

## Hướng nghiên cứu tiếp theo

Bài toán nội suy luôn là một bài toán bắt nguồn từ các bài toán thực tế và đang có nhiều lĩnh vực ứng dụng. Việc vận dụng kiến trúc mạng và các thuật toán phải tùy thuộc vào tính đặc thù của từng bài toán, trên cơ sở đã nghiên cứu và hiểu rõ nó, để có thể cài đặt và hiệu chỉnh thích hợp. Theo hướng này, trong thời gian tới chúng tôi tìm hiểu các bài toán thực tế, bắt đầu từ các bài toán đã sử dụng mạng nơron RBF có hiệu quả đến các bài toán mới để nâng cao hiệu quả giải quyết ứng dụng. Bên cạnh đó, nhờ phát triển ứng dụng, chúng tôi hy vọng có các cải tiến và đề xuất các thuật toán, kiến trúc mạng mới thích hợp cho từng loại bài toán được nghiên cứu.

## MỞ ĐẦU

### 1. Tính cấp thiết của luận án

Nội suy hàm số là một bài toán cổ điển nhưng quan trọng trong giải tích số, nhận dạng mẫu và có nhiều ứng dụng rộng rãi. Do các khó khăn trong xử lý toán học và nhu cầu ứng dụng trước đây chưa nhiều nên bài toán nội suy nhiều biến mới được quan tâm nhiều trong 50 năm gần đây. Các phương pháp *k- lân cận gần nhất* và *hồi quy trọng số địa phương* cho một giải pháp đơn giản, dễ sử dụng cho bài toán này và đang là một công cụ tốt. Nhược điểm lớn nhất của phương pháp này là chỉ xác định được thu hẹp địa phương của hàm nội suy khi biết điểm cần tính giá trị hàm, do đó không áp dụng được cho các bài toán cần xác định trước hàm nội suy để sử dụng cho nội suy hàm số tại điểm tùy ý.

Trong 30 năm gần đây, mạng nơron nhân tạo là một công cụ tốt để giải quyết bài toán nội suy nhiều biến, trong đó thông dụng nhất là mạng MLP và mạng RBF. Nhược điểm chính của mạng MLP là thời gian huấn luyện lâu, chất lượng mạng tùy thuộc vào hiệu quả giải bài toán cực trị và đến nay chưa có phương pháp tốt nào để xác định kiến trúc *đủ tốt* cho mạng.

Như mạng MLP, mạng RBF chủ yếu dùng để xấp xỉ hàm (mà nội suy là bài toán riêng). Việc xác định tâm, số hàm cơ sở thế nào là tốt vẫn đang là bài toán mở. Trường hợp dữ liệu huấn luyện ít (Looney khuyên là nhỏ hơn 200) thì có thể lấy các mốc nội suy là tâm hàm RBF ở tầng ẩn và mạng này có thể cho nghiệm đúng của hệ phương trình nội suy nên gọi là mạng nội suy RBF. Chất lượng của mạng cũng tùy thuộc vào thuật toán huấn luyện. Nhu cầu ứng dụng thực tế trong các bài toán thời gian thực, đòi hỏi có các thuật toán huấn luyện mạng nhanh, chất lượng tốt và kiến trúc mạng phù hợp với dữ liệu lớn, đặc biệt khi thường xuyên có dữ liệu bổ sung.

### 2. Mục tiêu của luận án

Nghiên cứu và đề xuất các thuật toán huấn luyện mạng nội suy RBF hiệu quả cho các trường hợp có số mốc nội suy lớn và tìm giải pháp cho bài toán thời gian thực, đặc biệt là các bài toán có dữ liệu bổ sung thường xuyên.

### 3. Các đóng góp của luận án.

1) Đề xuất một thuật toán lặp hai pha huấn luyện mạng nội suy RBF. Phân tích toán học và kết quả thực nghiệm cho thấy thuật toán có những ưu điểm vượt trội so với những thuật toán thông dụng: dùng được khi số mốc nội suy lớn (hàng chục ngàn mốc), dễ ước lượng sai số huấn luyện, thời gian huấn

luyện ngắn, tính tổng quát cũng tốt hơn và dễ song song hoá. Đề xuất này đã được đăng trên tạp chí quốc tế *Signal Processing*.

2) Trong trường hợp bài toán nội suy có mốc cách đều, thay cho khoảng cách Euclide, chúng tôi dùng khoảng cách Mahalanobis thích hợp và cải tiến thuật toán hai pha thành thuật toán một pha. Phân tích toán học và kết quả thực nghiệm cho thấy thuật toán này cải thiện đáng kể chất lượng mạng so với thuật toán hai pha cả về thời gian huấn luyện và tính tổng quát. Kết quả này đã được công bố trong kỷ yếu hội thảo quốc gia về công nghệ thông tin.

3) Đề xuất kiến trúc mạng địa phương cho các bài toán thời gian thực, đặc biệt là bài toán động. Phân tích toán học và kết quả thực nghiệm cho thấy chất lượng mạng có nhiều ưu điểm nổi trội. Kiến trúc mới đề xuất này đã được công bố ở hội nghị quốc tế của IEEE và tạp chí quốc tế *International Journal of Data Mining, Modelling and Management Science*.

#### 4. Bố cục của luận án

Luận án được tổ chức thành năm chương. Chương 1 giới thiệu về bài toán nội suy hàm số và mạng nơron nhiều tầng truyền tới. Chương 2 giới thiệu về mạng nơron RBF. Chương 3 trình bày thuật toán hai pha mới (gọi là thuật toán HDH) để huấn luyện mạng nội suy RBF. Chương 4 trình bày thuật toán một pha mới áp dụng cho bài toán nội suy với mốc cách đều. Chương 5 trình bày đề xuất kiến trúc mới mạng địa phương RBF. Cuối cùng chúng tôi đưa ra một số kết luận và đề xuất các nghiên cứu tiếp theo.

## CHƯƠNG 1. NỘI SUY HÀM SỐ VÀ MẠNG NƠRON

### 1.1 Nội suy hàm số

#### 1.1.1 Bài toán nội suy tổng quát

Bài toán nội suy tổng quát được phát biểu như sau. Xét hàm nhiều biến chưa biết  $f: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$  nhưng xác định được một tập mẫu  $\{x^k, y^k\}_{k=1}^N$  trong đó  $x^k \in \mathbb{R}^n$ ,  $y^k \in \mathbb{R}^m$  (với mọi  $k=1, \dots, N$ ) thỏa mãn  $f(x^k)=y^k$ . Ta cần tìm hàm  $g$  có dạng *đủ tốt* đã biết thỏa mãn hệ phương trình nội suy:  $g(x^i) = y^i, \forall i = 1, \dots, N$ . Các điểm  $x^k$  sẽ được gọi là các mốc nội suy còn hàm  $g$  được gọi là hàm nội suy của  $f$ . Giá trị hàm nội suy tính được tại điểm  $x$  bất kỳ trên miền  $D$  gọi là giá trị nội suy của hàm  $f$  tại  $x$ .

