TẠP CHÍ KHOA HỌC
Khoa học Tự nhiên và Công nghệ

Phan Trung Kiên, Đặng Thị Vân Chi (2023)
(30): 16-22

HUẤN LUYỆN MÔ HÌNH NHẬN DẠNG CÔNG THỨC TOÁN HỌC VỚI NHIỀU GPU

Phan Trung Kiên, Đặng Thị Vân Chi
Trường Đại học Tây Bắc

Tóm tắt: *Trong bài báo này, chúng tôi trình bày về cách sử dụng nhiều GPU để huấn luyện mô hình trong học sâu (Deep Learning). Chúng tôi khảo sát các chiến lược học sâu trên mạng nơ-ron tích chập (Convolutional Neural Network – CNN). Tập dữ liệu được huấn luyện là IM2LATEX-100K, được cung cấp bởi WYGIWYS, chứa 103.556 biểu thức LaTeX khác nhau được trích xuất từ hơn 60.000 bài báo khoa học. Phần mềm chúng tôi sử dụng thực nghiệm là Pytorch Lingting với mạng ConvMath trên hệ thống máy tính có 4 bộ xử lý đồ họa (Graphics Processing Units - GPU). Kết quả thực nghiệm cho thấy tốc độ huấn luyện tăng lên đáng kể khi sử dụng nhiều GPU..*

Từ khóa: *Học sâu, CNN, GPU, nhận dạng công thức toán học.*

1. GIỚI THIỆU

Nhận dạng kí tự quang học (Optical Character Recognition – OCR) là lĩnh vực nghiên cứu cách chuyển đổi ảnh số được chụp hay quét từ tài liệu viết tay, đánh máy hay in thành dạng văn bản máy tính có thể hiểu được. Trên thế giới, công nghệ OCR đã có những tác động sâu sắc đến nhiều lĩnh vực trong sản xuất và đời sống. Ngày nay, với sự phát triển nhanh chóng của công nghệ số, việc thực hiện chuyển đổi số trong các lĩnh vực nhất là lĩnh vực giáo dục, đào tạo và nghiên cứu khoa học là tất yếu. Lượng thông tin trong các lĩnh vực này đang trở nên vô cùng lớn và chủ yếu được lưu trữ, chia sẻ ở định dạng PDF. Việc ứng dụng OCR giúp cho ta có thể truy cập, chỉnh sửa thông tin một các dễ dàng.

Trong các tài liệu trên có chứa một lượng không nhỏ các công thức toán học nên việc nhận dạng công thức là rất cần thiết. Hệ thống nhận dạng biểu thức toán học đã được nghiên cứu trong nhiều thập kỷ để chuyển đổi hình ảnh của các biểu thức thành một định dạng máy có thể đọc được từ đó chuyển đổi sang các dạng mà ta có thể chỉnh sửa. Các phương pháp hiện có đạt độ chính xác cao đối với các biểu thức độc lập nhưng độ chính xác thấp đối với các biểu thức có trong tài liệu PDF cùng nhiều các dữ liệu khác. Một phương pháp gần đây được sử dụng nhiều gần đây và cho kết quả tốt là ứng dụng học máy, nhất là kỹ thật học sâu, để nhận dạng công thức.

Một kiến trúc được sử dụng rộng rãi trong kỹ thuật học sâu là mạng nơ-ron tích chập đã đạt được mức hiệu suất mới cho các tác vụ OCR [1]. Các kỹ thuật như phân loại thời gian liên kết [2] mô hình hóa các phụ thuộc liên nhãn một cách ngầm định, làm cho nó có thể huấn luyện một mạng nơ-ron trực tiếp với dữ liệu chưa được phân đoạn. Các giải pháp hiện có để dự đoán trình tự từ đầu vào hình ảnh có thể được tìm thấy trong các nhiệm vụ nhận dạng văn bản và chú thích hình ảnh [2], [3], thường kết hợp CNN với một mô hình tuần tự để tạo một bộ mã hóa - bộ giải mã (seq2seq).

