

SWishLU: Một phương pháp kích hoạt khác của hàm ReLU để nâng cao tính năng của mạng nơ-ron

Nguyễn Văn Trường* và Trần Duy Hưng
Trường Cao đẳng Công nghiệp Quốc phòng

TÓM TẮT

Trong bài viết này, dựa trên cơ sở nghiên cứu hàm số kích hoạt ReLU và các hàm số tương tự khác, chúng tôi đề xuất ra một hàm số kích hoạt mới mang tên SWishLU để tránh sự xuất hiện “nơ-ron chết” trong hàm số ReLU (Dying ReLU). SWishLU sử dụng hàm số Swish để kích hoạt phần âm của đầu vào, do đó hàm số này bao hàm tất cả các ưu điểm của hàm số kích hoạt ReLU và hàm số kích hoạt Swish. Hàm số này không những có thể tránh được sự xuất hiện của “nơ-ron chết” mà còn sử dụng hợp lý các thông tin của đầu vào âm để nâng cao hiệu quả nhận dạng của mạng nơ-ron. Kết quả thử nghiệm trên các tập dữ liệu MNIST, VNCD và CIFAR-10 cho ta thấy, độ nhận dạng chính xác của mạng nơ-ron tích chập Lenet-5 lần lượt đạt tới 99.18%, 82.82% và 63.81%, lần lượt cao hơn 0.09%, 2.08% và 0.74% so với hàm số ReLU.

Từ khóa: hàm số kích hoạt, mạng nơ-ron, hàm ReLU, hàm SWishLU

1. GIỚI THIỆU

Trong những năm gần đây, trí tuệ nhân tạo đã và đang được áp dụng rất thành công trong rất nhiều lĩnh vực như nhận dạng hình ảnh [1, 2], nhận dạng giọng nói [3, 4], phát hiện đối tượng [5, 6], thị giác máy tính [7] v.v... Một trong những nhân tố làm nên sự thành công đó chính là các hàm số kích hoạt, được dùng để kích hoạt tín hiệu của các nơ-ron thần kinh ở các tầng tích chập, tầng liên kết đầy đủ và tầng đầu ra, từ đó nâng cao đặc tính biểu đạt phi tuyến tính của mạng nơ-ron [8-10]. Trong những năm gần đây, hàm Sigmoid [11] và hàm ReLU [12] là hai hàm kích hoạt được các nhà khoa học, các nghiên cứu viên sử dụng phổ biến nhất trong các mạng nơ-ron hiện hành.

Hàm ReLU có công thức biểu đạt như công thức (1). Chúng ta có thể thấy rằng ReLU có các đặc điểm thừa thớt. Khi nơ-ron được kích hoạt bởi ReLU, chỉ một phần nơ-ron được kích hoạt, có thể tránh được hiện tượng giống nhau quá mức dễ xảy ra trong quá trình huấn luyện mạng

nơ-ron. Cụ thể, khi nơ-ron nhận được tín hiệu dương, ReLU không trải qua bất kỳ tính toán nào mà trực tiếp ánh xạ tới đầu ra để giảm đáng kể độ phức tạp trong tính toán, cải thiện tốc độ hội tụ của mạng nơ-ron. Tuy nhiên, ReLU thiết lập toàn bộ phần kích hoạt âm của các nơ-ron thành 0, do đó độ dốc của nơ-ron bán trục âm cũng luôn là 0. Điều này sẽ khiến lượng cập nhật cũng bằng 0, dẫn tới các nơ-ron sẽ không bao giờ được cập nhật và kích hoạt, hình thành nên các “nơ-ron chết”. Vì vậy, trong giai đoạn huấn luyện, các tham số của mạng chẳng hạn như tốc độ học tập phải được điều chỉnh cẩn thận để tránh các giá trị đầu vào của các nơ-ron đều nằm trong phần bán trục âm để tránh tình trạng “nơ-ron chết” ở trên.

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (1)$$

Trong vài năm qua, các nhà nghiên cứu đã đề xuất nhiều phương pháp cải tiến cho ReLU để

Tác giả liên hệ: TS. Nguyễn Văn Trường
Email: nvtruong.25890@gmail.com

giải quyết vấn đề nơ-ron chết như hàm Leaky ReLU (LReLU) có công thức biểu đạt như công thức (2) [13], Parametric ReLU (PReLU) có công thức biểu đạt như công thức (3) [14], hàm ELU có công thức biểu đạt như công thức (4) [15], hàm PReLU có công thức biểu đạt như công thức (5) [16], hàm Tanh ReLU [17], hàm Arctan ReLU có công thức biểu đạt như công thức (6) [18], hàm PoLU có công thức biểu đạt như công thức (7) [19], hàm V-shape ReLU có công thức biểu đạt như công thức (8) [20] v.v...