## KẾT LUẬN

### Các kết quả đạt được

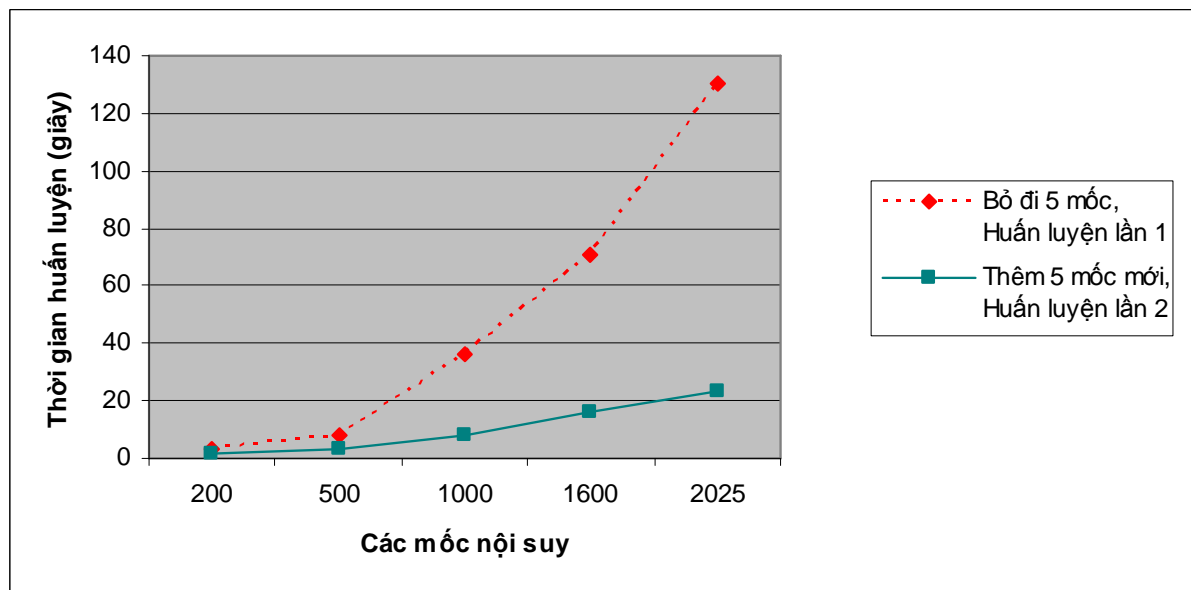
Trong thời gian qua, mặc dù có những hạn chế về thời gian và điều kiện làm việc, chúng tôi đã hoàn thành mục tiêu luận án. Các kết quả cụ thể đạt được như sau.

1) Đề xuất thuật toán hai pha đơn giản để huấn luyện mạng nội suy RBF. Pha thứ nhất xác định tham số độ rộng bán kính phù hợp với từng mốc nội suy, pha thứ hai dùng phương pháp lặp để tính trọng số tầng ra. Phân tích toán học và thực nghiệm chỉ ra rằng thuật toán luôn hội tụ, thời gian chạy chỉ phụ thuộc vào việc khởi gán giá trị ban đầu  $q, \alpha, \epsilon, \dots$ , phân bố của mốc nội suy và chuẩn của vectơ.

Qua kết quả thực nghiệm ta thấy thuật toán có ưu điểm nổi trội so với các phương pháp thông dụng hiện nay: thời gian huấn luyện mạng rất nhanh kể cả khi số mốc lớn, dễ dàng thực hiện và có hiệu quả cao, đánh giá sai số huấn luyện, điều khiển cân bằng giữa tốc độ hội tụ và tính tổng quát của mạng bằng việc điều chỉnh các tham số. Một ưu việt nữa của thuật toán là các bán kính tầng ẩn có thể huấn luyện độc lập và ở pha hai trọng số tầng ra cũng có thể huấn luyện độc lập, điều này làm cho chúng có thể song song hoá thuật toán.

2) Trong trường hợp các mốc nội suy cách đều nhau, để khai thác được ưu điểm phân bố này chúng tôi dùng metric Mahalanobis và cải tiến thuật toán hai pha thành thuật toán một pha. Nhờ các phân tích toán học, chất lượng mạng nội suy RBF được cải thiện rõ rệt so với mạng huấn luyện bằng thuật toán HDH và các thuật toán huấn luyện nhanh thông dụng. Không những có ưu thế về thời gian huấn luyện và tính tổng quát mà một hiệu quả dẫn xuất của mạng là có thể dùng cho trường hợp số mốc nội suy lớn hơn nhiều so với thuật toán HDH (và do đó với các thuật toán khác).

3) Đề xuất kiến trúc mạng mới, chúng được gọi là mạng RBF địa phương. Với kiến trúc này, thời gian huấn luyện mạng rất nhanh và tính xấp xỉ của mạng cũng tăng, thuật toán huấn luyện đơn giản và dễ song song hoá. Loại mạng này thích hợp cho các bài toán thời gian thực, trong đó đòi hỏi thời gian huấn luyện ngắn. Đặc biệt, đối với bài toán động, các mốc nội suy thường xuyên được bổ sung thì nhờ kỹ thuật cây  $k-d$  ta dễ dàng và nhanh chóng tái huấn luyện mạng.



Hình 5.9: Đồ thị so sánh thời gian huấn luyện tăng cường khi có mốc mới

Thời gian huấn luyện tăng cường rất nhỏ so với huấn luyện lại. Điều này là do chúng ta không phải tốn thời gian để xác định lại bán kính cho pha 1.

### 5.6 Nhận xét chung

Qua thực nghiệm ta thấy với mạng RBF địa phương như trên, thời gian huấn luyện mạng rất nhanh và tính xấp xỉ của mạng cũng tốt. Thời gian huấn luyện tăng cường cũng giảm đáng kể. Thuật toán huấn luyện đơn giản và dễ song song hoá. Loại mạng này thích hợp cho các bài toán thời gian thực, các bài toán động, trong đó các mốc nội suy thường xuyên được bổ sung.

### 1.1.2 Nội suy hàm một biến

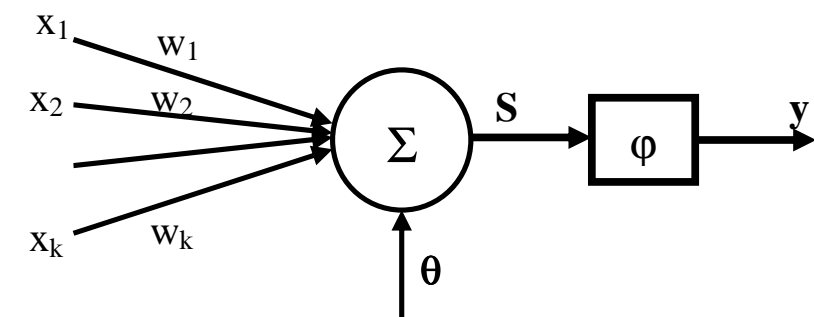
Với bài toán có  $n$  mốc nội suy, thông thường người ta tìm hàm  $g$  dưới dạng  $g(x) = \Phi(c_0, c_1, \dots, c_n, x)$  trong đó  $\Phi$  là hàm đã biết và các tham số  $c_k$  cần tìm để  $g$  thỏa mãn hệ phương trình nội suy. Khi  $g$  lấy trong lớp đa thức bậc  $n$  ta dễ dàng xác định được nhờ đa thức nội suy Lagrange  $L_n(x)$ . Trường hợp các mốc nội suy cách đều, với một phép đổi biến ta có đa thức nội suy Newton. Khi số mốc nội suy lớn, để khắc phục hiện tượng phù hợp trội (overfitting), người ta dùng các đa thức bậc thấp trên mỗi đoạn con của đoạn  $[a, b]$  và ghép tron đến mức cần thiết trên toàn đoạn làm hàm nội suy, các hàm nội suy này có tên gọi là hàm Spline.

