ISSN 2354-0575

NHẬN DẠNG KÝ TỰ VIẾT TAY SỬ DỤNG MẠNG NƠRON TÍCH CHẬP

Nguyễn Quang Hoan[1], Phạm Ngọc Hưng[2], Nguyễn Đình Tài[3]

1,2 Trường Đại học Sư phạm Kỹ thuật Hưng Yên

3. Học viện Công nghệ Bưu chính Viễn thông

Ngày tòa soạn nhận được bài báo: 14/10/2019

Ngày phản biện đánh giá và sửa chữa: 14/11/2019

Ngày bài báo được duyệt đăng: 11/12/2019

Tóm tắt:

Bài báo phát hiện và nhận dạng ảnh ký tự viết tay sử dụng mạng nơron tích chập (CNN – Convolutional Neural Network) và các giải thuật xử lý ảnh. Đóng góp chính của bài báo là nêu một phương pháp nhận dạng ký tự Latinh viết tay sử dụng mạng nơron tích chập. Dữ liệu thử nghiệm nhận dạng được lấy từ bộ cơ sở dữ liệu ký tự mẫu viết tay Viện Tiêu chuẩn, Kỹ thuật Quốc gia Hoa Kỳ (NIST). Kết quả quả thử nghiệm nhận dạng ký tự viết tay đạt độ chính xác khả quan.

Từ khoá: *Mạng nơron học sâu, mạng nơron tích chập, nhận dạng ký tự viết tay.*

1. Giới thiệu

Nhận dạng ký tự quang học (Optical Character Recognition: OCR) là quá trình xử lý, chuyển đổi ảnh các ký tự viết tay hoặc ký tự đánh máy thành các dữ liệu đã được số hoá sau đó trích chọn đặc trưng và nhận dạng. Thực tế cho thấy, tỉ lệ nhận dạng các ký tự đánh máy của nhiều hệ thống sử dụng mạng học sâu hiện nay đạt độ chính xác tới 99% [8]. Tuy nhiên, việc nhận dạng ký tự viết tay hiện nay là bài toán chưa có phương pháp giải quyết triệt để và vẫn là một thách thức đối với các nhà nghiên cứu do những khác biệt, biến đổi quá đa dạng trong cách viết, độ nghiêng ký tự viết tay của mỗi người, của màu mực, của chất liệu giấy v.v…

Các bước xây dựng hệ thống nhận dạng ký tự viết tay từ ảnh gồm 2 thành phần chính:

+ Bộ tiền xử lý và phát hiện ký tự.

+ Bộ phân loại, nhận dạng ký tự.

Tiền xử lý là quá trình đọc ảnh đầu vào, xử lý các loại nhiễu, tăng độ tương phản, khử độ lệch, phân ngưỡng… với mục đích làm ảnh tốt hơn và thường được thực hiện bởi những bộ lọc. Đầu ra của quá trình này là ảnh các ký tự đa mức xám hoặc nhị phân. Tiếp theo, hệ thống khoanh vùng đối tượng, sao cho nó chỉ chứa duy nhất một ký để nhận dạng. Giai đoạn này còn gọi là bước phát hiện đối tượng (Object Detection) nếu ảnh được xử lý tốt, tức các ký tự được tách rời nhau hoàn toàn và không chứa nhiễu. Nếu ảnh còn nhiễu, hoặc chứa nền phức tạp hoặc chứa các đối tượng khác, ta có thể sử dụng các mạng như R-CNN (Region Convolutional Neural Network) [1]; mạng Fast R-CNN [2], Faster R-CNN [3], RetinaNet để tách, lọc tiếp các ký tự đó. Vì ký tự viết tay đa phần chỉ viết trên các mặt phẳng có phần nền tách biệt với phần chữ nên trong nghiên cứu này, ta giả sử ảnh vào là chữ viết trên giấy thông thường, không chứa các đối tượng và vật thể gây nhiễu khác (như cây cối, xe cộ, con người…) nên không sử dụng các mạng nêu trên vì chúng tương đối phức tạp và làm tăng đáng kể thời gian cũng như chi phí tính toán. Khi bước tiền xử lý ảnh được thực hiện tốt, có thể tiến hành nhận dạng nhanh hơn so với việc dùng các mạng CNN [1, 2. 3].

