

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP #01
CÁC HỆ CƠ SỞ TRI THỨC
MẠNG ĐA LỚP TRUYỀN THẮNG
TRONG XỬ LÝ BÀI TOÁN NHẬN
DẠNG KÝ SỐ VIẾT TAY



08/04/2017

NGƯỜI THỰC HIỆN

1412050

NGUYỄN XUÂN BÌNH

1412547

LÊ QUANG TIẾN

1412630

ĐỖ KHÁNH LONG TƯỜNG

I. Giới thiệu bài toán nhận dạng chữ số viết tay:

- Nhận dạng chữ số viết tay là cần thiết và được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng các chữ số trên chi phiếu ngân hàng, mã số trên bì thư của dịch vụ bưu chính, hay các chữ số trên các biểu mẫu nói chung. Vấn đề nhận dạng chữ viết tay nói chung và nhận dạng chữ số viết tay nói riêng là một thách thức lớn đối với các nhà nghiên cứu. Bài toán lớn luôn đặt ra phía trước vì sự phức tạp của việc nhận dạng chữ viết phụ thuộc nhiều vào phong cách viết và cách thể hiện ngôn ngữ của người viết. Chúng ta không thể luôn luôn viết một ký tự chính xác theo cùng một cách. Do vậy, xây dựng hệ thống nhận dạng chữ viết có thể nhận dạng bất cứ ký tự nào một cách đáng tin cậy trong tất cả các ứng dụng là điều không dễ dàng.
- Hệ thống nhận dạng thường bao gồm hai bước: rút trích đặc trưng từ ảnh và học tự động từ các đặc trưng để có thể nhận dạng ký tự. Hiệu quả của hệ thống nhận dạng phụ thuộc vào các phương pháp sử dụng ở hai bước trên.
- Tập dữ liệu thực MNIST là tập dữ liệu sẽ được nhóm sử dụng để thử nghiệm và kiểm tra độ chính xác. Phương pháp học tự động mà nhóm sẽ chọn là mạng Nơ ron truyền thẳng đa lớp, phù hợp với việc học có giám sát dữ liệu trên tập MNIST.

II. Mô hình nhận dạng chữ số viết tay:

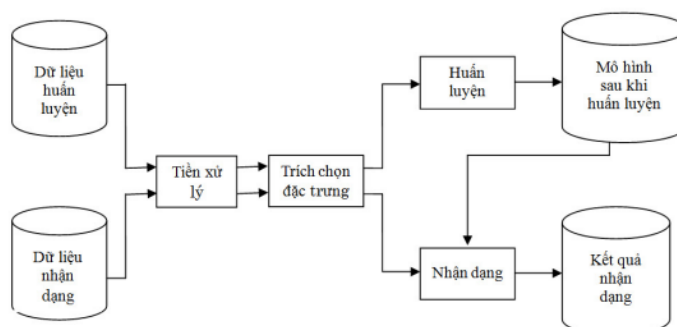
- Xử lý theo hai bước chính

+ *Bước 1: Xây dựng mô hình huấn luyện.*

Tập dữ liệu huấn luyện sau khi qua các khâu tiền xử lý và trích chọn đặc trưng sẽ được đưa vào hệ thống học (hệ thống huấn luyện). Quá trình huấn luyện tiêu tốn khá nhiều thời gian, tốc độ huấn luyện nhanh hay chậm tùy thuộc vào từng thuật toán huấn luyện cũng như số lượng mẫu tham gia huấn luyện.

+ *Bước 2: Phân lớp nhận dạng.*

Dựa vào giá trị các tham số của hàm quyết định thu được ở Bước 1, một mẫu mới x sau khi đã qua các khâu tiền xử lý và trích chọn đặc trưng sẽ được đưa vào tính toán thông qua hàm quyết định để xác định lớp của mẫu x.



Xét trong mô hình:

- Dữ liệu huấn luyện và dữ liệu nhận dạng được rút ra từ tập dữ liệu thực MNIST theo một tỉ lệ phù hợp.
- Bước tiền xử lý: rút trích các đặc trưng bằng nhiều phương pháp khác nhau.
- Bước trích chọn đặc trưng: thông qua tiền xử lý, áp dụng thuật toán lọc để giữ lại các đặc trưng quan trọng, có tính bất biến cao.
- Đặc trưng trích chọn được của tập huấn luyện sẽ được đưa vào thuật toán huấn luyện mạng nơron truyền thẳng đa lớp để học ra hàm quyết định.
- Đặc trưng trích chọn được của tập nhận dạng (tập kiểm thử) sẽ được dùng để kiểm tra độ chính xác của hàm quyết định học được.