### 1.1.3 Nội suy hàm nhiều biến

Bài toán nội suy nhiều biến là bài toán thường gặp trong thực tiễn. Các hướng tiếp cận chính để giải bài toán này: Thoạt tiên là dùng đa thức nhưng không hiệu quả do tính toán phức tạp. Tiếp theo sử dụng phương pháp  $k$ - lân cận gần nhất và hồi quy trọng số địa phương, những phương pháp này đơn giản dễ sử dụng nhưng cũng có những hạn chế với những bài toán cần xác định trước hàm nội suy. Mạng nơron nhân tạo là cách tiếp cận tốt để khắc phục những nhược điểm trên. Hiện nay mạng nơron nhân tạo là một công cụ hữu hiệu để giải các bài toán nội suy hàm nhiều biến. Trong đó thông dụng nhất là mạng MLP và mạng RBF.

### 1.2 Giới thiệu về mạng nơron nhân tạo

Các nghiên cứu về nơron sinh học khởi đầu bởi Ramón Y Cajál (1911) đặt tiền đề cho hướng mô phỏng mạng nơron tự nhiên gọi là các mạng nơron nhân tạo (về sau cũng gọi là mạng nơron). Các mạng nơron được xây dựng nhờ các kết nối theo nhiều cách của các nơron. Mỗi nơron là một đơn vị xử lý thông tin có kiến trúc như hình 1.4.



Hình 1.4. Mô hình một nơron nhân tạo

Tín hiệu vào kết hợp với trọng số kết nối nhờ hàm tổng và qua hàm kích hoạt cho đầu ra. Các mạng nơron với kiến trúc kết nối khác nhau cho các tính năng khác nhau. Trong đó mạng nơron nhiều tầng truyền tới là dạng thông dụng nhất và là công cụ hữu hiệu để xấp xỉ và nội suy hàm nhiều biến. Mạng perceptron nhiều tầng MLP (Multi-Layer Perceptrons) với hàm tổng của mỗi nơron là tổ hợp tuyến tính (hệ số là trọng số kết nối) của các tín hiệu vào và các hàm kích hoạt có thể chọn tùy theo các bài toán cụ thể. Với mạng nơron MLP cách huấn luyện lan truyền ngược sai số nhờ cực tiểu hóa tổng bình phương sai số theo một biến thể của phương pháp gradient đang được dùng rộng rãi. Tuy vậy, chọn kiến trúc tốt cho mạng MLP còn là bài toán mở, hơn nữa, việc huấn luyện nó dễ bị rơi vào các cực trị địa phương nên chất lượng thấp.

## CHƯƠNG 2. MẠNG NƠRON RBF

Mạng RBF là một mạng truyền tới 3 tầng (2 tầng nơron). So với mạng MLP, mạng RBF tỏ ra là một phương pháp hiệu quả và được ứng dụng rộng rãi để nội suy và xấp xỉ hàm nhiều biến. Mỗi nơron tầng ẩn là một hàm phi tuyến của khoảng cách giữa véc tơ vào  $\vec{X}$  và véc tơ tâm  $\vec{C}^j$  kết hợp với nơron  $j$  có bán kính tương ứng là  $\sigma_j$ .

### 2.1. Hàm cơ sở bán kính RBF và bài toán nội suy

Xét hàm nhiều biến  $f : D(\subset R^n) \rightarrow R^m$  cho bởi tập mẫu  $\{x^k, y^k\}_{k=1}^N$  ( $x^k \in R^n; y^k \in R^m$ ) sao cho:  $f(x^k) = y^k; k = 1, \dots, N$ .

Ta có thể giả thiết  $m=1$  và tìm hàm  $\varphi$  dạng

$$\varphi(x) = \sum_{k=1}^M w_k h(\|x - v^k\|, \sigma_k) + w_0, \quad \varphi(x^k) = y^k; \forall k = 1, \dots, N \quad (2.1)$$

trong đó hàm thực  $h(\|x - v^k\|, \sigma_k)$  được gọi là *hàm cơ sở bán kính* với tâm là  $v^k$  và  $M(\leq N)$  là số hàm bán kính sử dụng để xấp xỉ hàm  $f$ ;  $w_k$  và  $\sigma_k$  là các giá trị tham số cần tìm. Chúng tôi xét hàm cơ sở bán kính dạng Gauss:

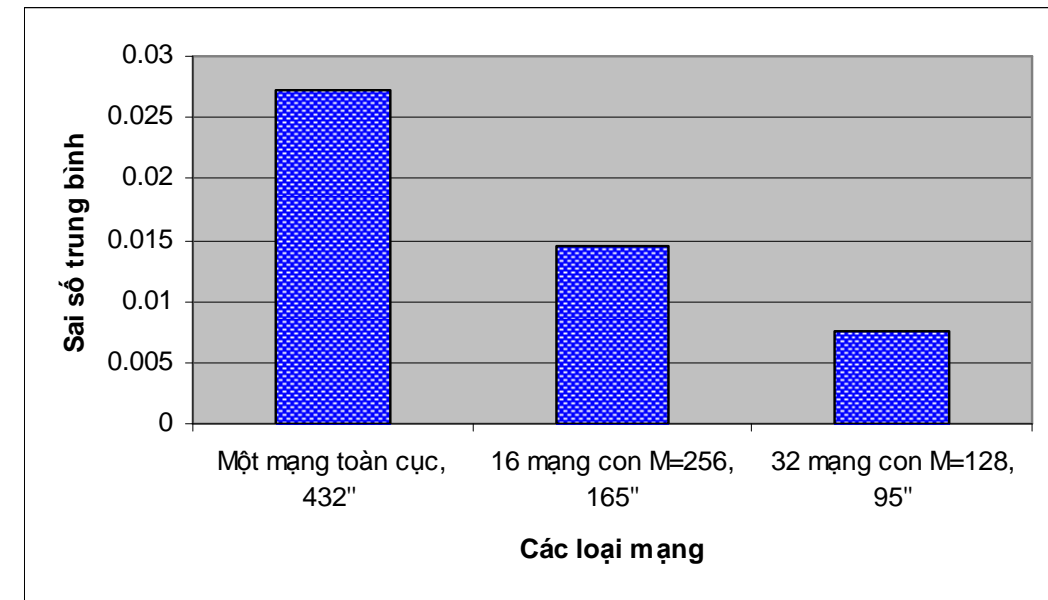
$h(r, \sigma) = e^{-\frac{r^2}{2\sigma^2}}$ . Khi số mốc nội suy không quá nhiều ta tìm  $\varphi$  dưới dạng

$$\varphi(x) = \sum_{k=1}^N w_k \varphi_k(x) + w_0 \quad (2.2)$$

Thực nghiệm cho thấy: Thời gian và sai số huấn luyện mạng giảm nhanh khi số cụm tăng, đặc biệt khi cỡ dữ liệu của cụm con thực sự giảm. Nếu song song hoá việc huấn luyện cụm con thì thời gian huấn luyện giảm đáng kể.

### 5.5.2 Tính tổng quát

Kết quả thực nghiệm trình bày chi tiết trong các bảng và hình của luận án. Thực hiện so sánh trên các hàm 2 biến và 3 biến phi tuyến.