Bộ nhận dạng ký tự là lớp tiếp theo ngay sau khi dữ liệu ảnh các ký tự đầu vào đã được tách ra từ bước trước. Việc nhận dạng ký tự trên thực tế có rất nhiều phương pháp như: đối sánh mẫu, phương pháp tiếp cận cấu trúc, phương pháp đồ thị, mô hình Markov ẩn, máy vectơ hỗ trợ, mạng nơron v.v…[9]. Ở đây, chúng ta nghiên cứu phương pháp sử dụng mạng nơron tích chập trong nhận dạng ký tự viết tay La tinh do mạng vừa có khả năng lọc, thu nhỏ kích thước ảnh, độ chính

xác cao.

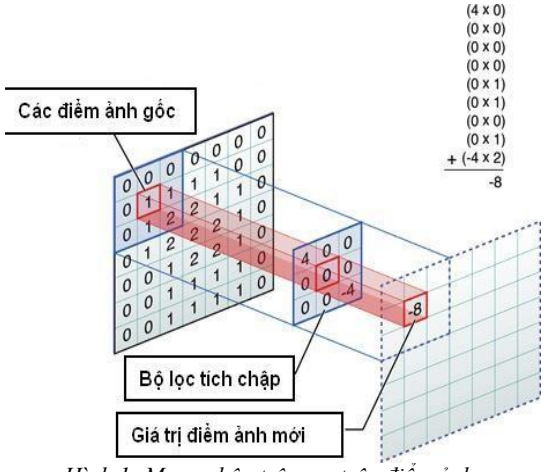
2. Mạng nơron tích chập

CNN là một trong những mô hình học sâu (Deep Learning) tiên tiến. Hiện nay các hãng phần mềm nổi tiếng đã xây dựng các hệ thống thông minh với độ chính xác cao như hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon. CNN sử dụng giải thuật nhân chập bằng một cửa sổ trượt được gọi là nhân (Kernel) hay bộ lọc hay bộ trích chọn đặc trưng để trích chọn đặc ảnh ký tự viết tay (đồng thời có thể giảm kích thước ma trận điểm ảnh) nhằm thu được một ma trận mới gọi là đặc trưng chập (Convoled Features).

2.1 Kiến trúc tổng quát của CNN

Kiến trúc cơ bản của một CNN thông thường gồm 4 lớp: Lớp tích chập (Convolutional Layer), Lớp kích hoạt phi tuyến, Lớp co mẫu (Pooling), Lớp kết nối đầy đủ (Fully Connected). Tuỳ theo thiết kế và cài đặt cũng như mục đích sử dụng mà mỗi mô hình được người thiết kế thêm hoặc bớt các lớp trên để đạt được mô hình với độ chính xác cao và chi phí tính toán thấp. Dưới đây là trình bày chi tiết về 4 lớp cơ bản của một mạng CNN

Lớp tích chập: đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước nhỏ.



Hình 1: Mạng chập trên ma trận điểm ảnh.

Trong Hình 1, bộ lọc được sử dụng là một ma trận có kích thước 3x3, bộ lọc này dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào. Kích thước ảnh đầu ra *O* được xác định tùy theo kích thước các khoảng trắng được thêm ở đường viền ảnh gốc theo công thức sau:

$$O = \frac{i+2*p-k}{s} + 1 \qquad (1)$$

trong đó:

i: kích thước ảnh đầu vào;

p: kích thước khoảng trắng phía ngoài viền; của ảnh gốc;

k: kích thước bộ lọc;

s: bước trượt của bộ lọc.

Khi đưa ảnh vào lớp tích chập, đầu ra của nó ứng với một loạt điểm ảnh. Các bộ lọc được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên và cập nhật trong quá trình huấn luyện.

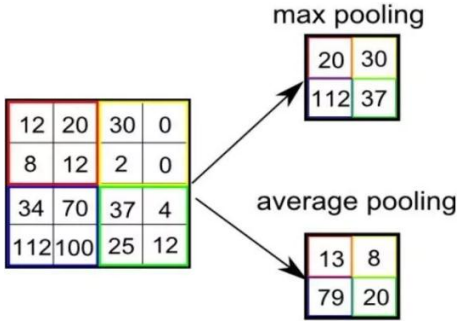
Lớp kích hoạt phi tuyến đảm bảo tính phi tuyến của mô hình huấn luyện sau khi thực hiện một loạt các phép tính toán tuyến tính qua các lớp tích chập. Lớp kích hoạt phi tuyến sử dụng các hàm kích hoạt phi tuyến như *ReLU* hoặc *Sigmoid*, *tanh*… để giới hạn biên độ cho phép của đầu ra. Trong số các hàm kích hoạt này, hàm *ReLU* được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Phép tính hàm *ReLU* đơn giản là chuyển tất cả các giá trị âm thành giá trị 0. Lớp *ReLU* được áp dụng ngay sau lớp tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.

