



KẾT HỢP MẠNG NƠN VÀ GIẢI THUẬT DI TRUYỀN ỨNG DỤNG CHO LỚP BÀI TOÁN NHẬN MẪU

Nguyễn Quang Hoan¹, Nguyễn Thị Trang², Nguyễn Thị Huyền¹,
Trương Quốc Khánh³, Nguyễn Thị Hoa⁴

¹ Trường Đại học Sư phạm Kỹ thuật Hưng Yên

² Trường Đại học Công nghệ Đông Á

³ Trường Trung Cấp nghề số 1 Hà Nội

⁴ Trường Cao đẳng Nghề Công nghệ Cao, Hà Nội

Ngày nhận: 15/06/2016

Ngày sửa chữa: 10/08/2016

Ngày xét duyệt: 19/09/2016

Tóm tắt:

Bài báo trình bày phương pháp tích hợp giải thuật di truyền (Genetic Algorithm: GA) và mạng nơron (Neural Network: NN) lan truyền ngược (Back Propagation: BP) để tránh cực tiểu cục bộ trong kỹ thuật học theo phương pháp hạ Gradient cho lớp bài toán nhận mẫu hoặc dự báo. Thực hiện vấn đề đó, quá trình học NN được chia thành hai pha: pha đầu sử dụng GA để xác định tập trọng số (weights) khởi đầu của mạng nơron nằm trong vùng tối ưu; pha hai: dùng thuật toán lan truyền ngược để thu được tập trọng số tối ưu toàn cục cuối cùng. Các thử nghiệm phương pháp cho nhận mẫu vân tay được thực hiện. Kết quả học bằng cách tích hợp GA và mạng nơron được so sánh với việc học của mạng nơron không tích hợp và định hướng nghiên cứu tiếp được đề xuất

Từ khóa: Mạng nơron, thuật toán lan truyền ngược, giải thuật di truyền, nhận mẫu, dự báo.

1. Giới thiệu

Có nhiều phương pháp và công cụ khác nhau để giải bài toán dự báo và nhận mẫu [4, 5, 13, 14, 15]. Để giải quyết những trường hợp phi tuyến tốt, hoặc tránh rơi vào tối ưu cục bộ, chúng ta có thể dùng NN lai với GA [6, 7, 8, 9, 10, 11, 15].

NN có nhiều cấu trúc khác nhau, nhưng thường dùng nhất là mạng BP [2, 4, 5] do hội tụ nhanh. Hạn chế của nó là kết quả học chỉ tìm được tập trọng số tối ưu cục bộ [8]; do đó, một biện pháp là dùng GA để xác định giá trị các trọng số khởi đầu thuộc vùng tối ưu toàn cục, đảm bảo mạng có bộ trọng số tối ưu toàn cục.

Nội dung bài báo này là thử nghiệm kết hợp NN và GA để giải lớp bài toán nhận mẫu hoặc lớp bài toán dự báo.

2. Giải thuật di truyền

GA được đề xuất như là quá trình tìm kiếm dựa trên luật chọn lọc tự nhiên và di truyền học. Thuật toán được sử dụng để tìm các giá trị trọng số ban đầu quanh vùng tối ưu của mạng nơron. Hình 1 mô tả quá trình hoạt động của của GA [6] với các bước:

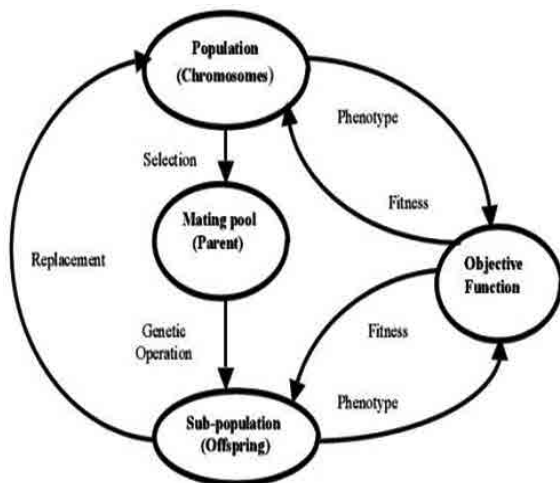
1. Khởi tạo ngẫu nhiên tạo một quần thể ban đầu (thế hệ 0): $P^0 = (w_{11}^0, w_{12}^0, \dots, w_{ij}^0)$ trong đó w_{ij}^0 : tập các NST (trọng số) thế hệ 0.