III. Đặc trưng rút trích:

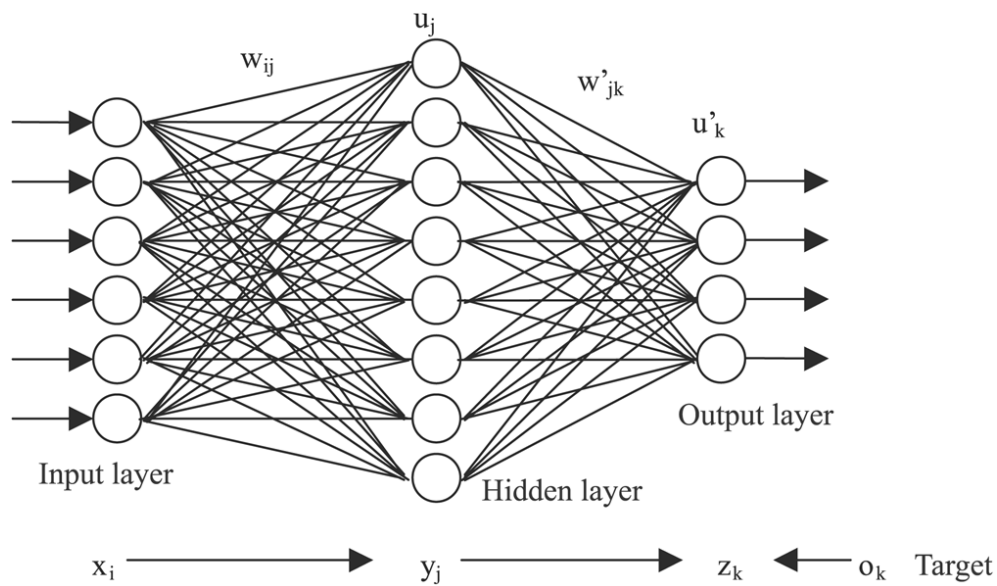
- Trong hệ thống nhận dạng, bước rút trích đặc trưng là rất quan trọng, ảnh hưởng lớn đến hiệu quả của việc huấn luyện mô hình học tự động. Các đặc trưng rút trích từ ảnh phải đạt được mục đích quan trọng là dựa trên các đặc trưng đó mà giải thuật học có thể phân biệt tốt nhất một ký tự số này với một ký tự số khác.
- Vào thập niên chín mươi, người ta thường sử dụng các đặc trưng cục bộ kết hợp với nhau như giá trị mức xám của từng điểm ảnh, đường biên, cạnh, độ dày, đến các phương pháp xử lý đặc biệt khác như lấy mẫu, dao động các điểm ảnh, biến đổi ảnh.
- Trong mấy năm trở lại đây, một số đặc trưng tiên tiến được phát triển, trong đó có hai kiểu đặc trưng rất hiệu quả là đặc trưng cục bộ không đổi (SIFT) và đặc trưng toàn cục GIST.
 - + Các véc tơ mô tả SIFT rút trích từ ảnh có tính chất quan trọng là: không bị thay đổi trước những biến đổi tỉ lệ, tịnh tiến, phép quay, không bị thay đổi một phần đối với phép biến đổi hình học affine (thay đổi góc nhìn) và khả năng chịu đựng với những thay đổi về độ sáng, sự che khuất hay nhiễu. Tuy nhiên bất biến với phép quay của đặc trưng SIFT lại gây ra sự bất lợi cho nhận dạng ký tự số (số 9 và 6 có thể như nhau). Hơn nữa, phương pháp SIFT cung cấp rất nghèo nàn về số lượng các đặc trưng từ ảnh ký tự số (ít hơn 10 đặc trưng).
 - + Trong khi đó sử dụng đặc trưng toàn cục GIST thì không gặp khó khăn như SIFT. Chính vì lý do đó, ta sử dụng đặc trưng toàn cục GIST để giải quyết vấn đề nhận dạng số viết tay. Phương pháp GIST rút trích từ ảnh tập hợp các đặc trưng quan trọng như tính tự nhiên, mở rộng, độ nhám, độ chắc chắn, cho phép trình bày cấu trúc không gian của một cảnh. Để tính toán đặc trưng mô tả GIST, ảnh đầu vào được đưa về dạng vuông, chia thành lưới 4 x 4, các tổ

chức đồ theo hướng tương ứng được trích ra. Nguyên lý trích đặc trưng dựa vào phép biến đổi Gabor theo các hướng và tần số khác nhau. Mỗi ảnh ký tự số được rút trích các đặc trưng GIST (véc tơ có 960 chiều). Sau bước trích đặc trưng này, tập dữ liệu ảnh đưa về dạng bảng hay ma trận mà ở đó mỗi ảnh là một dòng có 960 cột (chiều), mỗi ký tự số được gán nhãn (lớp tương ứng là 0, 1, ..., 9).