Các thuật toán học sâu, với lượng dữ liệu và khối lượng tính toán lớn, đạt được nhiều thành công nhờ sử dụng các phần cứng hiệu năng cao như GPU. GPU có khả năng tính toán cao hơn nhiều so với CPU. Trong nghiên cứu này chúng tôi khảo sát việc huấn luyện mô hình nhận dạng công thức toán học với mạng ConvMath do Z. Yan và cộng sự [4] đề xuất trên máy tính với 4 GPU.

2. MẠNG NƠ-RON TÍCH CHẬP CNN

Thị giác máy tính (Computer Vision – CV) là lĩnh vực khoa học xác định cách máy diễn giải ý nghĩa của hình ảnh và video. Các thuật toán CV phân tích các tiêu chí nhất định trong hình ảnh và video, sau đó áp dụng các diễn giải cho các nhiệm vụ dự đoán hoặc ra quyết định.

Ngày nay, mạng nơ-ron tích chập CNN là một trong những mô hình học sâu phổ biến nhất và có ảnh hưởng nhiều nhất trong lĩnh vực thị giác máy tính. CNN được dùng trong nhiều bài toán như nhân dạng ảnh, phân tích video, ảnh MRI (ảnh cộng hưởng từ - Magnetic Resonance Imaging), hoặc cho bài các bài của lĩnh vực xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này.

Kiến trúc gốc của mô hình CNN được giới thiệu bởi một nhà khoa học máy tính người Nhật vào năm 1980 [5]. Sau đó, năm 1998, Yan LeCun [6] lần đầu huấn luyện mô hình CNN với thuật toán lan truyền ngược (backpropagation) cho bài toán nhận dạng chữ viết tay. Tuy nhiên, mãi đến năm 2012, khi một nhà khoa học máy tính người Ukraine, Alex Krizhevsky xây dựng mô hình CNN (AlexNet) [1] và sử dụng GPU để tăng tốc quá trình huấn luyện với độ lỗi phân lớp giảm hơn 10% so với những mô hình truyền thống trước đó, đã tạo nên làn sóng mạnh mẽ sử dụng CNN với sự hỗ trợ của GPU để giải quyết càng nhiều các vấn đề trong CV.

2.1. Kiến trúc mạng nơ-ron CNN

Mạng CNN có cấu trúc như hình 1, với đầu vào sẽ được nhân chập với các ma trận lọc, công việc này có thể được xem như phép lọc ảnh với ma trận lọc khi sử dụng dạng ma trận, cũng như phép lọc ảnh bình thường trong không gian 2D thì tích chập này cũng được ứng dụng trong trong không gian ảnh màu 3D và trong cả không gian n chiều. Sau khi thu gọn ma trận dưới dạng một véc tơ thì nó sẽ được kết hợp với một mạng MLP (mạng truyền thẳng nhiều lớp - Multi Layer Perceptron) đầy đủ, các ảnh xám ma trận đầu vào là 2 chiều, còn với ảnh màu ma trận vào sẽ là 3 chiều.

