

NGHIÊN CỨU MẠNG NƠ-RON NHÂN TẠO VÀ ỨNG DỤNG VÀO NHẬN DẠNG CHỮ SỐ VIẾT TAY

Trần Thị Kiều*, Đặng Xuân Vinh, Vương Quang Phước

Khoa Điện tử viễn thông, Trường Đại học Khoa học, Đại học Huế

*Email: trankieudtvtk34@gmail.com

Ngày nhận bài: 30/10/2019; ngày hoàn thành phản biện: 14/11/2019; ngày duyệt đăng: 20/12/2019

TÓM TẮT

Trong nghiên cứu này, nhóm tác giả đã tìm hiểu về một số mô hình mạng nơ-ron nhân tạo để ứng dụng vào việc nhận dạng chữ số viết tay. Mô hình được lựa chọn là mô hình mạng nơ-ron nhân tạo đa lớp MLP (Multi-Layer Perceptron). Mô hình mạng nơ-ron này là một mô hình không quá phức tạp và phù hợp để ứng dụng vào nhận dạng chữ số viết tay. Bên cạnh đó, mô hình mạng nơ-ron này cũng là một mô hình cơ bản, cho nên việc tìm hiểu mô hình này là nền tảng để nghiên cứu những mô hình mạng nơ-ron khác phức tạp hơn. Các tham số của mô hình như tỷ lệ học, chu kỳ học, số lớp ẩn, số nơ-ron trên mỗi lớp ẩn được khởi tạo và lần lượt thay đổi để tìm ra bộ thông số tối ưu với mục đích xây dựng một mô hình mạng nơ-ron nhân tạo đa lớp MLP. Sau quá trình huấn luyện và kiểm định, mô hình đã đạt được độ chính xác khá cao (95.40%).

Từ khóa: AI, Mạng MLP; MNIST; Nhận dạng chữ số viết tay.

1. MỞ ĐẦU

Khái niệm trí thông minh nhân tạo (Artificial Intelligence: AI) [1] đã xuất hiện khá lâu, nhưng đến những năm gần đây nó mới có sự phát triển vượt bậc nhờ vào sự phát triển của mạng Internet, dữ liệu lớn (big data) và các công nghệ phần cứng hỗ trợ. Những ứng dụng phổ biến của AI mà chúng ta vẫn đang sử dụng hàng ngày đó là các trợ lý ảo trên các điện thoại thông minh (Siri, Google assistant, Cortana...).

Công nghệ AI cũng đang được áp dụng trong lĩnh vực chẩn đoán y khoa để giúp các bác sỹ nâng cao khả năng chữa trị các bệnh nan y [2]. Trí thông minh nhân tạo do IBM phát triển (IBM -Watson) đã được ứng dụng để phát hiện ung thư não chỉ trong 10 giờ phân tích, nhanh gấp 16 lần so với bác sỹ thông thường [3]. Trí thông minh nhân tạo cũng đang được ứng dụng một cách mạnh mẽ vào công nghệ xe tự lái, cho phép xe hoạt động một cách thông minh hơn, an toàn hơn (ví dụ như nền tảng NVIDIA Drive). Ngoài ra, AI cũng được áp dụng nhiều trong xử lý ngôn ngữ tự nhiên

Nghiên cứu mạng nơ-ron nhân tạo và ứng dụng vào nhận dạng chữ số viết tay

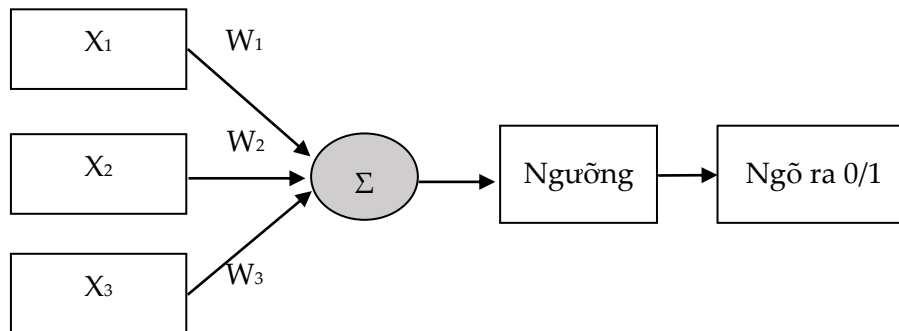
(Nature Language Process: NLP) để hỗ trợ phân tích các cấu trúc câu và ý nghĩa trong văn bản thông qua các phương pháp thống kê và máy học [4] [5].

Trong những năm gần đây, nhu cầu nhận dạng chữ số viết tay ngày càng lớn vì điều này có nhiều ứng dụng thực tế như phân loại thư, xử lý dữ liệu y tế, số hóa hồ sơ sổ sách. Tuy nhiên, điều này là một thách thức do đòi hỏi nguồn nhân lực rất lớn và tiềm ẩn nhiều sai sót trong quá trình thực hiện. Với sự phát triển của trí tuệ nhân tạo - AI (Artificial Intelligence), đặc biệt trong lĩnh vực xử lý ảnh bằng AI, việc số hóa dữ liệu đã đơn giản và nhanh chóng hơn rất nhiều [6]. AI giúp tiết kiệm thời gian và công sức, cũng như cải thiện độ chính xác trong việc xử lý, chuyển đổi định dạng dữ liệu hình ảnh sang dạng số hóa để dễ lưu trữ và phân tích hơn [7].

Các nghiên cứu gần đây sử dụng các mô hình quen thuộc để xử lý nhận dạng chữ số viết tay như mô hình mạng đa lớp MLP [8] [9] [10], mô hình mạng tích chập CNN (Convolutional Neural Network) hay mô hình học sâu DNN (Deep Learning Network) [11] [12]. Các mô hình này đều có ưu điểm là thời gian huấn luyện nhanh, độ chính xác cao, phù hợp với việc phân loại ảnh. Tuy nhiên, với mức độ phức tạp của bài toán nhận diện chữ số viết tay không quá cao và giới hạn trong khuôn khổ bài báo, nhóm tác giả sử dụng mô hình mạng nơ-ron nhân tạo đa lớp MLP để nghiên cứu vì mô hình này đơn giản, dễ hiệu chỉnh và đáp ứng được mục tiêu nghiên cứu.