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \quad (2)$$

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ \alpha_i x_i, & \text{if } x_i \leq 0 \end{cases} \quad (3)$$

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases} \quad (4)$$

$$f(x) = \begin{cases} \frac{a}{b} x, & \text{if } x > 0 \\ \alpha(e^{\frac{x}{b}} - 1), & \text{if } x \leq 0 \end{cases}, \quad a, b > 0 \quad (5)$$

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \frac{2}{\pi} \arctan(x), & \text{if } x \leq 0 \end{cases} \quad (6)$$

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ -|x|^\alpha, & \text{if } x \leq 0 \end{cases} \quad (7)$$

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ -x, & \text{if } x \leq 0 \end{cases} \quad (8)$$

Nhằm mục đích giải quyết vấn đề “nơ-ron chết” do hàm ReLU gây ra, bài viết này đề xuất ra một hàm kích hoạt cải tiến của ReLU - gọi là SWishLU. Hàm này duy trì phần kích hoạt dương của hàm ReLU và sử dụng hàm kích hoạt SWish cho phần kích hoạt âm để tránh vấn đề “nơ-ron chết”. Đồng thời, các tham số của hàm đưa ra có thể điều chỉnh trong phần kích hoạt âm để tiến hành kiểm soát, khống chế phạm vi thay đổi của

hàm kích hoạt, cải thiện chức năng kích hoạt của SWishLU, tăng sự linh hoạt trong xử lý các nhiệm vụ khác nhau. Hàm SWishLU được đề xuất trong bài viết này được sử dụng để phân loại các bộ dữ liệu MNIST, VNCD và CIFAR-10 trong cấu trúc mạng nơ-ron tích chập Lenet-5. So với ReLU và các hàm được cải tiến được đề xuất trong những năm gần đây, hiệu quả nhận dạng của mạng Lenet-5 khi sử dụng hàm kích hoạt SWishLU đã có sự cải thiện nhất định.

2. HÀM KÍCH HOẠT SWishLU ĐỀ XUẤT

2.1. Hàm SWish

Hàm số kích hoạt SWish [21] có công thức như sau:

$$f(x) = \frac{x}{1 + e^{-x}} \quad (9)$$

$$= x \text{Sigmoid}(x)$$

Khi tín hiệu đầu vào nhỏ hơn 0, giá trị đầu ra được kích hoạt với giá trị khác 0 và có đặc tính không đơn điệu. Như vậy hàm SWish không chỉ giải quyết vấn đề nơ-ron đi vào trạng thái chết và giảm bớt vấn đề sai lệch giá trị trung bình ở đầu ra, mà còn có thể lợi dụng huấn luyện mạng nơ-ron để tìm ra giá trị tối ưu nhất. Khi tín hiệu đầu vào lớn hơn 0, đầu ra của hàm kích hoạt không bị giới hạn và tăng khi tín hiệu đầu vào x tăng; và khi biến x có xu hướng tiến đến dương vô cùng, giá trị đầu ra của hàm SWish tiến gần tới giá trị của x , do đó có hiệu quả cải thiện hiệu suất nhận dạng của mạng nơ-ron. Ngoài ra, không giống như hàm kích hoạt ReLU (đạo hàm của ReLU luôn là 0 hoặc 1), đạo hàm bậc một của hàm SWish có công thức biểu đạt như công thức (10) luôn thay đổi khi biến đầu vào thay đổi và giá trị tối đa của đạo hàm hàm SWish nhỏ hơn 1 và chứa giá trị âm, thuận lợi hơn cho việc truyền tải thông tin ở giai đoạn truyền dẫn ngược.

$$f'(x) = \frac{1}{1 + e^{-x}} + \frac{x e^{-x}}{(1 + e^{-x})^2} \quad (10)$$

$$= f(x) + \text{Sigmoid}(x)(1 - f(x))$$

Nhóm Brain đã thực hành thử nghiệm xác định hiệu quả kích hoạt của hàm SWish trên các

mạng nơ-ron với các tập dữ liệu khác nhau, kết quả cho thấy hàm SWish với ưu điểm chỉ có giới hạn dưới, không đơn điệu và phi tuyến tính đã cho ra kết quả bằng và ưu việt hơn so với các hàm kích hoạt khác [21].