$$(x) = m(0, x) \qquad (2)$$

Lớp lấy mẫu: được đặt sau lớp tích chập và lớp *ReLU* để làm giảm kích thước ảnh đầu ra trong khi vẫn giữ các thông tin quan trọng của ảnh vào. Việc giảm kích thước dữ liệu có tác dụng bớt số các tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu sử dụng cửa sổ trượt để quét các vùng ảnh giống như lớp tích chập, và lấy mẫu thay vì phép tích chập, sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hình 3 thể hiện các kỹ thuật lấy mẫu thường sử dụng là Max Pooling (MP): lấy giá trị mức xám lớn nhất và Avarage Pooling (AP): lấy giá trị

mức xám trung bình của các điểm ảnh trong vùng cục bộ.



Hình 2: Phương thức Avarage Pooling và Max Pooling.

Như vậy, với mỗi ảnh vào, qua lấy mẫu thu được ảnh ra tương ứng, có kích thước giảm đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình nhận dạng.

Lớp kết nối đầy đủ: được thiết kế tương tự như mạng nơ ron truyền thống. Tất cả các điểm ảnh được kết nối đầy đủ với các nơ ron trong lớp tiếp theo.

So với mạng nơ ron truyền thống [4], các ảnh vào của lớp này có kích thước giảm rất nhiều. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ ron truyền thống.

2.2 Xây dựng CNN cho bài toán

Dưới đây chúng tôi nêu một kiến trúc CNN với 8 lớp gồm 1,7 triệu trọng số. Kết quả cho độ chính xác nhận dạng xấp xỉ 90% với các ký tự La tinh viết tay theo dữ liệu đã nêu, cụ thể:

***Ảnh đầu vào*:** Kích thước 28x28x1.

Lớp thứ nhất: *Lớp tích chập* (32 bộ lọc, kích thước 5x5, stride=1, padding=2, hàm kích hoạt=*ReLU*); Input: 28x28x1; Số lượng tham số: (5x5+1)x32=832; Output: 28x28x32

Lớp thứ hai: *Lớp tích chập* (32 bộ lọc, kích thước 5x5, stride=1, padding=2 hàm kích hoạt=*ReLU*); Input: 28x28x32; Số lượng tham số: (5x5x32+1)x32=25.632; Output: 28x28x32

Lớp thứ ba: Lớp Pooling (Kỹ thuật: MP, kích thước 2x2, stride=2); Input: 28x28x32; Ouput: 14x14x32.

Lớp thứ tư: *Lớp tích chập* (64 bộ lọc, kích thước 3x3, stride=1); Input: 14x14x32; Số lượng tham số: (3x3x32+1)x64=18.496; Output: 14x14x64

Lớp thứ năm: *Lớp tích chập* (64 bộ lọc, kích thước 3x3, stride=1); Input:14x14x64; Số lượng tham số: (3x3x64+1)x64=36,928; Output: 14x14x64

Lớp thứ sáu: *Lớp Pooling* (Kỹ thuật: MP, kích thước 2x2, stride=2); Input: 14x14x64; Output: 7x7x64

Lớp thứ bảy: Lớp kết nối đầy đủ (Số nơron: 512, hàm kích hoạt=ReLU)

Input: 7x7x64=3136; Số lượng tham số: 3136x512=1,606,144; Output: 512

Lớp thứ tám: *Lớp kết nối đầy đủ* (số nơron: 62, Hàm kích hoạt = *Softmax*); Input: 512; Số lượng tham số:512x62=31.744; Output: 62.

3. Cài cài đặt chương trình

3.1 Dữ liệu cho bài toán

Dữ liệu bài báo sử dụng được thu thập từ bộ cơ sở dữ liệu ký tự và biểu mẫu viết tay của Viện tiêu chuẩn và kỹ thuật quốc gia Hoa Kỳ (NIST). Tên bộ dữ liệu: EMIST ByClass; tổng số lượng mẫu: 814,255 mẫu; trong đó số mẫu huấn luyện: 697,932 mẫu (chiếm 85%); số mẫu kiểm thử: 116,323 mẫu (chiếm 15%); cho 62 ký tự ký tự viết tay của (*A-Z, a-z, 0-9).*

3.2 Các công cụ sử dụng trong nhận dạng

Trong bài báo, chúng tôi thiết kế hệ nhận dạng ký tự với phần huấn luyện (Hình 3), xử lý dữ liệu và ảnh trên ngôn ngữ Python. Để thiết kế giao diện, sử dụng ngôn ngữ C#.Net Framework của Microsoft; thư viện: Open CV, Keras, Tensorflow; Framework: .NET Framework (C#).

