Tạp chí Tin học và Điều khiển học, T.22, S.1 (2006), 123–133

HAI TIẾP CẬN TỚI BÀI TOÁN XẤP XỈ HÀM PHI TUYẾN

BÙI CÔNG CƯỜNG[1], DƯƠNG THĂNG LONG[2], HOÀNG VIỆT LONG[3]

[1] *Viện Toán học*

[2]*Khoa Công nghệ thông tin, Viện Đại học mở Hà Nội*

[3]*Đại học Giao thông Vận tải Hà Nội*

Abstract. In this paper we present some new results on the nonlinear functions approximation problem by using some Mamdani fuzzy systems and two constructive algorithms for structure learning in neural networks.

Tóm tắt. Bài báo trình bày hai tiếp cận tới bài toán xấp xỉ hàm phi tuyến. Mục 1 giới thiệu một lớp hệ mờ Mamdani cụ thể để xấp xỉ hàm khả tích. Mục 2 tiếp tục bài báo [11], trình bày hai thuật toán luyện cấu trúc mạng nơron để xấp xỉ hàm thực phi tuyến.

ĐẶT VẤN ĐỀ

Một kết quả nổi bật của hệ mờ và mạng nơron nhân tạo là khả năng xấp xỉ phổ dụng tới các hàm số liên tục bất kỳ (xem [3, 4, 6, 9]). Tuy nhiên các kết quả này chỉ được chứng minh dạng toán học dưới góc độ giải tích toán học (chủ yếu dùng định lý Stone–Weierstrass ([7]). Trong bài báo này chúng tôi trình bày hai tiếp cận tới cũng bài toán đó, nhưng dưới góc độ cụ thể hơn và có tính kiến thiết hơn. Hơn nữa trong tiếp cận bằng mạng nơron, các mạng truyền thẳng được xây dựng bằng những thuật toán luyện cấu trúc và được xấp xỉ cho hàm phi tuyến bất kỳ. Một vài kết quả chạy thử nghiệm các thuật toán sẽ minh hoạ cuối bài báo.

1. TIẾP CẬN BẰNG MỘT LỚP HỆ MỜ MAMDANI

Kết quả trình bày dưới đây trực tiếp phát triển kết quả trong [4, Mục 8.4]. Trong phần này chúng ta sẽ xấp xỉ tới hàm khả tích bằng một lớp hệ mờ Mamdani thông qua hàm tuyến tính từng mảnh.

Cho hàm số $S : R_+^2 \to R$. Ta ký hiệu $Supp(S) = \{(t, s) : S(t, s) \neq 0\}$.

Định nghĩa 1.1. Hàm S được gọi là tuyến tính từng mảnh nếu nó liên tục và thỏa các điều kiện sau:

i) Có một số nguyên dương m sao cho $Supp(S) \subset D$, ở đây $D = [0, m]^2$.

ii) Có một số nguyên dương n_0 và các đa giác $\Delta_1, ..., \Delta_{n_0}$ sao cho $\Delta_1 \cup ... \cup \Delta_{n_0} = D$, và S tuyến tính trên các Δ_i, tức là ta có $S(t.s) = \lambda_i t + \beta_i s + \rho_i$, với mọi $(t, s) \in \Delta_i$, với $i = 1, ..., n_0, \lambda_i, \beta_i, \rho_i$ là các hằng số.

Với các hàm tuyến tính từng mảnh ta có mệnh đề sau:

Mệnh đề 1.2. *Cho μ là độ đo trên σ-đại số của R_+^2 và $f : R_+^2 \to R$ thỏa mãn điều kiện*

$\int\limits_{R_+^2} |f(t,s)|^2 d\mu < +\infty$. *Khi đó với mỗi* $\varepsilon > 0$, *tồn tại* $m \in N$ *và hàm tuyến tính từng mảnh trên* S *có* $Supp(S) \subset [0,m]^2$ *sao cho:*

$$\left\{ \int\limits_{R_+^2} |f(t,s) - S(t,s)|^2 d\mu \right\}^{\frac{1}{2}} < \varepsilon.$$

Thật vậy, do hàm $|f(t,s)|^2$ khả tích nên với $\varepsilon > 0$ đã cho có thể chọn m_0 để cho

$$\int\limits_{\{(t,s)\in R_+^2 : \|(t,s)\| \geqslant m_0\}} |f(t,s)|^2 d\mu < \frac{\varepsilon^2}{2}.$$

Bây giờ ta chọn m sao cho $D \equiv [0,m]^2 \supset \{(t,s) \in R_+^2 : \|(t,s)\| \leqslant m_0\}$.

Do $f(t,s)$ khả tích nên có thể xấp xỉ trong L^2 bởi hàm liên tục trên D với độ chính xác tùy ý và ta có thể coi $f = 0$ ngoài D và trên biên ∂D.

Công việc xây dựng hàm tuyến tính từng mảnh có thể tiến hành theo các bước sau:

i) Trước tiên ta chia $[0,m]$ thành k đoạn bằng nhau, như vậy D được chia thành k^2 hình vuông nhỏ.

ii) Nối các đỉnh đối diện của hình vuông thành $2k^2$ tam giác cân.

iii) Nối 3 đỉnh tương ứng của mỗi tam giác cân trên mặt $z = f(t,s)$ tạo thành một tam giác và ta thu được mặt nối tuyến tính từng mảnh $z = S(t,s)$.