2.2. Hàm SWishLU đề xuất

Trong bài viết này, xuất phát từ ý tưởng của hàm kích hoạt SWish, chúng tôi đề xuất một hàm số kích hoạt cải tiến của hàm ReLU mang tên là hàm SWishLU. Hàm số này có công thức như sau:

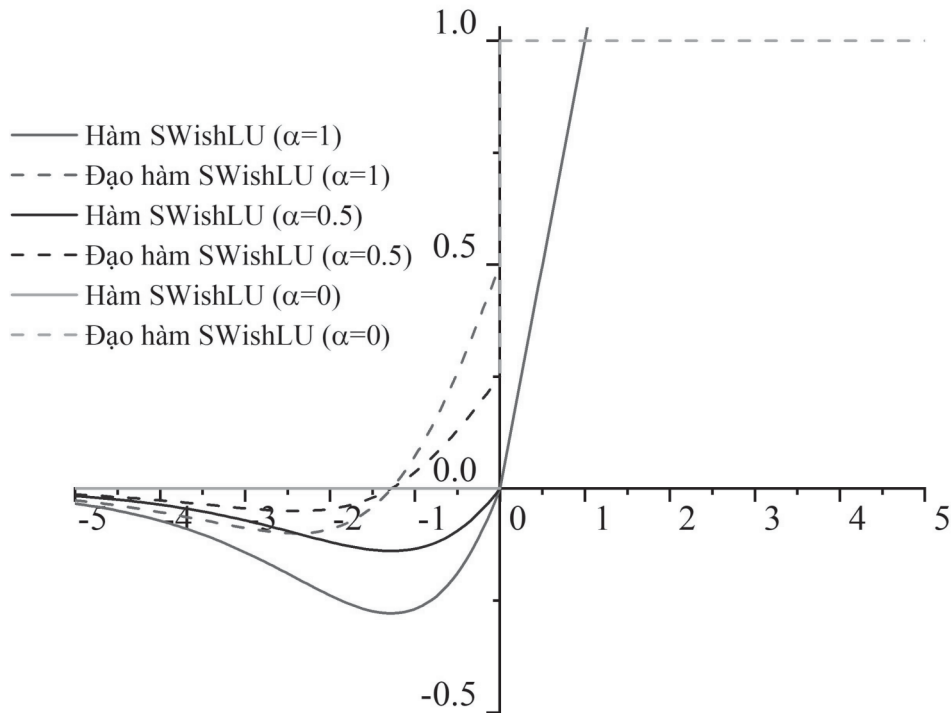
$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \frac{\alpha x}{1 + e^{-x}}, & \text{if } x \leq 0 \end{cases} \quad (11)$$

Trong đó, α là tham số có thể điều chỉnh được, có thể khống chế hệ số biến đổi của hàm số ở phần biến âm. Ở hàm số này, khi $x > 0$, chúng tôi duy trì hình thức kích hoạt của hàm ReLU để đảm bảo rằng số lượng tính toán của hàm kích hoạt được giảm thiểu nhất trong khi vẫn đảm bảo nâng cao được tính năng của hàm; khi $x < 0$, SWishLU sử dụng hình thức kích hoạt của hàm SWish đồng thời thêm vào tham số có thể điều

chỉnh α . Hình 1 cho thấy đường cong của hàm kích hoạt SWishLU dưới các giá trị tham số α khác nhau. Từ hình vẽ trên chúng ta có thể thấy tham số α trực tiếp ảnh hưởng đến hiệu quả kích hoạt ở phần âm của hàm SWishLU, trong đó α càng cao thì biên độ đầu ra của hàm số càng lớn; còn khi $\alpha = 0$ thì hàm SWishLU có công thức tương tự như hàm ReLU.

Từ những phân tích trên, chúng ta có thể rút ra kết luận hàm kích hoạt SWishLU có những đặc điểm chính sau:

- (1) Phần kích hoạt dương đầu ra sử dụng luôn giá trị của đầu vào, do đó đơn giản được phép toán;
- (2) Phần kích hoạt âm sử dụng hàm SWish thay thế, góp phần tránh khỏi sự xuất hiện của “nơ-ron chết”, đồng thời làm cho hàm SWishLU có đặc tính phi đơn điệu giúp nâng cao hơn nữa năng lực nhận dạng của mạng nơ-ron;
- (3) Hàm SWishLU đã thêm tham số có thể điều chỉnh α vào phần kích hoạt âm, thông qua thay đổi giá trị của α có thể thay đổi công thức của hàm số, làm tăng tính linh hoạt của hàm số này trong nhận dạng các nhiệm vụ khác nhau.



