

NHẬN DẠNG KÝ TỰ BẰNG MẠNG NEURAL LAN TRUYỀN NGƯỢC

Character recognition by backpropagation neural network

Trần Văn Nam¹

Tóm tắt

Nhận dạng ký tự quang học (Optical Character Recognition, viết tắt là OCR), là chương trình được tạo ra để chuyển các hình ảnh của chữ viết tay hoặc chữ đánh máy (thường được quét bằng máy scanner) thành các văn bản tài liệu. Bài báo giới thiệu một phương pháp nhận dạng ký tự, đó là kỹ thuật mạng Neural lan truyền ngược. Kỹ thuật này hiện nay được ứng dụng rộng rãi giúp người dùng không phải mất thời gian và công sức cho việc nhập lại văn bản từ file ảnh chứa văn bản (đã được scan hoặc dùng các thiết bị khác chụp lại). Kết quả thực nghiệm trên 90 ký tự với hai loại font : Arial và Tahoma cho thấy phương pháp này đạt được kết quả nhận dạng chính xác đến 98.89 phần trăm.

Từ khóa: nhận dạng ký tự quang học, mạng neural nhân tạo, mạng truyền thẳng, mạng hồi qui, mạng tự tổ chức.

Abstract

Optical Character Recognition (Optical Character Recognition, abbreviated as OCR), a program is created to convert images of handwritten or typewritten text (usually scanned by a scanner) to text documents. This paper presents a method of character recognition, which is the technical back-propagation neural network. This technique is now widely applied to enable users not to spend time and effort to re-enter the text from the image file (scanned or use other capture devices). The experimental results on 90 characters with two types of fonts of Arial and Tahoma showed that this method gives the accurate recognition result to 98.89 percent.

Keys words: optical Character Recognition, Artificial Neural Networks, feed-forward, feedback, self-organizing.

1. Dẫn nhập

Mạng neural nhân tạo (Artificial Neural Networks: ANN) ra đời xuất phát từ ý tưởng mô phỏng hoạt động của bộ não con người. ANN là sự tái tạo bằng kỹ thuật những chức năng của hệ thần kinh con người với vô số các neural được liên kết truyền thông với nhau qua mạng. Giống như con người, ANN được học bởi kinh nghiệm, lưu những kinh nghiệm đó và sử dụng trong những tình huống phù hợp. Trong một vài năm trở lại đây, mạng neural đã được nhiều người quan tâm và áp dụng thành công trong nhiều lĩnh vực khác nhau. Chẳng hạn như ở các cơ quan tài chính, y tế, địa chất và vật lý. Thật vậy, bất kỳ ở đâu có vấn đề về dự báo, phân loại và điều khiển, mạng neural đều có thể ứng dụng được. Mạng neural dựa trên việc mô phỏng cấp thấp hệ thống neural sinh học. Trong tương lai với sự phát triển mô phỏng neural sinh học, chúng ta có thể có loại máy tính thông minh thật sự.

Trong phạm vi bài báo này, chúng tôi vận dụng lý thuyết mạng neural để thiết kế chương trình nhận dạng ký tự quang học.

2. Phương pháp huấn luyện mạng neural²

2.1. Phương pháp học

Mạng neural nhân tạo phỏng theo việc xử lý thông tin của bộ não người. Do vậy, đặc trưng cơ bản của mạng là có khả năng học, khả năng tái tạo các hình ảnh và dữ liệu khi đã học. Trong trạng thái học, thông tin được lan truyền theo hai chiều nhiều lần để học các trọng số. Có 3 kiểu học chính, mỗi kiểu học tương ứng với một nhiệm vụ học trừu tượng. Đó là học có giám sát (có mẫu), học không giám sát và học tăng cường. Thông thường, loại kiến trúc mạng nào cũng có thể dùng được cho các nhiệm vụ.