IV. Mạng neuron nhân tạo.

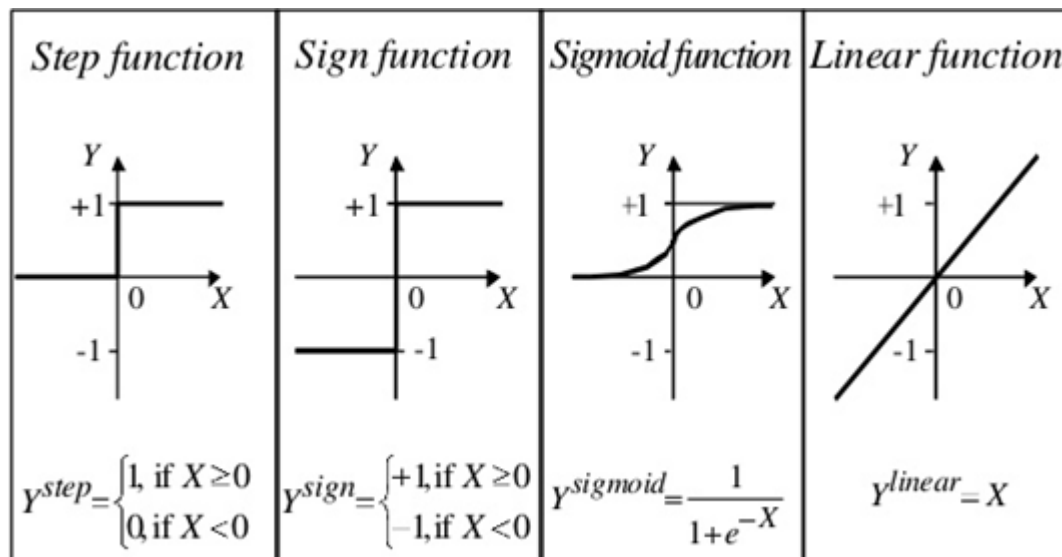
1. Khái niệm:

- Mạng Neuron nhân tạo là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh sinh vật, bao gồm số lượng lớn các Neuron được gắn kết để xử lý thông tin. Mạng neuron giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).
- Kiến trúc tổng quát:



Một ví dụ về mạng Neural

- + Input: Các tham số đầu vào
- + Output: Kết quả đầu ra
- + Trọng số liên kết: thể hiện độ mạnh của dữ liệu dùng điều chỉnh tham số đầu vào mỗi nơon
- + Hàm tổng (hàm kết hợp tuyến tính): $y = f(x)$ tính tổng trọng số của nơon, cho biết độ mạnh, khả năng kích hoạt của nơon đó
- + Hàm chuyển đổi (hay hàm kích hoạt): cho phép xác định output của 1 nơon có được tham gia vào layer tiếp theo không. Một số hàm chuyển đổi: sigmoid, step (threshold), linear,...
- + Lớp ẩn: 1 mạng nơon có thể có 1, 2, 3,... nhiều lớp, thông qua các lớp ẩn, ta có thể thu được kết quả output chính xác hơn.
- Dạng hàm kích hoạt: các dạng hàm kích hoạt thường dùng:



2. Perceptron – Một mô hình mạng nơon đơn giản nhất

a. Giới thiệu:

- Perceptron là một thuật toán Classification cho trường hợp đơn giản nhất: chỉ có hai class (lớp) (bài toán với chỉ hai class được gọi là binary classification) và cũng chỉ hoạt động được trong một trường hợp rất cụ thể. Tuy nhiên, nó là nền tảng cho một mảng lớn quan trọng của Machine Learning là Neural Networks và sau này là Deep Learning.

b. Bài toán:

- Bài toán Perceptron được phát biểu như sau: Cho hai class được gán nhãn, hãy tìm một đường thẳng sao cho toàn bộ các điểm thuộc class 1 nằm về

1 phía, toàn bộ các điểm thuộc class 2 nằm về phía còn lại của đường phẳng đó. Với giả định rằng tồn tại một đường thẳng như thế.

- Nếu tồn tại một đường phẳng phân chia hai class thì ta gọi hai class đó là linearly separable.

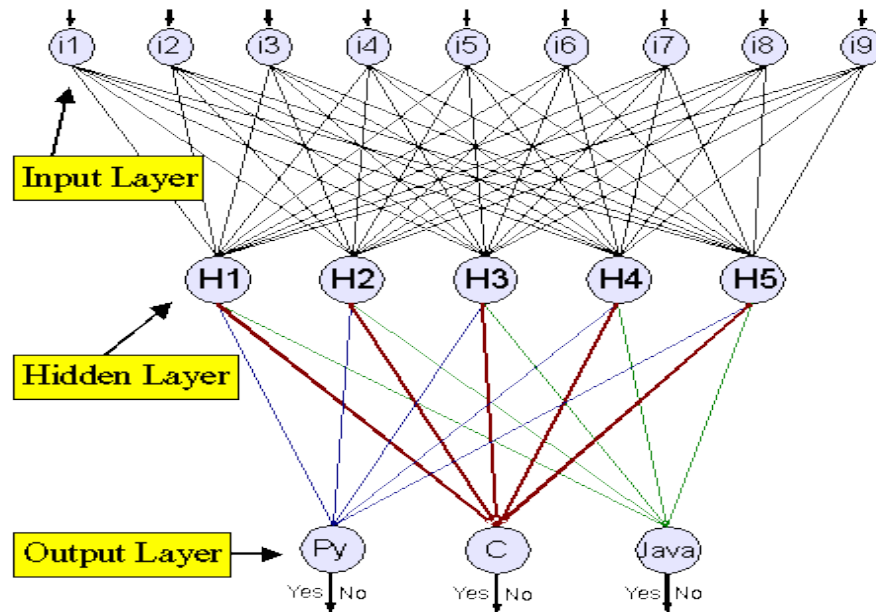
c. Thuật toán học Perceptron:

- Cho tập huấn luyện $D = \{(x_1, d_1), (x_2, d_2), \dots, (x_n, d_n)\}$ với x_i là vector thể hiện đối tượng i , d_i là giá trị thể hiện nhóm tương ứng của x_i . Sau khi có tập huấn luyện, ta bắt đầu khởi tạo giá trị cho vector w để tạo nên một hàm $f: f(x) = w^T x = w_1.x_1 + w_2.x_2 + \dots + w_n.x_n$. Cách thức khởi tạo có thể là ngẫu nhiên hoặc ràng buộc tùy vào người cài đặt. Tuy rằng nếu cách thức khởi tạo tương đối tốt thì quá trình học sẽ nhanh hơn nhưng việc tìm ra một cách thức khởi tạo tốt là khá mơ hồ nên đa phần chúng ta sẽ khởi tạo ngẫu nhiên.
- Sau khi đã khởi tạo hàm và có tập huấn luyện, chúng ta sẽ bắt đầu huấn luyện cho máy tính học. Tính các giá trị $f(x)$ với mỗi x_i trong tập huấn luyện (i chạy từ 1 đến n). Kết quả y_i phân nhóm dựa vào hàm $f: f(x)$ sẽ được đem so sánh với giá trị thực d_i . Điều chắc chắn là sẽ có sự sai biệt giữa y_i và d_i vì hàm $f(x)$ khởi tạo ban đầu chưa phải là kết quả mong muốn. Để có được hàm $f(x)$ đạt yêu cầu, tiến hành cập nhật giá trị cho vector w như sau: chọn ra một trong các vector x_i có y_i sai khác d_i và cập nhật $w = w + (d_i - y_i) \cdot x_i$. Bước cập nhật này chính là bước học từ tập huấn luyện.
- Sau khi có vector w mới, chúng ta lại tiến hành lặp lại các bước tính toán và cập nhật cho đến khi trung bình các độ sai lệch nhỏ hơn một ngưỡng nào đó cho trước (threshold). Trực quan hóa lên, chúng ta có thể hiểu rằng đường thẳng phân cắt mặt phẳng sẽ được điều chỉnh lên xuống sao cho phân chia hai nhóm riêng biệt một cách chính xác nhất. Chúng ta chỉ có thể tạo ra một hàm f sao cho gần nhất với đáp án thật sự (nghĩa là rất ít các điểm sai lệch) vì việc tạo ra một hàm f thật sự hoàn hảo tương đối khó khăn. Nếu tập huấn luyện có càng nhiều mẫu để huấn luyện với sự phân phối đều (có đối tượng thuộc cả hai nhóm tương đối đều) thì hàm f thu được sẽ càng tốt.
- Một lưu ý rằng trong mô tả thuật toán trên, ngầm định là hàm f sẽ đi qua gốc tọa độ. Điều này đôi khi sẽ không đúng cho nhiều bài toán và có thể điều chỉnh bằng cách thêm hằng số vào biểu thức: $f: f(x) = c + w \cdot x$ (trong đó c là hằng số thêm vào, trường hợp $c = 0$ sẽ giống như mô tả ban đầu). Các bước còn lại đều không thay đổi.

V. Mạng nơ-ron truyền thẳng đa lớp:

1. Giới thiệu:

- Là mạng nơ-ron có từ hai lớp ẩn trở lên. Cụ thể: 1 lớp nhập, ít nhất 1 lớp giữa, 1 lớp xuất:



- + Lớp nhập: nhận tín hiệu đầu vào và tái phân phối đến tất cả các nơ-ron trong lớp ẩn.
- + Lớp giữa (các lớp ẩn): giúp xác định, rút trích các đặc trưng, các nơ-ron biểu diễn các đặc trưng ẩn.
- + Lớp xuất: nhận tín hiệu xuất (các đặc trưng) từ lớp ẩn và xác định mẫu đầu ra.
- + Các nơ-ron trong lớp ẩn không thể được quan sát dựa trên đầu vào/ đầu ra của mạng. Vì thế không có cách nào biết rõ đầu ra mong muốn của lớp ẩn là gì.

2. Các loại học mạng nơ-ron:

- + 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.

+ 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 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.

+ 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.

+ 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 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. Thuật toán huấn luyện mạng – thuật toán lan truyền ngược:

- Bước 1: Tạo mạng truyền thẳng có n_{in} Noron đầu vào, n_{hidden} Noron trên mỗi lớp ẩn và n_{out} lớp ẩn trong mạng, với n_{out} đầu ra.
- Bước 2: Khởi tạo bộ trọng cho mạng với giá trị nhỏ
- Bước 3: Trong khi <điều kiện kết thúc chưa thỏa> làm:

Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:

- + Xét lớp nhập (*): truyền x qua mạng, tại mỗi lớp xác định đầu ra của mỗi Noron. Quá trình này được thực hiện đến lớp xuất dựa theo cấu trúc mạng cụ thể.