2. MẠNG NƠ-RON NHÂN TẠO

Một Perceptron có các ngõ vào nhị phân x_j và được gán tương ứng các trọng số w_j - thể hiện mức tác động của ngõ vào đến ngõ ra. Hình 1 mô tả cấu trúc của một perceptron đơn giản với 03 giá trị ngõ vào x_1, x_2 và x_3 và 01 ngõ ra nhị phân.



Hình 1. Mô hình Perceptron đơn giản

Nhiệm vụ của mỗi Perceptron là xác định các giá trị ngõ ra là 0 hoặc 1 dựa vào các giá trị ngõ vào. Trong mạng nơ-ron nhân tạo, ngõ ra của các nơ-ron ở lớp trước là ngõ vào của các nơ-ron ở lớp tiếp theo. Việc quyết định giá trị ngõ ra là 0 hoặc 1 phụ thuộc vào nhiều tham số như trọng số (Weight - w), Bias (b) hay việc chọn hàm kích hoạt.

2.1. Trọng số (Weight - w)

Trọng số (Weight - w) là con số biểu thị mức độ quan trọng của ngõ vào so với ngõ ra [4]. Giá trị ngõ ra của perceptron phụ thuộc vào tổng giữa trọng số và ngõ vào:

$$Ngõ\ ra = \begin{cases} 0 & \text{nếu } \sum_i x_i w_i \leq \text{ngưỡng} \\ 1 & \text{nếu } \sum_i x_i w_i > \text{ngưỡng} \end{cases} \quad (1)$$

Trong đó: x_i : ngõ vào thứ i của perceptron; w_i : trọng số của ngõ vào x_i ; ngưỡng (threshold): mức ngưỡng quyết định giá trị ngõ ra. Hàm xác định ngõ ra ở (1) còn được gọi là hàm step.

2.2. Bias - w

Để đơn giản cho perceptron trong việc quyết định giá trị đầu ra là 0 hoặc 1, ta có thể thay thế giá trị ngưỡng bằng nghịch đảo số Bias hay ngưỡng = -b.

Do đó, biểu thức (1) có thể được viết lại như sau:

$$Ngõ\ ra = \begin{cases} 0 & \text{nếu } \sum_i x_i w_i + b \leq 0 \\ 1 & \text{nếu } \sum_i x_i w_i + b > 0 \end{cases} \quad (2)$$

2.3. Neuron sigmoid và hàm kích hoạt sigmoid

Tương tự như perceptron, neuron sigmoid có nhiều ngõ vào (x_1, x_2, x_3, \dots) tương ứng có các trọng số cho mỗi đầu vào khác nhau (w_1, w_2, w_3, \dots) và bias - b. Nhưng thay vì chỉ nhận một trong hai giá trị 0 hoặc 1, mỗi neuron có thể có bất kì giá trị thực nào giữa 0 và 1, ngõ ra được xác định bởi hàm kích hoạt sigmoid:

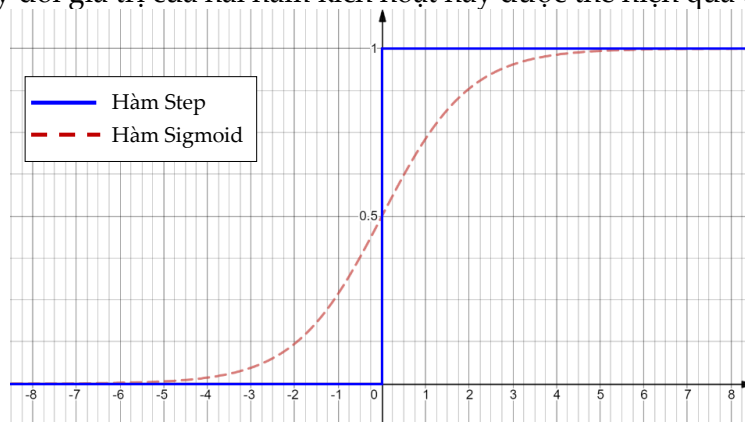
$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (3)$$

Trong đó: $z = \sum_i x_i w_i + b$

Giá trị ngõ ra của hàm sigmoid có thể được viết lại như sau:

$$Ngõ\ ra = \frac{1}{1+e^{-(\sum_i x_i w_i)}} \quad (4)$$

Sự thay đổi giá trị của hai hàm kích hoạt này được thể hiện qua đồ thị ở Hình 2.



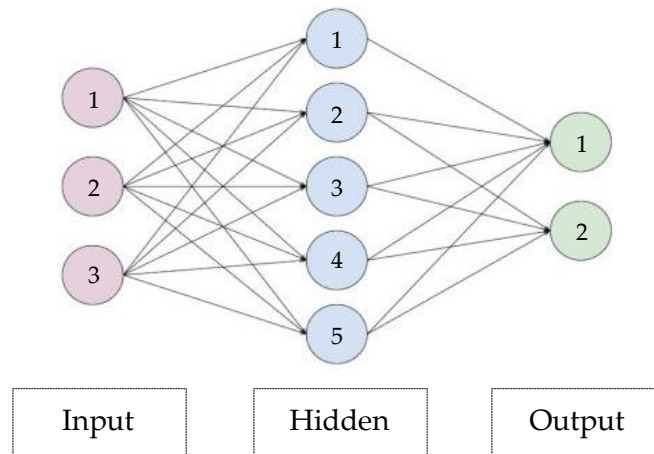
Hình 2. Đồ thị thể hiện sự thay đổi giá trị của hàm sigmoid và hàm step

3. MÔ HÌNH MẠNG ĐA LỚP MLP

3.1. Cấu trúc mô hình MLP

Mô hình MLP [8] là mô hình perceptron nhiều lớp. Các perceptron được chia thành nhiều nhóm, mỗi nhóm tương ứng với 1 lớp trong mạng. Một mô hình MLP gồm tối thiểu ba lớp, bao gồm một lớp ngõ vào (Input layer), một lớp ngõ ra (Output layer) và một hoặc nhiều lớp ẩn (Hidden layer). Thông thường khi giải quyết một bài toán ta chủ yếu quan tâm đến lớp ngõ vào và lớp ngõ ra của mô hình. Một sơ đồ cấu tạo mô hình MLP đơn giản nhất được thể hiện ở Hình 3.

