NGHIÊN CỨU ỨNG DỤNG GIẢI THUẬT DI TRUYỀN KẾT HỢP VỚI THUẬT TOÁN " VƯỢT KHE" ĐỂ CẢI TIẾN QUÁ TRÌNH HỌC CỦA MẠNG NEURAL MLP CÓ MẶT LỖI ĐẶC BIỆT

RESEARCH ON THE APPLOCATION OF GENETIC ALGORITHM COMBINED WITH THE "CLEFT-OVERSTEP" ALGORITHM FOR IMPROVING LEARNING PROCESS OF MLP NEURAL NETWORK WITH SPECIAL ERROR SURFACE

Nguyễn Thị Thanh Nga

Trường Đại học Kỹ thuật Công nghiệp Thái Nguyên

TÓM TẮT

Sự thành công của một mạng neural nhân tạo phụ thuộc rất lớn vào quá trình luyện mạng. Các kỹ thuật luyện mạng dựa trên gradient đã phần nào thỏa mãn và được sử dụng rất nhiều trong thực tế. Tuy nhiên, trong một số trường hợp khi mặt lỗi đặc biệt có dạng lòng khe, các thuật toán này còn chậm và gặp phải vấn đề cực trị địa phương. Trong bài báo này, các tác giả đề xuất việc sử dụng giải thuật di truyền kết hợp với thuật toán "vượt khe" để cải tiến quá trình luyện mạng neural có mặt lỗi đặc biệt và minh họa thông qua ứng dụng nhận dạng chữ. Trước tiên, một mạng neural nhân tạo MLP có mặt lỗi dạng lòng khe được luyện với các thuật toán lan truyền ngược, các kết quả luyện mạng được đánh giá. Tiếp đó, bài báo mô tả việc sử dụng phương pháp mới để cải tiến quá trình luyện mạng neural theo hai tiêu chí: tốc độ hội tụ và độ chính xác. Việc cài đặt mạng cũng như các kết quả có được đều thực hiện trên môi trường Matlab.

The success of an artificial neural network depends much on the training phase. Techniques for training neural network based on gradient are partially satisfying and are widely used in practice. However, in several cases which has special error surface similar to a deep cleft, these algorithms seem to work slowly and encounter local extreme values. Authors of this paper propose the use of genetic algorithm in combination with the "cleft-overstep" algorithm to improve the training process of neural network which has special error surface and illustrate this usage through a simple application in text recognition. First, An MLP artificial neural network with cleft-similar error surface is trained using back propagation algorithm and the results are analyzed. Next, the paper describes the usage of the proposed method to improve the training process of neural network on two aspects: correctness and rate of convergence. Implementation is conducted in and results obtained from Matlab environment.

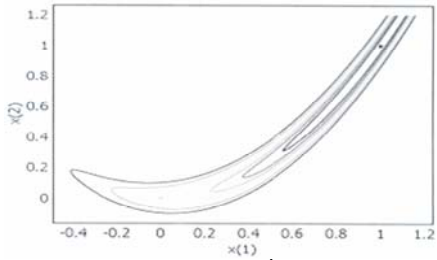
Keywords: MPL, Genetic Algorithm, cleft-overstep Algorithm, Character recognition.

1. Đặt vấn đề

Quá trình luyện mạng nơron thực chất là giải bài toán tối ưu nhằm cập nhật các trọng số sao cho hàm lỗi đạt cực tiểu, hoặc nhỏ hơn một giá trị cho phép nào đó.

Thuật toán hiện nay thường được sử dụng trong quá trình luyện mạng nơron là thuật toán gradien liên hợp hay thuật toán Levenberg – Marquardt với kỹ thuật lan truyền ngược và còn có thể gọi là thuật toán lan truyền ngược. Việc tìm kiếm thuật toán tối ưu nhằm tối thiểu hóa thời gian hội tụ cũng như thoát khỏi cực tiểu yếu, cực tiểu cục bộ đều được xuất phát điểm từ vấn đề nghiên cứu đặc điểm của mặt sai số hay còn gọi là mặt chất lượng, mặt lỗi. Một dạng mặt sai số để nhiều nhà khoa học quan tâm là có tính đa trị, tính kéo dãn trong không gian tham số,.... Bằng thuật toán lan truyền ngược giảm dốc nhất sẽ có vấn đề về sự hội tụ đối với mặt sai số phức tạp dạng lòng khe, tức là mặt sai số mà các đường đồng mức bị kéo

dài, cong tạo thành khe và trục khe và độ dốc có thể thay đổi ở một dải rộng trong các vùng khác nhau của không gian tham số.