Hình 1. Đường biểu thị của hàm SwishLU với các tham số α khác nhau

2.3. Phương thức truyền dẫn ngược dựa trên hàm SWishLU

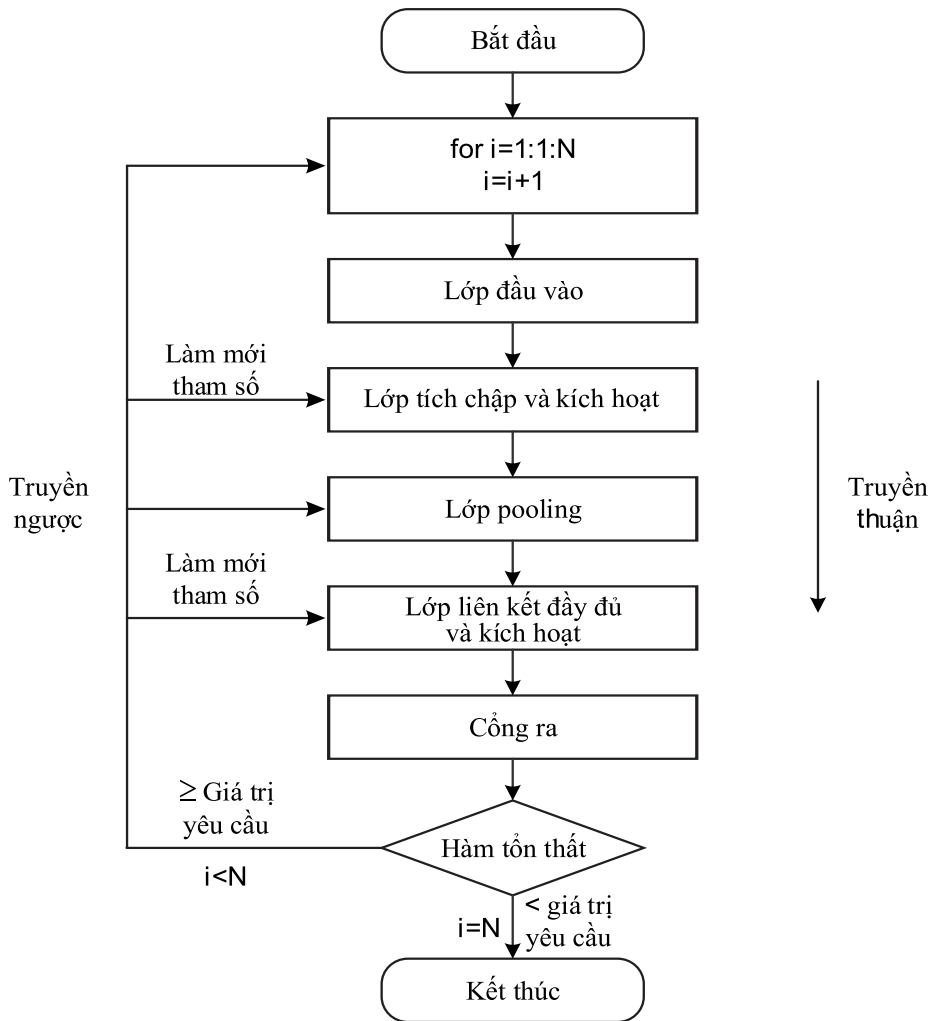
Quy trình huấn luyện của một mạng nơ-ron được thực hiện qua hai bước gồm truyền dẫn thuận (đầu vào - đầu ra) và truyền dẫn ngược (đầu ra - đầu vào). Phương thức truyền dẫn ngược của mạng nơ-ron như Hình 2 thể hiện.

Quá trình truyền dẫn ngược là quá trình truyền sai số đầu ra quay ngược lại các nơ-ron của các

tầng cho tới tầng đầu vào với mục đích là thay đổi giá trị của các trọng số từ đó làm giảm giá trị sai số của đầu ra. Phương pháp truyền dẫn ngược cụ thể như sau:

$$\frac{\partial E}{\partial W^l} = (f'_i(z^l) ((W^{l+1})^T \delta^{l+1})) (x^{l-1})^T \quad (12)$$

$$\frac{\partial E}{\partial b^l} = f'_i(z^l) ((W^{l+1})^T \delta^{l+1}) \quad (13)$$



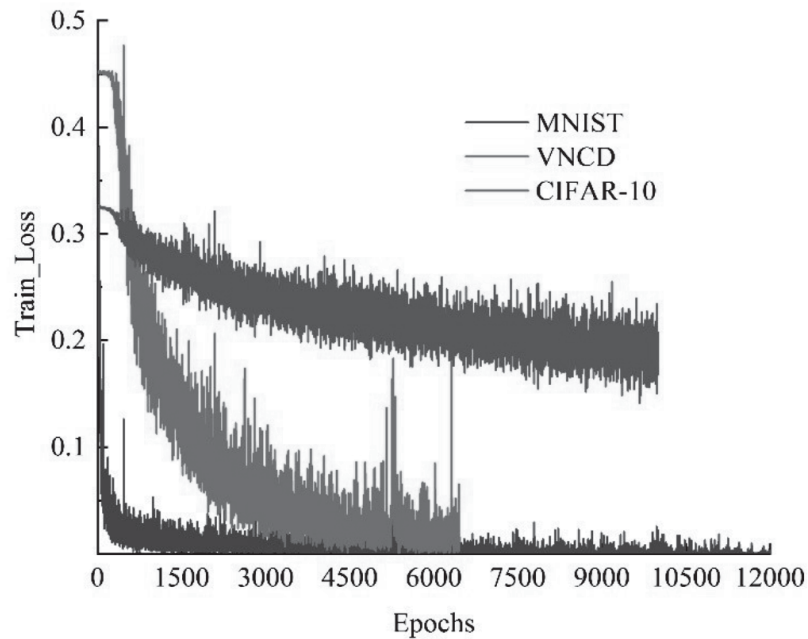
Hình 2. Quá trình truyền dẫn ngược của mạng nơ-ron

