

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**

\* \* \* \* \*



**CHỦ ĐỀ: Mạng neuron và bài toán phân loại chữ viết tay**

Môn học: Cấu trúc dữ liệu và giải thuật – IT003.M21.KHTN

Giảng viên: Nguyễn Thanh Sơn

Sinh viên: Nguyễn Đức Nhân

MSSV: 21520373

Lớp: KHTN2021

## Mục lục

I.	Tổng quan về đề án và lược sử của học sâu: .....	3
II.	Định nghĩa về mạng neuron và các thành phần của mạng học sâu: .....	4
1.	<i>Mạng neuron</i> : .....	4
2.	<i>Các thành phần của mạng neuron</i> : .....	4
III.	Bài toán phân loại chữ viết tay và hiệu quả của các thuật toán học máy khác so với neural network: .....	6
IV.	Giải thích lý do cần mạng neural và các activity function .....	7
1.	Nhược điểm của các thuật toán học máy: .....	7
2.	Sự ưu thế của mạng neuron: .....	7
V.	Bài toán phân loại chữ số viết tay trên tập dữ liệu MNIST: .....	7
1.	Phát biểu bài toán: .....	7
2.	Giới thiệu về một số đặc trưng của tập dữ liệu: .....	8
3.	Định hướng giải quyết bài toán: .....	8
4.	Cấu trúc mạng được đề xuất để giải quyết bài toán: .....	8
5.	Thuật toán hiệu chỉnh và độ đo: .....	8
6.	Nghiệm thu kết quả: .....	9
VI.	Tổng kết: .....	9

## I. Tổng quan về đề án và lược sử của học sâu:

Những từ học sâu, mạng neuron, mạng thần kinh nhân tạo,... đang trở nên quen thuộc và xuất ở rất nhiều lĩnh vực, giải quyết được rất nhiều bài toán trong thực tiễn. Nhưng những ai đó hiểu điều gì đang hoạt động ở dưới những cái tên đó thì thật sự không nhiều. Vì lý do đó nên nội dung tôi sẽ đề cập đến trong đề án này sẽ xoay quanh hai chủ đề chính là học sâu (Deep learning) và mạng thần kinh nhân tạo (Neural network).

Cách đơn giản nhất để ta hiểu về một thứ gì đó là ta hãy tìm hiểu lịch sử của nó. Quả thực khi nói đến học sâu thì mọi người đều cảm thấy nó có tính ứng dụng nhưng lại khá mới (chỉ vài năm gần đây học sâu mới thật sự được ứng dụng nhiều). Và hẳn nhiều người sẽ bất ngờ hơn khi học sâu đã có khởi nguồn từ những năm 1940. Học sâu từng mang một cái tên là mạng neuron nhân tạo – thứ đã trở thành một phần trong học sâu.

Ở thuở sơ khai của học sâu, các mô hình được thiết kế với tính chất đặc trưng của một mô hình tuyến tính (linear model) nhưng chúng vấp phải những hạn chế có thể đến như việc không học được hàm XOR (với  $f([1,0]) = f([0,1]) = 1$  nhưng  $f([1,1]) = f([0,0]) = 0$ ). Và vì vấp phải hạn chế khi học những hàm phi tuyến tính (nonlinear) dẫn đến sự thoái trào trong quá trình nghiên cứu học sâu.

Bước qua giai đoạn thoái trào đầu tiên của việc nghiên cứu học sâu là sự khai sinh của một định hướng nghiên cứu dựa trên thuyết kết nối (**Connectivism**) vào năm 80 của thập kỉ trước. Tư tưởng cốt lõi của thuyết kết nối là khi có một lượng lớn các đơn vị tính toán đơn giản được kết nối với nhau thành một mạng lưới, chúng có thể thông minh. Khẳng định này đúng với các tế bào trong hệ thần kinh sinh học lẫn các đơn vị ẩn trong các mô hình tính toán. Và đây cũng là phong trào đã áp dụng thành công và lan rộng việc sử dụng giải thuật lan truyền ngược – một giải thuật sẽ được đề cập trong đề án này.