Trong mô hình MLP, các nơ-ron ở các lớp kề nhau được liên kết hoàn toàn với nhau (fully connected layer), ngõ ra của các lớp trước sẽ là ngõ vào của các lớp sau. Chính vì cách tính toán theo một chiều từ ngõ vào cho tới ngõ ra như trên mà mô hình MLP còn được gọi là Mạng lan truyền tiến (Feed-forward Neural Network - FNN). Ưu điểm của mô hình này là đơn giản, dễ thực hiện tuy nhiên yêu cầu nhiều tài nguyên tính toán do chứa nhiều thông số và khối lượng phép tính rất lớn.



Hình 3. Cấu tạo mô hình MLP đơn giản

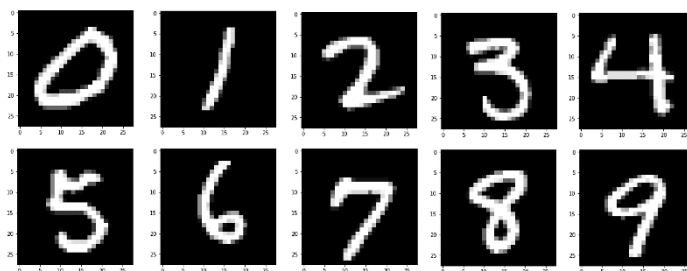
Sơ đồ mô tả ở Hình 3 cho thấy, nếu chưa kể đến thông số bias, số lượng trọng số trong mô hình đã là: $(3 \times 5) + (5 \times 2) = 25$ trọng số cho một mô hình đơn giản. Do vậy trong quá trình thiết kế mô hình cần tối ưu số lượng lớp và số nơ-ron ở mỗi lớp để giảm khối lượng tính toán, từ đó tăng hiệu năng của mô hình.

3.2. Huấn luyện mô hình MLP

3.2.1. Cơ sở dữ liệu

Việc tự tạo cơ sở dữ liệu (database) cho việc huấn luyện mô hình rất tốn công sức và thời gian. Do vậy, trong nghiên cứu này, nhóm tác giả sử dụng bộ cơ sở dữ liệu nổi tiếng MNIST (Modified National Institute of Standards and Technology) [13] [14] do Yann LeCun chia sẻ.

Bộ cơ sở dữ liệu MNIST bao gồm 70,000 hình ảnh chữ số viết tay đã được hiệu chỉnh thành các ảnh xám đơn sắc. Nó được chia thành hai tập: tập huấn luyện gồm 60,000 ảnh, tập kiểm tra gồm 10,000 ảnh. Mỗi ảnh có kích thước 28 pixel x 28 pixel, ảnh đã được khử răng cưa cũng như thông tin của ảnh đã được đưa về trung tâm nhằm loại bỏ các trường hợp các ảnh nằm ở góc trái, góc phải, lật ngược... Mỗi ảnh được gắn nhãn (label) từ 0 đến 9 tương ứng với các chữ số. Bộ cơ sở dữ liệu MNIST là bộ cơ sở đơn giản nhất được dùng phổ biến trong các mạng nơ-ron nhân tạo nhận dạng hình ảnh cơ bản hoặc nhằm mục đích nghiên cứu. Một số hình ảnh về chữ số viết tay trong bộ dữ liệu MNIST được thể hiện ở Hình 4.



Hình 4. Hình ảnh một số chữ số viết tay từ bộ dữ liệu MNIST

3.2.2. Phương pháp huấn luyện trong mô hình MLP

Hiệu suất của mô hình được đánh giá bằng cách sử dụng hàm tổn hao (cost function) được thể hiện ở công thức (5). Ta dựa vào hàm này để đánh giá một mô hình mạng được huấn luyện tốt hay không tốt. Giá trị của hàm tổn hao có giá trị lớn nếu ngõ ra dự đoán của mạng khác với nhãn thực sự và ngược lại.

$$C(w,b) = \frac{1}{2n} \sum_x (y(x) - a)^2 \quad (5)$$

Trong đó: $C(w,b)$: hàm tổn hao; n : tổng số nhãn dùng để kiểm tra; a : ngõ ra của mạng dự đoán; $y(x)$: giá trị ngõ ra thực tương ứng với ngõ vào x .

Việc huấn luyện hay cho mô hình “học” thực chất là việc tinh chỉnh và tìm ra giá trị của trọng số và bias sau mỗi chu kỳ học từ những giá trị ngẫu nhiên được khởi tạo ban đầu để có kết quả ngõ ra mong muốn. Ở mỗi chu kỳ học, các trọng số và bias sẽ được cập nhật theo phương pháp Gradient Descent bằng cách lấy đạo hàm của hàm tổn hao [9]. Việc cập nhật trọng số w và bias b được thể hiện ở công thức (6) và (7).

$$w' = w - \frac{\eta}{m} \sum \frac{\partial C}{\partial w} \quad (6)$$

$$b' = b - \frac{\eta}{m} \sum \frac{\partial C_x}{\partial w} \quad (7)$$

Trong đó w' , b' : tương ứng là các trọng số và bias mới của các nơ-ron được cập nhật sau mỗi chu kỳ học; η : tỷ lệ học; m : mini-batch size được tạo ra bằng cách chia nhỏ số lượng dữ liệu.

4. THỰC NGHIỆM VÀ KẾT QUẢ

Mô hình nhận dạng chữ số viết tay với các tham số khác nhau đã được huấn luyện và kiểm định trên hệ thống Cloud Computing của Google, với cấu hình như sau: CPU Intel(R) Xeon(R) CPU @ 2.30GHz (2 cores), RAM 12GB, GPU Nvidia Tesla K80. Mô hình mạng MLP đã được mô hình hóa bằng ngôn ngữ lập trình Python 3 kết hợp với thư viện Keras và Panda. Thời gian huấn luyện mỗi mô hình khoảng 30 phút ứng với 100 chu kỳ học. Kết quả đã tìm ra bộ tham số tối ưu cho mô hình, đưa ra kết quả nhận dạng chữ số viết tay có độ chính xác cao.