Có thể thấy, hàm kích hoạt là một phần không thể thiếu của truyền dẫn ngược và đạo hàm đầu tiên của hàm ảnh hưởng trực tiếp đến hiệu quả của truyền dẫn ngược. Từ công thức (11) ta có thể tính toán ra công thức đạo hàm đối với biến z của hàm SWishLU như sau:

$$f'_i(z^l) = \begin{cases} 1, & z^l > 0 \\ f(z^l) + Sigmoid(z^l)(\alpha - f(z^l)), & z^l \leq 0 \end{cases} \quad (14)$$

3. KẾT QUẢ THỬ NGHIỆM VÀ PHÂN TÍCH

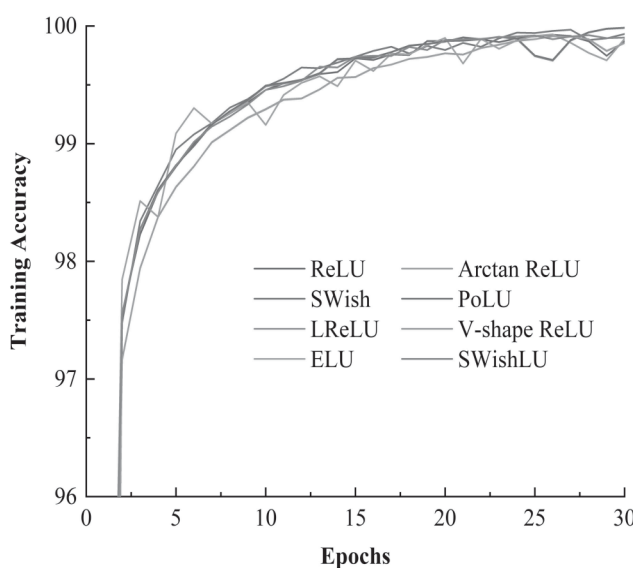
Trong bài viết này, chúng tôi thông qua thử nghiệm nhận dạng các tập dữ liệu MNIST, VNCD, CIFAR-10 để chứng minh hàm SWishLU có hiệu quả nhận dạng ưu việt hơn hàm ReLU và các hàm cải tiến trước của nó. Hình 3 cho thấy kết quả sai số ở tầng đầu ra giảm dần theo số lần huấn luyện của mạng nơ-ron tích chập Lenet-5 khi sử dụng hàm SWishLU kích hoạt nơ-ron.



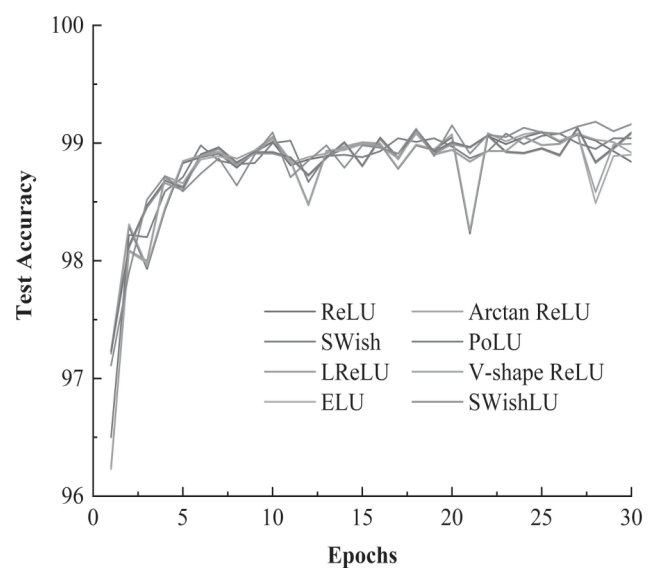
Hình 3. Sai số đầu ra trong huấn luyện sử dụng mạng Lenet-5 và hàm SwishLU

Dựa trên cấu trúc mô hình mạng nơ-ron tích chập Lenet-5, bài viết này tiến hành thử nghiệm so sánh hiệu quả nhận dạng của mạng nơ-ron trên các tập dữ liệu MNIST, VNCD và CIFAR-10 bằng cách thay đổi hàm kích hoạt của mạng nơ-ron. Các hàm kích hoạt khác nhau ảnh hưởng đến độ chính xác của quá trình huấn luyện và kiểm tra của mạng nơ-ron khi xử lý tập dữ liệu MNIST được thể hiện trong Hình 4. Sau