Như vậy , cuối cùng chỉ cần để ý rằng có $n(f) \in N$, sao cho với mọi $n > n(f)$ ta có

$$f(p_1\frac{m}{n}, p_2\frac{m}{n}) = S(p_1\frac{m}{n}, p_2\frac{m}{n}); \forall p_1, p_2 = 0, 1, ..., n.$$

Bây giờ chúng ta sẽ định nghĩa một lớp hệ mờ dạng Mamdani cụ thể.

Cho $m, n \in N$ được chọn thích hợp. Ta sẽ chia đoạn $[0,m]$ thành n đoạn bằng nhau.

Lớp hệ mờ sau đây sẽ dùng các số mờ thỏa Định nghĩa 1.3.

Định nghĩa 1.3. Với $i = 1, 2$, $j = 0, 1, ..., n$, $A_{i,j}$ là các số mờ trên R_+ thỏa các điều kiện sau:

i) Hàm thuộc $A_{i,j}(.)$ khả tích trên R_+ và $A_{i,j}(t) = 0$ với $t > m$.

ii) $A_{i,j}(\frac{mj}{n}) = 1$, và nếu với mỗi j sao cho $j, j-1, j+1 \in \{0, 1, ..., n\}, t \in [0,m]$, nếu $\min(A_{i,j-1}(t), A_{i,j+1}(t)) > 0$ thì $A_{i,j}(t) > 0$.

iii) Với $i = 1, 2, \forall t \in [0,1]$ thì $\sum\limits_{j=0}^{n} A_{i,j}(t) = 1$.

iv) Có $c_0 \in N$, không phụ thuộc m, n sao cho với $i = 1, 2$ và $\forall t \in [0,m]$, ta có

$$\operatorname{card}(\{j : A_{i,j}(t) > 0\}) \leqslant c_0.$$

Để gọn trình bày ta sẽ sử dụng t-chuẩn dạng tích $T(x,y) = x.y$, và ký hiệu $T_{p1,p2}(t_1,t_2) = A_{1,p_1}(t_1).A_{2,p_2}(t_2)$, với mọi $p_1, p_2 \in \{0, 1, ..., n\}, (t_1,t_2) \in [0,m]^2$.

Định nghĩa 1.4. (Lớp hệ mờ sẽ dùng)

Ký hiệu $B_r(y)$ là số mờ trên R thỏa điều kiện $B_r(r(p_1,p_2)) = 1$, ở đây $r : R_+^2 \to R$.

Cho với mỗi cặp $(p_1,p_2) \in \{0, 1, ..., n\}, (t_1,t_2) \in [0,m]^2, A_{1,p_1}(t_1), A_{2,p_2}(t_2)$ là các số mờ thỏa mãn các điều kiện trong Định nghĩa 1.3.

Hệ mờ Mamdani cho bởi $(n+1)^2$ luật mờ $R(p_1,p_2)$, với $(p_1,p_2), p_i = 0,1,...,n, i = 1,2$, có dạng

$$R(p_1,p_2): IF\ t_1\ is\ A_{1,p_1}\ AND\ t_2\ is\ A_{2,p_2}\ THEN\ y\ is\ B_r(p_1,p_2)(y).$$

Quy trình suy diễn trong hệ mờ này theo quy tắc max-prod như sau.

Với mỗi luật mờ ta có một tập mờ mới trên $R_+ \times R_+ \times R$ với hàm thuộc có dạng $W_{p_1,p_2}(t'_1,t'_2,z) = A_{1,p_1}(t'_1).A_{2,p_2}(t'_2).B_r(p_1,p_2)(z)$.

Với tín hiệu vào dạng $I_0(t'_1,t'_2)$ dạng đơn giản $I_0(t_1,t_2) = 1$, còn $I_0(t'_1,t'_2) = 0$ khi $(t'_1,t'_2) \neq (t_1,t_2)$ ta sẽ có phép hợp thành $*$

$$\begin{aligned} I_0 * W_{p_1,p_2}(z) &= \sup_{t'_1,t'_2 \in R_+} \{I_0(t'_1,t'_2).A_{1,p_1}(t'_1).A_{2,p_2}(t'_2).B_{r(p_1,p_2)}(z)\} \\ &= T_{p_1,p_2}(t_1,t_2).B_{r(p_1,p_2)}(z). \end{aligned}$$

Sử dụng phép giải mờ theo trọng tâm ta có kết quả đầu ra:

$$M_{n,m}(t_1,t_2) = \frac{\sum\limits_{p_1,p_2=0}^{n} y^*_{p_1,p_2}.(I_0 * W_{p_1,p_2})(y^*_{p_1,p_2})}{\sum\limits_{p_1,p_2=0}^{n} (I_0 * W_{p_1,p_2})(y^*_{p_1,p_2})}$$

ở đây $y^*_{p_1,p_2} = r(p_1,p_2)$ là điểm đảm bảo $B_{r(p_1,p_2)}(y^*_{p_1,p_2}) = 1$. Do đó $(I * W_{p_1,p_2})(r(p_1,p_2)) = T_{p_1,p_2}(t_1,t_2))$, và như vậy đầu ra của hệ có dạng, nếu để ý đẳng thức $\sum\limits_{p_1,p_2=0}^{n} T_{p_1,p_2}(t_1,t_2) = 1$,

$$M_{n,m}(t_1,t_2) = \sum_{p_1,p_2=0}^{n} T_{p_1,p_2}(t_1,t_2).r(p_1,p_2), t_1,t_2 \in R_+.$$