2.2. Học có giám sát

Một thành phần không thể thiếu của phương pháp này là sự có mặt của một người thầy (ở bên ngoài hệ thống). Người thầy này có kiến thức về môi trường thể hiện qua một tập hợp các cặp đầu vào - đầu ra đã được biết trước. Hệ thống học (ở đây là mạng neural) sẽ phải tìm cách thay đổi các tham số bên trong của mình (các trọng số và các

² Trần, Hùng Cường. 2013. Nhận dạng ký tự quang học. Trường Đại học Công nghiệp Hà Nội.

¹ Thạc sĩ, Trường Đại học Trà Vinh

ngưỡng) để tạo nên một ánh xạ có khả năng ánh xạ các đầu vào thành các đầu ra mong muốn. Sự thay đổi này được tiến hành nhờ việc so sánh giữa đầu ra thực sự và đầu ra mong muốn.

2.3. Học không giám sát

Trong học không có giám sát, ta được cho trước một số dữ liệu x và hàm chi phí cần được cực tiểu hóa có thể là một hàm bất kỳ của dữ liệu x và đầu ra của mạng, f – hàm chi phí được quyết định bởi phát biểu của bài toán. Phần lớn các ứng dụng nằm trong vùng của các bài toán ước lượng như mô hình hóa thống kê, nén, lọc, phân cụm.

2.4. Học tăng cường

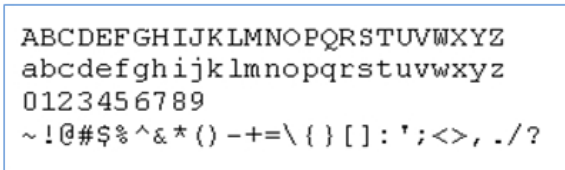
Dữ liệu x thường không được tạo trước mà được tạo ra trong quá trình một agent tương tác với môi trường. Tại mỗi thời điểm t , agent thực hiện hành động y_t và môi trường tạo một quan sát x_{t+1} với một chi phí tức thời C_t , theo một quy trình động nào đó (thường là không được biết). Mục tiêu là một sách lược lựa chọn hành động để cực tiểu hóa một chi phí dài hạn nào đó, nghĩa là chi phí tích lũy mong đợi. Quy trình hoạt động của môi trường và chi phí dài hạn cho mỗi sách lược thường không được biết, nhưng có thể ước lượng được. Mạng neural nhân tạo thường được dùng trong học tăng cường như một phần của thuật toán toàn cục. Các bài toán thường được giải quyết bằng học tăng cường là các bài toán điều khiển, trò chơi và các nhiệm vụ quyết định tuần tự (sequential decision making) khác.

3. Phương pháp và thuật toán nhận dạng ký tự

Ở đây, chúng tôi sử dụng mạng feed-forward và thuật toán lan truyền ngược sai số Back Propagation để xử lý bài toán.

3.1. Cơ sở dữ liệu

Cơ sở dữ liệu cho bài toán nhận dạng ký tự quang gồm 90 ký tự theo chữ cái Latin với hai loại font: *Times New Roman* và *Arial*, trong bảng mã Unicode tương ứng.



Hình 1. Một ví dụ về mẫu các ký tự trong nhận dạng ký tự quang học

3.2. Phương pháp nhận dạng

Phương pháp nhận dạng ký tự quang bằng mạng neural bao gồm các bước được mô tả như sau:

- Thu nhận ảnh
- Tiến hành phân tích ảnh để tìm ký tự
- Tiền xử lý ký tự
- Mạng Neural nhận dạng ký tự
- Hậu xử lý dữ liệu

3.2.1. Thu nhận ảnh

Input văn bản, tài liệu có thể được thu nhận bằng máy quét scanner, webcam, hoặc các thiết bị thu nhận ảnh thông dụng khác.