30 lần huấn luyện, độ chính xác trong nhận dạng được hiển thị trong Bảng 1, trong đó "Nil" có nghĩa là không có cấu hình tham số trong hàm kích hoạt. Kết quả cho thấy hiệu quả nhận dạng của mạng nơ-ron khi sử dụng hàm kích hoạt SWishLU cao hơn so với khi sử dụng các hàm kích hoạt khác. Tỷ lệ nhận dạng chính xác khi sử dụng hàm SWishLU là 99.18%, cải thiện được 0.09% so với mạng nơ-ron sử dụng hàm ReLU.



(a) Huấn luyện



(b) Kiểm tra

Hình 4. Kết quả nhận dạng tập MNIST với các hàm kích hoạt khác nhau

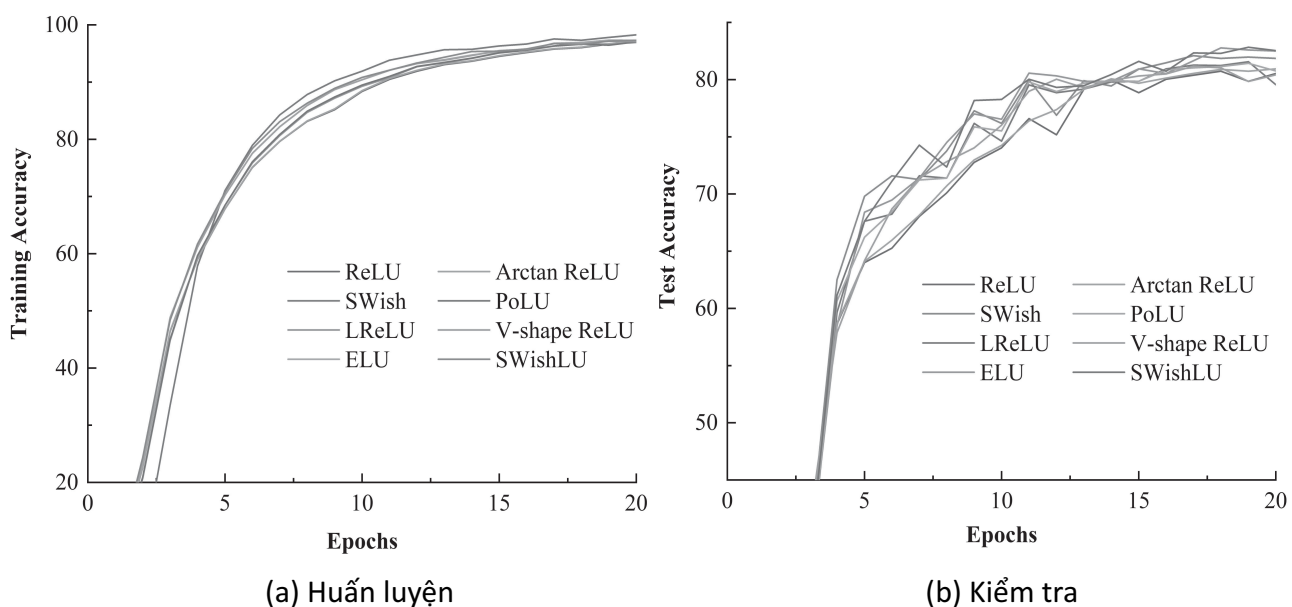
Bảng 1. Kết quả nhận dạng tập MNIST với các hàm kích hoạt khác nhau

Hàm kích hoạt	Tham số	Độ chính xác trong huấn luyện (%)	Độ chính xác trong kiểm tra (%)
ReLU [12]	Nil	99.92	99.09
LReLU[13]	0.01	99.98	99.12
ELU[15]	Nil	99.91	99.09
Arctan ReLU[18]	Nil	99.93	99.10
PoLU[19]	0.5	99.98	99.13
V-shape ReLU[20]	Nil	99.91	99.08
SWish[21]	Nil	99.93	99.08
SWishLU	0.1	99.78	99.18

Đối với tập dữ liệu VNCD, trong thử nghiệm này chúng tôi tiến hành huấn luyện trong 20 lần. Kết quả nhận dạng của mạng Lenet-5 khi sử dụng các hàm kích hoạt khác nhau được thể hiện trong Hình 5 và Bảng 2. Từ kết quả này ta thấy được, tương tự như nhận dạng trên tập MNIST, mạng Lenet-5 sử dụng hàm SWishLU kích hoạt nơ-ron thu được kết quả cao hơn so với sử dụng các hàm kích hoạt khác. Cụ thể, tỷ lệ nhận dạng chính xác thu được khi sử dụng hàm SWishLU là 82.82%, cải thiện được 2.08%