2. Tính giá trị từ hàm thích nghi $f(w_{ij}^t)$ của mỗi NST w_{ij}^t trong quần thể P^t hiện tại.

3. Dựa vào giá trị thích nghi, tạo ra các NST mới bằng cách chọn lọc các NST cha mẹ, áp dụng

thuật toán lai và đột biến.

4. Loại NST có độ thích nghi kém, thay NST mới vào quần thể.



Hình 1. Chu trình hoạt động của GA

5. Tính các giá trị thích nghi cho NST mới $f(w_{ij}^1)$ và chèn vào quần thể.

6. Tăng số lượng thế hệ t nếu chưa đạt điều kiện dừng và lặp lại từ bước 3. Khi đạt điều kiện kết thúc thì dừng và cho NST tốt nhất (là bộ trọng số cần tìm).

3. Kết hợp mạng nơron với GA

Cho cấu trúc NN. Dùng GA tìm tập các trọng

số nằm trong miền dẫn chúng tới tối ưu toàn cục cho NN. Một quần thể gồm N chuỗi trọng số của mạng được khởi tạo ngẫu nhiên. Sau G thế hệ tiến hoá, cá thể tốt nhất trong G thế hệ được giữ lại. Các cá thể này, sau đó, được giải mã và đưa vào NN để học tiếp. Trong hệ tích hợp này, quá trình học có thể phân thành hai giai đoạn huấn luyện. Giai đoạn đầu sử dụng GA với hàm thích nghi (hàm mục tiêu) được xác định theo:

$$E = \frac{1}{2} \sum_{i=1}^m (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^m (d_i - f(\sum_{j=1}^m (w_{ij}x_j - w_0)))^2 \quad (1)$$

trong đó: d_i và y_i tương ứng là đầu ra mong muốn và thực tế của neuron thứ i ; x_j là đầu vào thứ j ; w_{ij} là trọng số liên kết giữa đầu vào thứ j đến đầu ra thứ i . Giai đoạn hai sử dụng thuật toán lan truyền ngược để tìm các trọng số tối ưu toàn cục từ kết quả của giai đoạn 1.

Giai đoạn 1: Pha học dùng GA

Thuật toán:

Bước 0. Khởi tạo các NST ngẫu nhiên (trọng số khởi đầu) cho thế hệ hiện tại và các tham số làm việc. Đặt NST đầu tiên là tốt nhất `best_chromosome`

Bước 1. Lập từ $i=1$ đến kích thước quần thể n . Khởi tạo `sub_total_fitness` bằng 0 và `sub_best_chromosome` là rỗng.

Bước 2. Lập từ $j=1$ đến độ dài của NST m

- Thực hiện thủ tục truyền thẳng cho mạng 3 lớp sử dụng hàm tương tác dạng sigmoid:

$$(u_i) = \frac{1}{1 + e^{-\lambda u_i}}; u_i = \sum_{j=1}^m w_{ij}x_j - w_0 \quad (2)$$

- Tính hàm mục tiêu (1)

- Tính tổng sai số `total_fitness` bằng cách tích lũy `sub_total_fitness`

Bước 3. Lưu `best_chromosome` vào `sub_best_chromosome`

Bước 4. So sánh các `sub_best_chromosome`. Đặt `sub_best_chromosome` lớn nhất là `best_chromosome`.

Bước 5. Lập từ $i=0$ đến kích thước quần thể n

- Khởi tạo `sub_total_fitness` bằng 0 và `sub_best_chromosome` là rỗng;

- Lập từ $j=1$ tới độ dài NST m

* Chọn các NST cha mẹ sử dụng phương pháp chọn theo bánh xe Roulette

* Áp dụng các phép lai và đột biến

- Lập từ $k=1$ đến độ dài nhiễm sắc thể:

* Thực hiện thủ tục truyền thẳng cho mạng

* Tính các giá trị hàm mục tiêu cho các NST cha mẹ.