Để cho gọn ta sẽ gọi đây là *hệ mờ Mamdani hai chiều.*

Mệnh đề 1.5. *Cho $S : R_+^2 \to R$ là hàm tuyến tính từng mảnh. Khi đó với mỗi $\varepsilon > 0$ bất kỳ sẽ có một hệ mờ Mamdani 2 chiều $M_{n,m}(.,.)$ sao cho*

$$\left\{ \int_{R_+^2} \left|S(t,s) - M_{n,m}(t,s)\right|^2 d\mu \right\}^{1/2} < \varepsilon.$$

Thật vậy, ta chọn m sao cho $Supp(S) \subset [0,m]^2$. Chia $[0,m]$ thành n đoạn con. Chọn $r : R_+^2 \to R$ bằng biểu thức $r(t,s) = S(\frac{mt}{n}, \frac{ms}{n})$ và chọn hệ mờ dạng

$$M_{n,m}(t_1,t_2) = \sum_{p_1,p_2=0}^{n} T_{p_1,p_2}(t,s).r(p_1,p_2).$$

Phần còn lại chỉ là ước lượng sai số.

Phối hợp với các kết quả trên ta có mệnh đề xấp xỉ sau.

Mệnh đề 1.6. *Giả sử μ, f thỏa mãn diều kiện của Mệnh đề 1.2 với $\int |f(t,s)|^2 d\mu < +\infty$ và cho $\varepsilon > 0$ nhỏ tùy ý. Giả sử chọn $m \in N$ sao cho*

$$\int_{\{(t,s)\in R^2: t^2+s^2 > m^2/2\}} |f(t,s)|^2 d\mu < \varepsilon^2/2.$$

Khi đó có số nguyên n_0 và tồn tại hệ mờ Mamdani cho bởi

$$M_{n,m}(t,s) = \sum_{p_1,p_2=0}^{n} T_{p_1,p_2}(t,s).f(mp_1/n, mp_2/n)$$

sao cho với $n > n_0$ thì $\left(\int\limits_{R_+^2} |f(t,s) - M_{n,m}(t,s)|^2 d\mu\right)^{1/2} < \varepsilon.$

Nhận xét. Quy trình xấp xỉ đã trình bày ở trên có tính kiến thiết và có triển vọng mở rộng cho hàm nhiều biến và hệ mờ với các luật mờ có nhiều biến vào.

2. XÂY DỰNG MẠNG NƠRON ĐỂ XẤP XỈ HÀM PHI TUYẾN

Tính xấp xỉ phổ dụng của mạng nơron nhân tạo tới các hàm thực liên tục đã trình bày trong [8-10]. Bây giờ chúng ta xây dựng một kiến trúc mạng và thực hiện thuật toán luyện trên mạng đó để thu được mạng xấp xỉ tùy ý tới các hàm phi tuyến, điều ấy có nghĩa chúng ta cần luyện cấu trúc để thu được mạng nơron xấp xỉ. Trong bài báo [11] chúng tôi đã trình bày một số kiến trúc mạng và một thuật toán luyện mạng nơron, ta sẽ quy ước đó là thuật toán T_1.

Chúng ta xác định tập dữ liệu huấn luyện mạng là D, $(p_q, t_q) \in D$ là các mẫu dữ liệu với $q = 1, 2, ..., K$ trong đó đầu vào $p_q \in R^N$ và đầu ra mong muốn $t_q \in R$.

Để khắc phục hiện tượng quá khớp (overfit) chúng ta chia tập D một cách ngẫu nhiên thành 10 phần bằng nhau: lấy 9 phần tạo thành tập huấn luyện T, còn lại một phần dùng để làm tập đánh giá X.

2.1. Xây dựng mạng nơron theo hướng thêm vào (constructive learning)

Cách tiếp cận xây dựng mạng nơron theo hướng thêm vào cố gắng tìm kiếm một mạng nơron hoạt động tốt. Phương pháp này bắt đầu với một mạng nơron nhỏ và sau đó thêm các nút ẩn cùng với các trọng số liên kết cho đến khi đạt được yêu cầu. Cách tiếp cận này không tìm ra một kiến trúc mạng nhỏ nhất, thường ở lớp NP-khó, mà chỉ dừng lại ở việc tìm một mạng phù hợp cho bài toán.

Theo một số tác giả (xem tài liệu trích dẫn trong [11]) và một số kết quả thực nghiệm, cách tiếp cận này sẽ có một số thuận lợi sau đây:

Thứ nhất, cách tiếp cận này sẽ xuất phát từ một mạng nơron có kích thước nhỏ cho nên rất dễ chọn và việc tính toán sẽ tốn ít thời gian hơn, trong khi đó ở cách tiếp cận trước việc chọn một mạng đủ lớn thường khó thực hiện.

Thứ hai, có thể có nhiều mạng nơron với kích thước khác nhau sẽ cùng cho một kết quả của bài toán, tuy nhiên thuật toán này sẽ luôn cho ta một mạng nơron nhỏ nhất trong số đó. Các hàm của các nút ẩn riêng biệt sẽ dễ quan sát hơn. Hơn nữa, bằng việc tìm kiếm mạng nơron nhỏ, số lượng tập dữ liệu huấn luyện yêu cầu sẽ giảm xuống.

Thứ ba, thuật toán T_1 luôn phải xác định sự thay đổi sai số khi một nút ẩn hoặc trọng số bị loại bỏ. Tuy nhiên sự thay đổi này có thể được xấp xỉ cho hiệu quả tính toán, vì thế sẽ dẫn đến sai số lớn, đặc biệt khi có nhiều nút nơron bị loại.