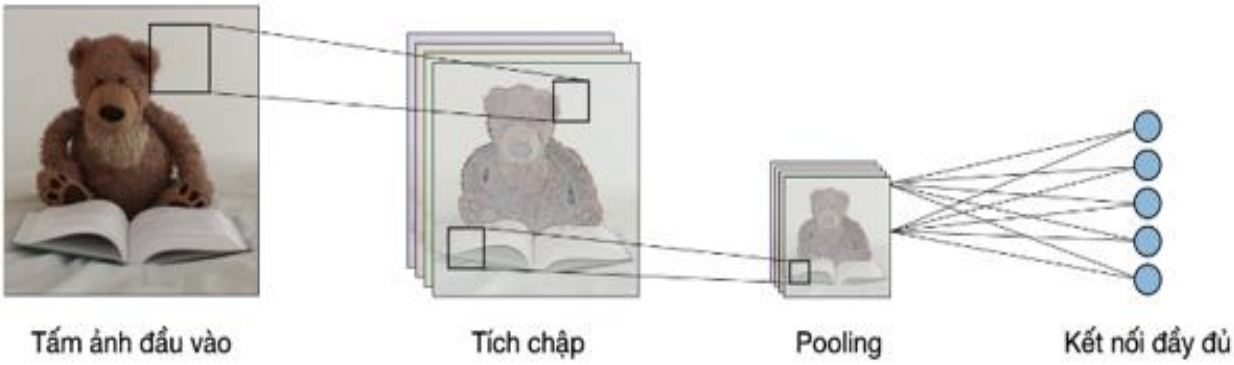
Tầng tích chập (Convolution Layer) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc F và độ trượt S. Kết quả đầu ra O được gọi là feature map hay activation map.

Tầng tích chập có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh. Những đặc trưng này bao gồm đặc trưng cơ bản là góc, cạnh, màu sắc, hoặc đặc trưng phức tạp hơn như texture của ảnh. Vì bộ filter quét qua toàn bộ bức ảnh, nên những đặc trưng này có thể nằm ở vị trí bất kì trong bức ảnh, cho dù ảnh bị xoáy trái/phải thì những đặc trưng này vẫn bị phát hiện.

Tầng pooling (Pooling Layer) là một phép lấy mẫu xuống , thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra. Ý tương đằng sau tầng pooling là vị trí tuyết đối của những đặc trưng trong không gian ảnh không còn cần thiết, thay vào đó vị trí tương đối giữ các đặc trưng đã đủ để phân loại đối tượng. Hơn nữa, giảm tầng pooling có khả năng giảm chiều nhiều, làm hạn chế quá khớp, và giảm thời gian huấn luyện.

Tầng kết nối đầy đủ (Fully Connected Layer) là tầng cuối cùng của mô hình CNN trong bài toán phân loại ảnh. Tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán. Ví dụ, trong bài toán phân loại số viết tay MNIST có 10 lớp tương ứng 10 số từ 0-9, tầng fully connected layer sẽ chuyển ma trận đặc trưng của tầng trước thành vector có 10 chiều thể hiện xác suất của 10 lớp tương ứng.

Cuối cùng, quá trình huấn luyện mô hình CNN cho bài toán phân loại ảnh cũng tương tự như huấn luyện các mô hình khác. Chúng ta cần có hàm độ lỗi để tính sai số giữa dự đoán của mô hình và nhãn chính xác, cũng như sử dụng thuật toán lan truyền ngược cho quá trình cập nhật trọng số.