0	0	1	1	0
0	1	0	0	0
1	0	0	0	0
1	1	1	1	0
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

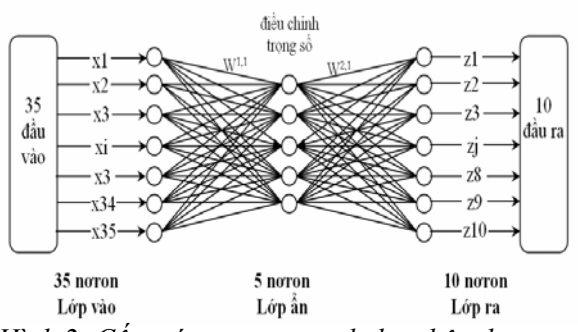
Hình 1: Mặt sai số dạng lòng khe và ví dụ về nhận dạng chữ viết

Bài báo trình bày về giải thuật di truyền(GA), kết hợp với thuật toán "vượt khe" để chế ngự quỹ đạo và rút ngắn thời gian của quá trình tìm kiếm tối ưu với mặt sai số phức tạp dạng lòng khe.

Để minh họa, nhóm tác giả đề xuất cấu trúc mạng nơ ron để nhận dạng các chữ số: 0, 1, 2, ...,9. Trong đó hàm sigmoid được sử dụng với mục đích dễ sinh ra mặt sai số có dạng lòng khe.

Để biểu diễn các chữ số, chúng ta sử dụng một ma trận 5×7 =35 để mã hóa cho mỗi ký tự. Tương ứng với mỗi vectơ đầu vào x là một vectơ có kích thước 35×1, với các thành phần nhận các giá trị hoặc 0 hoặc 1.Như vậy, ta có thể lựa chọn lớp nơ-ron đầu vào có 35 nơ-ron. Để phân biệt được mười ký tự, chúng ta cho lớp đầu ra của mạng là 10 nơ-ron. Đối với lớp ẩn ta chọn 5 nơ ron, ta được cấu trúc mạng như hình 2, trong đó:

- Véc tơ đầu vào x, kích thước 35×1
- Véc tơ đầu ra lớp ẩn y, kích thước 5×1
- Véc tơ đầu ra lớp ra z, kích thước 10×1
- Ma trận trọng số lớp ẩn: $W^{1,1}$, kích thước 35×5
- Ma trận trọng số lớp ra: $W^{2,1}$, kích thước 5×10



Hình 2: Cấu trúc mạng neural cho nhận dạng

Hàm f được chọn là hàm sigmoid vì thực tế hàm này cũng hay được dùng cho mạng nơ ron nhiều lớp và hơn nữa do đặc điểm của hàm sigmoid rất dễ sinh ra mặt sai số có dạng lòng khe hẹp. Phương trình của hàm sigmoid là: $f = 1/(1+\exp(-x))$

Hàm sai số sử dụng cho luyện mạng : $J = 0.5*(z-t)^2$ với z là đầu ra của nơron lớp ra và t là giá trị đích mong muốn.

1.Xây dựng thuật toán luyện mạng

2.1. Luyện mạng nơron theo các phương pháp lan truyền ngược

Thuật học lan truyền ngược với mạng MLP được mô tả như sau:

- Bước 1: Cung cấp tập mẫu huấn luyện gồm K cặp mẫu vào và kết quả ra đích
- Bước 2: Khởi tạo giá trị ban đầu cho các trọng số và thiết lập các tham số của mạng
- Bước 3: Lần lượt cho K mẫu lan truyền qua mạng từ lớp vào tới lớp ra. Ta có thể diễn tả việc tính toán tín hiệu ra ở từng lớp như sau:

$a_0 = P_K$ *(mẫu vào)*

$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1})$ *với chỉ số lớp m=0,1,2...M-1*

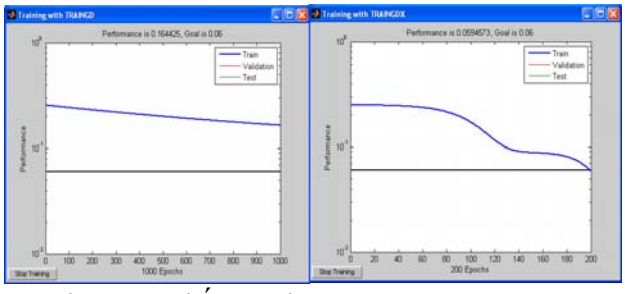
$a = a^M$ *(a-tín hiệu ra của mạng)*

- Bước 4: Tính sai số trung bình bình phương và lan truyền ngược sai số này về các lớp trước.
- Bước 5: Cập nhật các trọng số liên kết theo hướng giảm dốc nhất Gradient.

Quá trình lặp lại từ bước 3 cho đến khi giá trị sai số trung bình bình phương là nhỏ ở mức chấp nhận được.

Thuật toán lan truyền ngược hội tụ đến một giải pháp mà nó tối thiểu hoá được sai số trung bình bình phương vì cách thức hiệu chỉnh trọng số và hệ số bias của thuật toán là ngược hướng với vectơ Gradient của hàm sai số trung bình bình phương đối với trọng số. Tuy nhiên, đối với mạng MLP thì hàm sai số trung bình bình phương thường phức tạp và có nhiều cực trị cục bộ, vì thế các phép lặp huấn luyện mạng có thể chỉ đạt đến cực trị cục bộ của hàm sai số trung bình bình phương mà không đạt đến cực trị tổng thể.