3.2.2. Tiến hành phân tích ảnh để tìm ký tự

Việc phân tích ảnh để tìm ký tự bao gồm các bước sau:

- Đầu tiên, tiến hành tách dòng ký tự ra khỏi ảnh ký tự
- Thứ hai, tách từ riêng biệt ra khỏi dòng ký tự
- Cuối cùng, tách riêng từng ký tự ra khỏi từ

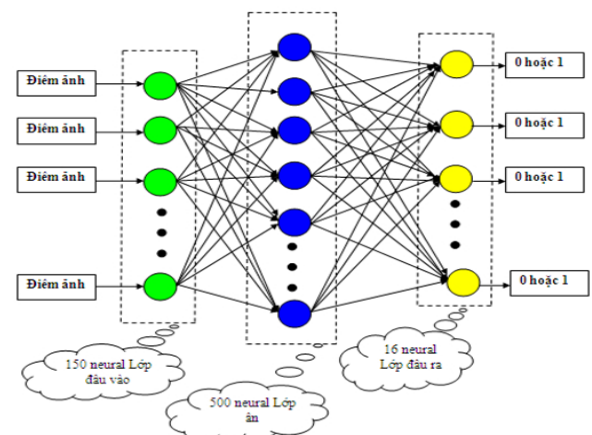
3.2.3. Tiền xử lý ký tự

Quá trình tiền xử lý ký tự giải quyết vấn đề ánh xạ giá trị pixel ảnh ký tự vào ma trận 10x15 và tuyến tính hóa ma trận thành 150 giá trị đưa vào 150 Neural ở lớp vào của mạng.

3.2.4. Mạng Neural nhận dạng ký tự

Hiện nay, các loại mạng Neural thông dụng gồm có: mạng truyền thẳng (feed-forward), mạng hồi qui (feedback), mạng tự tổ chức (self-organizing). Mạng truyền thẳng feed-forward bao gồm nhiều lớp các đơn vị xử lý phi tuyến (non-linear processing unit).

Trong phương pháp này, chúng tôi thiết kế chương trình nhận dạng sử dụng mạng MLP có 3 lớp: lớp vào có 150 nút tương ứng với 250 phần tử vectơ ma trận pixel, lớp ẩn có 500 Neural và lớp ra có 16 Neural với 16 bit.



Hình 2. Quá trình nhận dạng ký tự

3.2.5. Hậu xử lý dữ liệu

Đây là giai đoạn sau cùng, giai đoạn này làm nhiệm vụ chuyển đổi giá trị sang dạng ký tự tương ứng và sắp xếp lại các ký tự dưới dạng text theo dạng văn bản ban đầu.

4. Thực nghiệm và kết quả

4.1. Tách dòng

Thuật toán:

Bước 1: Xác định giới hạn dưới của dòng:

Bước 2: Bắt đầu duyệt từ giới hạn trên (đỉnh) vừa tìm thấy của dòng (0, top_line).

Bước 3: Tương tự như xác định giới hạn trên, chúng ta duyệt hết chiều rộng của ảnh trên cùng một giá trị y.

– Nếu duyệt hết dòng mà không tìm thấy ký tự pixel đen nào thì ghi nhận y-1 là giới hạn dưới của dòng (bottom_line). Dừng duyệt. Tăng số dòng lên (lines++).

– Nếu chưa tìm thấy bottom_line, tiếp tục duyệt đến dòng tiếp theo (tăng y, reset x=0).

Bắt đầu từ giới hạn dưới y (bottom_line) vừa tìm thấy sau cùng, lặp lại các bước a,b để xác định các giới hạn của các dòng tiếp theo, cho đến khi duyệt hết chiều cao của ảnh thì dừng, quá trình xác định dòng ký tự hoàn tất.

4.2. Tách ký tự

Thuật toán:

Bước 1: Bắt đầu từ ký tự đầu tiên của hàng trên cùng với giá trị x đầu tiên.

Bước 2: Quét hết chiều rộng với một giá trị y

– Nếu phát hiện điểm đen đánh dấu y như là đỉnh của hàng đầu tiên

– Nếu không xét điểm tiếp theo

Bước 3: Bắt đầu từ giới hạn trên của ký tự phát hiện được và giá trị x đầu tiên. (0, giới hạn trên ký tự)

Bước 4: Quét đến giới hạn dưới của dòng, giữ nguyên x

– Nếu phát hiện điểm đen đánh dấu x là phía trái của ký tự

– Nếu không xét điểm tiếp theo

– Nếu không thấy điểm đen nào tăng x và khởi động lại y để xét đường thẳng đứng tiếp theo