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

- Lưu `best_chromosome` vào `sub_best_chromosome`

Bước 6. Thay thế hệ cũ bằng thế hệ mới nếu chưa thỏa mãn điều kiện dừng.

Giai đoạn 2: Pha học dùng thuật toán lan truyền ngược

Bước 0 (Khởi tạo)

Chọn ngưỡng lỗi $E_{threshold} = 0$. Khởi tạo giá trị các trọng số: lấy kết quả thu được từ pha 1.

Bước 1 (Bắt đầu một chu kỳ học)

Áp dụng véc tơ vào đối với lớp vào k của NN ($k=1$): ${}^k y_i = {}^k Out_i = {}^1 Out_i = x_i, \forall i$;

Bước 2 (Lan truyền tiến)

Lan truyền tiến các tín hiệu vào qua mạng, cho đến khi nhận được các giá trị ra của mạng (ở lớp đầu ra) ${}^k Out_i$

Bước 3 (Tính toán lỗi đầu ra)

Tính sai số ra của mạng và tín hiệu lỗi ${}^k \delta_i$ của mỗi neuron ở lớp ra i .

$$E = \frac{1}{2} \sum_{i=1}^n (d_i^{(k)} - {}^k Out_i)^2$$

$${}^k \delta_i = (d_i^{(k)} - {}^k Out_i) f'^{({}^k Net_i)}$$

ở đây, $f' \equiv \frac{\partial f}{\partial w_{ij}}$; ${}^k u_i = {}^k Net_i$ (4)

Bước 4 (Lan truyền ngược lỗi)

Lan truyền ngược sai số để cập nhật trọng số và tính tín hiệu sai số ${}^{k-1} \delta_i$ cho các lớp trước.

$$\Delta {}^k W_{ij} = \eta \cdot ({}^k \delta_i) \cdot ({}^{k-1} Out_j); \quad (5)$$

$${}^k W_{ij}(t+1) = {}^k W_{ij}(t) + \Delta {}^k W_{ij}(t)$$

$${}^{k-1} \delta_i = f'^{({}^{k-1} Net_i)} \sum_j {}^k W_{ji} \delta_{ij};$$

Bước 5. Kiểm tra tập học đã được sử dụng. Nếu tập dữ liệu học đã được dùng, chuyển đến *Bước 6*; ngược lại, chuyển đến *Bước 1*.

Bước 6 (Kiểm tra lỗi tổng thể)

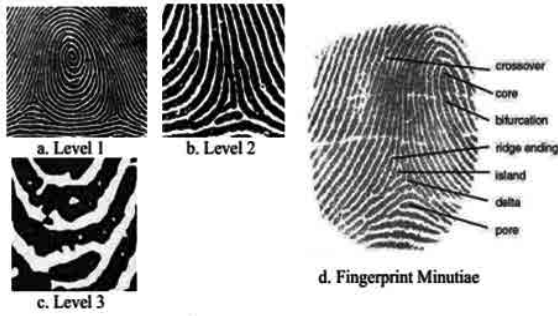
Nếu lỗi tổng thể E đạt ngưỡng lỗi chấp nhận ($E_{threshold}$), quá trình học kết thúc và trả về các trọng số tối ưu. Ngược lại, gán lại lỗi bằng 0, bắt đầu chu kỳ học mới (quay về *Bước 1*).

4. Ứng dụng

Trong bài báo, chúng tôi thử nghiệm phương pháp tích hợp đã nêu cho nhận mẫu vân tay theo các đặc trưng mức hai. Bài toán dự báo cũng có thể dùng GA tương tự để tính trọng ban đầu.

4.1. Các đặc trưng vân tay

Dấu vân tay là một thể hiện bằng những đường vân đầu ngón tay của một người. Có thể chia đặc trưng vân tay (hình 2 a, b, c) thành ba mức độ (level): level 1: mẫu - *Pattern*; level 2: các điểm đầu, kết thúc, chạc ba - *Minutiae*; level 3: các lỗ chân lông và kiểu vân - *Pores and Ridge Shape*. Hầu hết các hệ thống nhận mẫu vân tay sử dụng các đặc trưng mức 1 và mức 2 [5, 6, 11, 12].