Tuy nhiên, cũng có những cản trở đối với cách tiếp cận thêm vào. Ví dụ, phải xác định khi nào thì dừng việc thêm các nút ẩn vào. Hơn nữa, có nhiều thuật toán lại sử dụng tiếp cận ăn tham để xây dựng mạng, có thể dẫn đến trường hợp tối ưu cục bộ của giải pháp.

Trong phần này chúng tôi trình bày thuật toán xây dựng mạng nơron cho bài toán tiệm cận hàm phi tuyến dưới dạng bài toán tìm kiếm. Ta sẽ sử dụng các định nghĩa như trong [11]. Ký hiệu:

n là số lượng nút ẩn trong mạng nơron,

C là đồ thị liên kết nơron trong mạng, là sự liên kết giữa các nút vào, nút ra và nút ẩn,

Γ là các dạng hàm của các nút ẩn,

W là các tham số của toàn mạng, bao gồm trọng số liên kết giữa các nơron, độ lệch và các tham số trong Γ.

Một bộ bốn (n, C, Γ, W) sẽ chỉ ra một mạng duy nhất và một hàm f tương ứng.

Theo kiến trúc mạng nơron, chúng ta thấy bộ bốn thành phần n, C, Γ và W không hoàn toàn độc lập nhau. Ví dụ, với $C = (V, E)$ tương ứng là các đỉnh (nút nơron trong mạng) và các cạnh (các liên kết giữa các nơron), ta có

$$n = |V| - n_{\text{in}} - n_{\text{out}}$$

trong đó n_{in} là các nơron đầu vào, n_{out} là các nơron đầu ra. Hơn nữa chúng ta cho C và Γ thì W có thể được xác định bằng cách sử dụng một thuật toán huấn luyện tham số nào đó, ví dụ BP. Ta giả sử rằng W có thể xác định duy nhất từ C và Γ.

Ta xem xét mỗi đồ thị C (là một trạng thái trong không gian tìm kiếm) là bộ gồm hai thành phần là V và E, ký hiệu $s = C = (V, E)$.

Chúng ta xem mỗi dữ liệu đầu vào thu thập được là một véc tơ p, thể hiện của một giá trị của biến ngẫu nhiên nhiều chiều X, đầu ra t là thể hiện của biến ngẫu nhiên Y. Một xấp xỉ khá tốt đó là lấy kỳ vọng trên tập dữ liệu huấn luyện với kích thước là N mà người quan sát được.

$$\text{err}(f_{n,N}) = E(\|f - f_{n,N}\|).$$

Chú ý rằng sai số trên vẫn không tính được bởi chúng ta không biết rõ hàm f như thế nào. Vì vậy chúng ta sử dụng thuật toán thêm vào này để có được một sai số $\text{err}(f_{n,N})$ tốt nhất.

2.2. Thuật toán xây dựng mạng

Với kiến trúc truyền thẳng 3 lớp: lớp vào, lớp ẩn, lớp ra; số nơron ở lớp vào và lớp ra có thể xác định thông qua tập dữ liệu mẫu. Thuật toán thể hiện việc xây dựng mạng đó chính là xác định số nơron trong lớp ẩn.

Như đã trình bày, vì chọn C là liên kết đầy đủ giữa các nơron ở các lớp và ở mỗi trạng thái ta sinh trạng thái tiếp bằng cách thêm một nơron vào V (trong lớp ẩn), dẫn đến ánh xạ trạng thái là đơn ánh. Tuy nhiên ta cũng có thể chọn giải pháp là đồ thị liên kết C không đầy đủ và số nơron có thể không thêm ở trạng thái tiếp. Hàm kích hoạt của mỗi nơron trong lớp ẩn là như nhau và ta chọn là tansig:

$$\text{tansig}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Trong thuật toán này sử dụng hai tham số α và β, α để xác định dừng cho thuật toán luyện tham số được sử dụng BP, β để xác định dừng cho quá trình tìm kiếm.

Thuật toán T_2

Vào: Tập dữ liệu huấn luyện $D = \{(p_i, t_i)$, với $i = 1, 2, ..., Q\}$.

Ra: Một mạng nơron NN đã được xây dựng.

Các bước thực hiện:

Bước 1. Tạo một mạng nơron tương ứng với trạng thái xuất phát S, có số nơron $V = N_i + N_o + 1$ (số nơron ở lớp vào + số nơron ở lớp ra + 1 nơron lớp ẩn), và liên kết đầy đủ E.

$$N_i = |p_i| \text{ và } N_o = |t_i|.$$

Bước 2. Huấn luyện mạng nơron S theo thuật toán BP, với hệ số giảm sai số là α, tức là nếu sai số của mạng sau mỗi lần luyện không nhỏ hơn α lần so với sai số của mạng trước đó thì sẽ dừng thuật toán BP.

Bước 3. Đặt $k = 1$ và tính sai số của mạng trên tập D

$$E(S_1) = \sum_D (t_i - a_i)^2.$$

Bước 4. Xác định trạng thái tiếp theo S_{k+1} bằng cách thêm một nơron ở lớp ẩn và tạo liên kết đầy đủ với các nơron ở lớp vào và lớp ra cho nơron mới thêm đó, các liên kết mới có trọng số ngẫu nhiên.

Bước 5. Thực hiện thuật toán BP luyện tham số cho mạng S_{k+1} theo hệ số giảm α.