Một thành tựu khác của giai đoạn này là việc đưa ra mạng LTSM (long short-term memory network) – một mạng được ứng dụng rộng rãi cho đến tận bây giờ. Nhưng rồi, đó cũng là hồi kết cho giai đoạn này. Cái kết này là điều không thể tránh khỏi bởi những tiến bộ trong các ngành học máy khác. Học sâu lại rơi vào một giai đoạn thoái trào cho

đến năm 2006. Và đây là lúc mà một đột phá mới, khai sinh cho một hướng đi mới với người khơi nguồn là Geoffrey Hinton với nghiên cứu về mạng niềm tin đa tầng (deep belief network) và chiến lược huấn luyện trước tham lam theo tầng (greedy layer – wise pre-training). Từ đó, phong trào nghiên cứu này đã phát triển và đem lại cho học sâu sự thành công như hiện tại.

Tuy trải qua nhiều giai đoạn thăng trầm và mang cái tên khiến nhiều người liên tưởng đến khoa học thần kinh nhưng học sâu đã vươn đến một góc độ mà ở đó bài toán được giải quyết bởi sự gợi ý của khoa học thần kinh, thống kê và toán ứng dụng. Và nhìn rộng hơn, học sâu vẫn còn nhiều lĩnh vực để ứng dụng cũng như những sự phát triển xa hơn.

## II. Định nghĩa về mạng neuron và các thành phần của mạng học sâu:

### 1. Mạng neuron:

Về định nghĩa thì mạng neuron là một tập hợp của rất nhiều node mỗi node sẽ mang một giá trị hoặc một hàm. Các node này được sắp xếp với nhau thành từng layer. Từ “deep” trong từ “Deep learning” ý chỉ độ sâu của các layer được xây dựng này.

Thật thú vị nhưng mô hình neural network đầu tiên được gọi với cái tên Perceptron. Một mạng với nhiều đầu ra, một node để tính toán và đưa ra kết quả output. Các mạng neuron hiện nay đều được tạo nên với sự tổng hợp của nhiều lớp và mỗi lớp tại có nhiều node (ta có thể coi những mạng neuron hiện tại là tổng hợp của nhiều perceptron).

Mạng neuron hiện nay được xây dựng dựa trên định lý xấp xỉ phổ quát (Universal Approximation Theorem) cho rằng: “Bất kỳ một *hàm số thực* liên tục trên tập [compact](#), đều có thể được xấp xỉ bằng một *Feed-forward DNN* một lớp với *hữu hạn* số “neuron”, nếu “*hàm kích hoạt*”, *activation function*, đảm bảo được các điều kiện sau trên tập xác định: (1) *bị chặn*, (2) *liên tục*, (3) *đơn điệu tăng*.”

### 2. Các thành phần của mạng neuron:

#### a) Input layer:

Lớp đầu vào của các mạng neuron nhìn chung chỉ là tập hợp các node trên đó có một giá trị được trích xuất và biến đổi từ dữ liệu đầu vào

b) Output layer:

Lớp đầu ra của mạng neuron thường sẽ là các số nhị phân (0 hoặc 1) hay những số thực biểu diễn xác suất của một đang vị nào đó trong với một lớp.

c) Các layer ẩn:

Các lớp ẩn nằm giữa lớp đầu ra và lớp đầu vào. Nhiệm vụ chính của chúng là tiếp nhận các giá trị từ lớp trước đó sau đó truyền vào các unit (các neuron hay node trong một layer) nơi có các hàm kích hoạt để đưa ra một giá trị cho lớp tiếp theo.

d) Các unit:

Mỗi node trong một layer sẽ mang một giá trị hoặc một hàm kích hoạt hay chỉ là một hàm tuyến tính nào đó. Sự hiệu quả của một mạng neuron lại đến từ các node này. Các node này còn có những tên gọi khác như unit hay neuron.