Hình 5.7: Đồ thị so sánh tính tổng quát hoá với hàm 2 biến có 4086 mốc nội suy tại 10 điểm xa tâm.

Thực nghiệm cho thấy rằng tính tổng quát của mạng tốt hơn khi tăng số cụm, đặc biệt khi cỡ dữ liệu ở cụm con thực sự giảm.

### 5.5.3 Huấn luyện tăng cường ở bài toán động

Bỏ đi 5 mốc và huấn luyện số mốc còn lại, sau đó bổ sung thêm 5 mốc mới, lấy giá trị bán kính  $\sigma$  của lần huấn luyện trước làm giá trị khởi tạo cho  $\sigma$  của lần huấn luyện sau. Với 5 mốc mới thêm thì ta khởi gán  $\sigma$  theo như thuật toán HDH. Bảng 5.6 trong luận án và hình 5.9 cho kết quả để so sánh về thời gian huấn luyện.

địa phương huấn luyện đủ tốt với tập dữ liệu này sẽ thỏa mãn:  
 $\forall x \in D, |f(x) - \varphi(x)| < \varepsilon, \varphi(x)$  là hàm nội suy cần tìm.

#### 5.4. Bài toán động

Bài toán động là các bài toán sau khi huấn luyện xong thường được bổ sung các mốc nội suy mới. Đối với các bài toán này, khi có các mốc nội suy mới, ta kiểm tra lại số lượng mốc trong hình hộp con tương ứng, nếu vượt quá  $M$  mốc thì ta thực hiện tách đôi hình hộp con này và huấn luyện lại mạng RBF địa phương tương ứng. Nhờ cấu trúc cây  $k-d$ , việc bổ sung dữ liệu vào cụm thích hợp mất rất ít thời gian và với cách xử lý đã nêu ta không phải huấn luyện lại toàn mạng.

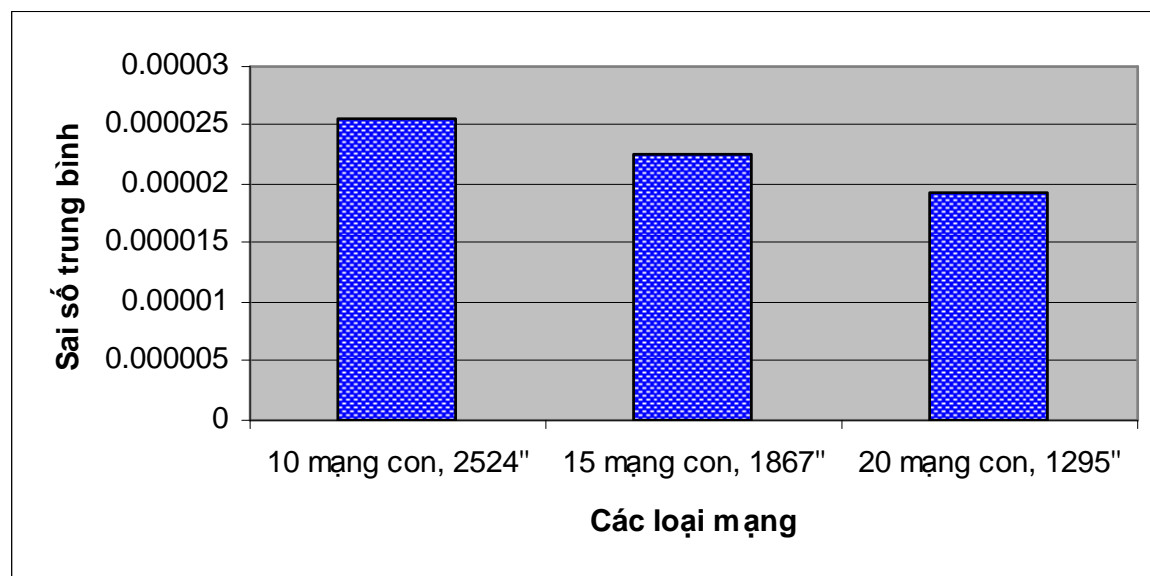
#### 5.5. Kết quả thực nghiệm

Thực nghiệm nhằm so sánh thời gian huấn luyện, sai số huấn luyện và tính tổng quát của mạng địa phương so với mạng nội suy RBF toàn cục (ở đây dùng từ toàn cục để phân biệt với mạng địa phương). Ngoài ra thực nghiệm cũng so sánh thời gian huấn luyện tăng cường bằng thuật toán HDH khi có dữ liệu mới và thời gian huấn luyện từ đầu để dùng cho bài toán động.

Thực nghiệm trên hàm hai biến và ba biến phi tuyến.

##### 5.5.1 So sánh thời gian và sai số huấn luyện

Kết quả thực nghiệm trình bày chi tiết trong các bảng và hình trong luận án. Các mạng sau khi được huấn luyện sẽ lấy 10 mốc nội suy để so sánh sai số huấn luyện. Thực hiện trên các hàm 2 biến và hàm 3 biến phi tuyến.



Hình 5.6: So sánh thời gian và sai số huấn luyện của hàm 3 biến có 19683 mốc

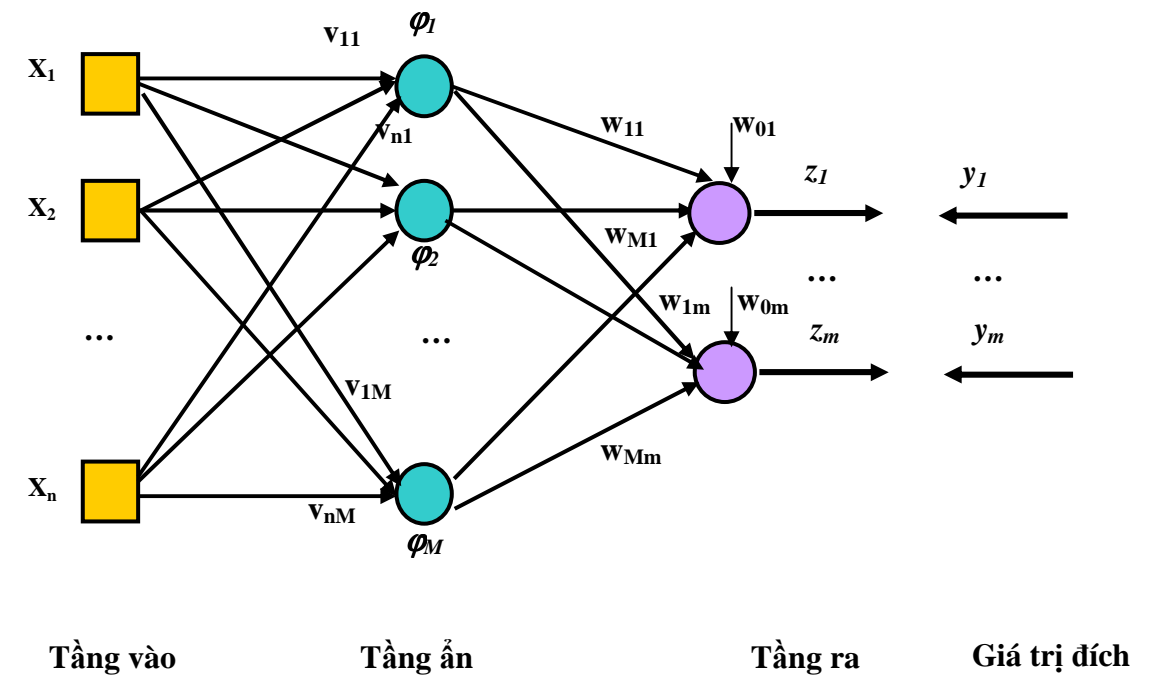
trong đó  $\varphi_k(x) = e^{-\|x-v^k\|^2/\sigma_k^2}$  (2.3)

$\|\cdot\|$  là kí hiệu chuẩn Euclide,  $v^k$  gọi là tâm của  $\varphi_k$ . Đối với bài toán nội suy, ta lấy tâm chính là các mốc nội suy  $v^k = x^k$  ( $M=N$ ). Với bài toán xấp xỉ thì  $M < N$ , các tham số  $w_k$  và  $\sigma_k$  cần tìm để  $\varphi$  cực tiểu tổng bình phương sai số hoặc thỏa mãn phương trình nội suy. Đến nay, việc xác định tâm tối ưu còn là bài toán đang được quan tâm. Với mỗi  $k$ , tham số  $\sigma_k$  được gọi là tham số *độ rộng* của hàm cơ sở bán kính.