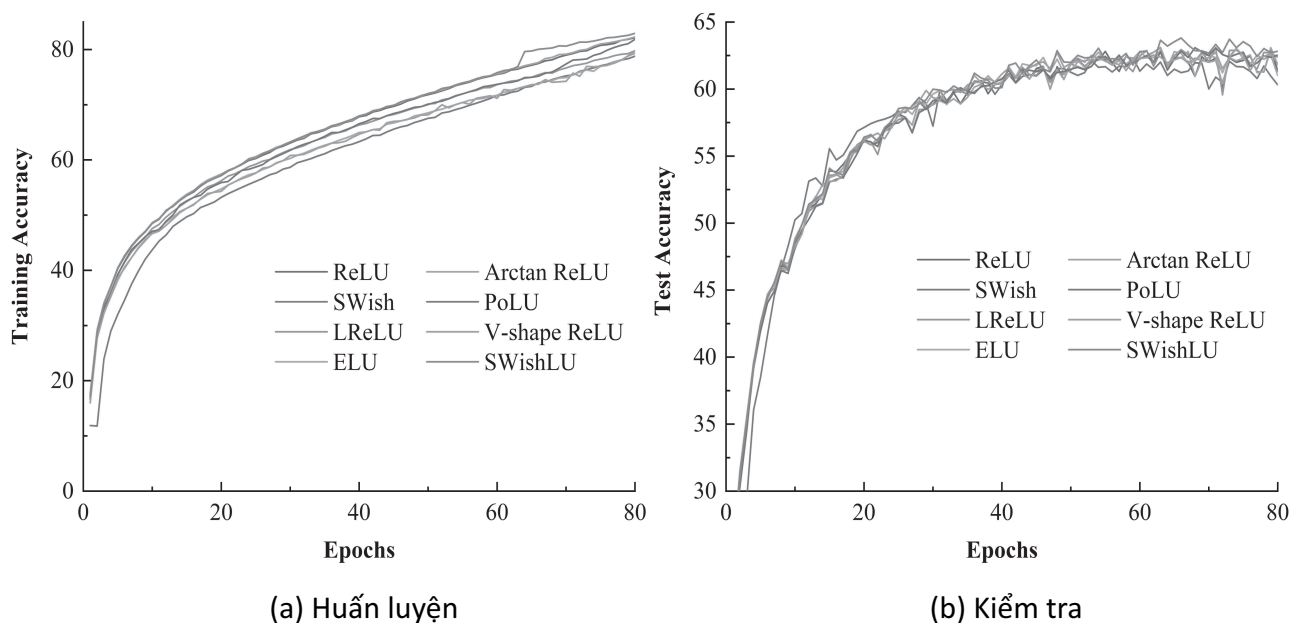
so với kết quả khi sử dụng hàm ReLU.

Hình 6 và Bảng 3 hiển thị kết quả nhận dạng tập CIFAR-10 của mạng Lenet-5 khi sử dụng hàm SWishLU và các hàm kích hoạt hiện hành khác sau 80 lần huấn luyện. Từ Bảng 3 ta thấy kết quả nhận dạng khi sử dụng hàm SWishLU kích hoạt đạt hiệu quả cao nhất là 61.38% cao hơn so với sử dụng các hàm kích hoạt hiện hành khác, đặc biệt là cao hơn 0.74% so với kết quả nhận dạng khi sử dụng hàm ReLU.

**Hình 5.** Kết quả nhận dạng tập VNCD với các hàm kích hoạt khác nhau

Bảng 2. Kết quả nhận dạng trên tập VNCD với các hàm kích hoạt khác nhau

Hàm kích hoạt	Tham số	Độ chính xác trong huấn luyện (%)	Độ chính xác trong kiểm tra (%)
ReLU[12]	Nil	96.94	80.74
LReLU[13]	0.01	97.25	81.56
ELU[15]	Nil	97.34	82.59
Arctan ReLU[18]	Nil	96.99	80.94
PoLU[19]	0.5	97.02	81.43
V-shape ReLU[20]	Nil	97.11	81.17
SWish[21]	Nil	98.24	82.08
SWishLU	0.1	97.25	82.82



Hình 6. Kết quả nhận dạng trên tập CIFAR-10 với các hàm kích hoạt khác nhau

Bảng 3. Kết quả nhận dạng tập CIFAR-10 với các hàm kích hoạt khác nhau

Hàm kích hoạt	Tham số	Độ chính xác trong huấn luyện (%)	Độ chính xác trong kiểm tra (%)
ReLU[12]	Nil	82.13	63.07
LReLU[13]	0.01	79.78	63.10
ELU[15]	Nil	79.27	63.04
Arctan ReLU[18]	Nil	82.29	63.11
PoLU[19]	0.5	81.84	63.09