e) Trọng số và độ lệch (**weights and bias**):

Một trong những ưu thế của mạng neuron là khả năng tự động học cũng như tự động điều chỉnh các hệ số. Và để có được khả năng này chúng ta cần các trọng số (weights) để điều chỉnh khả năng kết nối giữa các unit thuộc các lớp khác nhau trong khi độ lệch(bias) lại đem đến khả năng hiệu chỉnh trong các unit (node).

f) Các hàm kích hoạt (**activation function**):

Về định nghĩa thì các hàm kích hoạt quyết định xem có nên kích hoạt một unit nào đó bằng cách tính tổng trọng số được truyền vào unit (+ độ lệch). Khi một unit được kích hoạt thì có thể hiểu là ta đang cố gắng đưa tính phi tuyến tính vào trong mạng neuron.

3. Quá trình đào tạo (training neural network):

a) Feedforward:

Quá trình truyền thẳng (Feedforward) là quá trình ta thực hiện tính toán trên mạng một cách tuần tự từ lớp đầu vào đến lớp ẩn rồi đến lớp đầu ra. Ta cũng tính toán từng tự các unit trên lớp rồi mới thực hiện đến unit tiếp theo cho đến khi hoàn thiện tính toán tất cả các unit thì sẽ đến lớp tiếp theo.

b) Backpropagation:

Quá trình lan truyền ngược (Backpropagation) là quá trình ta tính toán và hiệu chỉnh các tham số của unit theo chiều từ các lớp đầu ra về lại lớp đầu vào. Giải thuật được dùng trong quá trình này là Gradient descent để tính toán đạo hàm và hiệu chỉnh từng unit.

Các định nghĩa đã được nói đến ở trên sẽ được hiện thực hóa ở phần thực nghiệm ứng dụng mạng neuron giải quyết bài toán phân loại chữ số viết tay trình bày ở phần V của đồ án.

### III. Bài toán phân loại chữ viết tay và hiệu quả của các thuật toán học máy khác so với neural network:

Ta thường nghe về hiệu quả của học sâu và mạng neuron và để minh chứng cho điều đó, tôi sẽ trình bày một bài toán phân loại chữ viết tay trên tập dữ liệu MNIST (Modified National Institute of Standards and Technology database) và trích dẫn một bảng so sánh độ chính xác với các thuật toán.

Type	Classifier	Distortion	Preprocessing	Error rate (%)
Convolutional neural network	Committee of 20 CNNs with Squeeze-and-Excitation Networks <sup>[35]</sup>	None	Data augmentation	0.17 <sup>[36]</sup>
Random Multimodel Deep Learning (RMDL)	10 NN-10 RNN - 10 CNN	None	None	0.18 <sup>[25]</sup>
Convolutional neural network	Committee of 5 CNNs, 6-layer 784-50-100-500-1000-10-10	None	Expansion of the training data	0.21 <sup>[22][23]</sup>
Convolutional neural network	Committee of 35 CNNs, 1-20-P-40-P-150-10	Elastic distortions	Width normalizations	0.23 <sup>[15]</sup>
Convolutional neural network (CNN)	13-layer 64-128(5x)-256(3x)-512-2048-256-256-10	None	None	0.25 <sup>[20]</sup>
Convolutional neural network	6-layer 784-50-100-500-1000-10-10	None	Expansion of the training data	0.27 <sup>[34]</sup>
Convolutional neural network (CNN)	6-layer 784-40-80-500-1000-2000-10	None	Expansion of the training data	0.31 <sup>[33]</sup>
Deep neural network	6-layer 784-2500-2000-1500-1000-500-10	Elastic distortions	None	0.35 <sup>[32]</sup>
K-Nearest Neighbors	K-NN with non-linear deformation (P2DHMDM)	None	Shiftable edges	0.52 <sup>[26]</sup>
Support-vector machine (SVM)	Virtual SVM, deg-9 poly, 2-pixel jittered	None	Deskewing	0.56 <sup>[30]</sup>
Deep neural network	2-layer 784-800-10	Elastic distortions	None	0.7 <sup>[31]</sup>
Boosted Stumps	Product of stumps on Haar features	None	Haar features	0.87 <sup>[27]</sup>
Deep neural network (DNN)	2-layer 784-800-10	None	None	1.6 <sup>[31]</sup>
Random Forest	Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC) <sup>[28]</sup>	None	Simple statistical pixel importance	2.8 <sup>[29]</sup>
Non-linear classifier	40 PCA + quadratic classifier	None	None	3.3 <sup>[10]</sup>
Linear classifier	Pairwise linear classifier	None	Deskewing	7.6 <sup>[10]</sup>