Trường hợp số hàm bán kính bằng số mốc nội suy ( $M=N$ ). Với các tham số  $\sigma_k$  đã chọn, Micchelli đã chứng minh rằng nếu các mốc  $x^k$  khác nhau thì với  $w_0$  cho trước tùy ý hệ phương trình nội suy luôn tồn tại duy nhất nghiệm  $w_1, \dots, w_N$ .

#### 2.2. Kiến trúc mạng RBF

Kiến trúc của một mạng RBF tổng quát được mô tả trong Hình 2.5.



Hình 2.5 Mô tả kiến trúc mạng neuron RBF

Với dữ liệu  $\{x^k, y^k\}_{k=1}^N$  ( $x^k \in R^n; y^k \in R^m$ ), ký hiệu  $w_{0j}$  là giá trị ngưỡng của neuron ra thứ  $j$ . Khi đó đầu ra của neuron  $j$  ở tầng ra như sau  $z_j = w_{1j}\varphi_1 + \dots +$

$w_{Mj}\varphi_M + w_{0j}$   $j=1, \dots, m$ , với  $\varphi_k = e^{-\frac{\|x-v^k\|^2}{2\sigma_k^2}}$ . Về sau ta quy ước gọi các mạng RBF có số hàm bán kính bằng số mốc nội suy ( $M=N$ ) và có tâm trùng với mốc tương ứng là *mạng nội suy RBF*.



### 2.3. Huấn luyện mạng RBF

Có thể chia các kiểu huấn luyện mạng RBF ra thành ba loại: huấn luyện một pha, huấn luyện hai pha và huấn luyện đầy đủ.

*Huấn luyện một pha hay gọi là huấn luyện nhanh:* Khởi gán các tham số bán kính là một hằng số, sau đó huấn luyện trọng số kết nối  $w$  của tầng ra bằng phương pháp giả nghịch đảo hoặc bình phương tối thiểu.

*Huấn luyện hai pha:* pha thứ nhất tìm các tham số bán kính và tâm (với mạng xấp xỉ), pha thứ hai tìm các trọng số kết nối tầng ra  $w$ . Đây là phương pháp thông dụng hiện nay.

*Huấn luyện đầy đủ:* thường dùng cho mạng xấp xỉ. Tất cả các tham số của mạng đều được hiệu chỉnh để cực tiểu hoá tổng sai số bình phương. Thường dùng phương pháp Gradient để tìm kiếm cả ba tham số (tâm, bán kính, trọng số tầng ra).

### 2.4. So sánh mạng RBF với mạng MLP

Mạng RBF là mạng chỉ có một tầng/lớp nơron ẩn, trong khi đó mạng MLP có thể có một hoặc nhiều tầng nơron ẩn và khó xác định số nơron đủ tốt cho tầng này. Mạng MLP có thể dùng để ngoại suy hàm số còn mạng RBF với các hàm bán kính chỉ có ảnh hưởng địa phương (như hàm Gauss,...) nên không dùng để ngoại suy được. Các thuật toán huấn luyện ảnh hưởng nhiều tới chất lượng của mạng. Thực tiễn cho thấy mạng RBF có thời gian huấn luyện nhanh hơn mạng MLP và không sợ rơi vào cực trị địa phương. Hiện nay sai số nội suy (hay xấp xỉ hàm) vẫn phải qua thực nghiệm, chứ chưa có phương pháp ước lượng hữu hiệu nào.

## CHƯƠNG 3. THUẬT TOÁN MỚI HUẤN LUYỆN MẠNG NỘI SUY RBF

Chương này trình bày thuật toán mới phát triển của chúng tôi để huấn luyện mạng RBF. Kết quả chủ yếu của chương đã được công bố trong hội thảo khoa học, và tạp chí quốc tế *Signal Processing*.

### 3.1. Nền tảng lý thuyết của thuật toán

*Phương pháp lặp đơn.* Giả sử ta cần giải hệ phương trình  $Ax=b$ . Nếu đưa được về hệ tương đương dạng  $x = Bx + d$ , trong đó  $B$  là ma trận vuông cấp  $n$

**Procedure** chia đôi hình hộp  $n$ -chiều  $D_j = \prod_{i=1}^n [a_i^j, b_i^j]$

**Begin**

**Bước 1.** Chọn  $i$  sao cho cạnh  $[a_i^j, b_i^j]$  của  $D_j$  là lớn nhất

**Bước 2.** Chia cạnh  $[a_i^j, b_i^j]$  bởi nhất cắt trực giao và qua ít nhất một điểm dữ liệu sao cho hai hình hộp nhận được từ nó chứa số đối tượng dữ liệu bằng nhau với quy ước các điểm dữ liệu thuộc nhất cắt có thể đồng thời tính là thuộc hai hình hộp.

Thủ tục chia đôi kết thúc khi số lượng đối tượng trong mỗi hình hộp con không vượt quá  $M$  chọn trước.

**End.**

Hình 5.3 Thủ tục chia đôi hình hộp  $n$ -chiều

### 5.2.3 Độ phức tạp thuật toán huấn luyện mạng

Quá trình xây dựng cây  $k$ - $d$  ở đây tương tự như thuật toán truyền thống, có độ phức tạp là  $O(N \log N)$  với  $N$  là số mốc nội suy. Thời gian huấn luyện mỗi mạng RBF trên mỗi miền con là  $O((T+c)nM^2)$  với  $n$  là số chiều của mốc nội suy,  $M$  là số mốc nội suy trong mỗi cụm. Nếu huấn luyện các cụm con được thực hiện tuần tự thì độ phức tạp tính toán để huấn luyện các mạng RBF là  $O((T+c)nMN)$ .

Nhờ cấu trúc cây  $k$ - $d$ , khi bổ sung dữ liệu mới, độ phức tạp tính toán để tìm ra cụm thích hợp cho nó là  $O\left(\log \frac{N}{M}\right)$ .

### 5.3. Tính xấp xỉ tổng quát của mạng nội suy RBF địa phương

Định lý dưới đây cho ta đánh giá tính xấp xỉ tổng quát của mạng nội suy RBF địa phương.