Hình 1. Kiến túc mạng nơ-ron CNN

2.2. Mạng CNN trong nhận dạng công thức toán học

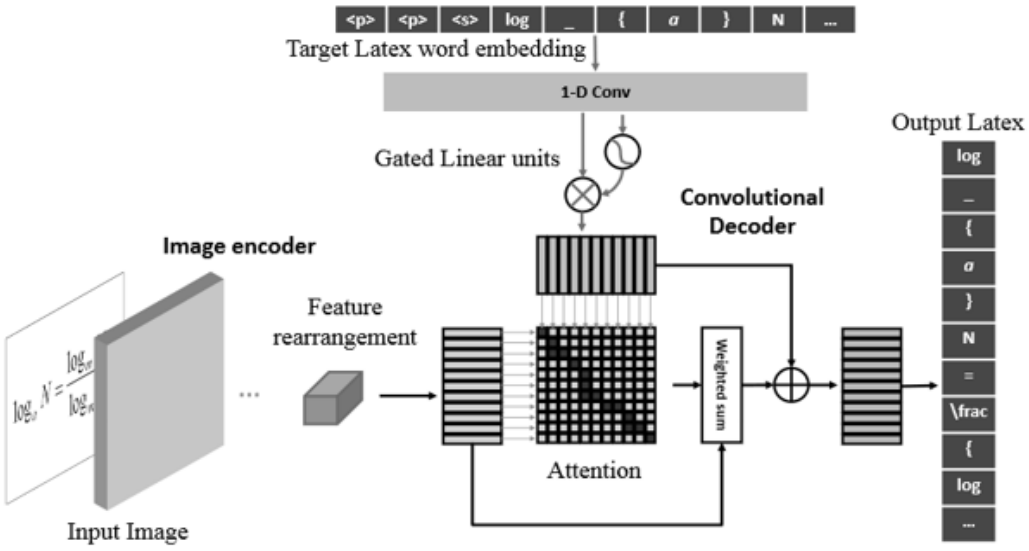
Gần đây nhiều nhà khoa học [4], [7], [8], [9] đã ứng dụng học sâu dùng mạng CNN để xử lý bài toán nhận dạng công thức toán học. Trong đó Z. Yan và cộng sự đề xuất mạng ConvMath [4], một mạng CNN chứa bộ mã hóa hình ảnh để trích xuất đặc trưng và bộ giải mã chập để tạo mã LaTeX. Với sự nhiều lớp, bộ giải mã có thể căn chỉnh hiệu quả các vectơ đặc trưng nguồn và các ký hiệu toán học đích và vấn đề thiếu vùng bao phủ trong khi đào tạo mô hình có thể được giảm bớt phần lớn. Mô hình đã đạt được nhiều kết quả khả quan.

A. Image encoder:

Là một mạng nhiều lớp CNN để tách các đặc trưng của ảnh đầu vào. Để trích xuất đặc trưng từ các hình ảnh biểu thức toán học, các tiêu chí sau phải được xem xét:

(1) Bản đồ tính năng cuối cùng yêu cầu sự kết hợp của cả biểu diễn mức cao và mức thấp của hình ảnh đầu vào. Như chúng ta đã biết, đặc điểm đại diện được CNN trích xuất ngày càng nhiều trừu tượng với sự gia tăng chiều sâu và lĩnh vực tiếp nhận trở nên ngày càng lớn hơn, điều này có lợi cho việc lập mô hình các mối quan hệ 2-D. Các ký hiệu toán học là các đối tượng thường nhỏ và thông tin chi tiết rất hữu ích để xác định chúng.

(2) Bộ mã hóa hình ảnh phải dễ dàng để tối ưu hóa và giữ dung lượng cùng một lúc. Toàn bộ mô hình, ConvMath, để đạt được hiệu suất cao, là tương đối tinh vi và sâu sắc. Vấn đề khét tiếng của độ dốc biến mất hoặc bùng nổ dễ dàng xảy ra khi mô hình đi sâu.



Hình 2. Tổng quan mô hình ConvMath

B. Convolutional decoder

Bộ giải mã này là phỏng theo mô hình của Gehring et al [10]. được phát triển cho phương pháp học theo trình tự. Điểm khác biệt là mạng này hoàn toàn tích chập, có nghĩa là các tính toán không phụ thuộc vào các bước thời gian trước đó và song song hóa trên mọi phần tử có sẵn. Bên cạnh đó, độ dài thay đổi của đầu vào và đầu ra được hỗ trợ, điều này rất quan trọng đối với biểu thức toán học công nhận bởi vì cả kích thước của hình ảnh và chiều dài của chuỗi LaTeX không cố định.