- + Xét lớp xuất: đối với đầu ra ok của Noron k trong lớp xuất K, xác định sai số:
- + Chuyển sang lớp ẩn L kế nó, đặt $L = K - 1$
 - Xét các lớp ẩn (**): với mỗi Noron l trên lớp ẩn thứ L, xác định sai số:
- + Cập nhật lại trọng số có trong mạng, w_{ji}
 - $w_{ji} = w_{ji} + \Delta w_{ji}$ với $\Delta w_{ji} = \eta \sigma_j o_{ji}$
- + Nếu ($L > 1$) thì: chuyển sang lớp ẩn trên nó : $L = L - 1$ và quay lại bước (**)
- + Ngược lại: chọn cặp (x, t) mới trong không gian mẫu học, quay lại bước (*).

VI. Áp dụng mạng nơron truyền thẳng đa lớp vào nhận dạng chữ số viết tay:

1. Bộ dữ liệu ký số viết tay MNIST:

- Bộ dữ liệu được sử dụng cho bài toán “Sử dụng mô hình mạng đa lớp để phân lớp ảnh chữ số viết tay” là MNIST. Mỗi mẫu trong bộ dữ liệu gồm 2 phần chính:
 - + Phần input: ảnh chữ số viết tay có kích thước 28x28 ở dạng grayscale (như vậy, mỗi vector input sẽ có $28 \times 28 = 784$ chiều).
 - + Phần correct output/label: các số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 tương ứng với từng ảnh. Như vậy, sẽ có tất cả 10 lớp.
- Do ảnh grayscale, mỗi pixel nhận giá trị thuộc khoảng số nguyên từ 0 đến 255. Để tăng tính đúng đắn cho bài toán cũng như tránh biến đổi ra các số quá lớn thì ta scale về khoảng [0, 1] bằng cách chia cho 255.
- Dữ liệu thực nghiệm chúng ta sử dụng được chia làm 3 tập dữ liệu:
 - + Tập training: 60,000 mẫu.
 - + Tập validation: 6,000 mẫu.
 - + Tập test: 10,000 mẫu.

2. Giới thiệu về thư viện Tensorflow, Keras và cài đặt mô hình thực nghiệm.

a. Thư viện hỗ trợ xây dựng mạng neural đa lớp:

- Tensorflow là một thư viện mã nguồn mở dùng cho các tác vụ liên quan đến học máy. Thư viện này được phát triển bởi Google nhằm mục đích xây dựng, huấn luyện mạng neural để phát hiện, nhận dạng và cũng như thống kê kết quả thực nghiệm cho học máy. Hiện tại, Tensorflow được sử dụng cho cả mục đích học tập, nghiên cứu lẫn thương mại. Tensorflow có thể chạy trên CPU và GPU, trên hệ điều hành khác nhau như Linux, Mac OS,... trên nhiều nền tảng di động như Android, iOS. Thư viện thuật toán này xử lý theo mô hình “data flow”, hỗ trợ nhiều thao tác để huấn luyện mạng neural.

- Keras là một thư viện mã nguồn mở nhỏ gọn về mạng neural được phát triển trên Python. Thư viện này thực thi dựa trên nền tảng của thư viện Tensorflow hoặc Theano nhằm mục đích tiếp cận nhanh hơn, dễ dàng hơn với mô hình huấn luyện mạng neural.
- + Modularity: Mỗi một mô hình có thể được xem xét như là một chuỗi các thao tác rời rạc có liên quan đến nhau. Mỗi một thao tác, mỗi bước thực thi, mỗi thành phần của mô hình có thể được rời rạc hóa và có thể được kết hợp tùy ý theo nhiều cách khác nhau.
- + Minimalism: Thư viện cung cấp đầy đủ các kết quả đạt được, không gây lãng phí và tối đa hóa việc đọc hiểu.
- + Extensibility: Các thành phần, các module mới có thể được thêm vào một cách dễ dàng và sử dụng trong một framework dành cho các nhà nghiên cứu thử nghiệm và khám phá ra nhiều ý tưởng mới.

b. Cài đặt chương trình thực nghiệm (code đính kèm).

3. Kết quả thực nghiệm và thống kê.

a. Cấu hình máy tính thực nghiệm thống kê:

- Hệ điều hành: Window 10 – 64bit.
- Môi trường lập trình: Ipython Notebook (bản Python 3.5).
- Cấu hình thực nghiệm (ảnh hưởng đến thời gian xử lý):

RAM	8GB
CPU	Intel core i7 – 6500U
Core speed	2.6 GHz
GPU	Nvidia GeForce 930M

b. Bộ thống kê thực nghiệm.

- Số nodes ở layer 1:
 - + Lần lượt nhận các giá trị: 5, 50, 500, 5000.
- Số nodes ở layer 2:
 - + Lần lượt nhận các giá trị: 5, 50, 500, 5000.
- Số lần lan truyền ngược:
 - + Lần lượt nhận các giá trị: 5, 10, 20, 40.

c. Bảng thống kê độ chính xác trên tập train, tập test và thời gian huấn luyện.