3.3 Cài đặt chương trình

* Khởi tạo các thông số để huấn luyện mạng:

Số lần học(Epochs): 10

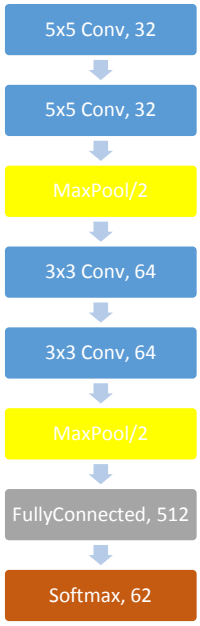
Số bó (Batch_size): 256 (trọng số được cập nhật lại sau mỗi bó)

* Huấn luyện mạng:

Môi trường huấn luyện là hệ điều hành Windows 10; ngôn ngữ Python phiên bản 3.7.5 cùng giao diện thiết kế dựa trên .NET Framework phiên bản 4.6.2;

Hàm mất mát: Cross Entropy;

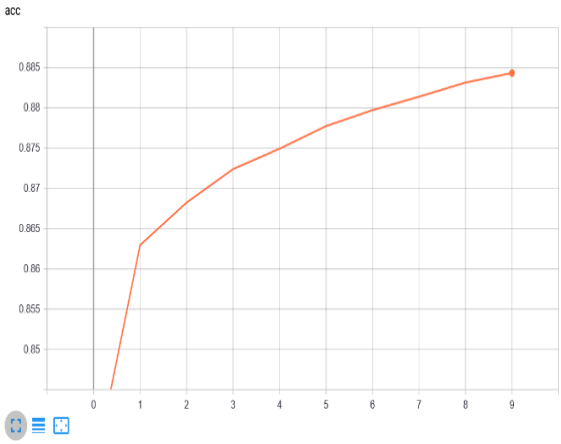
- Phần cứng: CPU Intel Core i5 – 9400, 8GB RAM, Card đồ hoạ Intel(R) UHD Graphics 630;
- Tốc độ huấn luyện: 3ms/mẫu;
- Thời gian huấn luyện: 6 giờ.



Hình 3: Mô hình huấn luyện

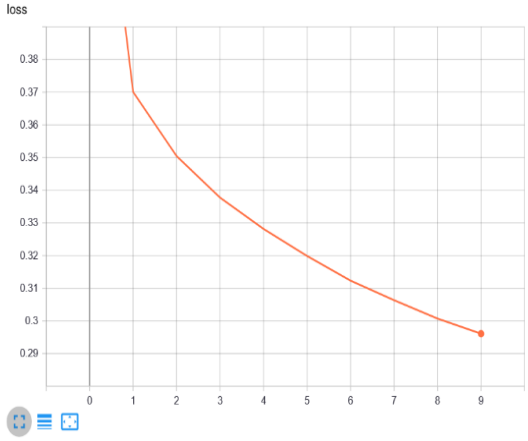
3.4 Kết quả nhận dạng

Sau khi huấn luyện dựa trên dữ liệu đã nêu, chúng ta được một cấu trúc có các trọng số đã gán giá trị cụ thể và hoàn toàn đủ thông số để tiến hành nhận dạng với độ chính xác trên dữ liệu kiểm thử là khoảng 90%.



Hình 4: Độ chính xác huấn luyện

Hàm mất mát (Loss Function) thu được xấp xỉ 0.3 (Hình 5).