3. HỌC SÂU VỚI NHIỀU GPU

Bộ xử lý đồ họa (GPU) là bộ xử lý máy tính thực hiện các phép tính nhanh để hiển thị hình ảnh và đồ họa. GPU sử dụng xử lý song song để tăng tốc hoạt động của chúng. Chúng chia các tác vụ thành các phần nhỏ hơn và phân bổ chúng cho nhiều lõi bộ xử lý (lên đến hàng trăm lõi) chạy trong cùng một GPU. Ban đầu, GPU chủ yếu được dùng để hiển thị hình ảnh, video và hoạt ảnh 2D và 3D, nhưng ngày nay đã mở rộng phạm vi sử dụng rộng hơn, bao gồm học sâu và phân tích dữ liệu lớn.

Trước khi GPU xuất hiện, các đơn vị xử lý trung tâm (CPU) đã thực hiện các tính toán cần thiết để kết xuất đồ họa. Tuy nhiên, CPU không hiệu quả đối với nhiều ứng dụng phải tính toán song song lớn. GPU giảm tải quá trình xử lý đồ họa và các tác vụ song song ồ ạt từ CPU để mang lại hiệu suất tốt hơn cho các tác vụ điện toán chuyên dụng.

3.1. Kiến trúc GPU

Kiến trúc GPU tập trung vào việc đưa các lõi hoạt động trên nhiều hoạt động nhất có thể và ít tập trung vào khả năng truy cập bộ nhớ nhanh vào bộ đệm của bộ xử lý, như trong CPU. Hình 3 là một kiến trúc GPU NVIDIA điển hình.

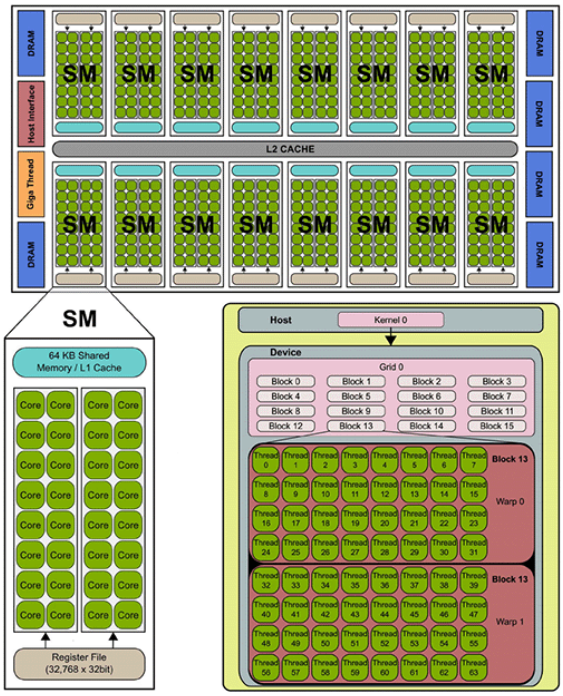
GPU NVIDIA có các thành phần chính:

Cụm bộ xử lý (PC) - GPU bao gồm một số cụm Bộ đa xử lý truyền trực tuyến.

Bộ đa xử lý truyền trực tuyến (SM) - mỗi SM có nhiều lõi bộ xử lý và bộ nhớ đệm lớp 1 cho phép nó phân phối hướng dẫn tới các lõi của nó.

Bộ nhớ đệm lớp 2 - đây là bộ đệm được chia sẻ kết nối các SM với nhau. Mỗi SM sử dụng bộ đệm lớp 2 để truy xuất dữ liệu từ bộ nhớ chung.

DRAM - đây là bộ nhớ chung của GPU, thường dựa trên công nghệ như GDDR-5 hoặc GDDR-6. Nó chứa các hướng dẫn cần được xử lý bởi tất cả các SM.



Hình 3. Kiến trúc GPU NVIDIA

Độ trễ bộ nhớ không phải là yếu tố quan trọng cần cân nhắc trong kiểu thiết kế GPU này. Mối quan tâm chính là đảm bảo GPU có đủ khả năng tính toán để giữ cho tất cả các lõi luôn làm việc.