4.1. Xây dựng mô hình – lựa chọn tham số

Trong nghiên cứu này, nhóm tác giả đã xây dựng mô hình mạng MLP với số đầu vào, đầu ra và số lớp ẩn như sau:

- Lựa chọn số nơ-ron ở ngõ vào và ngõ ra: Dữ liệu ảnh từ bộ dữ liệu MNIST gồm các ảnh có kích thước 28×28 như đã đề cập ở trên, do vậy số nơ-ron ở ngõ vào là $28 \times 28 = 784$. Tương tự ngõ ra gồm 10 nơ-ron tương ứng với các số nguyên từ 0 đến 9.
- Lựa chọn số lớp ẩn và số nơ-ron trên mỗi lớp ẩn: Theo Jeff Heaton [10], kích thước tối ưu của lớp ẩn thường nằm giữa kích thước lớp ngõ vào và kích thước lớp ngõ ra. Bên cạnh đó, đối với mạng MLP nếu càng nhiều lớp ẩn sẽ có số lượng trọng số càng lớn. Vậy để phù hợp với mục đích nghiên cứu và giảm thiểu thời gian huấn luyện mô hình, nhóm tác giả chọn số lớp ẩn bằng 01 và số nơ-ron trên mỗi lớp ẩn bằng 100.

Việc chọn lựa các tham số cho mô hình tùy thuộc vào yêu cầu mỗi bài toán và không có một quy luật chung nào. Do đó để đánh giá sự ảnh hưởng của các tham số đến mô hình, ở mỗi bước, nhóm tác giả lần lượt thay đổi từng tham số và giữ nguyên các tham số còn lại để lần lượt tìm ra bộ tham số tối ưu. Trong bài báo này, nhóm tác giả khảo sát và lựa chọn 4 thông số tối ưu cho mô hình là: Chu kỳ học (Epochs), tỷ lệ học (Learning rate – LR), số lớp ẩn và số nơ-ron trên mỗi lớp ẩn. Ở mỗi chu kỳ học, các giá trị sau sẽ được tính toán: độ chính xác của quá trình huấn luyện (train_acc), tỉ lệ lỗi của quá trình huấn luyện (train_loss), độ chính xác của quá trình kiểm tra (val_acc), tỉ lệ lỗi của quá trình kiểm tra (val_loss).

Trong quá trình học, khi chu kỳ học của nào đó của mô hình có giá trị val_loss bắt đầu cao hơn giá trị train_loss, lúc này mô hình bắt đầu bị “overfitting”. Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu huấn luyện. Việc quá khớp này có thể dẫn đến việc dự đoán nhầm lẫn, và chất lượng mô hình không còn tốt trên dữ liệu kiểm tra nữa. Do vậy, để tránh tình trạng overfitting, tham số về chu kỳ học sẽ được lựa chọn và đánh giá trước, sau đó các tham số tỉ lệ học, số lớp ẩn và số nơ-ron trên mỗi lớp ẩn sẽ lần lượt được lựa chọn.

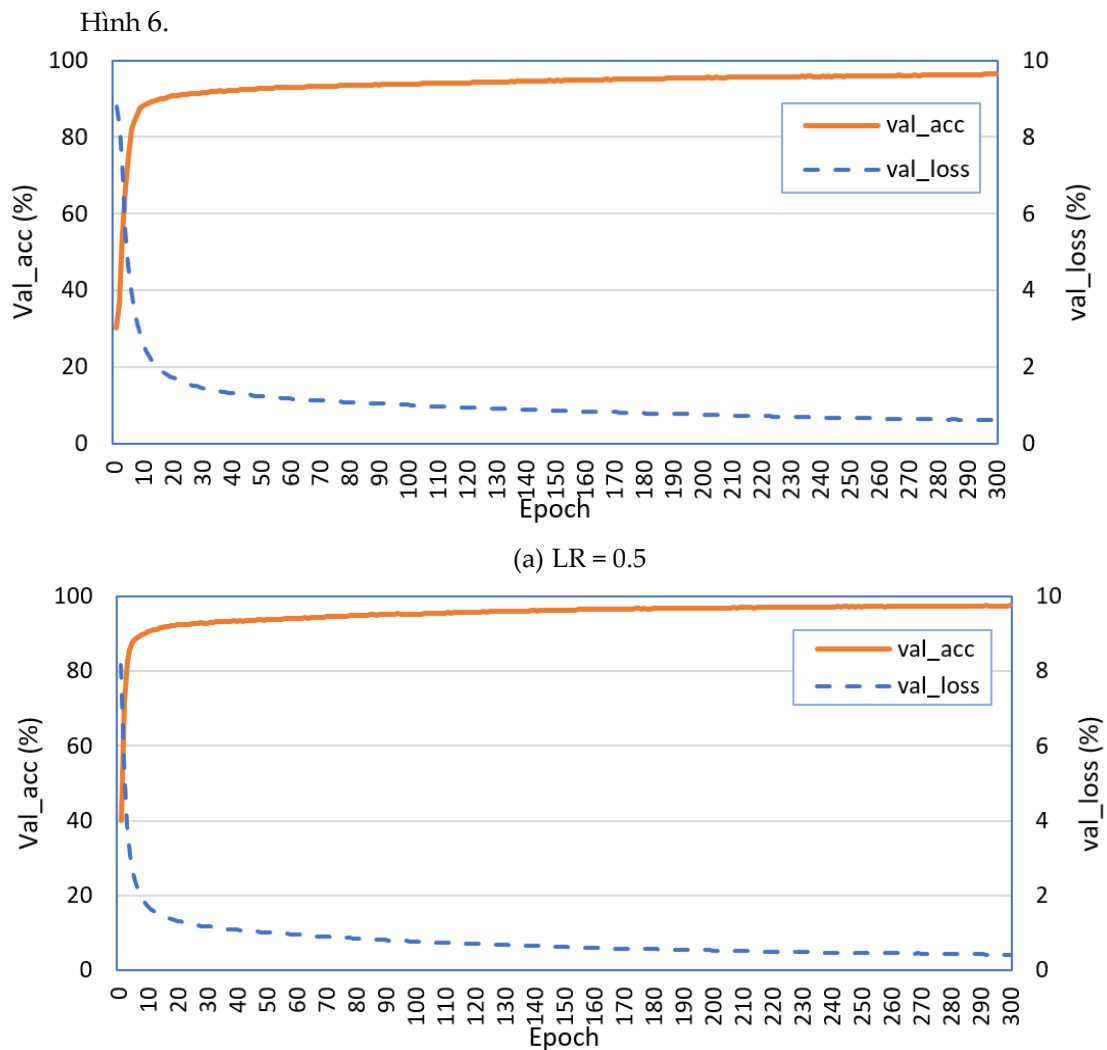
Để đánh giá và lựa chọn tham số chu kỳ học, tác giả sử dụng 4 mô hình với các bộ tham số như ở Bảng 1. Trong đó, chu kỳ học được lựa chọn ban đầu là 300. Chu kỳ học thích hợp là chu kỳ học trước khi hiện tượng overfitting xảy ra. Bên cạnh đó, theo nghiên cứu [9], giá trị tỷ lệ học bằng 0.8 cho kết quả độ chính xác nhận dạng rất cao (trên 95%). Từ đó, nhóm tác giả chọn sơ bộ 4 giá trị tỷ lệ học từ thấp đến cao là 0.5, 1.0, 1.5 và 2.0 và chu kỳ học là 300 để đánh giá đồng thời ảnh hưởng của hai tham số này đến mô hình.