Để phát biểu định lý, ta cần đến khái niệm  $\delta$ -lưới của miền  $D$ . Tập mốc gọi là  $\delta$ -lưới trên  $D$  nếu  $\forall x \in D$  tồn tại mốc  $x^i$  sao cho  $d(x, x^i) < \delta$ , trong đó  $d(x, x^i)$  là khoảng cách sinh bởi chuẩn xác định bởi chuẩn  $\|u\|_* = \max_{j \leq N} \{u_j\}$ .

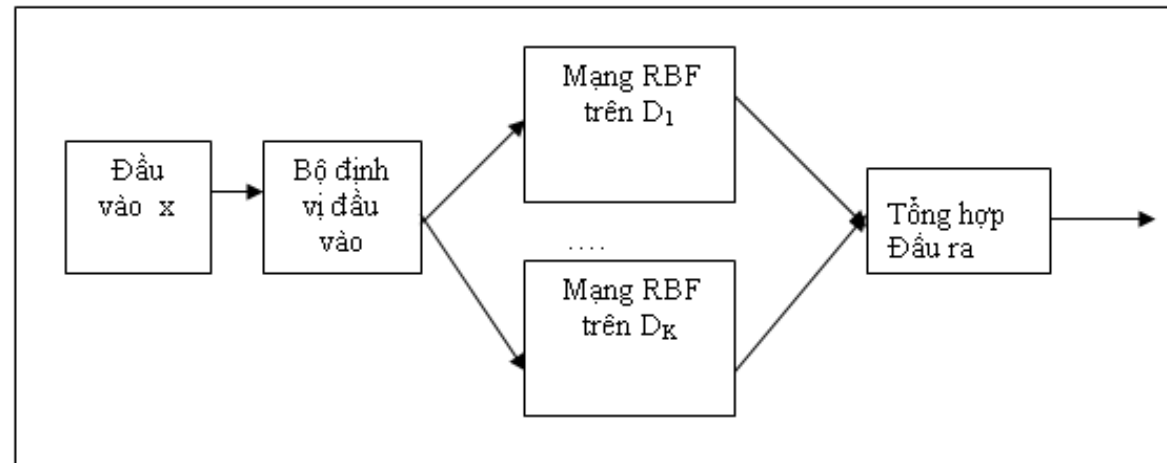
**Định lý 5.1.** (tính xấp xỉ tổng quát) Giả sử  $f$  là hàm liên tục trên miền  $D$  và  $M$  là số phần tử cực đại trong mỗi cụm con cho trước. Khi đó với mọi  $\varepsilon > 0$  tồn tại  $\delta > 0$  sao cho với bất kỳ tập dữ liệu  $\delta$ -lưới nào trên  $D$ , thì một mạng

miền con lớn hơn  $M$  thì thuật toán cây  $k-d$  sẽ được sử dụng để phân chia thành hai cụm có kích cỡ nhỏ hơn.

**Procedure** Xây dựng mạng RBF địa phương;  
**Begin**  
 1. Phân  $D$  thành các miền con  $D_1, \dots, D_k$ ; // dùng thuật toán phân cụm cây  $k-d$  để số mốc trong mỗi cụm con không vượt quá  $M$ .  
 2. Xây dựng bộ định vị đầu vào cho các mạng RBF con.  
 3. Huấn luyện các mạng RBF con; // Dùng thuật toán HDH.  
 4. Kết nối bộ định vị đầu vào với các mạng con để được mạng RBF địa phương.  
**End;**

Hình 5.1 Thủ tục xây dựng mạng RBF địa phương

Kiến trúc mạng được mô tả trong Hình 5.2. Trong pha huấn luyện, thành phần *tổng hợp đầu ra* không hoạt động. Mỗi mạng con được huấn luyện độc lập. Trong khi nội suy, với mỗi giá trị vào  $x$ , *bộ định vị* sẽ xác định mạng con tương ứng, còn tất cả các mạng con khác có giá trị vào rỗng. Và giá trị đầu ra của tất cả các mạng con đều là rỗng ngoại trừ mạng con có đầu vào  $x$ . Sau đó thành phần *tổng hợp đầu ra* là tổng đầu ra của các mạng RBF con.



Hình 5.2. Mô hình kiến trúc mạng RBF địa phương

### 5.2.2 Thuật toán phân cụm nhờ cây $k-d$ .

Thuật toán sau đây sửa đổi nhỏ kỹ thuật cây  $k-d$ , để phân hình hộp  $n$ -chiều  $D$  chứa  $N$  đối tượng dữ liệu thành các hình hộp con  $D_1, \dots, D_k$  sao cho mỗi hình hộp  $D_j$  chứa không quá  $M$  đối tượng.

thoả mãn  $\|B\|=q<1$ ;  $\|B\| = \max\{\|Bx\|/\|x\|=1\}$  thì nghiệm của hệ  $Ax=b$  có thể tìm bằng phương pháp lặp đơn như sau.

Với xấp xỉ ban đầu  $x^0$  tùy ý, nghiệm xấp xỉ ở bước  $k+1$  xác định bởi công thức:  $x^{k+1} = Bx^k + d$  hội tụ nhanh tới nghiệm và dễ dàng ước lượng sai số. Phương pháp này được cải tiến bằng phương pháp Seidel có tốc độ hội tụ nhanh hơn.

Ta xét bài toán nội suy với  $m = 1$ , và hàm cơ sở bán kính dạng Gauss:

$$\varphi(x) = \sum_{k=1}^N w_k \varphi_k(x) + w_0; \quad \varphi_k(x) = e^{-\|x-x^k\|^2/\sigma_k^2} \quad \forall k=1, \dots, N \quad (3.11)$$

Các tham số  $w_k$  và  $\sigma_k$  cần tìm để hàm  $\varphi$  thỏa các điều kiện nội suy:

$$\varphi(x^i) = \sum_{k=1}^N w_k \varphi_k(x^i) + w_0 = y^i; \quad \forall i=1 \dots N \quad (3.12)$$

$$\text{Hay là: } \sum_{k=1}^N w_k e^{-\|x^i-x^k\|^2/\sigma_k^2} = y^i - w_0 = z_i; \quad \forall i=1 \dots N \quad (3.13)$$

Với các tham số  $\sigma_k$  đã chọn ta xét ma trận cấp  $N \times N$ :  $\Phi = (\varphi_{k,i})_{N \times N}$  trong đó

$\varphi_{k,i} = \varphi_k(x^i) = e^{-\|x^i-x^k\|^2/\sigma_k^2}$ . Khi đó (3.13) trở thành:  $\Phi W = Z$  với  $W=(w_1, \dots, w_N)^T$  và  $Z=(z_1, \dots, z_N)^T$ . Ta xây dựng mạng neuron RBF giải bài toán nội suy này ( về sau gọi là *mạng nội suy RBF* cho gọn).

### 3.2. Thuật toán lặp hai pha huấn luyện mạng nội suy RBF

Ý tưởng chính của thuật toán mới là ở pha đầu tìm các bán kính  $\sigma_k$  được xác định nhờ cân bằng giữa tính tổng quát của mạng và tốc độ hội tụ của pha sau. Trong pha thứ hai, các tham số  $w_k$  được xác định nhờ tìm điểm bất động của một ánh xạ co.