- Với số vòng lặp epochs: 40

Nodes of Layer 1	Nodes of Layer 2	Train accuracy	Test accuracy	Time train (s)
5000	3000	0.995	0.983	10528.046
500		0.994	0.984	1660.752
50		0.958	0.964	438.128
5		0.675	0.510	355.414
5000	300	0.995	0.984	3388.925
500		0.994	0.984	354.945
50		0.953	0.971	68.784
5		0.655	0.591	50.839
5000	30	0.986	0.982	2340.105
500		0.986	0.981	215.087
50		0.931	0.963	44.785
5		0.501	0.727	29.729
5000	3	0.112	0.113	2299.910
500		0.488	0.772	215.594
50		0.486	0.898	44.005
5		0.289	0.552	30.399

- Với số vòng lặp epochs: 20

Nodes of Layer 1	Nodes of Layer 2	Train accuracy	Test accuracy	Time train (s)
5000	3000	0.992	0.977	5195.166
500		0.991	0.981	656.244
50		0.948	0.972	198.886
5		0.641	0.560	171.294
5000	300	0.992	0.985	1436.674
500		0.991	0.983	162.80
50		0.944	0.964	32.034
5		0.611	0.649	24.480
5000	30	0.984	0.980	1213.816
500		0.980	0.981	117.117
50		0.920	0.958	22.528
5		0.541	0.735	17.301
5000	3	0.529	0.842	1179.883
500		0.565	0.927	113.225
50		0.491	0.835	23.085
5		0.296	0.662	14.745

- Với số vòng lặp epochs: 10

Nodes of Layer 1	Nodes of Layer 2	Train accuracy	Test accuracy	Time train (s)
5000	3000	0.987	0.978	2514.199
500		0.985	0.982	328.278
50		0.942	0.968	103.499
5		0.644	0.750	88.664
5000	300	0.988	0.982	727.092
500		0.985	0.981	72.965
50		0.935	0.965	16.384
5		0.611	0.689	12.269
5000	30	0.978	0.977	579.367
500		0.977	0.978	53.804
50		0.899	0.950	11.686
5		0.487	0.721	10.113
5000	3	0.523	0.837	549.196
500		0.427	0.675	53.925
50		0.505	0.864	12.526
5		0.275	0.605	7.767

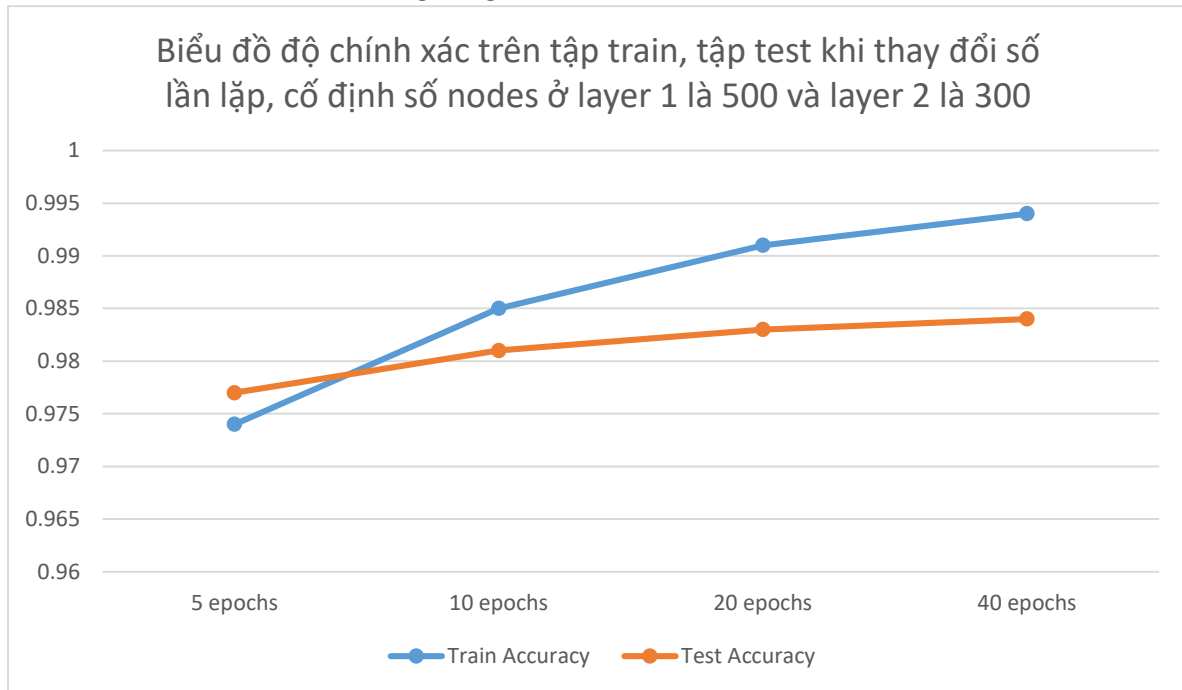
- Với số vòng lặp epochs: 5

Nodes of Layer 1	Nodes of Layer 2	Train accuracy	Test accuracy	Time train (s)
5000	3000	0.980	0.976	1251.806
500		0.978	0.978	166.321
50		0.924	0.959	56.857
5		0.596	0.700	46.368
5000	300	0.981	0.974	389.604
500		0.974	0.977	40.752
50		0.918	0.954	10.800
5		0.564	0.759	6.847
5000	30	0.970	0.977	298.080
500		0.957	0.972	28.329
50		0.874	0.941	6.732
5		0.464	0.827	4.960
5000	3	0.112	0.114	275.944
500		0.468	0.911	27.480
50		0.397	0.761	7.312
5		0.257	0.598	4.944