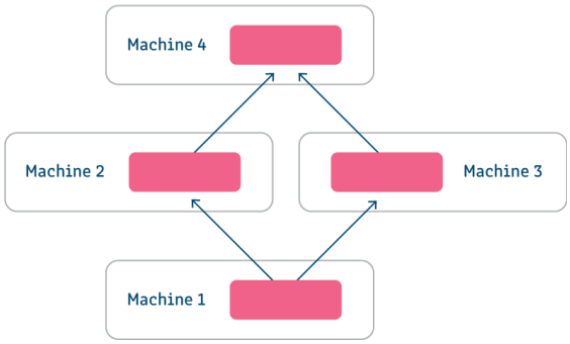
3.2. Học sâu với nhiều GPU

Có hai chiến lược sử dụng nhiều GPU cho phương pháp học sâu là song song hóa mô hình và song song hóa dữ liệu.

A. Song song hóa mô hình

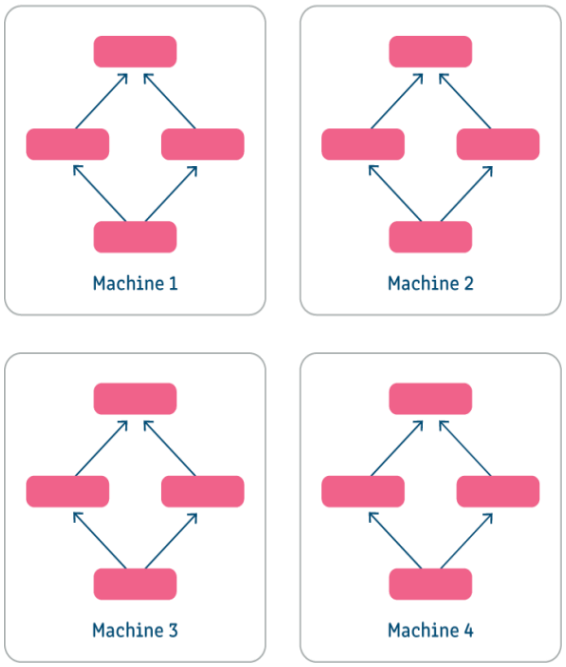
Song song hóa mô hình là một phương pháp bạn có thể sử dụng khi các tham số của bạn quá lớn so với giới hạn bộ nhớ của bạn. Sử dụng phương pháp này, ta chia các quy trình đào tạo mô hình của mình trên nhiều GPU và thực hiện song song từng quy trình (hình 4)

hoặc theo chuỗi. Tính song song của mô hình sử dụng cùng một tập dữ liệu cho từng phần trong mô hình của bạn và yêu cầu đồng bộ hóa dữ liệu giữa các phần tách.



Hình 4. Chiến lược song song hóa mô hình

B. Song song hóa dữ liệu



Hình 5. Chiến lược song song hóa dữ liệu

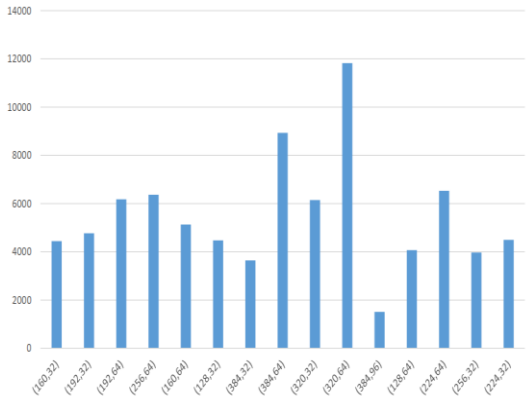
Song song hóa dữ liệu (hình 5) là một phương pháp sử dụng các bản sao mô hình trên các GPU. Phương pháp này hữu ích khi kích thước lô mà mô hình quá lớn để phù hợp với một máy duy nhất hoặc khi bạn muốn tăng tốc quá trình đào tạo. Với tính song song của dữ liệu, mỗi bản sao của mô hình của bạn được đào tạo đồng thời trên một tập hợp con của tập dữ liệu. Sau khi hoàn thành, kết quả của các mô hình được kết hợp và đào tạo tiếp tục như bình thường.