#### 3.2.1 Định lý cơ bản

Ký hiệu:  $I$  là ma trận đơn vị cấp  $N$  và đặt:  $\Psi = I - \Phi = \left| \Psi_{k,j} \right|_{N \times N} \quad (3.16)$

thì ta có  $\Psi_{k,j} = \begin{cases} 0 & \text{khi : } k = j \\ -e^{-\|x^j-x^k\|^2/\sigma_k^2} & \text{khi : } k \neq j \end{cases} \quad (3.17)$

Khi đó hệ phương trình (3.12) tương đương với hệ:

$$W = \Psi W + Z \quad (3.18)$$

Với mỗi  $k \leq N$ , ta có hàm  $q_k$  của  $\sigma_k$  xác định theo công thức:  $q_k = \sum_{j=1}^N |\Psi_{k,j}|$ . Việc xác định các bán kính  $\sigma_k$  này dựa vào định lý sau.

**Định lý 3.1.** Hàm  $q_k$  là đơn điệu tăng, hơn nữa với mọi số dương  $q < 1$ , luôn tồn tại giá trị  $\sigma_k$  sao cho  $q_k(\sigma_k)$  bằng  $q$ .

Định lý 3.1 cho thấy với  $q < 1$  tùy ý, ta có thể tìm được tập các  $\{\sigma_k\}_{k=1}^N$  để nghiệm đúng  $W^*$  của (3.12) là điểm bất động của ánh xạ co  $\Psi W + Z$  và hệ số co là  $q$ .

### 3.2.2 Mô tả thuật toán

Với sai số  $\varepsilon$  và các hằng số dương  $q, \alpha$  ( $0 < q < 1, 0 < \alpha < 1$ ) cho trước, thuật toán sẽ gồm 2 pha để xác định các tham số  $\sigma_k$  và  $W^*$ . Trong pha thứ nhất, các  $\sigma_k$  được xác định để  $q_k \leq q$ , sao cho  $q_k$  gần tới  $q$  nhất. Pha hai xác định  $W^*$  bằng phương pháp lặp tìm nghiệm của hệ phương trình (3.18).

#### Procedure Huấn luyện mạng RBF (HDH)

##### Begin

##### Pha 1: Xác định tham số bán kính

For k=1 to N do

Xác định các  $\sigma_k$  để  $q_k < q$ ;

sao cho nếu thay thế  $\sigma_k := \frac{\sigma_k}{\alpha}$  thì  $q_k > q$

End

##### Pha 2: Xác định các trọng số tầng ra

Xác định  $W^*$  bằng phương pháp lặp

##### End

Hình 3.2 Đặc tả thuật toán lặp hai pha huấn luyện mạng RBF

Với mỗi vectơ  $N$ -chiều  $u$ , ta ký hiệu chuẩn  $\|u\|_* = \max_{j \leq N} \{|u_j|\}$ , điều kiện kết thúc có thể chọn một trong biểu thức sau:

$$a) \frac{q}{1-q} \|W^1 - W^0\|_* \leq \varepsilon \quad (3.23)$$

## 4.4 Nhận xét chung về thuật toán một pha mới

Thuật toán mới một pha QHDH không những giảm đáng kể thời gian huấn luyện mà tính tổng quát của mạng cũng tốt hơn các thuật toán khác đã so sánh. Thuật toán này thực sự có ý nghĩa cho bài toán có mốc nội suy cách đều nhau và chạy tốt với số mốc lớn.

## CHƯƠNG 5. MẠNG RBF ĐỊA PHƯƠNG

Chương này xây dựng kiến trúc mới đó là mạng RBF địa phương, để giải quyết các bài toán động và thời gian thực.

Các kết quả chính của chương được công bố trong hội thảo Quốc tế của IEEE, và tạp chí quốc tế *International Journal of Data Mining, Modelling and Management Science*.

### 5.1. Giới thiệu

Hiện nay chưa có thuật toán hiệu quả nào để huấn luyện cho các bài toán động và thời gian thực. Thuật toán HDH cải thiện đáng kể thời gian huấn luyện và có nhiều ưu điểm nhưng thời gian huấn luyện tăng rất nhanh khi số mốc nội suy tăng. Đặc tính này gợi ý tưởng phân miền dữ liệu thành các miền con chứa mốc nội suy gần bằng nhau và không vượt quá  $M$  cho trước, rồi xây dựng mạng nội suy RBF trên mỗi cụm con này. Nếu có dữ liệu bổ sung thì chỉ huấn luyện lại trên mỗi cụm con.

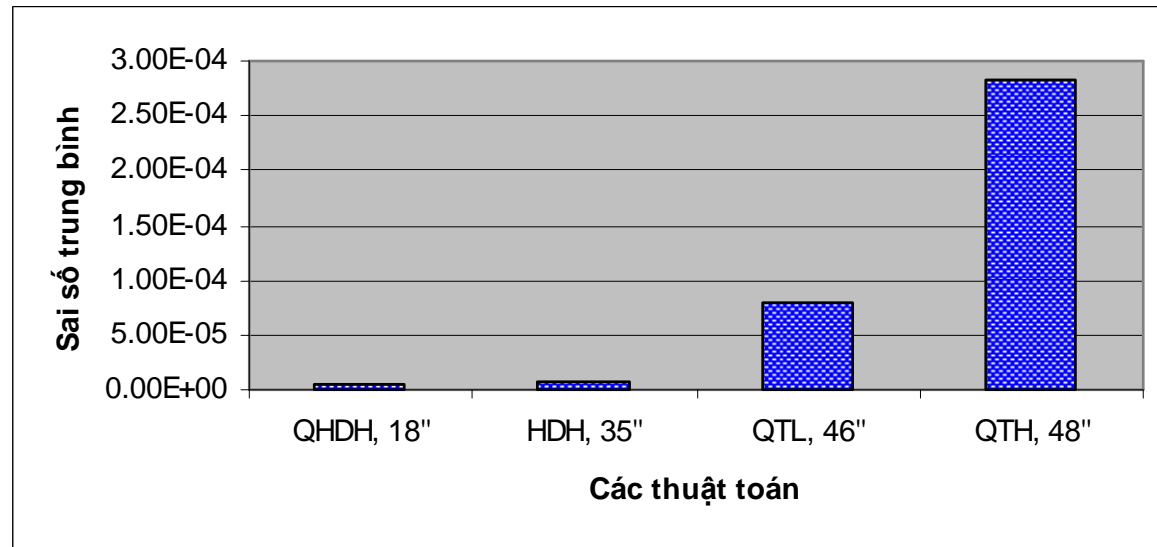
### 5.2. Mạng RBF địa phương

#### 5.2.1 Kiến trúc và thủ tục xây dựng mạng

Giả sử tập mốc nội suy  $\{x^k\}_{k=1}^N$  nằm trong miền đóng giới nội  $D = \prod_{i=1}^n [a_i, b_i]$  và  $N$  lớn. Ta chọn trước số nguyên dương  $M$  cho số điểm trong mỗi cụm con và chia miền  $D$  thành các hình hộp  $n$  chiều  $D_j$  ( $j=1,2,\dots,k$ ), với số mốc trong mỗi cụm nhỏ hơn  $M$  theo thuật toán phân cụm nhờ cây  $k-d$ . Sau đó sử dụng thuật toán lặp HDH để huấn luyện các mạng RBF cho mỗi miền con  $D_j$ . Xây dựng thủ tục để xác định mỗi  $x$  trong  $D$  thuộc miền con  $D_j$  và mạng RBF địa phương sẽ là kết nối giữa thủ tục này với các mạng RBF con. Với mỗi dữ liệu mới thuộc  $D_j$  thì chỉ có mạng nội suy địa phương của miền  $D_j$  phải huấn luyện lại. Khi bổ sung dữ liệu mới, nếu số mốc nội suy trong