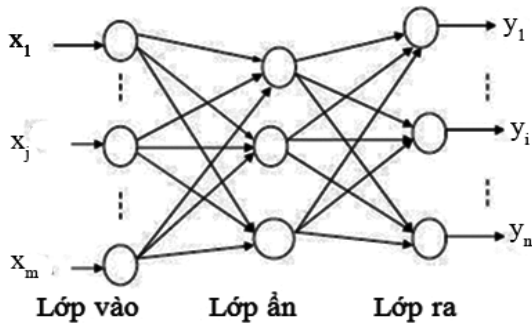
Hình 2. Các mức độ đặc trưng của vân tay

Bài báo sử dụng mức hai. Mức hai có nhiều kỹ thuật trích chọn đặc trưng khác nhau. Trong phạm vi thử nghiệm trên máy tính xách tay, chúng tôi chọn *Core* hoặc *Delta* (Hình 2.d) làm gốc tọa độ, chia vân trên một ngón tay thành 4 góc phần tư, mỗi góc phần tư trong mặt phẳng hai chiều; tìm điểm trung bình của các giá trị kết thúc, bắt đầu hoặc chạc ba của vân tay. Điểm trung bình có 2 giá trị theo tọa độ trục (x_p, y_p) , 4 góc phần tư sẽ có 8 giá trị đặc trưng trở thành 8 đặc trưng đầu vào $x = [x_1, x_2, \dots, x_8]$ cho NN.

4.2. Nhận dạng vân tay

4.2.1. Thiết kế mạng nơron nhân tạo

Chọn BP NN ba lớp-đủ khả năng nhận biết các mẫu đầu vào với hàm tương tác đầu ra liên tục, khả vi, bị chặn [4, 14, 15] như (2).



Hình 3. Cấu trúc NN ba lớp lan truyền ngược

a. Lớp vào (Input Layer)

Với tập mẫu đầu vào x , nơron thứ q của lớp ẩn được xác định:

$$net_q = \sum_{j=1}^m v_{jq}x_j \quad j = 1, 2, \dots, m; q = 1, 2, \dots, l \quad (6)$$

trong đó m là số nơron của lớp vào, q là số nơron thứ q trong l nơron của lớp ẩn.

b. Lớp ẩn. Đầu ra của lớp ẩn

$$z_q = f(net_q) = f\left(\sum_{j=1}^m v_{jq}x_j\right) \quad (7)$$

trong đó, $f(.)$ là hàm tương tác đầu ra (2).

$$f(net_q) = \frac{1}{1 + e^{-\lambda net_q}} \quad (8)$$

Chọn 1000 nơron cho lớp ẩn qua thực nghiệm từ những kết quả tối ưu và giá trị lỗi trong quá trình huấn luyện.

c. **Lớp ra:** chọn hàm tương tác đầu ra của lớp ra giống các lớp khác (2), khi đó:

$$net_i = \sum_{q=1}^l w_{iq}z_q = \sum_{q=1}^l w_{iq}f\left(\sum_{j=1}^m v_{jq}x_j\right) \quad (9)$$

$$y_i = f(net_i) = f\left(\sum_{q=1}^l w_{iq}z_q\right) = f\left(\sum_{q=1}^l w_{iq}f\left(\sum_{j=1}^m v_{jq}x_j\right)\right)$$

Trong đó: i là nơron thứ i trong n nơron đầu ra được chọn tùy thuộc vào cách mã hoá đặc trưng đầu ra. Trong bài toán, chọn số nơron lớp ra $n = 10$, vì $2^n = 2^{10}$ lớn hơn số mẫu thử.

4.2.2. Nhận mẫu vân tay kết hợp NN và GA

Nhận mẫu được thực hiện trong 4 thao tác:

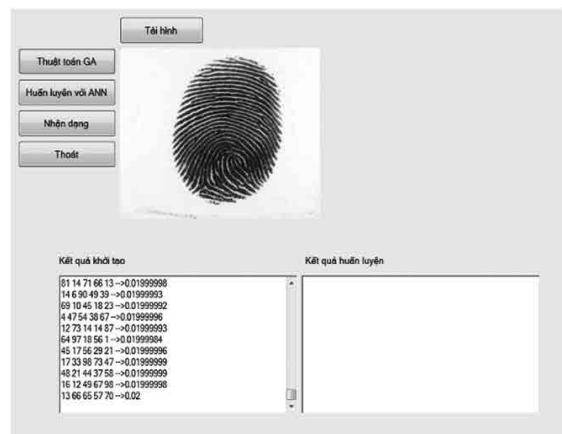
- a) Tải hình ảnh vân tay cần nhận dạng từ tập cơ sở dữ liệu ảnh thu thập được đã chuẩn hóa