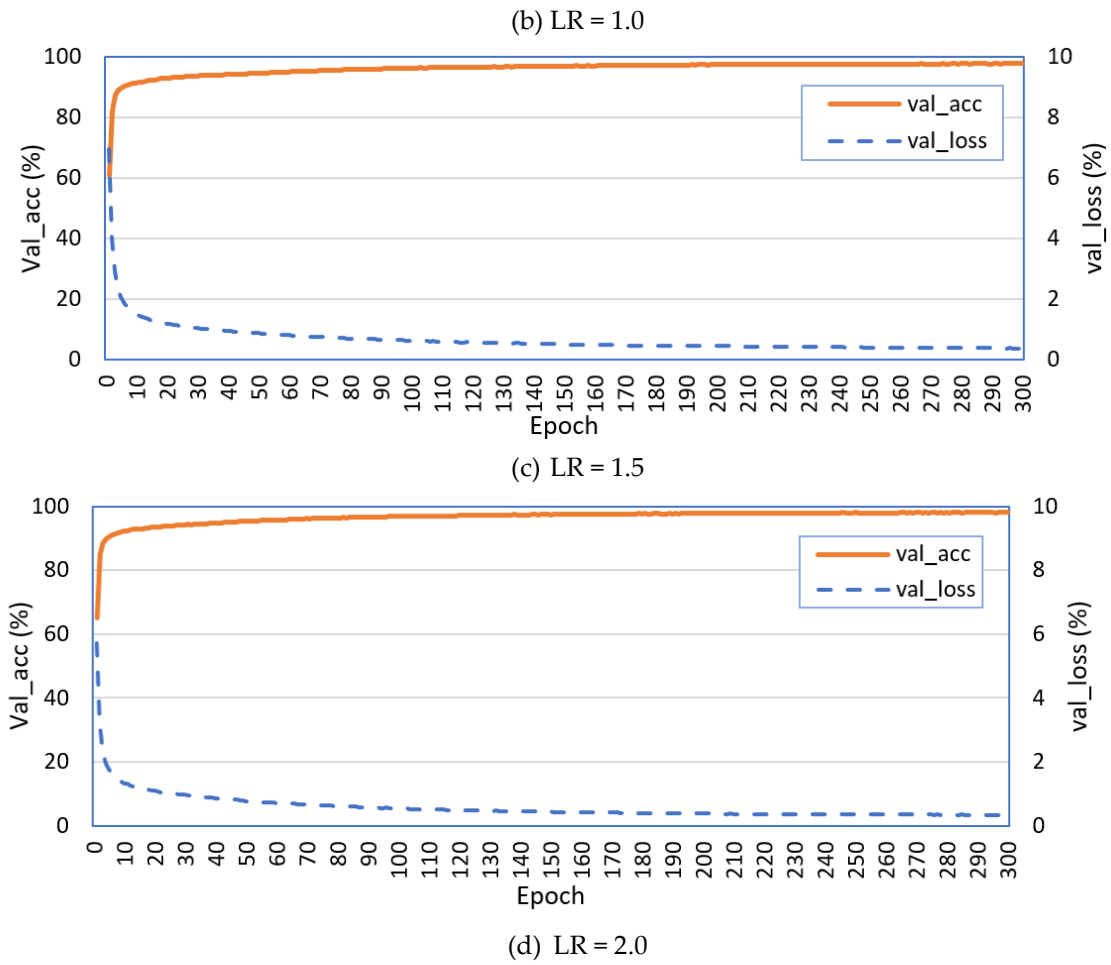
Bảng 1. Các tham số của 4 mô hình để lựa chọn chu kỳ học

Tỷ lệ học	Chu kỳ học	Số lớp ẩn	Số nơ-ron mỗi lớp ẩn	Số tổ hợp mô hình
0.5; 1.0; 1.5; 2.0	300	1	100	4