Figure 1 Bảng so sánh độ chính xác giữa các thuật toán trên bộ data MNIST

Có thể thấy các thuật toán sử dụng mạng thần kinh đem lại hiệu quả từ cao đến rất cao và chúng sẽ càng hiệu quả với nhiều lớp hơn. Và không chỉ có nhiều lớp hơn, các lớp cũng chứa nhiều unit hơn. Và tại đây tôi có đặt ra hai câu hỏi:

1. Tại sao ta không thử mở rộng mạng với nhiều layer và thêm nhiều unit trong mỗi layer? Liệu chúng có đem lại độ sai lệch nhỏ hơn không?
2. Đâu là nguyên nhân đem lại sự hiệu quả cho các mạng neuron?

Có rất nhiều lớp bài toán đã và đang ứng dụng mạng neuron để đem lại hiệu quả rất cao và bài toán phân loại hình ảnh chữ viết tay chỉ là một trong số những bài toán đánh dấu sự thành công của mạng neuron. Tuy nhiên ta sẽ cần tìm hiểu nhiều hơn để hình dung lý do mà mạng neuron hiệu quả. Và tôi sẽ trả lời câu hỏi số 2 đã được nêu ở trên trong phần tiếp theo của báo cáo.

#### **IV. Giải thích lý do cần mạng neural và các activity function**

##### **1. Nhược điểm của các thuật toán học máy:**

Nhìn chung các thuật toán học máy được liệt kê trên bảng so sánh độ sai lệch ở phần trước có nhược điểm chung là khó phù hợp với những tập dữ liệu lớn cũng như rất nhạy cảm với những dữ liệu nhiễu. Riêng với thuật toán Random forest thì nhược điểm là thích ứng tốt với tập dữ liệu lớn nhưng có rất nhiều tham số cần hiệu chỉnh dẫn đến tốn nhiều chi phí đào tạo.

##### **2. Sự ưu thế của mạng neuron:**

Ưu điểm lớn nhất của mạng neuron so với các thuật toán học máy trước đó hẳn là việc mạng neuron có khả năng tự học và tự động điều chỉnh tham số qua phương pháp lan truyền ngược. Nhờ tính rộng trong mỗi layer và tính sâu trong mỗi mạng dẫn đến khả năng xử lý được cái dữ liệu không hoàn chỉnh cũng là một ưu thế của mạng. Và may mắn hơn khi nhờ phát kiến xoay quanh việc phát triển phần cứng, việc phát triển và vận hành các mạng neuron cũng đã trở nên dễ dàng hơn.

#### **V. Bài toán phân loại chữ số viết tay trên tập dữ liệu MNIST:**

##### **1. Phát biểu bài toán:**

Bài toán phân loại chữ viết tay trên tập dữ liệu MNIST với dữ liệu đầu vào là một tập các input với mỗi một giá trị đầu vào là một mảng có kích thước là một mảng 784 biểu diễn cho một ảnh có kích thước là  $28 \times 28$  pixels.

Tập dữ liệu với kích thước tập đào tạo là 60000 ảnh và 10000 ảnh cho phần kiểm tra độ chính xác. Phần nhãn của tập đào tạo sẽ được lưu trữ tại ô số 785 của hàng là một số trong phạm vi từ 0 đến 9.