Hình 4. Cửa sổ tải hình ảnh vân tay

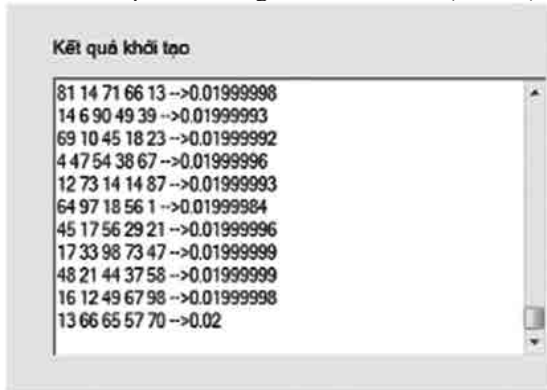
- b) Học theo GA: (Hình 4. Chọn Nút: “Thuật toán GA”): gọi chương trình học theo GA:

- Cho trọng số khởi đầu (xem Hình 5, cột bên phải “Kết quả khởi tạo”)
- “Kết quả huấn luyện” ở Hình 5 cột trái.



Hình 5. Thực hiện thuật toán GA: pha học 1

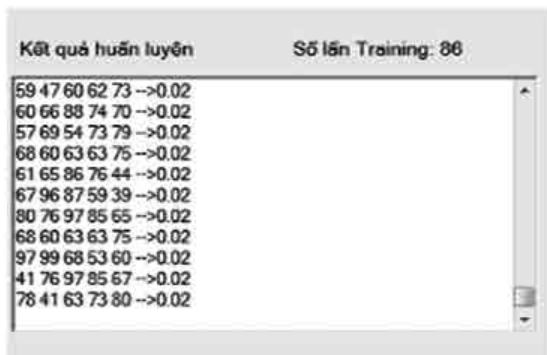
Kết quả học dùng thuật toán GA1 (Hình 6).



Hình 6. Kết quả khởi tạo trọng số thực hiện bởi GA

Kết quả này dùng làm khởi tạo cho pha 2: học dùng thuật toán lan truyền ngược của NN. Trên Hình 6, số lần học tiến hành được hiển thị trên màn hình tương tác. Tỷ lệ lai được thử với 25% và đột biến 1% của NST [11].

c) Huấn luyện NN: pha học 2. Nhấn nút “Huấn luyện với ANN”, ta có kết quả thử nghiệm của một lần học như Hình 7.



Hình 7. Học bằng thuật toán lan truyền ngược

d) Nhận dạng. Mẫu nhận dạng được đưa vào trên NN đã học. Mẫu được gán tên người, giới tính (Nam/nữ), ngày sinh, số chứng minh thư, địa chỉ...



Hình 8. Nhận mẫu vân tay bằng mạng nơ ron với GA

4.2.3. Thử nghiệm nhận mẫu

Để áp dụng thử nghiệm nhận mẫu vân tay sử dụng phương pháp trên (Hình 8), chúng tôi áp dụng hệ thống cho Trường Đại học Công nghệ Đông Á, cơ sở dữ liệu vân tay là của các nhân viên Nhà trường trên 40 nhân viên và gần 400 mẫu vân tay. Các thông tin của các nhân viên và mẫu vân tay được lưu trữ trong cơ sở dữ liệu của hệ thống. Trong 400 mẫu, 300 mẫu (75%) vân tay dùng cho việc học và 100 (chiếm 25%) mẫu dùng để thử (Test). Mẫu thử chứa cả mẫu được học và mẫu chưa được học để xem xét khả năng suy diễn của NN. Các mẫu đã tiến hành tiền xử lý bằng các kỹ thuật xử lý ảnh, và chuẩn hóa độ tương phản, khuôn dạng [1, 3].