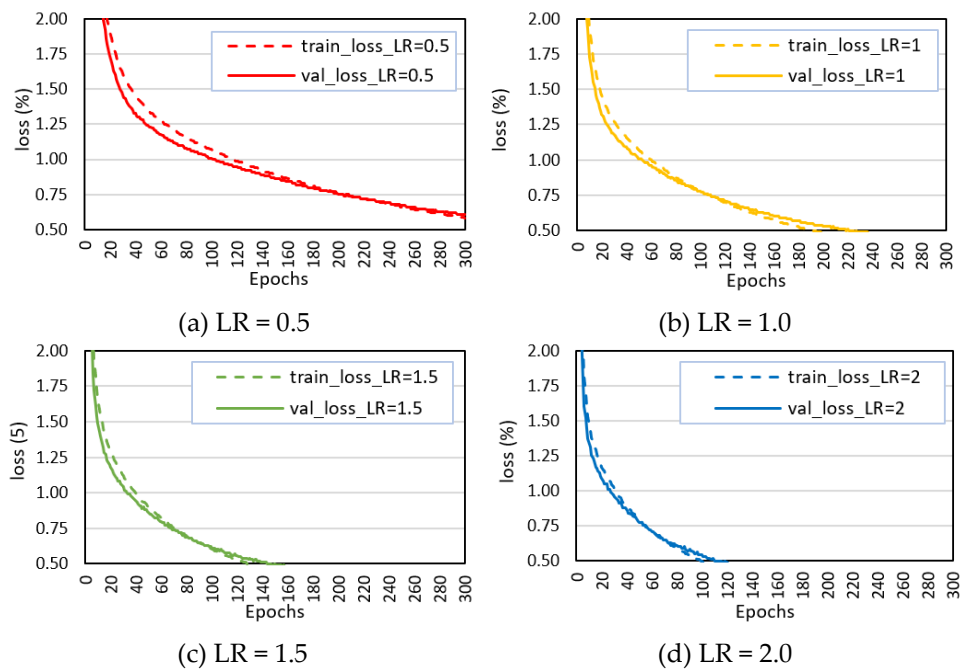
Kết quả độ chính xác của các mô hình được thể hiện ở hình Hình 5 a, b, c, d. Các giá trị train_loss và val_loss của các mô hình được thể hiện ở (c) LR = 1.5

(d) LR = 2.0





Hình 5. Độ chính xác của mô hình với 300 chu kì học và thông số LR khác nhau



Hình 6. Đồ thị thể hiện sự thay đổi val_loss và train_loss của các mô hình

Kết quả ở Hình 5 cho thấy, mặc dù với tỷ lệ học khác nhau, độ chính xác của các mô hình đều được cải thiện qua 3 giai đoạn tương đồng nhau:

- Giai đoạn 1: Từ chu kỳ học 1 đến 20, độ chính xác của các mô hình tăng rất nhanh. Tuy nhiên với tỉ lệ học thấp (LR = 0.5) thì độ chính xác của chu kỳ học đầu chỉ khoảng 30.27%, trong khi đó với tỉ lệ học cao (LR = 2.0) thì độ chính xác của chu kỳ học đầu tiên lên đến 65.07%.

- Giai đoạn 2: Từ chu kỳ học 21 đến 100, độ chính xác của các mô hình vẫn tăng nhưng không đáng kể, chỉ được cải thiện từ 3.11% (LR = 0.5) đến 3.35% (LR = 2.0).

- Giai đoạn 3: Từ chu kỳ học 101 trở đi, độ chính xác vẫn được cải thiện nhưng rất ít (khoảng 1.65%) và hầu như không thay đổi từ chu kỳ học thứ 250 trở đi.

Kết quả ở (c) LR = 1.5

(d) LR = 2.0

Hình 6 cho thấy, các mô hình sau các chu kỳ học nhất định đều bị hiện tượng overfitting. Các chu kỳ học bắt đầu xuất hiện hiện tượng overfitting của các mô hình được thể hiện ở Bảng 2. Trong đó, với tỉ lệ học nhỏ (LR = 0.5), phải đến chu kỳ học thứ 208 mô hình mới bắt đầu bị overfitting, với tỉ lệ học lớn (LR = 2.0) thì mới chỉ 61 chu kỳ học là mô hình đã bị overfitting. Vậy có thể kết luận: Với tỷ lệ học càng lớn thì mô hình càng nhanh bị overfitting.

Bảng 2. Các giá trị chu kỳ học của các mô hình khi bắt đầu bị overfitting

Tỉ lệ học	Chu kỳ học	val_loss (%)	train_loss (%)
0.5	208	0.75	0.74
1.0	106	0.59	0.58
1.5	91	0.65	0.64
2.0	61	0.71	0.70

Từ các phân tích trên, để tăng tối đa độ chính xác của mô hình cũng như tránh hiện tượng overfitting, tác giả chọn giá trị tham số chu kỳ học bằng 100.

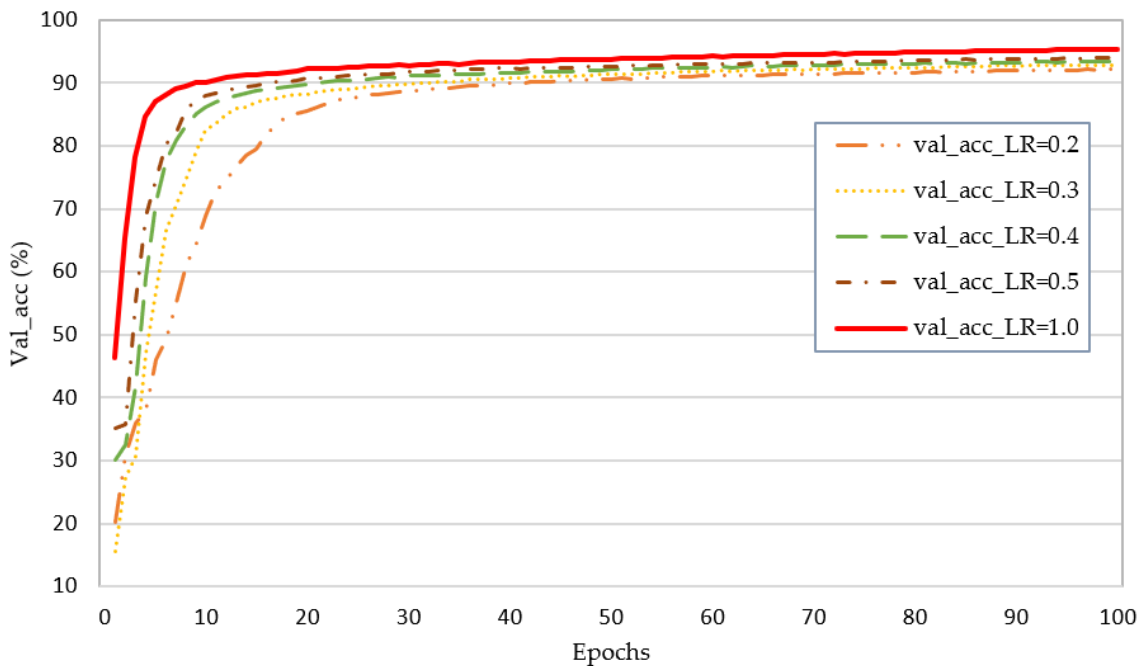
4.1.1. Lựa chọn tỷ lệ học LR

Với chu kỳ học đã lựa chọn ở trên, tác giả tạo ra các mô hình mới với tham số tỷ lệ học được thay đổi để đánh giá sự ảnh hưởng của tham số này tới mô hình. Sự thay đổi của tỷ lệ học và số lượng các mô hình được thể hiện ở Bảng 3.

