN là những giá trị xung nhị phân. Khi lọc ra giá trị cực đại trong một cửa sổ chỉ bao gồm những giá trị nhị phân, việc mất mát dữ liệu có thể xảy ra. Do đó, trong những nghiên cứu ban đầu về việc sử dụng lớp lọc trong SNN, các bộ lọc trung bình thường được sử dụng. Tuy nhiên những bộ lọc này dễ gây suy giảm về độ chính xác.

Các công trình nghiên cứu gần đây được thực hiện bởi các tác giả Rueckauer et al. và Guo et al. đã đề xuất các phương pháp thực hiện phép lọc cực đại. Phần này trình bày cụ thể các phương pháp hiện nay được sử dụng để thực hiện phép lọc cực đại trong CSNN.

4.3 Phép lọc cực đại đề xuất

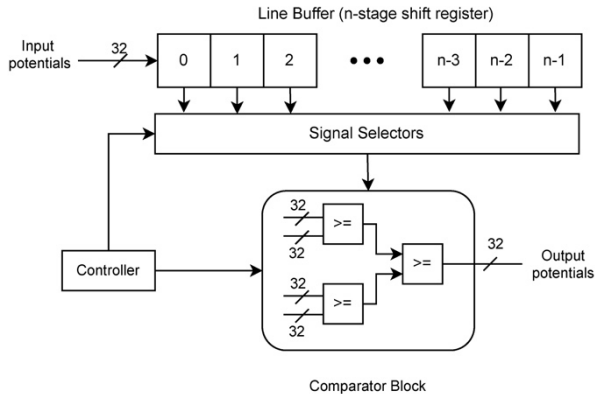
NCS đề xuất một phép lọc cực đại mới, trong đó điện thế ở thân nơ ron sau mỗi lớp tích chập được sử dụng để xác định các kết quả của lớp lọc. Hình 4.4 thể hiện phép lọc đề xuất.



Hình 4.4 Phép lọc đề xuất

4.4 Kiến trúc phần cứng cho phép lọc đề xuất

Một kiến trúc phần cứng để đánh giá hiệu quả của phép lọc đề xuất được trình bày ở đây. Hình 4.6 thể hiện kiến trúc phần cứng của khối lọc được đề xuất.



Hình 4.6 Kiến trúc phần cứng cho khối lọc đề xuất

4.5 Kết quả đánh giá

4.5.1 Tập dữ liệu và mô hình mạng

NCS đánh giá kết quả nhận dạng với phép lọc đề xuất trên các tập dữ liệu MNIST và CIFAR, với các mô hình mạng nông và sâu (dựa trên mô hình mạng VGG).

4.5.2 Kết quả mô phỏng

Với tập dữ liệu MNIST, phương pháp lọc đề xuất đạt được độ chính xác cao nhất là 99.2%, so với độ chính xác 99.38% của mạng DNN và 99.3% của phương pháp lọc của Rueckauer et al. Với tập dữ liệu CIFAR-10, phương pháp lọc đề xuất làm giảm độ chính xác lần lượt là 5.9% và 4.3% khi so sánh với hai phương pháp nêu trên.

4.5.3 Kết quả thực thi phần cứng

Khối kiến trúc phần cứng cho phép lọc đề xuất đã được mô tả với ngôn ngữ Verilog và tổng hợp với các phần mềm của Synopsys trên thư viện NANGATE 45nm. Khối kiến trúc phần cứng đề xuất hỗ trợ một dữ liệu đầu vào với kích thước tối đa là 32×32 nơ ron. Kết quả thực thi phần cứng cho thấy khối phần cứng là gọn nhẹ với kết quả cổng logic tương đương là 15.3k gate, với thông lượng tối đa là 326k ảnh/s.

Kết luận chương

Trong chương này, NCS đã đề xuất một phương pháp lọc cực đại mới cho mạng CSNN để giải quyết các vấn đề tồn tại với phép lọc trong các mạng CSNN. Các vấn đề chính bao gồm việc yêu cầu nhiều bộ nhớ để lưu trữ lịch sử của các xung, đồng thời các yêu cầu tính toán tăng cường của nơ ron LIF. NCS cũng đã đề xuất một mô hình kiến trúc phần cứng mới cho phương pháp lọc đề xuất. Kết quả mô phỏng phần mềm trên tập dữ liệu MNIST và CIFAR10 đã chứng minh hiệu quả của đề xuất. Kết quả thực thi phần cứng của khối lọc đề xuất đã chứng minh tính gọn nhẹ của kiến trúc, và sẽ được sử dụng như một khối tính toán cơ bản trong các kiến trúc phần cứng mới cho CSNN của NCS.

Kết luận và hướng phát triển

Mô hình mạng SNN đã được đề xuất và nghiên cứu từ lâu. SNN chủ yếu được ứng dụng trong việc mô phỏng hoạt động của não bộ con người. Tuy nhiên với những tiến bộ gần đây trong việc tìm ra các thuật toán huấn luyện cho SNNs, SNN đã được ứng dụng rộng rãi cho một số tác vụ học máy. Nó cũng đã được chứng minh là ngang bằng về khả năng tính toán, đồng thời cho thấy tiềm năng có thể đạt mức tiêu thụ điện năng và chi phí diện tích phần cứng thấp hơn nhiều khi được triển khai trong phần cứng. Điều này có được là do cơ chế hoạt động hướng sự kiện vốn có của mạng SNN và cơ chế hoạt động đơn giản của các nơ ron LIF. Mặc dù có những lợi thế nhất định, tuy nhiên SNN vẫn còn tồn tại một số vấn đề. Đầu tiên là việc yêu cầu rất nhiều bộ nhớ lưu trữ khi tính toán của các mạng SNN tân tiến hiện nay. Việc này cản trở việc thực thi SNN trên các nền tảng tính toán biên vì năng lượng từ việc truy cập bộ nhớ sẽ chi phối mức tiêu thụ năng lượng của cả nền tảng. Nhược điểm thứ hai của SNN là hiện nay đang thiếu các phương pháp lọc tối đa cho các mạng SNN có tô pô mạng phức tạp như mạng SNN tích chập.