Vấn đề quá trình huấn luyện sẽ hội tụ như thế nào sẽ phụ thuộc vào các điều kiện ban đầu của quá trình huấn luyện. Đặc biệt là việc chọn hệ số học α như thế nào để tăng khả năng hội tụ của mạng. Với mỗi bài toán ta lại có phương án chọn hệ số học khác nhau. Như vậy, khi một quá trình huấn luyện theo thuật toán lan truyền ngược hội tụ, ta chưa thể khẳng định được nó đã hội tụ đến phương án tối ưu. Ta cần phải thử với một số điều kiện ban đầu để đảm bảo thu được phương án tối ưu.



Hình 3: Các kết quả luyện mạng nơ ron với các phương pháp lan truyền ngược khác nhau

Hình 3 trình bày kết quả của quá trình luyện mạng cho bài toán nhận dạng chữ với các thuật toán lan truyền ngược sai số theo phương pháp Batch Gradient Descent (*traingd*), Batch Gradient Descent with Momentum (*traingdm*), Variable Learning Rate (*traingda, traingdx*).

Các phương pháp này đều được tích hợp trên Neural Network Toolbox của Matlab. Nhìn chung các phương pháp đều cho kết quả khá tốt, tuy nhiên để đạt được độ chính xác như mong muốn thì thời gian cần thiết cho luyện mạng là khá lớn. Thậm chí có trường hợp tín hiệu lỗi hầu như thay đổi rất ít quá các chu kỳ luyện mạng. Để giải quyết vấn đề này, cần thiết phải tìm ra một thuật toán hiệu chỉnh các bước học nhằm rút ngắn thời gian hội tụ của mạng đồng thời cũng tránh được vấn đề cực trị địa phương. Phần tiếp theo của bài báo sẽ trình bày:

- Nguyên lý của thuật toán vượt khe điều chỉnh hệ số học của mạng nhằm nâng cao độ chính xác, tốc độ hội tụ của mạng .
- Nguyên lý thuật di truyền nhằm thiết kê mạng neural với độ chính xác cao.

2.2.Thuật toán vượt khe cho bài toán luyện mạng neural

Nguyên lý vượt khe

Xét bài toán cực tiểu hoá không ràng buộc: $J(u) \rightarrow \min,\ u \in E^n$ (2.1)

trong đó u là vectơ cực tiểu hoá trong không gian Euclide n chiều, J(u) là hàm mục tiêu bị chặn dưới và thoả mãn điều kiện:

$$\lim_{\|u\| \to \infty} J(x) = b \qquad (2.2)$$

Thuật toán tối ưu hoá bài toán (2.1) có phương trình lặp như dạng sau:

$u^{k+1} = u^k + \alpha_k s^k$, k = 0,1... (2.3)

trong đó u^k và u^{k+1} là điểm đầu và điểm cuối của bước lặp thứ k, s^k là véctơ chỉ hướng thay đổi các biến số trong không gian n chiều ; α_k là độ dài bước.

α_k được xác định theo nguyên lý vượt khe thì được gọi là bước vượt khe, còn

phương trình (2.3) gọi là thuật toán vượt khe.

Sự khác biệt cơ bản của phương pháp vượt khe so với các phương pháp khác là ở quy tắc điều chỉnh bước. Theo quy tắc điều chỉnh bước của phương pháp vượt khe thì độ dài bước của điểm tìm kiếm ở mỗi bước lặp không nhỏ hơn độ dài bước nhỏ nhất mà tại đó hàm mục tiêu đạt giá trị cực tiểu (địa phương) theo hướng chuyển động tại bước lặp đó.

Quỹ đạo tìm kiếm tối ưu theo nguyên lý vượt khe tạo ra một bức tranh hình học, tựa như điểm tìm kiếm tại mỗi lần lặp đều bước vượt qua lòng khe của hàm mục tiêu. Để cụ thể hoá nguyên lý vượt khe, ta xét hàm một biến sau đối với mỗi bước lặp :

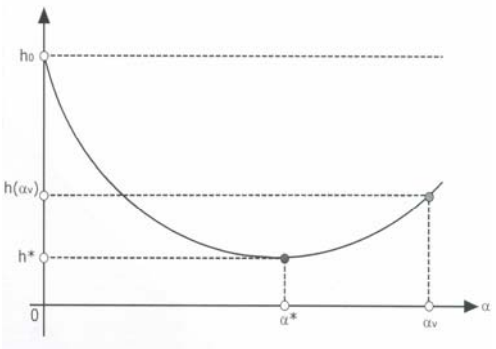
$$h(\alpha) = J\left(u^k + \alpha . s^k\right) \qquad (2.4)$$