V-shape ReLU[20]	Nil	79.68	63.03
SWish[21]	Nil	78.80	62.32
SWishLU	0.1	82.93	63.81

4. KẾT LUẬN

Trong bài viết này, chúng tôi đã thảo luận vấn đề “nơ-ron chết” của mạng nơ-ron khi sử dụng hàm kích hoạt ReLU. Hàm số mà chúng tôi đề xuất ra với tên gọi là SWishLU đã sử dụng hàm SWish thay thế phần kích hoạt âm của ReLU ban đầu và một siêu tham số có thể điều chỉnh được thêm vào phần kích hoạt âm đã có thể loại trừ được vấn đề “nơ-ron chết” trên cũng như cải

thiện tính năng nhận dạng của mạng nơ-ron. Kết quả thử nghiệm sử dụng mạng nơ-ron tích chập Lenet-5 nhận dạng các tập dữ liệu MNIST, VNCD và CIFAR-10 đã chứng minh rằng sử dụng hàm kích hoạt SWishLU có hiệu quả làm tăng độ nhận dạng chính xác của mạng nơ-ron trên ba tập dữ liệu trên lần lượt là 99.18%, 82.82% và 63.81%, lần lượt tăng 0.09%, 2.08% và 0.74% so với kết quả nhận dạng sử dụng hàm ReLU.

TÀI LIỆU THAM KHẢO

- [1] Q. Chen, J. Xu, and V. Koltun, “Fast image processing with fully-convolutional networks,” *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2497-2506.
- [2] L.Q. Liu, C.H. Shen and A.V. Hengel, “Cross-Convolutional-Layer Pooling for Image Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2305-2313, Nov. 2017.
- [3] Y.M. Qian, M.X. Bi, T. Tan and K. Yu, “Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 2263-2276, Dec. 2016.
- [4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.
- [5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.
- [6] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 142-158, 2016.
- [7] N. Akhtar and A. Mian, “Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey,” *IEEE Access*, vol. 6, pp. 14410-14430, 2018.
- [8] S.S. Liew, M. Khalil-Hani and R. Bakhteri, “Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems,” *Neurocomputing*, vol. 216, pp. 718-734, 2016.
- [9] Ö.F. Ertuğrul, “A novel type of activation function in artificial neural networks: Trained activation function,” *Neural Networks*, vol. 99, pp. 148-157, 2018.
- [10] C.E. Nwankpa, W. Ijomah, A. Gachagan and

S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," *ArXiv e-prints*, pp. 1-20, 2018.

[11] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, pp. 251-257, 1991.

[12] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," *In Proc. 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807-814.

[13] A.L. Maas and A.Y. Hannun, "Rectifier nonlinearities improve neural network acoustic models," *In Proc. International Conference on Machine Learning*, 2013.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026-1034.

[15] D.A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *International Conference on Learning Representations*, 2016.

[16] L. Trottier, P. Giguere and B. Chaib-draa, "Parametric Exponential Linear Unit for Deep Convolutional Neural Networks," *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 207-214.

[17] 张涛, 杨剑和宋文爱, "关于改进的激活函数TReLU的研究," *小型微型计算机系统*, vol.40, pp. 58-63, 2019.

[18] 许赞杰和徐菲菲, "基于ArcReLU函数的神经网络激活函数优化研究," *数据采集与处理*, vol.34, pp. 517-529, 2019.

[19] 骆训浩和李培华, "一种改进的基于幂线性单元的激活函数," *计算机应用研究*, vol. 36, pp. 3145-3147, 2019.

[20] H. Hu, "vReLU Activation Functions for Artificial Neural Networks," *14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2018, pp. 856-860.

[21] P. Ramachandran, B. Zoph and V.Q. Le, "Swish: a Self-Gated activation function," *arXiv: Neural and Evolutionary Computing*, 2017.

SWishLU: Another activation approach on ReLU function to improve neural network performance

Nguyen Van Truong* and Tran Duy Hung

ABSTRACT

Based on the study of ReLU and its existing modified methods, SWishLU, an improved method of ReLU activation function is proposed to avoid neuronal death caused by ReLU (dying ReLU) in this paper. This function is activated by Swish function in the negative axis part, so the function has the advantages of both Swish and ReLU. SWishLU function not only can avoid the dying ReLU but also reasonably apply the negative input information of the network to improve the network performance. The experimental results show that the Lenet-5 convolutional neural network using

SwishLU function achieves high accuracy of 99.18%, 82.82% and 63.81% in MNIST, VNCD and CIFAR-10 datasets respectively, which are 0.09%, 2.08% and 0.74% higher than ReLU function.

Keywords: *activation function, neural network, ReLU function, SWishLU function*

Received: 04/08/2021

Revised: 29/10/2021

Accepted for publication: 11/11/2021