Bước 6. Tính sai số $E(S_{k+1})$ và kiểm tra, nếu $E(S_{k+1}) < \beta.E(S_k)$ thì sẽ đặt $k = k + 1$ và quay lại Bước 4, ngược lại sẽ sang Bước 7.

Bước 7. Đặt NN $= S_{k+1}$ và dừng thuật toán.

Trong thuật toán này không sử dụng tập dữ liệu X để kiểm tra và khắc phục hiện tượng overfit, tuy nhiên chúng ta có thể thực hiện điều đó một cách đơn giản như Thuật toán T1.

2.3. Một thuật toán kết hợp T_1 và T_2

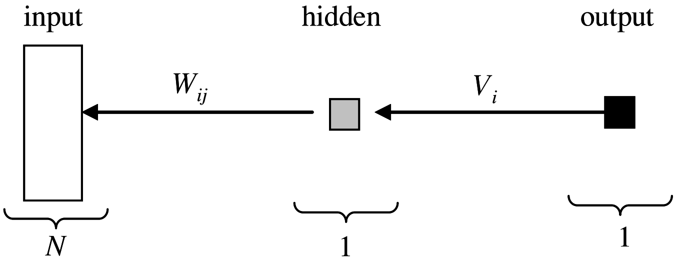
Một giải pháp tổng hợp là chúng ta sử dụng thuật toán loại bỏ T_1 trong [11], tuy nhiên trong bước xác định các nơron ở lớp ẩn ta sẽ thay bằng thuật toán T_2. Như chúng ta biết thuật toán thêm vào hiệu quả hơn ở các điểm sau:

- Việc chọn một kiến trúc mạng xuất phát là đơn giản.
- Việc huấn luyện mạng trong các bước của thuật toán sẽ nhanh hơn.
- Thuật toán sẽ cho một kiến trúc mạng đủ nhỏ để thực hiện bài toán.

Do vậy việc thay thế này sẽ làm cho thuật toán đơn giản và thực hiện nhanh hơn.

Các mẫu dữ liệu thu thập được bởi người dùng có thể có các thuộc tính phụ thuộc với nhau, dẫn đến nếu chúng ta chọn số lượng đầu vào cho mạng bằng số lượng các thuộc tính của mẫu dữ liệu thì dẫn đến việc dư thừa làm cho mạng hoạt động kém. Do vậy chúng ta phải thực hiện loại bỏ các dư thừa ở đầu vào theo thuật toán loại bỏ ở trên.

Chọn một kiến trúc mạng xuất phát chỉ với 3 lớp, lớp vào có số nơron bằng số thuộc tính dữ liệu thu thập được, số nơron ở lớp ra dựa vào bài toán cần thực hiện (trong bài chúng tôi chọn 1 nơron đầu ra cho đơn giản), số nơron ở lớp ẩn là 1. Kiến trúc mạng xuất phát được thể hiện trên Hình 1.



Hình 1

Ta gọi tập mẫu dữ liệu thu thập được là $D, (p_q, t_q) \in D$, đánh giá sai số của mạng trên tập dữ liệu là

$$\mathrm{ET} = \frac{1}{|T|} \sum_{(p_q, t_q) \in T} |\tilde{t}_q - t_q|, \tag{1}$$

trong đó $\tilde{t}_q$ là đầu ra của mạng, t_q là đầu ra mong muốn.

Trong thuật toán này chúng ta sử dụng hai tham số ε, ϖ là các hệ số giảm. Thuật toán huấn luyện mạng nơron theo BP sẽ dừng khi sai số lần sau không nhỏ hơn sai số lần trước ε lần. Việc thêm các nơron sẽ dừng khi sai số của mạng lần sau không nhỏ hơn lần trước ϖ lần, $0 \leqslant \varepsilon \leqslant 1$ và $0 \leqslant \varpi \leqslant 1$.

Thuật toán T_3

Vào: Tập T, một mạng nơron xuất phát theNET.

Ra: Mạng nơron đã được huấn luyện theNET để thực hiện bài toán hồi quy.

Các bước thực hiện:

Bước 1: Huấn luyện mạng nơron xuất phát theNET

a) Thực hiện thuật toán luyện mạng theNET cho đến khi sai số của mạng không giảm theo một hệ số ε.

b) Tính sai số ET của mạng theo công thức (1).

Bước 2: Thêm các nơron trong lớp ẩn.

a) Lưu cấu hình mạng hiện tại: oldNET = theNET.

b) Thêm một nơron vào lớp ẩn của mạng theNET và đặt các giá trị ban đầu cho các trọng số liên kết ứng với nơron vừa thêm. Đặt $H = H + 1$.

c) Thực hiện luyện mạng theNET cho đến khi sai số mạng không giảm theo một hệ số ε (ε là một hệ số giảm nhỏ hơn 1). Tính sai số EE của mạng vừa luyện theNET theo công thức (1).

d) Nếu EE/ET $< \varpi$ (ϖ là một hệ số giảm nhỏ hơn 1) thì chấp nhận mạng mới theNET và quay lại Bước 2a.

Ngược lại ta sẽ khôi phục cấu hình mạng cũ và sang Bước 3: theNET = oldNET.