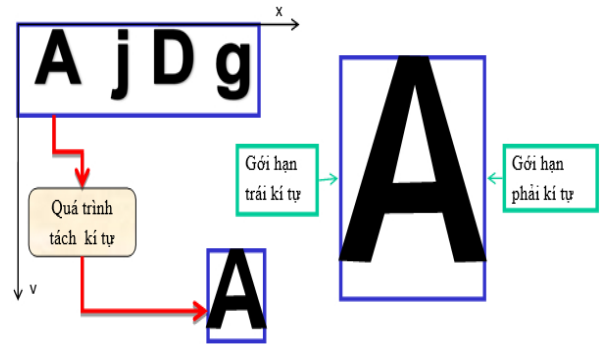
Bước 5: Bắt đầu từ phía trái của ký tự tìm thấy

và đỉnh của dòng hiện thời, (giới hạn trái ký tự, giới hạn trên dòng)

Bước 6: Quét hết chiều rộng của ảnh trên cùng một giá trị x

– Nếu không có điểm đen nào thì đánh dấu x-1 là bên phải của ký tự

Nếu phát hiện điểm đen tăng x và khởi động lại y để xét đường thẳng đứng tiếp theo



Hình 3. Quá trình tách ký tự

4.3. Tìm giới hạn ký tự

Thuật toán:

Bước 1: Bắt đầu từ đỉnh của dòng hiện thời và bên trái của ký tự

Bước 2: Quét đến bên phải của ký tự cùng một giá trị y

– Nếu phát hiện điểm đen thì đánh dấu y và thay đổi lại giới hạn trên

– Nếu không xét điểm tiếp theo

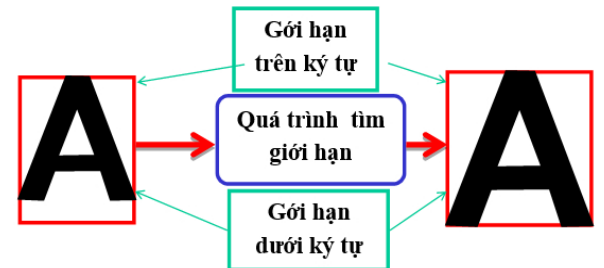
– Nếu không tìm thấy điểm đen nào tăng y và khởi động lại x, xét đường thẳng ngang tiếp theo

Bước 3: Bắt đầu từ giới hạn dưới của dòng và bên trái của ký tự

Bước 4: Quét tới bên phải của ký tự trên một giá trị y

– Nếu phát hiện điểm đen, đánh dấu y là giới hạn dưới của ký tự

– Nếu không phát hiện điểm đen giảm y và khởi động lại x xét đường thẳng ngang tiếp theo



Hình 4. Quá trình tìm giới hạn ký tự

4.4. Ảnh xạ vào ma trận

Thuật toán:

Bước 1: Đối với chiều rộng:

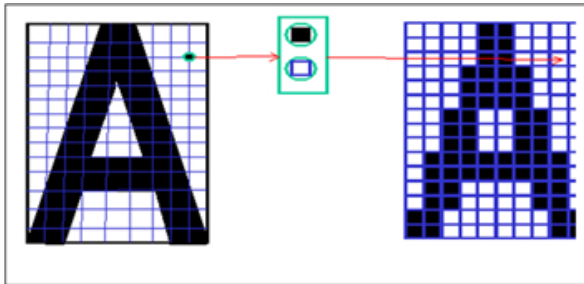
- Khởi tạo với 19 phần tử tương ứng
- Ảnh xạ điểm đầu (0,y) và điểm cuối (C_rong,y) của ảnh kí tự tương ứng với giá trị đầu (0,y) và giá trị cuối (10,y) của ma trận

- Chia nhỏ chiều rộng thành 19 giá trị tương ứng

Bước 2: Đối với chiều cao:

- Khởi tạo với 29 phần tử tương ứng
- Ảnh xạ điểm đầu (x,0) và điểm cuối (x,C_cao) của ảnh kí tự tương ứng với giá trị đầu (x,0) và giá trị cuối (x,29) của ma trận

- Chia nhỏ chiều cao thành 19 giá trị tương ứng



Hình 5. Quá trình chia lưới kí tự

Để đưa giá trị vào mạng neural, chúng ta cần chuyển ma trận điểm ảnh sang ma trận giá trị.