Hình 5: Hàm mất mát

Tích hợp mô hình huấn luyện vào phần mềm nhận dạng ta nhận kết quả như hình 6:

Hình 6: Kết quả phần mềm nhận dạng

3.5 Đánh giá độ chính xác nhận dạng

Có rất nhiều cách đánh giá mô hình phân lớp. Tuỳ vào những bài toán khác nhau mà chúng ta sử dụng các phương pháp khác nhau. Các phương pháp thường được sử dụng là: Accuracy Score, Confusion Matrix [9], ROC Curve, Area Under the Curve, Precision and Recall, F1 score, Top R error [5, 9] v.v…Bài báo quan tâm chủ yếu tới các tiêu chí đánh giá như độ chính xác (Accuracy) theo Ma trận nhầm lẫn (Confusion Matrix)

Độ chính xác (Accuracy) trong bài toán nhận dạng (Hình 4) đơn giản là tính tỉ lệ giữa số mẫu được dự đoán đúng trên tổng số mẫu trong tập kiểm thử [5, 9].

$$A = \frac{T}{T+F} \qquad (3)$$

3.6. Hàm mất mát

Hàm mất mát (Loss Function): $L(\hat{y}, y)$ cho một số thực, không âm thể hiện sự chênh lệch

giữa đại lượng: $\hat{y}$ dự đoán và y thực tế.

$$L(\hat{y}, y) = |\hat{y} - y| \quad (4)$$

Hàm mất mát là hình thức buộc hệ thống điều chỉnh (hay phạt) mỗi lần dự đoán sai; số mức phạt tỉ lệ thuận với độ lớn của sai sót [6]. Trong học hay huấn luyện có giám sát, mục tiêu luôn giảm thiểu tổng mức sai số. Trong trường hợp lý tưởng, hàm mất mát trả về giá trị cực tiểu bằng 0. Một cách khác xã định hàm mất mát là lấy bình phương của sai số:

$$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2 \quad (5)$$

Với bài toán nhận dạng, ta có thể dùng chỉ tiêu bình phương lỗi (*MSE – Mean Square Error)* bằng cách tính tổng bình phương lỗi của $\hat{y}$ và y [7, 8] và là chỉ tiêu đánh giá độ chính xác của bài toán nhận dạng.

3. Kết luận và hướng phát triển

Mô hình mạng CNN với kiến trúc như trên có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến các nút trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ ron truyền thẳng, làm tăng khả năng xử lý và đạt tỷ lệ cao trong nhận dạng ký tự.

Độ chính xác của mô hình khá cao; tuy nhiên chất lượng của hệ thống còn bị ảnh hưởng bởi một số yếu tố khác nhau như: độ sáng ảnh, chất lượng ảnh, góc chụp v.v… Ứng dụng mới có khả năng phát hiện và nhận dạng các ký tự riêng lẻ. Trong tương lai, các tác giả sẽ sử dụng mạng đề xuất khu vực như Faster RCNN [6] để phát hiện ký tự; đồng thời sử dụng một bộ nhớ ngắn hạn [7] (LSTM – Long Short Term Memory) để xây dựng từ điển các từ nhằm nhận dạng các từ thay vì các ký tự riêng lẻ. Ngoài ra, chúng tôi cũng sẽ nghiên cứu ứng dụng nhận dạng ký tự tiếng Việt, kết quả nếu đạt được sẽ công bố ở các công trình sau bài báo này.

Tài liệu tham khảo

[1] Ross, Girshick (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation" (PDF). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE. doi:10.1109/CVPR.2014.81.

[2] Girschick, Ross (2015). "Fast R-CNN" (PDF). Proceedings of the IEEE International Conference on Computer Vision: 1440–1448. arXiv:1504.08083.

[3] Shaoqing, Ren (2015). "Faster R-CNN" (PDF). Advances in Neural Information Processing Systems. arXiv:1506.01497.

[4] Đoàn Hồng Quang, Lê Hồng Minh, Chu Anh Tuấn (2015), "Nhận dạng bàn tay bằng mạng nơ ron nhân tạo", Tuyển tập báo cáo diễn đàn "Đổi mới - Chìa khóa cho sự phát triển bền vững", Viện Ứng dụng Công nghệ, Bộ Khoa học và Công nghệ.

[5] Creus, Antonio (1994): "Accuracy (Trueness and Precision) of Measurement Methods and Results - Part 1: General Principles and Definitions.", p.1

[6] Nikulin, M. S. (2001) [1994], "Risk of a Statistical Procedure", in Hazewinkel, Michiel (ed.), Encyclopedia of Mathematics, Springer Science+Business Media B.V./Kluwer

[7] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long Short-Term Memory". Neural Computation

[8] https://vi.wikipedia.org/wiki/Nhan_dang_ky_tu_quang_hoc

[9] Nguyễn Quang Hoan, Lý Đông Hà, Ngô Xuân Trang, Lê Công Hiếu (2014), "Ứng dụng mạng nơron trong nhận dạng và dự báo", *Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*, ISSN 2354-0575, số 16/12-2017.