Bước 3: Loại bỏ các nơron ở lớp vào dư thừa.

a) Tính sai số mạng ETbest theo (1), lưu cấu hình mạng oldNET = theNET.

b) Với mỗi nơron vào $j = 1, 2, ..., N$, đặt $w_{ij} = 0$ cho tất cả các nơron lớp ẩn $i = 1, 2, ..., H$ và tính ET_j theo (1).

c) Tìm $w_{in} = \min\{\mathrm{ET}_j\}$ và đặt $w_{in} = 0$. Huấn luyện lại mạng nơron và tính ET.

d) Nếu ET $\leqslant (1+\alpha)$ETbest thì: Loại bỏ nút vào thứ n trên, đặt ETbest $= \min\{\text{ET}, \text{ETbest}\}$, đặt $N = N - 1$ và quay lại Bước 3b.

Ngược lại khôi phục cấu hình mạng trước đó theNET = oldNET và kết thúc thuật toán.

Trong thuật toán này chúng tôi không chia tập dữ liệu mẫu thành 2 tập ET và EX, tuy nhiên chúng ta có thể thực hiện điều này để khắc phục hiện tượng overfit trong huấn luyện mạng.

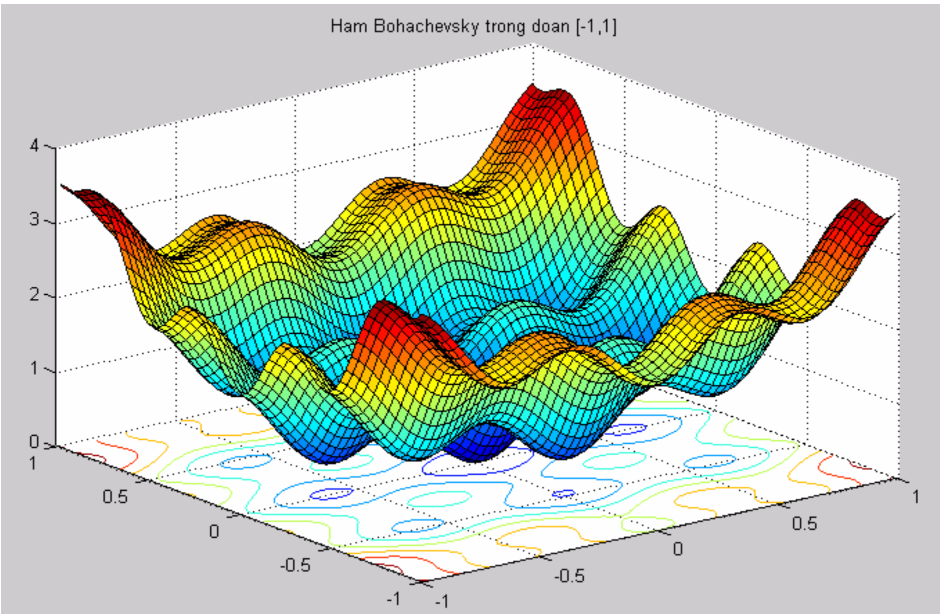
2.4. Kết quả thử nghiệm

Trong phần này chúng tôi sẽ thử nghiệm cho hai thuật toán tiêu biểu đó là thuật toán T1 và T2, các thử nghiệm được thực hiện trên 3 hàm từ đơn giản đến phức tạp. Ký hiệu 3 hàm như sau:

$$
\begin{aligned}
&F1 : f(x) = \sin(x),\\
&F2 : f(x) = \sqrt{1+\sin(x)} * (\cos(x) + 3 * x - x^2) + 7,\\
&F3 : f(x, y) = x^2 + 2y^2 - 0{,}3\cos(3\pi.x) - 0{,}4\cos(4\pi.y) + 0{,}7.
\end{aligned}
$$

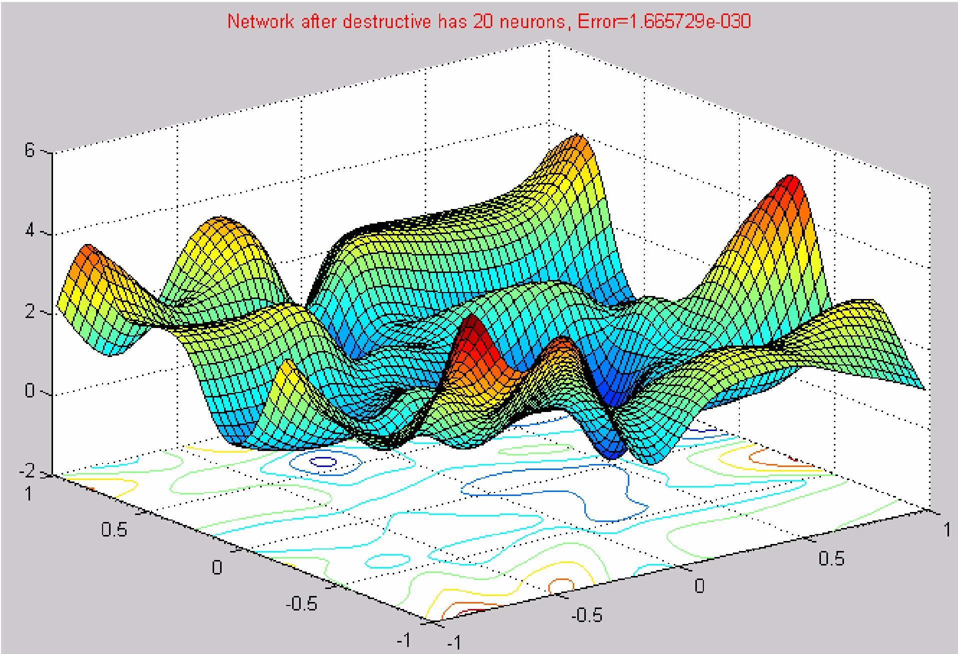
Tập dữ liệu mẫu được lấy ngẫu nhiên, với độ lớn các tập mẫu khác nhau. Tập mẫu dữ liệu sau khi lấy sẽ được làm nhiễu hóa theo một hệ số nhiễu α tự đặt. Chúng tôi tiến hành trên 2 tập có kích thước tương ứng là 50 và 150. Sau đây là kết quả thử nghiệm với hàm $F3$.