Nếu $J(u^k)$, cũng tức là $h(\alpha)$, khả vi liên tục, ta có định nghĩa bước vượt khe như sau:

$$h'(\alpha)\big|_{\alpha=\alpha^v} > 0,\ \ h\left(\alpha^v\right) \le h(0) \qquad (2.6)$$

(α^v là bước vượt quá, tức bước vượt khe)

Giả sử s^k là hướng của hàm mục tiêu tại điểm u^k. Theo điều kiện (2.2), tồn tại một giá trị $\alpha^* > 0$ bé nhất, sao cho $h(\alpha)$ đạt cực tiểu : $\alpha^* = \arg\min h(\alpha),\ \ \alpha > 0$ (2.5)



Hình 4: Xác định bước vượt khe α^v

Đồ thị biến thiên của hàm $h(\alpha)$, khi quỹ đạo tối ưu thay đổi từ điểm đầu u^k đến điểm cuối u^{k+1} thể hiện ở *hình 4*. Ta thấy rằng, khi giá trị α tăng dần từ 0 vượt qua điểm cực tiểu α^* của $h(\alpha)$ tới giá trị α^v, quỹ đạo tối ưu hoá tương ứng tiến dọc theo hướng s^k theo quan hệ $u^{k+1} = u^k + \alpha_k s^k$, thực hiện một độ dài bước $\alpha = \alpha^v \ge \alpha^*$. Đồ thị này cũng chỉ ra rằng, xét theo hướng chuyển động, thì hàm mục tiêu thay đổi theo chiều giảm dần từ điểm u^k, còn khi đạt điểm u^{k+1} thì nó đã chuyển sang xu hướng tăng.

Nếu ta sử dụng bước chuyển động theo điều kiện (2.5) thì có thể tắc ở trục khe và thuật toán tối ưu hoá tương ứng bị tắc lại ở đó. Còn nếu quá trình tối ưu hoá theo điều kiện (4.6) thì sẽ không cho phép điểm tìm kiếm rơi vào lòng khe trước khi đạt lời giải tối ưu, đồng thời nó vẽ ra một quỹ đạo luôn vượt lòng khe. Để quá trình lặp có hiệu quả hội tụ cao và ổn định, điều kiện (2.6) được thay đổi bởi điều kiện (2.7)

$$\alpha^v \ge \alpha^* = \arg\min_{\alpha>0} h(\alpha),\ \ h\left(\alpha^v\right) - h^* \le \lambda\left[h^0 - h^*\right] \qquad (2.7)$$

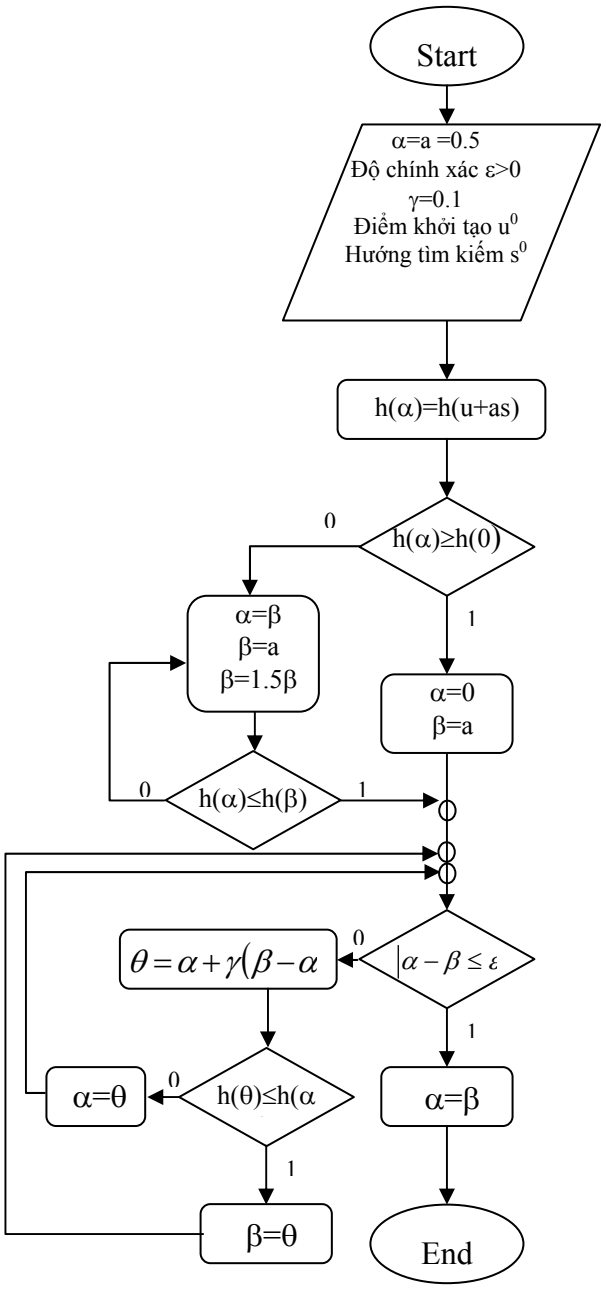
Trong đó: $0 < \lambda < 1$ được gọi là hệ số vượt