4.3. Kết quả đạt được

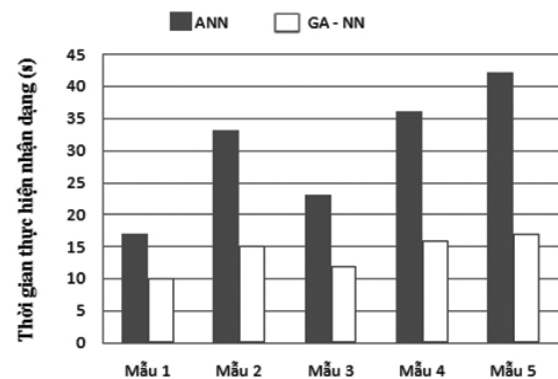
Thử nghiệm tiến hành trên máy tính có cấu hình: Intel® Core™i5-6500U CPU 2.3 GHz.

a) Thời gian thực hiện:

Phương pháp tích hợp mạng nơ ron và giải thuật di truyền GA-NN có thời gian thực hiện nhanh hơn phương pháp nhận mẫu bằng NN (Bảng 1). Điều này dễ giải thích vì GA-NN đã được đưa về vùng tối ưu trước. Trong một số trường hợp GA-NN đã cho đúng tập trọng tối ưu, nên thời gian học pha hai nhanh. Tuy nhiên, thời gian học ở giai đoạn một với GA dài. Có thể đề xuất chiến lược học: cho pha học của GA học lần đầu (có thể kéo dài), các mẫu vân tay cập nhật có thể bỏ qua giai đoạn này, tránh thời gian kéo dài.

Bảng 1. Thời gian nhận mẫu của ANN, GA-NN

Mẫu	Thời gian thực hiện nhận dạng (s)	
	ANN	GA - NN
Mẫu 1	17	10
Mẫu 2	33	15
Mẫu 3	23	12
Mẫu 4	36	16
Mẫu 5	42	17



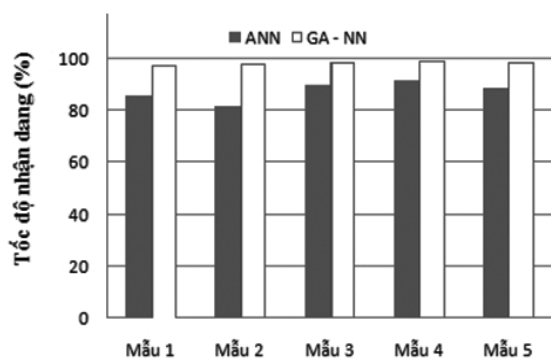
Hình 9. So sánh thời gian thực hiện nhận mẫu của ANN và GA-NN

b) Tốc độ nhận mẫu (%):

Tốc độ nhận mẫu của phương pháp GA-NN cao hơn so với phương pháp ANN.

Bảng 2. Tốc độ nhận mẫu của ANN và GA-NN

Mẫu	Tốc độ nhận dạng (%)	
	ANN	GA - NN
Mẫu 1	86	98
Mẫu 2	82	98.22
Mẫu 3	90	99.21
Mẫu 4	92	99.33
Mẫu 5	89	98.85