Các đóng góp chính của luận án có thể được tóm tắt như sau

- Một kiến trúc phần cứng cho mạng SNN truyền thẳng đã được đề xuất. NCS đề xuất một thiết kế số đơn giản của một nơ ron, và đóng vai trò là khối phần cứng chính để xây dựng một hệ thống SNN truyền thẳng gồm 3 lớp. Mạng SNN đã được huấn luyện với thuật toán chuyển đổi trực tiếp từ ANN sang DNN. Các kết quả mô phỏng đã cho thấy SNN khi được thực thi trên phần cứng, với các trọng số giới hạn ở độ chính xác dấu phẩy thập 10-b đã không làm giảm độ chính xác một cách đáng kể. Các kết quả thực thi phần cứng đã chứng minh hệ thống đề xuất

phù hợp cả về chi phí diện tích phần cứng và thông lượng khi so sánh với các công trình khác.

- Để giúp giảm hơn nữa yêu cầu về bộ nhớ lưu trữ cho các trọng số, NCS đề xuất TW-SNN, một thuật toán huấn luyện mới cho SNN với các trọng số được biểu diễn dưới dạng tam phân (cần 2-bit để biểu diễn). Điều này là rất quan trọng vì với yêu cầu bộ nhớ lưu trữ nhỏ hơn, các trọng số của mạng có thể được lưu trữ hoàn toàn trên chip, giúp giảm các năng lượng tiêu thụ từ việc truyền dữ liệu từ bộ nhớ ngoài sang bộ nhớ trên chip. Kết quả mô phỏng phần mềm cho thấy TW-SNN có thể đạt độ chính xác tốt khi huấn luyện các mạng SNN truyền thẳng và các mạng SNN tích chập. NCS cũng đề xuất một kiến trúc phần cứng cho SNN, với kiến trúc truyền thẳng ba lớp và một kiến trúc phần cứng có thể mở rộng với mạng trên chip 3D. Các kết quả thực thi phần cứng và ước lượng năng lượng tiêu thụ đã chứng minh tính hiệu quả về mặt sử dụng năng lượng của TW-SNN trên các nền tảng yêu cầu công suất tiêu thụ thấp.
- Để có thể sử dụng các phép lọc cực đại với mạng SNN, NCS cũng đề xuất một phương pháp lọc cực đại mới cho mạng SNN. Phương pháp đề xuất không yêu cầu thêm về lưu trữ bộ nhớ và tính toán. Kết quả mô phỏng phần mềm và thực thi phần cứng đã chứng minh tính hiệu quả của phương pháp lọc và kiến trúc mới được đề xuất.

Mặc dù luận án đã cố gắng giải quyết các vấn đề hiện nay đối với việc thực thi SNN trên các nền tảng tính toán biên, nhưng vẫn có một số hướng nghiên cứu trong tương lai. Mục tiêu tiếp theo của NCS là đề xuất và thực thi một thiết kế hoàn chỉnh và có thể mở rộng với mạng trên chip 3D, dựa trên một node của khối kiến trúc phần cứng được đề

xuất. Tuy nhiên, việc giải quyết các vấn đề của một mạng liên kết rất lớn cũng là một thách thức lớn. Một hướng nghiên cứu khác là áp dụng các công nghệ mới về bán dẫn để thiết kế các kiến trúc phần cứng cho SNN, như sử dụng dây nano siêu dẫn hoặc các memristive crossbar arrays. Tuy nhiên hầu hết các nghiên cứu vẫn đang trong giai đoạn phát triển ban đầu và các kết quả hiện nay đa phần vẫn là từ các kết quả mô phỏng. Các nghiên cứu về các kiến trúc phần cứng mới cho SNN hiện nay vẫn chưa thực sự chín muồi và vẫn còn rất nhiều cơ hội và thách thức trong tương lai.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ CỦA LUẬN ÁN

[C1] **Duy-Anh Nguyen**, Xuan-Tu Tran, Khanh N. Dang, Francesca Iacopi, "A lightweight max-pooling method and architecture for deep spiking convolutional neural networks", 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2020), pp. 209-212, Dec 2020, Halong Bay, Vietnam, ISBN: 978-17281-9396-0.

[C2] **Duy-Anh Nguyen**, Duy-Hieu Bui, Francesca Iacopi, Xuan-Tu Tran, "An Efficient Event-driven Neuromorphic Architecture for Deep Spiking Neural Networks" 2019 32nd IEEE International System-on-Chip Conference (SOCC 2019), pp. 144-149, Sep 2019, Singapore, ISBN: 978-1-7281-3282-6.

[J1] **Duy-Anh Nguyen**, Xuan-Tu Tran, Francesca Iacopi, "A review of algorithms and hardware implementations for spiking neural networks", Journal of Low Power Electronics and Applications, Apr. 2021, Vol. 11, 23, ISSN: 2079-9268 (ESCI, Q2).

[J2] **Duy-Anh Nguyen**, Xuan-Tu Tran, Khanh N. Dang, Francesca Iacopi, "A low-power, high-accuracy with fully on-chip ternary weight hardware architecture for Deep Spiking Neural Networks", Microprocessors and Microsystems, Vol. 90, 2022 104458, ISSN: 0141-9331 (SCIE, Q2).