Bảng 3. Các bộ tham số mô hình để đánh giá ảnh hưởng của tỷ lệ học

Tỷ lệ học	Chu kỳ học	Số lớp ẩn	Số nơ-ron mỗi lớp ẩn	Số tổ hợp mô hình
0.2; 0.3; 0.4; 0.5; 1.0	100	1	100	5

Kết quả sau khi chạy 5 mô hình, độ chính xác của các mô hình được thể hiện như ở Hình 7. Với tỉ lệ học thấp nhất (LR = 0.2), mô hình cần tới 42 chu kì học để đạt được độ chính xác 90%, độ chính xác sau khi quá trình học hoàn tất chỉ đạt 92.19%. Với tỷ lệ học trung bình (LR = 0.5), mô hình vẫn cần 20 chu kì học để đạt được độ chính xác 90%, độ chính xác tối đa là 94.01%. Ngược lại, với tỉ lệ học cao nhất (LR = 1.0), chỉ cần 10 chu kì học thì mô hình đã đạt độ chính xác 90%, độ chính xác cao nhất đạt được là 95.40%. Do đó, để quá trình học được nhanh chóng và đảm bảo độ chính xác tối đa, tác giả chọn tỷ lệ học bằng 1.0 là tham số tối ưu tiếp theo cho mô hình.



Hình 7. Độ chính xác của mô hình khi thay đổi tham số tỷ lệ học

4.1.2. Lựa chọn số lớp ẩn và số nơ-ron trên mỗi lớp ẩn

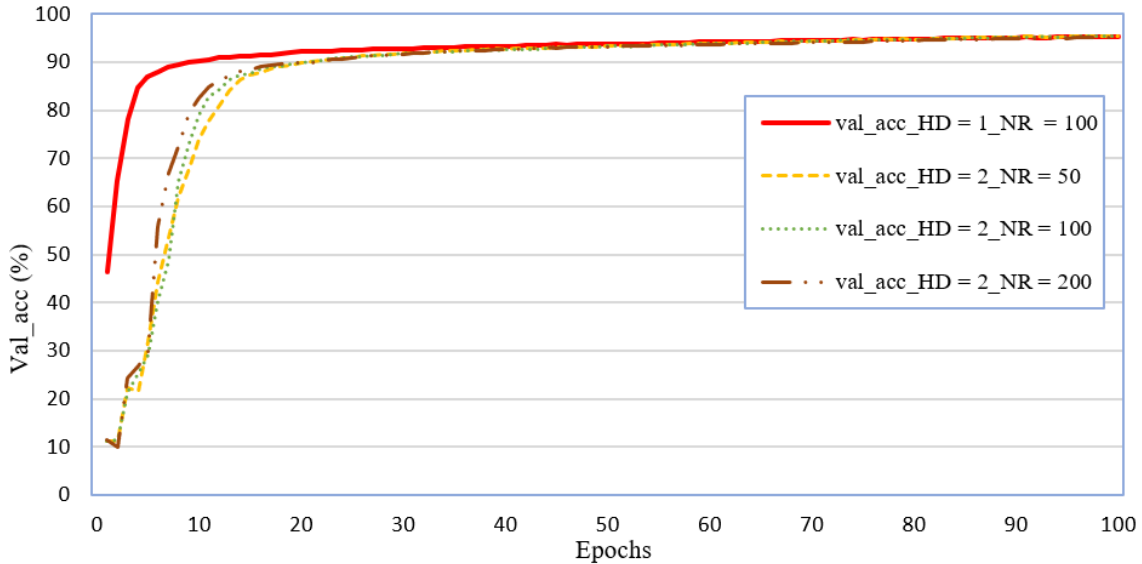
Ở bước này, tác giả tiến hành thay đổi số lớp ẩn cũng như số nơ-ron trên mỗi lớp ẩn để đánh giá ảnh hưởng của các tham số này đến độ chính xác của mô hình. Các bộ tham số được thể hiện ở Bảng 4. Kết quả độ chính xác của các mô hình được thể hiện ở Hình 8.

Bảng 4. Các bộ tham số mô hình để lựa chọn số lớp ẩn và số nơ-ron trên mỗi lớp ẩn

Tỷ lệ học	Chu kì học	Số lớp ẩn	Số nơ-ron mỗi lớp ẩn	Số tổ hợp mô hình
1.0	100	1	100	1
1.0	100	2	50; 100; 200	3

Kết quả ở Hình 8 cho thấy, mặc dù có sự khác nhau về độ chính xác trong giai đoạn “warming up” nhưng sau khoảng 50 chu kì học, độ chính xác của các mô hình là tương tự nhau. Điều này có thể giải thích là do bài toán đặt ra ở dạng đơn giản với ngõ

ra chỉ 10 lớp và độ chính xác của mô hình MLP gần như đạt tới giới hạn, kết quả này phù hợp với các nghiên cứu trước đây [9]. Do vậy, tác giả sử dụng tham số giống như lúc khởi tạo ban đầu với số lớp ẩn bằng 1 và số nơ-ron trên mỗi lớp ẩn bằng 100.



Hình 8. Độ chính xác của mô hình khi thay đổi số lớp ẩn và số nơ-ron trên mỗi lớp ẩn

4.2. Kết quả

Từ các phân tích trên, tác giả đã lựa chọn được bộ thông số tối ưu cho mô hình như ở Bảng 5. Với bộ thông số này, mô hình đã đạt được độ chính xác cao (95.40%).