4. KẾT QUẢ THỰC NGHIỆM

4.1. Dữ liệu thực nghiệm và cấu hình phần cứng

Chúng tôi thực nghiệm trên bộ dữ liệu, IM2LATEX-100K, được cung cấp bởi WYGIWYS [11], chứa 103.556 biểu thức LaTeX khác nhau được trích xuất từ hơn 60.000 bài báo từ khoa học. Các biểu thức LaTeX có phạm vi từ 38 đến 997 ký tự với giá trị trung bình là 118 và trung vị là 98. Chúng tôi loại bỏ các biểu thức có chiều rộng hình ảnh lớn hơn 480. Chúng tôi tuân theo các quy tắc tương tự như [7] để mã hóa các chuỗi LaTeX. Sau khi mã hóa, chúng tôi nhận được một từ điển ký hiệu có kích thước là 583. Tập dữ liệu được phân tách ngẫu nhiên thành tập huấn luyện (65.995 biểu thức), tập hợp lệ (8.181 biểu thức) và tập kiểm tra (8.301 biểu thức).

Tập dữ liệu được chia thành 15 nhóm ảnh: (160,32), (192,32), (192,64), (256,64), (160,64), (128,32), (384,32), (384,64), (320,32), (320,64), (384,96), (128,64), (224,64), (256,32), (224,32). Số lượng hình ảnh tương ứng với các nhóm khác nhau được hiển thị trong hình 6.



Hình 6. Số lượng hình ảnh tương ứng với các nhóm

Chúng tôi thực hiện việc huấn luyện mô hình nhận dạng công thức toán học với mạng ConvMath [1] bằng thư viện PyTorch Linghting trên máy tính với Intel(R)

Pentium(R) CPU G4560 @ 3.50GHz, 16GB RAM, 4 NVIDIA GeForce GTX 1060 6GB.

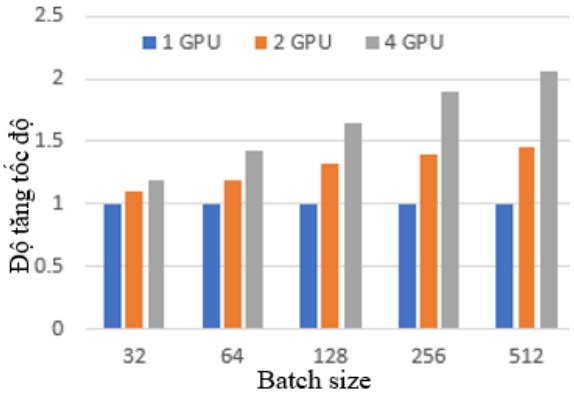
4.3. Kết quả thực nghiệm

Bảng 1 thể hiện các kết quả thực nghiệm khi chúng tôi huấn luyện mô hình với 1, 2 và 4 GPU.

Bảng 1. Kết quả thực nghiệm

Batch size	Thời gian huấn luyện (s)		
	1 GPU	2 GPU	4 GPU
32	202	184	170
64	298	251	209
128	352	265	214
256	394	283	208
512	421	289	205

Biểu diễn theo độ tăng tốc độ huấn luyện ta được hình 7.



Hình 7. Đồ thị độ tăng tốc độ huấn luyện theo batch size

Kết quả cho thấy, với kích thước batch nhỏ, độ tăng tốc thể hiện chưa rõ ràng; với kích thước batch lớn hơn, tốc độ huấn luyện tăng rõ rệt. Tuy nhiên, ngay cả khi kích thước batch lớn, với 4 GPU tốc độ cũng chỉ tăng khoảng 2 lần.