Thuật toán:

Bước 1: Bắt đầu từ phần tử (0,0)

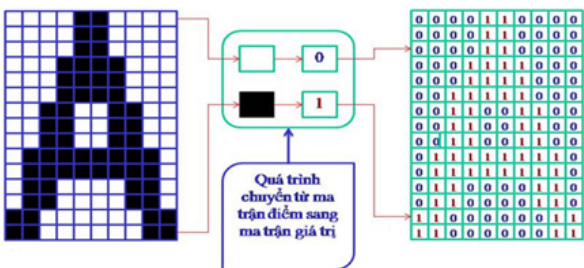
Bước 2: Tăng x giữ nguyên giá trị y cho tới khi bằng chiều rộng của ma trận

- Ảnh xạ mỗi phần tử tới một phần tử của mảng tuyến tính

- + Nếu là điểm đen thì nhận giá trị bằng 1
- + Ngược lại nhận giá trị bằng 0

- Nếu x = chiều rộng thì khởi động lại x và tăng y

Lặp lại cho tới khi (x,y)=(C_Rong, C_Cao)



Hình 6. Quá trình ảnh xạ từ ma trận điểm sang ma trận giá trị

4.5. Huấn luyện mạng neural

Thuật toán:

Bước 1: Xây dựng mạng tương ứng với mô hình tham số

Bước 2: Khởi tạo giá trị trọng số với giá trị ngẫu nhiên. Nạp file huấn luyện (cả ảnh đầu vào và ảnh đầu ra mong muốn)

Bước 3: Phân tích ảnh và ảnh xạ tất cả kí tự tìm thấy vào các mảng một chiều

Bước 4: Đọc giá trị đầu ra mong muốn từ file và chuyển đổi từng kí tự tới giá trị nhị phân Unicode và lưu trữ riêng biệt

Bước 5: Với mỗi kí tự:

- Tính toán giá trị đầu ra của mạng Feed Forward

- So sánh với giá trị đầu ra mong muốn tương ứng với từng kí tự và tính toán lỗi

- Truyền ngược giá trị từ đầu và với mỗi liên kết điều chỉnh trọng số liên kết

Bước 6: Chuyển sang kí tự tiếp theo và lặp lại “6” cho tới khi hết các kí tự

Bước 7: Tính toán trung bình lỗi cho tất cả các kí tự

Bước 8: Lặp lại từ bước 6 đến 8 cho tới khi đạt số đưa vào của số lần lặp tối đa

- Với phạm vi lỗi đạt đến ngưỡng. Nếu như vậy thì bỏ lặp lại

- Ngược lại tiếp tục lặp lại

4.6. Nhận dạng ảnh kí tự

Thuật toán:

Bước 1: Nạp file ảnh

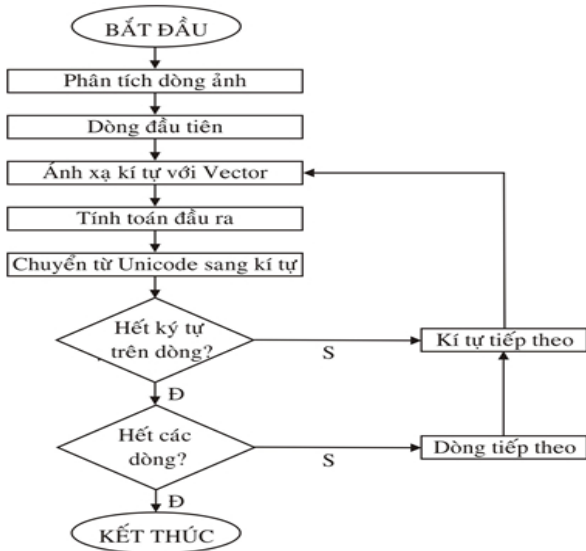
Bước 2: Phân tích ảnh cho các dòng kí tự