Hàm $F3$ gọi là hàm Bohachevsky, có 2 biến vào và có nhiều cực trị như hình vẽ sau, giới hạn trong đoạn $[-1, 1]$.



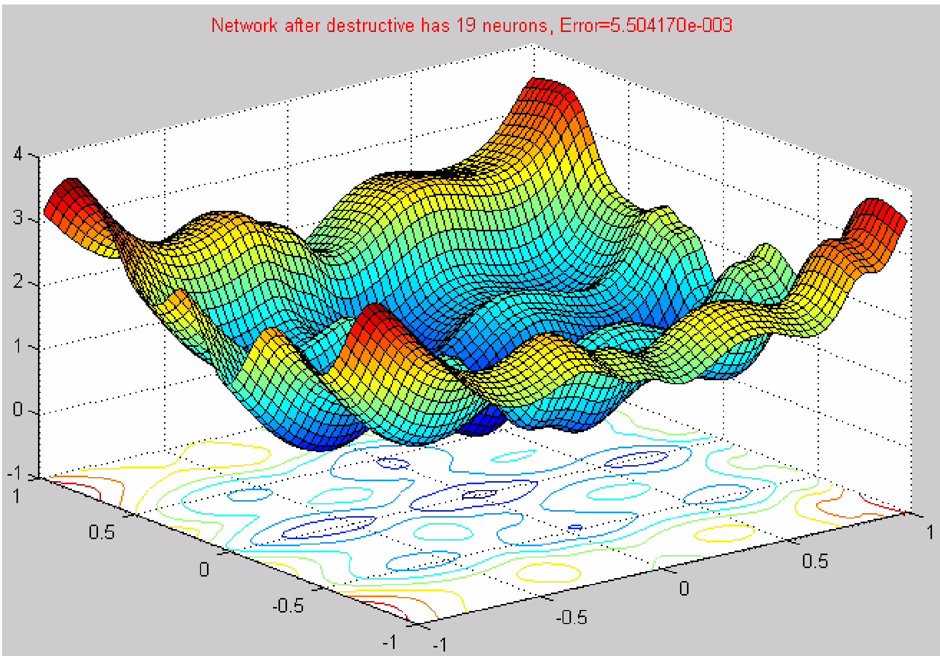
Hình 2. Hàm Bohachevsky trên đoạn $[-1, 1]$

Với trường hợp tập dữ liệu huấn luyện có 50 mẫu, tập mẫu được làm nhiễu với hệ số nhiễu là 0,1 và mạng nơron có 20 nơron trong lớp ẩn ta có kết quả số nơron trong lớp ẩn không thay đổi như hình sau:



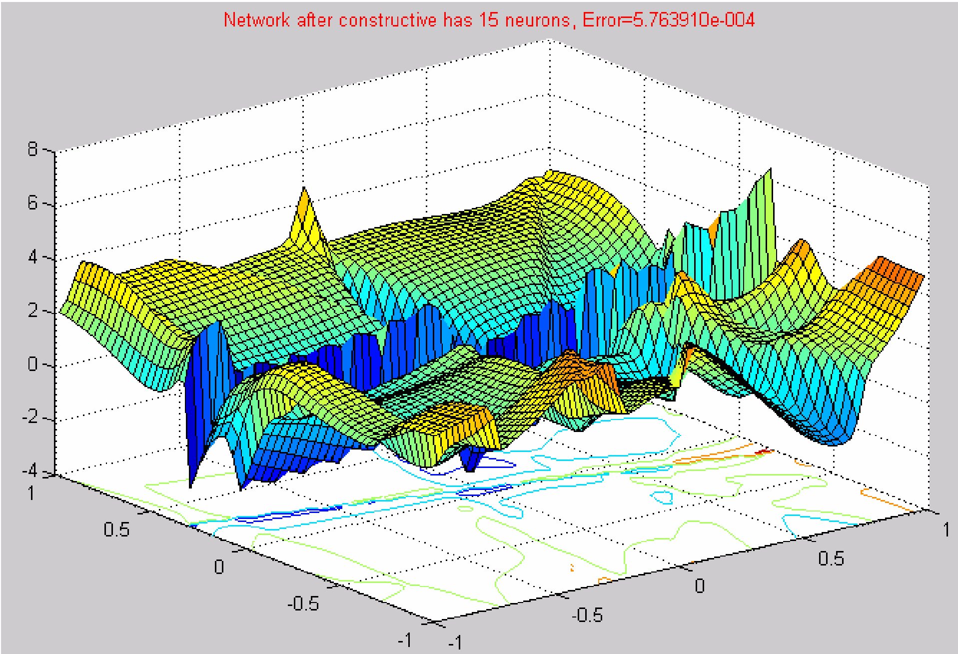
Hình 3. Xấp xỉ hàm Bohachevsky bằng thuật toán T1 với 50 mẫu

Với trường hợp tập huấn luyện có 150 mẫu, kết quả sẽ tốt hơn như hình vẽ sau, số nơron trong lớp ẩn còn lại 19.