$h^* = h\left(\alpha^*\right)\ \ h^0 = h\left(\alpha^0\right)$

Xác định bước vượt khe

Việc lựa chọn độ dài bước học trong bài toán khe hẹp có ý nghĩa đặc biệt quan trọng. Nếu độ dài bước học mà quá nhỏ thì tốn thời gian thực hiện của máy tính. Nếu độ dài bước học lớn thì có thể gây trở ngại cho việc tìm kiếm vì khó theo dõi được sự uốn lượn của khe hẹp. Do vậy, vấn đề bước học thích nghi với khe hẹp là cần thiết trong quá trình tìm kiếm nghiệm tối ưu. Trong mục này, ta đưa ra một cách rất hiệu quả và đơn giản tìm bước "vượt khe". Giả sử J(u) liên tục và thoả mãn điều kiện $\lim J(u) = \infty$ khi $|u| \to \infty$ và tại mỗi bước lặp k, điểm u^{k-1} và véc tơ chuyển

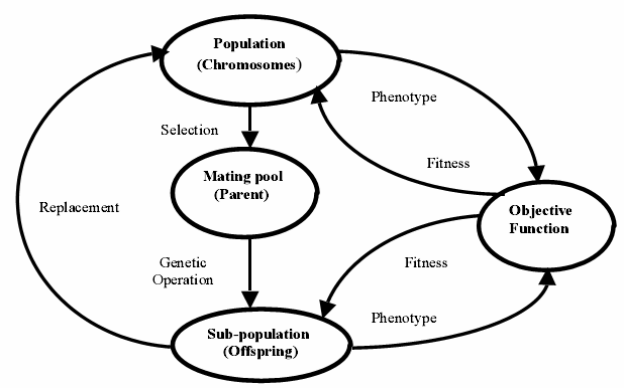
động s^{k-1} đã xác định. Cần phải xác định độ dài bước α_k thoả mãn điều kiện (2.7).



Hình5: Lưu đồ thuật toán tính bước học vươt khe

2.3.Giải thuật di truyền GA

Những thông số cần phải được tìm kiếm bởi GA trong luyện mạng bao gồm các trọng số liên kết mạng, giá trị ban đầu của tỷ lệ học,tyr leej momentum và kiến trúc mạng. Các trọng số ban đầu được tạo ra bằng cách sử dụng hàm ngẫu nhiên trong khoảng [0, 1]. Các kiến trúc mạng được mô tả bởi số lượng lớp ẩn (h), số lượng các nút trong mỗi lớp ẩn (ni; i = 0; 1;:::; h). GA được đề xuất như là quá trình tìm kiếm dựa trên luật chọn lọc tự nhiên và di truyền học. Thuật toán này trước tiên được sử dụng để có được những giá trị tối ưu của các nút ẩn và các giá trị ban đầu của tỷ lệ học tập momentum. Hình 6 mô tả quá trình hoạt động của của một GA đơn giản.



Hình6. Chu kỳ hoạt động của giải thuật di truyền

1. Khởi tạo ngẫu nhiên tạo một quần thể ban đầu $P^0 = (a_1^0, a_2^0, ..., a_\lambda^0)$

2. Tính toán giá trị thích nghi $f(a_i^t)$ của mỗi nhiễm sắc thể a_i^t trong quần thể P^t hiện tại.

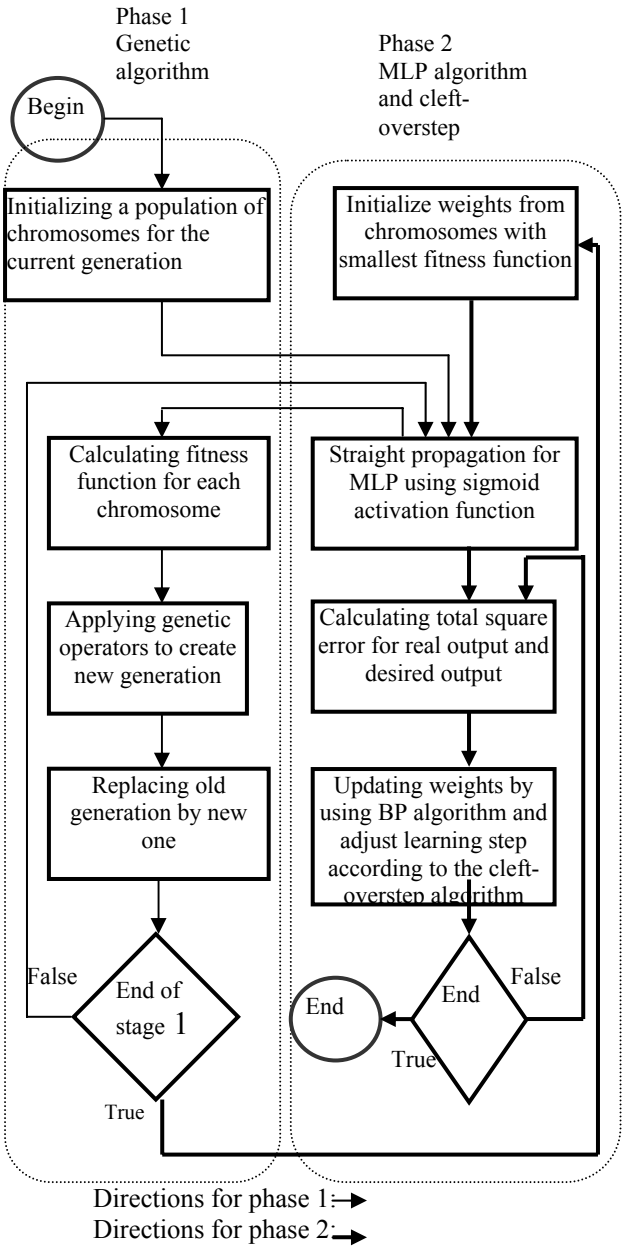
3. Căn cứ vào giá trị thích nghi tạo ra các nhiễm sắc thể mới bằng cách chọn lọc các nhiễm sắc thể cha mẹ, áp dụng các thuật toán lai tạo và đột biến.