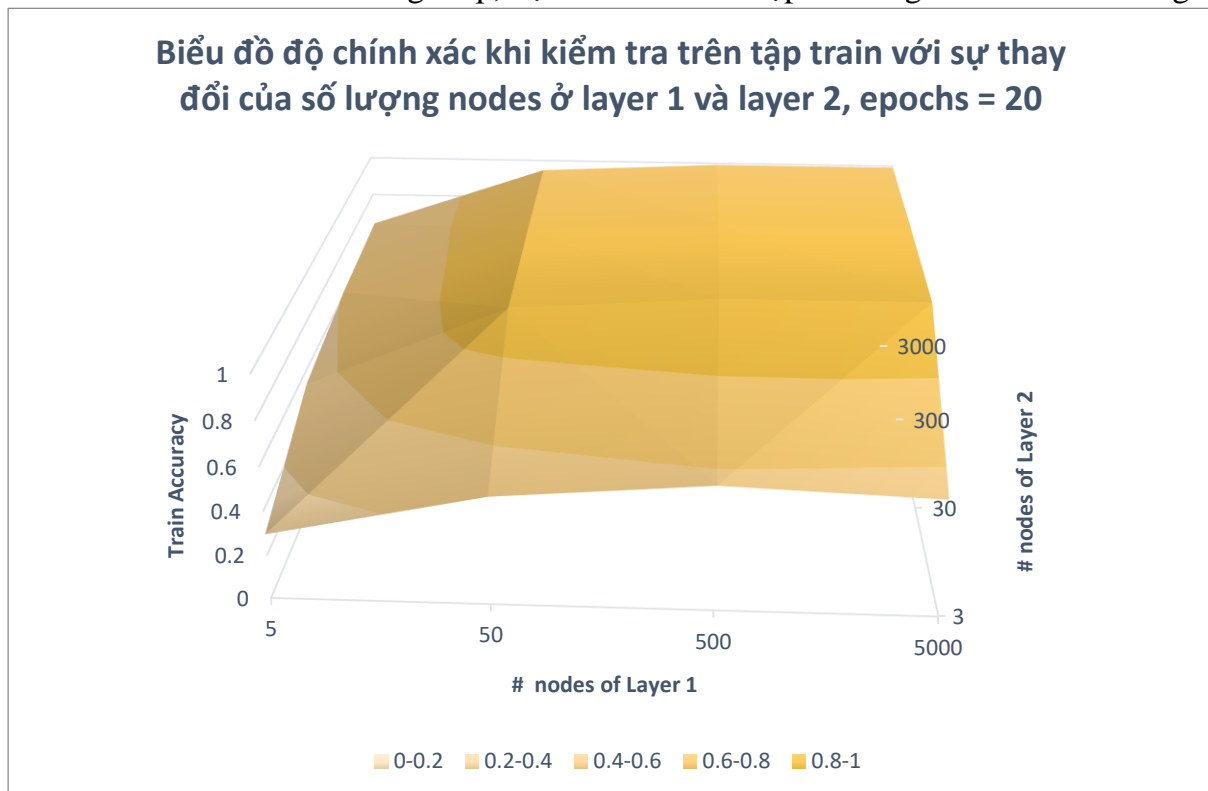
d. Đánh giá về kết quả thực nghiệm:

- Độ chính xác trên tập test mà mạng neural đa lớp có thể thực hiện được là 98,5% (tại Layer 1: 5000, Layer 2: 300 và Epochs: 20).