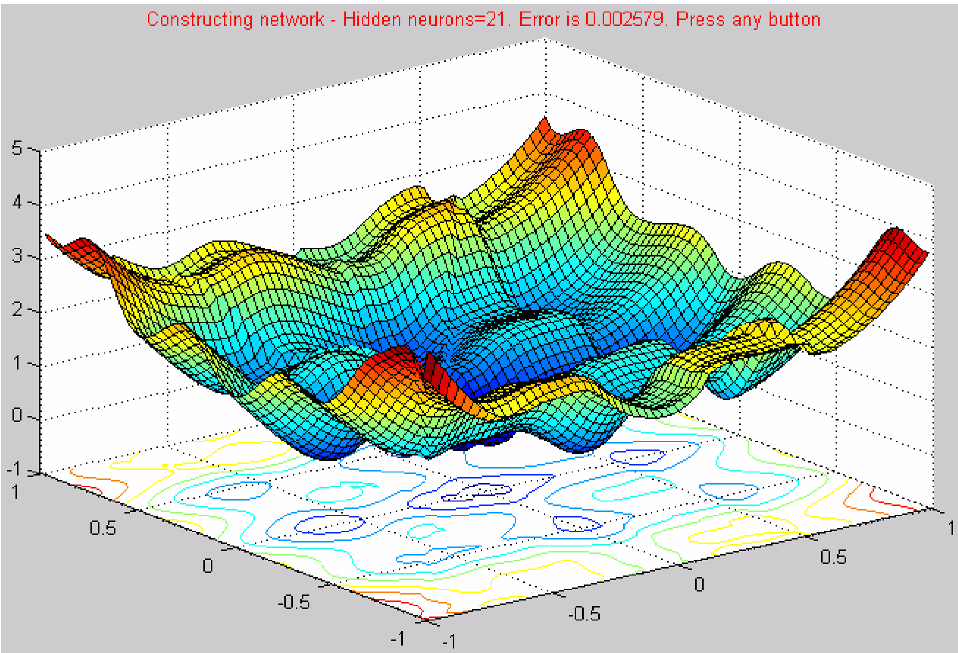


Hình 4. Xấp xỉ hàm Bohachevsky bằng thuật toán T1 với 150 mẫu

Đối với thuật toán T2 và trường hợp tập dữ liệu có 50 mẫu ta có kết quả sau đây.



Hình 5. Xấp xỉ hàm Bohachevsky bằng thuật toán T2 với 50 mẫu



Hình 6. Xấp xỉ hàm Bohachevsky bằng thuật toán T2 với 150 mẫu

Kết quả hàm xấp xỉ của thuật toán T1 và T2 với tập dữ liệu có 150 mẫu sẽ cho kết quả tốt hơn. Tuy nhiên với thuật toán T1 nếu ta xuất phát với số nơron quá lớn sẽ dẫn đến tình trạng thuật toán dừng quá sớm, cho kết quả sai số nhỏ trên tập dữ liệu huấn luyện nhưng rất nhạy cảm (thay đổi bất thường) với các dữ liệu ngoài tập huấn luyện. Thuật toán T2 sẽ luôn dừng lại ở một cấu hình mạng nơron mà cho kết quả sai số đủ tốt và đối với các dữ liệu ngoài tập học.

TÀI LIỆU THAM KHẢO

[1] Bùi Công Cường, Nguyễn Doãn Phước, *Hệ mờ, mạng nơron và ứng dụng*, Nhà xuất bản Khoa học Kỹ thuật, Hà Nội, 2001.

[2] Hung T. Nguyen and Michio Sugeno, *Fuzzy Systems Modeling and Control*, Kluwer Academic Pub., Massachusetts, 1998.

[3] V. Kreinovich, G. C. Mouzouris, Hung T. Nguyen, Fuzzy rule based modeling as a universal approximation tool, trong [2], 135–195.

[4] Hung T. Nguyen and E. Walker, *A First Course of Fuzzy Logic*, 2^{nd} edition, Chapman&Hall/CRC, London, 2000.

[5] E. H. Ruspini, P. P. Bonissone, W. Pedrycz,(Eds.), *Handbook of Fuzzy Computation,* Inst. of Physics Pub., Bristol, 1998.

[6] J. J. Buckley, Approximation aspects of fuzzy models, *Handbook of Fuzzy Computation,* Inst. of Physics Pub., Bristol, 1998 (phần C2.5).

[7] Walter Rudin, *Functional Analysis*, 2^{nd}edition, McGraw-Hill, Inc., New York, 1991.

[8] S. Haykin, *Neural Networks a Comrehenssive Foundation*, 2^{nd}edition, Prentice Hall, London, 1999.

[9] Martin T. Hagan, Howard B. Demuth, Mark Beale, *Neural Network Design*, PWS Publishing Company, 1996.

[10] L. Jin, M. M. Gupta, P. N. Nikiforuk, Neural networks and fuzzy basic functions for functional approximation, *Fuzzy Logic and Intelligent Systems*, H.Li and M. Gupta (Ed.), Kluwer Academic Pub., London, 1994, 17–67.

[11] Bùi Công Cường, Dương Thăng Long, Lê Quang Phúc, Hai tiếp cận dùng mạng nơron trong khai phá dữ liệu, *Kỷ yếu Hội thảo khoa học quốc gia lần thứ nhất,* Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin, NXB Khoa học và Kỹ thuật, Hà Nội, 2004, 102–115.

Nhận bài ngày 20 - 2 - 2006