Bước 3: Với mỗi dòng tách các kí tự liên tiếp

- Phân tích và xử lý ảnh kí tự cho việc ảnh xạ vào một vector đầu vào

- Đưa giá trị vector đầu vào cho mạng neural và tính toán giá trị đầu ra

- Chuyển đổi mã Unicode đầu ra từ nhị phân tới kí tự tương ứng và trả ra dưới dạng textbox



Hình 7. Quá trình nhận dạng ký tự

4.7. Kết quả

Thử nghiệm việc huấn luyện mạng với tập mẫu của 2 loại font: Latin Arial, Latin Times Roman.

4.7.1. Kết quả khi thay đổi số lần lặp (Epoch)

Số ký tự = 90, tốc độ học learning_rate = 150, hệ số góc hàm sigmoid $\alpha = 0.014$

| Font | Epoch=300 | | Epoch=500 | | Epoch=700 | |
|-------------------|--------------|---------|--------------|---------|--------------|---------|
| | Số ký tự sai | % Error | Số ký tự sai | % Error | Số ký tự sai | % Error |
| Latin Arial | 3 | 3.33 | 2 | 2.22 | 1 | 1.11 |
| Latin Times Roman | 0 | 0 | 0 | 0 | 1 | 1.11 |

4.7.2. Kết quả khi thay đổi tham số learning_rate

Số ký tự = 90, số Epoch = 150, hệ số góc $\alpha = 0.014$

| Font | 60 | | 100 | | 180 | |
|-------------------|--------------|---------|--------------|---------|--------------|---------|
| | Số ký tự sai | % Error | Số ký tự sai | % Error | Số ký tự sai | % Error |
| Latin Arial | 80 | 88.88 | 19 | 21 | 2 | 2.22 |
| Latin Times Roman | 76 | 84.44 | 14 | 15.56 | 0 | 0 |

5. Kết luận

Chương trình thực nghiệm đã được huấn luyện và nhận dạng hai loại font: Arial và Times Roman với nhiều kích thước khác nhau đã đạt được kết quả 98.89%, nhưng còn một số tồn tại cần được phát triển để đạt kết quả cao hơn.

Đối với quá trình huấn luyện, chúng ta cần chú ý nhiều và font Arial như đã nói ở trên chữ “I-Hoa” mã 49h và chữ “l-Thường” mã 6Ch khi tách ký tự, chia lưới và đưa kết quả vào mạng sẽ làm cho mạng không phân biệt được hai ký tự này, dẫn đến dễ nhận dạng sai. Phương pháp này cần tăng số lần lặp cho quá trình huấn luyện.

Ngoài ra, còn một số trường hợp ảnh của hai ký tự nằm chéo nhau như một số trường hợp sau:

AV
fa
LT LV

Nó sẽ dẫn đến quá trình tách ký tự bị dính do đó nhận dạng sai. Đối với trường hợp này, chúng ta cần phát triển phương pháp tách để tách ký tự.

Trong quá trình thực nghiệm nhận dạng ký tự, chúng ta thấy những ký tự sai là do quá trình huấn luyện mạng chưa học được nên những ký tự này trong ảnh nhận dạng sẽ bị nhận dạng sai, chỉ có một số ít là do trong quá trình tách ký tự.

Tài liệu tham khảo

Jain, A. 2000. *Personal Identification Based on Handwriting*. IIT Kapur, April.

Holst, A. 1997. “*The Use of Bayesian Neural Network Model for Classification Tasks*. Department of Numerical Analysis and Computing Science. Sweden.

Jain, A. K. 1998. *Texture Analysis*. East Lansing, MI 48824-1027 USA.

Cha Sung-Hynk. 2001. *Use of Distance Measures in Handwriting Analysis*. Buffalo UB: New York.

Trần, Hùng Cường. 2013. *Nhận dạng ký tự quang học*. Trường Đại học Công Nghiệp Hà Nội.