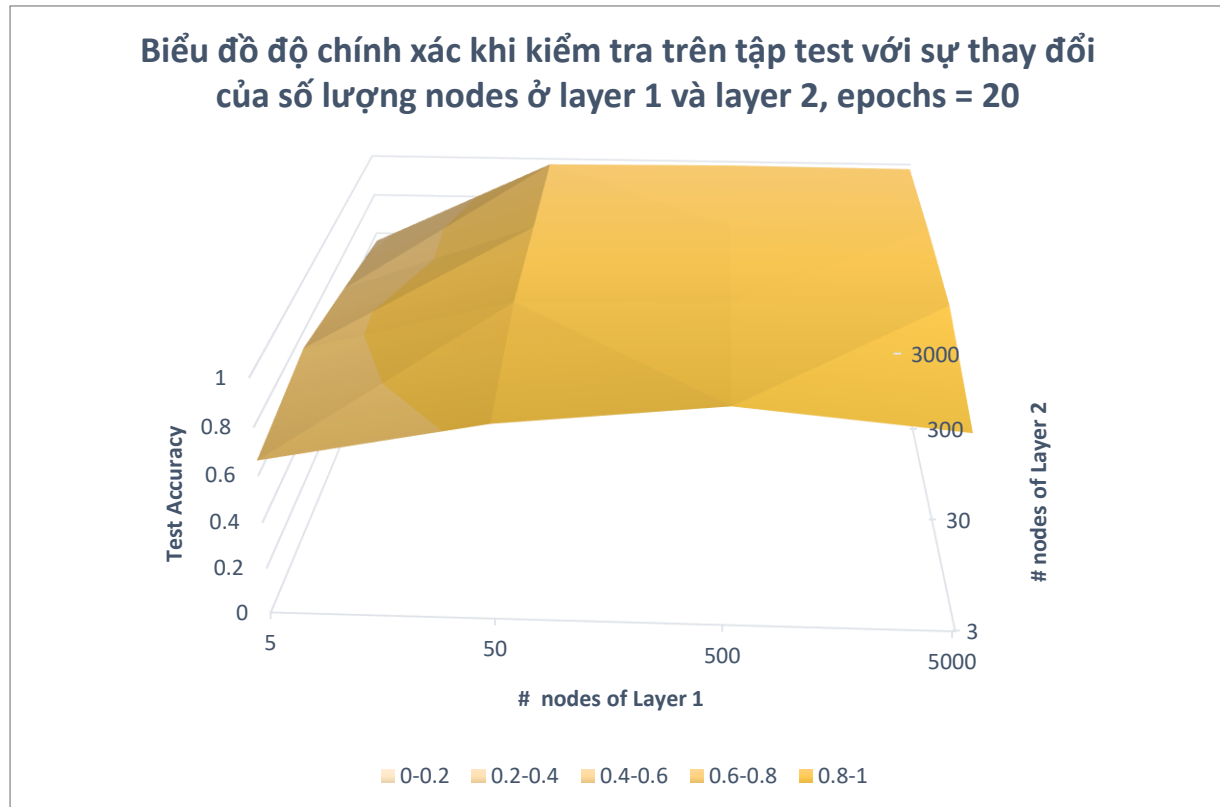
- Khi số vòng lặp (số lần áp dụng lan truyền ngược) càng tăng thì độ chính xác càng tăng.



- Khi số lượng node ở layer 1 và layer 2 càng lớn thì độ chính xác trên tập train càng lớn, tuy nhiên độ chính xác trên tập test sẽ giảm đi do có overfitting (mặc dù không nhiều).
- Khi số node ở layer 1 và layer 2 càng giảm thì độ chính xác trên tập train sẽ càng thấp, độ chính xác trên tập test sẽ giảm do underfitting.



- Khi số node ở layer 1, layer 2 giảm mà số epochs lại tăng thì độ chính xác trên tập test cao hơn trên tập train dù độ chính xác của cả hai không cao do tính chất độc lập trong xử lý mẫu của mạng nơron.
- Khi số node ở layer 1, layer 2 giảm mà số epochs giảm thì độ chính xác trên tập train giảm còn độ chính xác trên tập test sẽ không ổn định.



4. Đánh giá về mạng nơron truyền thẳng đa lớp.

- Mạng neural cho phép xây dựng một mô hình có khả năng học dữ liệu rất cao, có khả năng học được mối quan hệ, các ràng buộc giữa dữ liệu đầu vào và đầu ra.
- Một ưu điểm nổi trội của mạng neural là khả năng nhận dạng với số lượng mẫu output lớn.
- Việc sử dụng mạng neural trong bài toán nhận dạng chữ số viết tay đạt được nhiều ưu điểm hơn so với các phương pháp nhận dạng truyền thống khác.
- Mạng neural có khả năng học dữ liệu on-line, truyền trực tiếp dữ liệu vào mô hình học tập. Từ đó, cho ra kết quả với độ chính xác cao.
- Với số lượng nodes tại các lớp càng lớn, số lần học tập, lan truyền ngược càng nhiều thì độ chính xác trên tập huấn luyện mà mạng neural mang lại cho bài toán nhận dạng chữ số viết tay càng lớn. Hơn thế nữa, độ chính xác trên tập test cũng không thay đổi nhiều. Điều này chứng tỏ rằng mạng neural đa lớp có khả năng tổng quát hóa đặc trưng của đối tượng huấn luyện rất lớn. Góp phần xây dựng được một mô hình nhận dạng chữ số viết tay cực kỳ hiệu quả.

VII. Tài liệu tham khảo:

1. *Free online book: Neural Network and Deep Learning – Michael Nielsen.*
2. *Slide bài giảng tuần 4 về mạng nơ ron nhân tạo - Tiến sĩ Lê Hoàng Thái.*
3. *Giáo trình Cơ sở trí tuệ nhân tạo – Phó giáo sư, Tiến sĩ Lê Hoài Bắc, Thạc sĩ Tô Hoài Việt.*
4. *Mạng nơ ron nhân tạo truyền thống - KS. Phan Văn Khoa, TS. Lưu Trường Văn, GS. Lê Kiều*

Hồ Chí Minh, ngày 08 tháng 04 năm 2017

----- Hết -----