4. Loại bỏ nhiễm sắc thể có độ thích nghi kém để tạo chỗ cho quần thể mới.

5. Tính toán các giá trị thích nghi của các nhiễm sắc thể mới $f(a_i^{'t})$ chèn vào quần thể.

6. Tăng số lượng các thế hệ nếu chưa đạt đến điều kiện kết thúc và lặp lại từ bước 3. Khi đạt đến điều kiện kết thúc thì dừng lại và đưa ra nhiễm sắc thể tốt nhất.

2.4. Luyện mạng nơron kết hợp thuật toán vượt khe và giải thuật di truyền



Hình 7: Sơ đồ thuật toán kết hợp giải thuật vượt khe và di truyền cho luyện mạng MLP

Thuật toán kết hợp giải thuật vượt khe và giải thuật di truyền cho mạng MLP ứng dụng cho nhận dạng chữ được đề xuất trong hình7. Nó bao gồm hai giai đoạn luyện mạng. Giai đoạn đầu tiên sử dụng thuật toán di truyền với bước truyền thẳng nhằm đẩy nhanh toàn bộ quá trình luyện mạng . Thuật toán di truyền thực hiện tìm kiếm toàn cục và tìm kiếm tối ưu gần điểm ban đầu (trọng lượng vector) cho giai đoạn thứ hai. Trong đó, mỗi nhiễm sắc thể được sử dụng để mã hóa các trọng số của mạng nơron. Hàm thích nghi (hàm mục tiêu) cho các thuật toán di truyền được xác định là tổng bình phương lỗi (TSSE) của mạng nơron tương ứng. Do đó, bài toán sẽ trở thành tối ưu hóa không giới hạn nhằm tìm một tập hợp các biến quyết định giảm thiểu hàm mục tiêu. Trong giai đoạn thứ 2 sẽ sử dụng thuật toán lan truyền ngược với các bước học được thay đổi theo thuật toán vượt khe đã được đề xuất ở trên.

Việc cài đặt thuật toán trên Matlab được tiến hành như sau:

/* Giai đoạn 1*/

Khởi tạo các nhiễm sắc thể một cách ngẫu nhiên cho thế hệ hiện tại, khởi tạo các tham số làm việc và đặt nhiễm sắc thể đầu tiên là nhiễm sắc thể tốt nhất *best_chromosome*.

a- Lặp từ i=1 đến kích thước quần thể, thực hiện công việc sau:

- Khởi tạo *sub_total_fitness* bằng 0 và *sub_best_chromosome* là rỗng

b- Lặp từ j=1 đến độ dài của nhiễm sắc thể, thực hiện các công việc sau:

- Thực hiện thủ thục truyền thẳng cho mạng MLP (sử dụng hàm hoạt hóa là sigmoid).
- Tính toán hàm mục tiêu (lỗi hệ thống của mạng rron)
- Tính toán lỗi tổng cộng *total_fitness* bằng cách tích lũy *sub_total_fitness*

c- Lưu *best_chromosome* vào *sub_best_chromosome*

d- So sánh các *sub_best_chromosome* với nhau và đặt *sub_best_chromosome* lớn nhất là *best_chromosome*.

e- Lặp từ i=0 đến kích thước quần thể/2 , thực hiện các thủ tục sau:

- Khởi tạo *sub_total_fitness* bằng 0 và *sub_best_chromosome* là rỗng

- Lặp từ j=1 tới độ dài nhiễm sắc thể, thực hiện các công việc sau:

* Chọn các nhiễm sắc thể cha mẹ sử dụng phương pháp lựa chọn theo bánh xe roulette

* Áp dụng các phép lai tạo và đột biến

- Lặp từ k=1 đến độ dài nhiễm sắc thể, thực hiện các công việc sau:

* Thực hiện thủ tục truyền thẳng cho mạng MPL

* Tính toán các giá trị hàm mục tiêu cho các nhiễm sắc thể cha mẹ.