Bảng 5. Bộ tham số tối ưu của mô hình

Tỷ lệ học	Chu kỳ học	Số lớp ẩn	Số nơ-ron trên mỗi lớp ẩn
1.0	100	1	100

4. KẾT LUẬN

Bài báo đã giới thiệu về mô hình mạng nơ-ron nhân tạo, ý nghĩa của một số tham số trong mô hình. Bài báo cũng đã phân tích để tìm ra bộ tham số tối ưu để mô hình đạt được độ chính xác cao nhất cũng như đảm bảo độ ổn định. Kết quả nghiên cứu này có thể sử dụng để làm cơ sở cho nghiên cứu tiếp theo như ứng dụng mạng nơ-ron nhân tạo để số hóa bảng điểm cho học sinh hoặc số hóa các tài liệu chữ số viết tay khác. Bên cạnh đó, nghiên cứu vẫn còn một số hạn chế như chưa đánh giá được ảnh hưởng của sự thay đổi đồng thời các tham số đến độ chính xác, mô hình được lựa chọn còn đơn giản. Do vậy tác giả đề xuất sử dụng các mô hình tiên tiến hơn như CNN, DNN để thực hiện các nghiên cứu tiếp theo nhằm cải thiện độ chính xác cũng như hiệu suất của mô hình.

TÀI LIỆU THAM KHẢO

- [1]. Richard E. Neapolitan, Xia Jiang, *Artificial Intelligence: With an Introduction to Machine Learning*, Second Edition, Chapman and Hall/CRC, 2018.
- [2]. S. Kang, "CancerLocator: non-invasive cancer diagnosis and tissue-of-origin prediction using methylation profiles of cell-free DNA," *Genome Biology*, pp. 18-53, 2017.
- [3]. E. Strickland, "IBM Watson Makes a Treatment Plan for Brain-Cancer Patient in 10 Minutes; Doctors Take 160 Hours," *IEEE Spectrum*, 2017.
- [4]. Yuanzhi Ke and M. Hagiwara, "A natural language processing neural network comprehending English," in *International Joint Conference on Neural Networks (IJCNN)*, 2015.
- [5]. Lernout Stephen; Devos Geert; Kraze Andreas; Platteau Frank, "A non-biological AI approach towards natural language understanding," 2016.
- [6]. Đ. T. Nghị and P. N. Khang, "Nhận dạng ký tự số viết tay bằng giải thuật máy học," *Tạp chí Khoa học Trường Đại học Cần Thơ*, 2013.
- [7]. Stipe Celar, Zeljko Stojkic, Zeljko Seremet, Zeljko Marusic, Danijel Zelenika, "Classification of Test Documents Based on Handwritten Student ID's Characteristics," *Procedia Engineering*, pp. 782-790, 2015.
- [8]. Walter H. Delashmit and Michael T. Manry, "Recent developments in multilayer perceptron neural networks," *Proceedings of the 7 th Annual Memphis Area Engineering and Science Conference, MAESC*, 2005.
- [9]. S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "An MLP based Approach for Recognition of Handwritten Bangla'Numerals," *Proc. 2nd Indian International Conference on Artificial Intelligence*, pp. 407-417, 2005.
- [10]. Hossein Karimi, Azadeh Esfahanimehr, Mohammad Mosleh, Faraz Mohammadian, Simintaj Salehpour, Omid Medhati,, "Persian handwritten digit recognition using ensemble classifiers," *Procedia Computer Science*, pp. 416-425, 2015.
- [11]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, p. 1097-1105, 2012.
- [12]. Berkant Savas and Lars Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognition*, pp. 993-1003, 2007.
- [13]. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.
- [14]. L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Processing Magazine*, pp. 141-142, 2012.

A STUDY ABOUT NEURON NETWORK AND APPLYING TO RECOGNIZE HANDWRITTEN DIGITS

Tran Thi Kieu*, Dang Xuan Vinh, Vuong Quang Phuoc

Faculty of Electronics – Telecommunications, University of Sciences, Hue University

*Email: trankieudtvtk34@gmail.com

ABSTRACT

In this paper, the authors built an artificial neural network model to recognize handwritten digits. The proposed model is an artificial neural network model called MLP (Multi-Layer Perceptron). This neural network model is not too complicated and suitable for application in handwritten recognition. Besides, this neural network model is also a basic model, so learning about this one is the foundation for studying other more complex neural network models. The parameters of the model such as learning rate, epochs, number of hidden layers, the neuron of each hidden layers are initialized and modified to choose the optimal set of parameters to build a model. After training and testing process, the model achieved high accuracy (95.40%).

Keywords: AI, Handwriting number recognition, MLP Network, MNIST.



Trần Thị Kiều sinh ngày 29/02/1992 tại Quảng Nam. Năm 2015, bà tốt nghiệp kỹ sư chuyên ngành Điện tử viễn thông tại Trường Đại học Khoa học, Đại học Huế. Hiện nay, bà đang công tác tại Khoa Điện tử Viễn thông, Trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Thông tin quang.



Đặng Xuân Vinh sinh năm 1959 tại Nam Định. Ông nhận bằng cử nhân đại học ngành Vật lý Vô tuyến tại Đại học Tổng hợp Hà Nội năm 1978, bằng Thạc sĩ Vật lý Chất rắn năm 1986 và bằng Tiến sĩ Khoa học Vật liệu (Vật liệu điện tử) năm 2000. Hiện ông công tác tại khoa Điện tử -Viễn thông, trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Kỹ thuật điện tử, Kỹ thuật viễn thông và Kỹ thuật Điều khiển tự động (Tự động hóa).



Vương Quang Phước sinh ngày 14/08/1990 tại Thừa Thiên Huế. Năm 2013, ông tốt nghiệp kỹ sư chuyên ngành Điện tử viễn thông tại Trường Đại học Khoa học, Đại học Huế. Hiện nay, ông đang công tác tại Khoa Điện tử Viễn thông, Trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Thông tin quang, mạng máy tính, Deep Learning.