Hình 10. Tốc độ nhận mẫu của ANN và GA-NN

Tài liệu tham khảo

- [1]. Nguyễn Quang Hoan, *Xử lý ảnh*, Học viện Công nghệ Bru chính Viễn thông, 2013.
- [2]. Nguyễn Quang Hoan, Vũ Thị Giang, *Mạng nơron Kohonen và khả năng ứng dụng*. Tạp chí Khoa học Công nghệ Quân sự, 2013.
- [3]. Lương Mạnh Bá, Nguyễn Thanh Thủy, *Nhập môn xử lý ảnh số*, NXB Khoa học và Kỹ thuật, 2003.
- [4]. Nguyễn Quang Hoan, Nguyễn Thị Thảo, Vũ Thành Vinh, *Nhận mẫu ký tự sử dụng mạng nơron lan truyền ngược*, Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên, số 7/9/2015.
- [5]. Đỗ Thanh Long, Nguyễn Quang Hoan, Nguyễn Thị Hoa, *Hệ thống nhận mẫu vân tay sử dụng mạng nơron nhân tạo*, Tạp chí Khoa học và Công nghệ, Trường Đại học Sư phạm Kỹ thuật Hưng Yên, số 7/9/2015.
- [6]. Le Hoai Bac, Le Hoang Thai, *Neural Network & Genetic Algorithm Application Handwritten Character Recognition*, Tạp chí Tin học và Điều khiển học, T. 17, S. 4, 2001.
- [7]. Sowmya Ravi, Neeraja Ganesan, Vandita Raju, *Search Engines Using Evolutionary Algorithms*, International Journal of Communication Network Security ISSN: 2231 – 1882, Volume-1, Issue-4, 2012.
- [8]. David J. Montana, *Neural Network Weight Selection Using Genetic Algorithms*, Bolt Beranek and Newman Inc. 70 Fawcett Street, Cambridge, MA 02138, 2013.
- [9]. Liu Han-li, Li Lin, Zhu Hai-hong, *Genetic Neural Network Based Data Mining and Application in Case Analysis of Police Office*, School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan, P.R.China, 430079, 2011.
- [10]. David J. Montana and Lawrence Davis, *Training Feedforward Neural Networks Using Genetic Algorithms*, BBN Systems and Technologies Corp. 10 Moulton St. Cambridge, MA 02138, 2012.
- [11]. Dhia A. Alzubaydi, Thikra M. Abed, *Genetic Algorithm and Probabilistic Neural Networks for Fingerprint Identification*, International Journal of Computer Applications (0975 – 8887), Volume 101– No.11, September 2014.

5. Kết luận

NN tích hợp với GA có khả năng tối ưu quá trình học toàn cục, giúp ta xác định tốt bộ trọng số ứng với lỗi ra nhỏ nhất. NN tích hợp có những kết quả tốt hơn so với không tích hợp, tuy nhiên, thời gian học giai đoạn một (với GA khá lớn). Mỗi lần dùng GA, có thể cho kết quả khác nhau phụ thuộc vào điểm lai, quần thể lai và đột biến. Mỗi lần học pha 1 của GA cho kết quả, thời gian học, số vòng lặp là khác nhau. Trong thực nghiệm lai GA-NN, chúng tôi giải quyết bài toán nhận mẫu và hoàn toàn có thể sử dụng nó cho lớp bài toán dự báo và các bài toán khác kiểu tương tự.

Hướng tiếp theo, chúng tôi sẽ nghiên cứu cải tiến để chạy trên máy tính cấu hình lớn với nhiều tham số và cấp độ vân tay mức ba. Các thử nghiệm sẽ cố gắng phong phú số mẫu, chi tiết hơn. Các kết quả sẽ được thông báo trong những công bố tiếp theo.

- [12]. Roli, Priti Shgal and Punam Bedi, *Minutiae Extraction from Fingerprint Images - a Review*, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, 2011.
- [13]. Tu Le Anh, Hoan Nguyen Quang, Thai Le Son, *Hierarchy Supervised SOM Neural Network Applied for Classification Problem*, Journal of Computer Science and Cybernetics, V.24, N.3 (2014), pp. 278-290.
- [14]. Hoan Nguyen Quang, Tu Le Anh, *Clustering Hierarchical Data Using SOM Neural Network*, ICCASA 2012, LNICST 109, pp. 282–289, 2013.
- [15]. Rajendra Arvind Akerkar and Priti Srinivas Sajja, *Knowledge-Based Systems*, Jones and Bartlett Publisher, 2010.

NEURAL NETWORKS AND GENETIC ALGORITHM FOR FINGERPRINT PATTERN RECOGNITION

Abstract:

In this paper, we integrate Genetic Algorithm (GA) with Backpropagation Neural Network (BP NN) to remove local minimums when using Gradient Descent technique for some problems such as pattern recognitions or forecasting. With this method, learning process can be divided in two phases: first phase using GA to define initial NN weights; second phase using traditional BP Algorithm to receive global optimization weights. We to make some experiments using the intergrated technique on Fingerprint Recognition and to compare the results of the technique with the non-ingterated NN.

Keywords: *Backpropagation, Genetic Algorithm, Pattern Fingerprint Recognition, Forecasting.*