Yêu cầu bài toán là với mỗi ảnh trong tập kiểm tra ta sẽ đưa ra một giá trị từ 0 đến 9 đại diện cho giá trị ta sẽ dự đoán nhãn của ảnh sẽ mô tả số đấy.

## 2. Giới thiệu về một số đặc trưng của tập dữ liệu:

Tập dữ liệu có thể được lưu trữ dưới dạng file với định dạng .csv hoặc truy cập tập dữ liệu thông qua thư viện mnist của keras.datasets. Mỗi pixel của ảnh là một số có giá trị từ 0 đến 255 biểu diễn mức độ sáng của pixel trên thang ảnh xám.

## 3. Định hướng giải quyết bài toán:

Bộ dữ liệu sau khi được load từ server keras sẽ được tinh chỉnh để mỗi pixel chỉ mang 2 giá trị là 0 hoặc 1. Sau quá trình tiền xử lý dữ liệu ta sẽ kiến trúc mạng và tiến hành quá trình training cũng như hiệu chỉnh mạng.

## 4. Cấu trúc mạng được đề xuất để giải quyết bài toán:

Kiến trúc mạng được sử dụng sẽ là một mạng với:

- Một lớp đầu vào kích thước 60000 x 784
- Một lớp tuyến tính với kích thước 784 x 40
- Một hàm kích hoạt, hàm được chọn ở đây là hàm Tanh
- Một lớp tuyến tính với kích thước 40x10
- Một hàm kích hoạt, hàm được chọn là hàm Tank

## 5. Thuật toán hiệu chỉnh và độ đo:

Thuật toán hiệu chỉnh của mạng vẫn dựa trên Gradient descent với hàm mất mát được chọn là MSE (mean squared error) có công thức là:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

$n$  = number of data points

$Y_i$  = observed values

$\hat{Y}_i$  = predicted values



## 6. Nghiệm thu kết quả:

Thử nghiệm trên tập data MNIST với tập học là 60000 và tập kiểm nghiệm là 10000.

Các thông số khi training:

- Loss function: MSE
- Epoch: 100
- Learning rate: 0.1

Kết quả: Giá trị loss trong quá trình học tập hội tụ từ 0.17416 ở epoch đầu tiên đến 0.0199 ở epoch 100. Accuracy trên tập thử nghiệm là 0.9066.

Kết quả và thông số của các lần thực nghiệm sau đã được ghi lại trong file experiment.pdf được đính với code trong link github.

Qua thực nghiệm ta thấy vì mạng vẫn còn nhỏ và nông dẫn đến cần hiệu chỉnh số lượng epoch lớn và tốc độ học vừa phải (0.1). Khi tăng learning rate thì mạng thực hiện học nhanh hơn nhưng chỉ số error ở mỗi lượt train trở nên cao hơn dẫn đến kết quả lúc test thấp hơn dự kiến.

## VI. Tổng kết:

Nhìn chung mạng neuron là một thành tựu của tiến bộ khoa học ở cả phần cứng cũng như những nghiên cứu về kiến trúc, lý thuyết học tập. Tuy mạng được trình bày ở trên chỉ mang tính chất mô tả một mạng cơ bản nhưng cũng phần nào trình bày được phương thức học, cải thiện của một mạng.

Vì lý do thời gian cũng như phạm vi kiến thức còn hạn hẹp nên tôi sẽ tiếp tục nghiên cứu thêm về những kiến trúc mạng đã và đang được ứng dụng rất hiệu quả hiện nay.

Note: Trong link github tôi có upload một file mnist\_conv với kiến trúc mạng được sử dụng là mạng CNN. Qua quá trình thực nghiệm thì với một mạng nhiều layer hơn như CNN, với khoảng 7 layer thì kết quả đã được đẩy lên tới 99% trong bộ data MNIST có 12k ảnh train và 2k ảnh test.

Link github đồ án: <https://github.com/UIT-21520373/Neural-network>