- Tính toán *sub_total_fitness* bằng cách tích lũy giá trị hàm mục tiêu của mỗi nhiễm sắc thể.

- Lưu *best_chromosome* vào *sub_best_chromosome*

g- Thay thế thế hệ cũ bằng thế hệ mới nếu thỏa mãn điều kiện dừng.

/* Giai đoạn 2 */

- Đặt *best_chromosome* là véc tơ trọng số khởi tạo, thiết lập cấu trúc mạng rron MLP.

- Tính toán đầu ra thực tế của mạng MLP truyền thẳng.

- Tính toán lỗi giữa đầu ra thực tế và đầu ra mong muốn.

- Cập nhật các trọng số bằng thuật toán lan truyền ngược, cập nhật các hệ số học bằng thuật toán vượt khe.

END

3. Các kết quả thực nghiệm

Mạng MLP được luyện với bộ các ký tự mẫu chữ với kích thước 7 x 5 được trình bày ở trên. Các giá trị ban đầu như số đầu vào (35), số lượng lớp ẩn (1), số nơron lớp ẩn (5), các kỹ thuật luyện mạng khác nhau, mã hóa đầu vào và đầu ra nhằm khởi tạo các trọng số đã được đề cập ở trên. Để kiểm tra khả năng của mạng cho quá trình nhận dạng chữ, chúng tôi đề xuất một tham số đánh giá chất lượng của mạng là tỷ lệ lỗi nhận dạng được tính theo công thức :

$$\text{Ty le loi } (FR) = \frac{\text{So ky tu da kiem tra} - \text{So ky tu da nhan dang}}{\text{So ky tu da kiem tra}} 100\%$$

Test 1 : Luyện mạng MLP theo thuật toán BP thuần túy

Các tham số luyện mạng:

Kích thước ký tự = 5 x 7 Số đầu ra = 10

Số đầu vào = 35

Số nơron lớp ẩn = 5 Độ chính xác mong muốn = 90% Tỷ lệ học : 0.6

Lỗi hệ thống mong muốn=0.06

Kết quả luyện mạng như sau:

Số chu kỳ luyện	20	60	100	130	200
Tỷ lệ lỗi %	93.33	60.33	40.67	37.33	0
TSSE	0.8136	0.6848	0.2834	0.2823	0.06

Test 2 : Luyện mạng MLP theo thuật toán BP kết hợp với thuật toán vượt khe và di truyền

Các tham số luyện mạng:

Kích thước quần thể = 20 Xác suất lai tạo = 0.46 Mã hóa bằng số thực

Độ dài nhiễm sắc thể = 225 Độ chính xác mong muốn = 90% Số thế hệ: 20

Lỗi hệ thống mong muốn=0.06

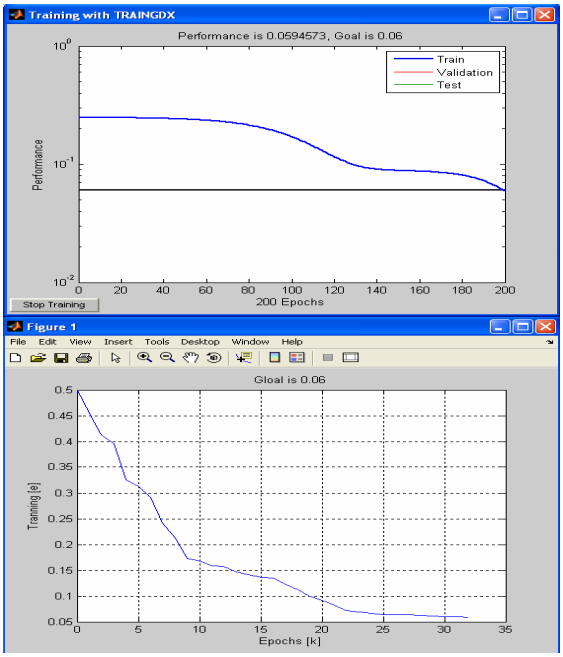
Kết quả luyện mạng như sau:

Số thế hệ	1	5	10	15	20
Tổng thích nghi	9.5563	8.1638	6.1383	5.724	5.697

Số chu kỳ luyện	5	10	15	20	33
Tỷ lệ lỗi %	93.33	60.33	40.67	37.33	0
TSSE	0.3186	0.1674	0.1387	0.0864	0.0589