5. KẾT LUẬN

Trong bài báo này, chúng tôi trình bày các kết quả thực nghiệm khi huấn luyện mô hình nhận dạng công thức toán học. Mặc dù với nhiều GPU cho tốc độ huấn luyện càng cao nhưng ta thấy tốc độ chưa tăng như mong muốn. Trong thời gian tới chúng tôi sẽ tiếp tục nghiên cứu và thử nghiệm với các thuật toán, phần mềm khác cũng như phân tích sâu hơn sự ảnh hưởng của kích thước batch đến tốc độ huấn luyện.

6. Lời cảm ơn

Chúng tôi xin cảm ơn đề tài nghiên cứu khoa học cấp cơ sở tại Trường Đại học Tây Bắc *Nhận dạng công thức toán học bằng phương pháp học máy (mã số TB 2021 – 31)* đã hỗ trợ chúng tôi thực hiện bài báo này.

TÀI LIỆU THAM KHẢO

1. Krizhevsky A., Sutskever I., và Hinton G.E. (2012). Imagenet classification with deep convolutional neural networks. 2012 Advances in Neural Information Processing Systems (NIPS). *Neural Inf Process Syst Found Jolla CA*.
2. Graves A., Fernández S., Gomez F., và Schmidhuber J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the 23rd international conference on Machine learning*, 369–376.
3. Shi B., Bai X., và Yao C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans Pattern Anal Mach Intell*, 39(11), 2298–2304
biological tissues and fluids. *Anal Biochem*, 102(1), 176–188.
4. Yan Z., Zhang X., Gao L., Yuan K., và Tang Z. (2020). ConvMath: A Convolutional Sequence Network for Mathematical Expression Recognition. .
5. Fukushima T. và Nixon J.C. (1980). Analysis of reduced forms of biopterin in
6. LeCun Y., Bottou L., Bengio Y., và Haffner P. (1998). Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11), 2278–2324.
7. Gao L., Yi X., Liao Y., Jiang Z., Yan Z., và Tang Z. (2017). A Deep Learning-Based

Formula Detection Method for PDF Documents. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, IEEE, 553–558.

8. Wang Z. và Liu J.-C. (2020). PDF2LaTeX: A Deep Learning System to Convert Mathematical Documents from PDF to LaTeX. *Proceedings of the ACM*
10. Gehring J., Auli M., Grangier D., Yarats D., và Dauphin Y.N. (2017). Convolutional sequence to sequence learning. *International conference on machine learning*, PMLR, 1243–1252.

Symposium on Document Engineering 2020, New York, NY, USA, Association for Computing Machinery, 1–10.

9. Wang Z. và Liu J.-C. (2020). Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training. *Int J Doc Anal Recognit IJDAR*.

11. Deng Y., Kanervisto A., và Rush A.M. (2016). What you get is what you see: A visual markup decompiler. *ArXiv Prepr ArXiv160904938*, 10, 32–37.

TRAINING MATHEMATICAL EXPRESSION RECOGNITION MODEL WITH MULTIPLE GPUs

Phan Trung Kien, Đang Thi Van Chi
Tay Bac University

Abstract: *This paper reports the findings of an trial in which multiple GPUs were used to train mathematical expression recognition in Deep Learning. The different learning strategies were investigated using a Convolutional Neural Network (CNN). The dataset used for the training was IM2LATEX-100K, provided by WYGIWYS containing 103,556 different LaTeX expressions extracted from more than 60,000 articles from arXiv. Pytorch Lingting library with a convolutional sequence modeling network with ConvMath was employed on a computer system with 4 GPUs. The findinds showed an increased training speed with multiple GPUs.*

Keywword: *Deep Learning, CNN, GPU, mathematical expression recognition model.*

Ngày nhận bài: 28/11/2022. Ngày nhận đăng: 20/12/2022
Liên lạc: Phan Trung Kiên. e-mail: kienptr@utb.edu.vn