Như vậy, lỗi hệ thống ở test 1 khi sử dụng luyện mạng MLP bằng giải thuật BG thuần túy là 0.06 sau 200 chu kỳ luyện mạng. Đối với test 2, sau 20 thế hệ đã đạt đến yêu cầu của bài toán. Giá trị thích nghi trung bình đạt được là 5.679. Kết quả của

giai đoạn 1 được sử dụng để khởi tạo trọng số cho giai đoạn 2. Với sự thay đổi bước học theo giải thuật vượt khe, sau 33 chu kỳ luyện mạng lỗi hệ thống đã đạt đến mục đích 0.0589 và độ chính xác của quá trình nhận dạng là 100% . Hoạt động của mạng MLP thuần túy và mạng MLP có kết hợp giải thuật vượt khe và di truyền cho nhận dạng chữ được thể hiện trên hình 8



Hình 8: Hoạt động của mạng MLP thuần túy và MLP cải tiến

4. Kết luận

Trong bài báo này, các tác giả đề xuất việc sử dụng giải thuật di truyền kết hợp với thuật toán "vượt khe" để cải tiến quá trình luyện mạng neural có mặt lỗi đặc biệt và minh họa thông qua ứng dụng nhận dạng chữ.

Qua việc nghiên cứu và thực nghiệm trên máy tính cho ta thấy: với những cấu trúc mạng nơ ron mà mặt lỗi có dạng lòng khe, vẫn sử dụng kỹ thuật lan truyền ngược nhưng việc áp dụng giải thuật di truyền kết hợp với thuật toán " vượt khe" để luyện mạng sẽ cho ta độ chính xác và tốc độ hội tụ nhanh hơn nhiều so với phương pháp gradient.

Kết quả nghiên cứu này được giải thích như sau:

-Kết quả luyện mạng nơron phụ thuộc rất lớn vào giá trị ban đầu của vecto trọng số. Việc sử dụng giải thuật di truyền thực hiện quá trình tìm kiếm toàn cục cho phép có được vecto trọng số ban đầu tốt cho giai đoạn sau của quá trình luyện mạng.

-Khi mặt lỗi đặc biệt có dạng lòng khe, nếu luyện mạng bằng thuật toán gradien liên hợp hay thuật toán Levenberg – Marquardt sẽ chậm hội tụ và gặp phải vấn đề cực trị địa phương. Thuật toán "vượt khe" nhằm tìm kiếm các bước học tối ưu trong giai đoạn 2 của quá trình luyện mạng nên đã khắc phục các nhược điểm này và do đó làm tăng tốc độ hội tụ cũng như độ chính xác của quá trình luyện mạng.

Việc sử dụng giải thuật di truyền kết hợp với thuật toán "vượt khe" có thể ứng dụng để luyện một số cấu trúc mạng nơ ron mà có mặt lỗi đặc biệt khác. Vì vậy kết quả nghiên cứu này có thể ứng dụng cho nhiều bài toán khác trong lĩnh vực viễn thông, điều khiển, và công nghệ thông tin.

Bài báo cần được nghiên cứu thêm về việc xác định hướng của véc tơ tìm kiếm trong thuật toán "vượt khe", việc thay đổi chuẩn đánh giá của hàm chất lượng để giảm độ phức tạp của quá trình tính toán trên máy tính[6]. Tuy nhiên, các kết quả nghiên cứu này, bước đầu đã phản ánh tính đúng đắn của thuật toán đề xuất và mở ra khả năng ứng dụng vào thực tiễn.

Bài báo cần được nghiên cứu thêm về việc xác định hướng của véc tơ tìm kiếm trong thuật toán "vượt khe", việc thay đổi chuẩn đánh giá của hàm chất lượng để giảm độ phức tạp của quá trình tính toán trên máy

tính[6]. Tuy nhiên, các kết quả nghiên cứu này, bước đầu đã phản ánh tính đúng đắn của thuật toán đề xuất và mở ra khả năng ứng dụng vào thực tiễn.

Tài liệu tham khảo

[1] D.E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Pub. Comp. Inc., Reading, MA, 1989

[2] D. Anthony, E. Hines, "The use of genetic algorithms to learn the most appropriate inputs to neural network", Application of the International Association of Science and Technology for Development-IASTED, June, 1990, 223–226.

[3] Nguyen Van Manh and Bui Minh Tri, "Method of "cleft-overstep" by perpendicular direction for solving the unconstrained nonlinear optimization problem", Acta Mathematica Vietnamica, vol. 15, N02, 1990.

[4] L. Davis, "Hand book of Genetic Algorithms", Van Nostrand Reinhold, New York, 1991.

[5] L. Fauselt, "Fundamentals of Neural Networks", Prentice-Hall, International Inc., Englewood CliGs, NJ, 1994.

[6] R.K. Al Seyab, Y. Cao (2007)"Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation", School of Engineering Cranfield University, College Road, Cranfield, Bedford MK43 0AL, UK, Science Direct

[7] Cong Nguyen Huu; Nam Nguyen Hoai, "Optimal control for a distributed parameter and delayed – time system based on the numerical method", Teth international conference on Control, Automotion, Robotics and vision(ICARCV'2008)..