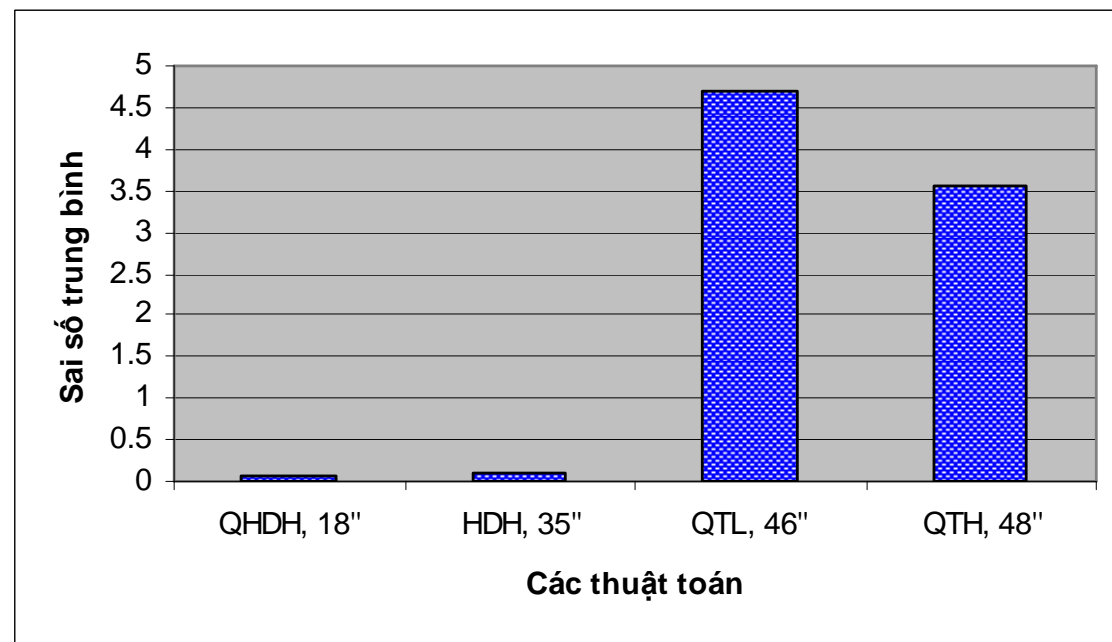
### 4.3.2 So sánh tính sai số huấn luyện



Hình 4.3: So sánh sai số và thời gian huấn luyện của các thuật toán QHDH, HDH, QTL, QTH với 1331 mốc của hàm 3 biến.

Theo thực nghiệm ta thấy sai số huấn luyện và thời gian huấn luyện của thuật toán QHDH là tốt nhất.

### 4.3.3 So sánh tính tổng quát



Hình 4.4: Đồ thị So sánh tính tổng quát hoá của mạng huấn luyện bởi các thuật toán QHDH, HDH, QTL và QTH với 1331 mốc của hàm 3 biến.

Theo thực nghiệm ta thấy tính tổng quát của thuật toán QHDH là tốt nhất.

$$b) t \geq \frac{\ln \frac{\varepsilon(1-q)}{\|Z\|_*}}{\ln q} = \frac{\ln q - \ln \|Z\|_* + \ln(1-q)}{\ln q} \quad (3.24)$$

với  $t$  là số lần lặp.

### 3.2.3 Đặc tính hội tụ

Định lý đảm bảo tính hội tụ và cho phép ước lượng sai số của thuật toán.

**Định lý 3.2.** Thuật toán huấn luyện mạng HDH luôn kết thúc sau hữu hạn bước và đánh giá sau là đúng:  $\|W^1 - W^*\|_* \leq \varepsilon$  (3.25)

### 3.2.4 Độ phức tạp của thuật toán

Đặt  $c = \max \left\{ \log_\alpha \frac{\sigma_{\min}}{\sigma_0}, \log_\alpha \frac{\sigma_{\max}}{\sigma_0} \right\}$  và  $T$  là số lần lặp trong pha 2 của thuật

toán ( $T$  có thể được ước lượng theo công thức:  $T = \frac{\ln \frac{\varepsilon(1-q)}{\|Z\|_*}}{\ln q} = \log_q \frac{\varepsilon(1-q)}{\|Z\|_*}$ )

thì độ phức tạp của thuật toán là  $O((T+c)nN^2)$ .

### 3.3. Thử nghiệm thuật toán.

Hiệu quả của thuật toán huấn luyện mạng neuron dựa trên các tiêu chí sau:

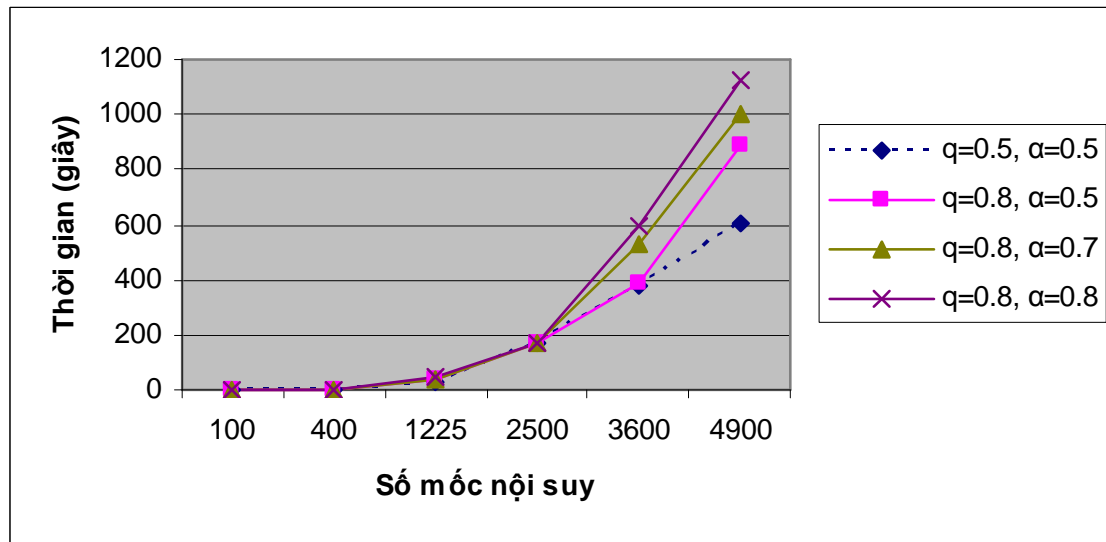
- 1) Tốc độ hội tụ của thuật toán biểu hiện qua thời gian chạy và sai số huấn luyện.
- 2) Tính tổng quát của mạng biểu hiện qua sai số kiểm tra các điểm mới sau khi huấn luyện.

Thực nghiệm với hàm 3 biến  $y = x_1^2 x_2 + \sin(x_2 + x_3 + 1) + 4$  (3.33)

Chạy trên máy tính có cấu hình: Intel Pentium 4 Processor, 3.0GHz, 512MB DDR RAM. Dữ liệu thử nghiệm được lấy cách đều theo hai chiều và tích hợp ngẫu nhiên với chiều thứ ba.

#### 3.3.1 Tốc độ hội tụ

Thử nghiệm với số mốc khác nhau với các tham số  $q$ ,  $\alpha$  và  $\varepsilon$  thay đổi tuần tự.

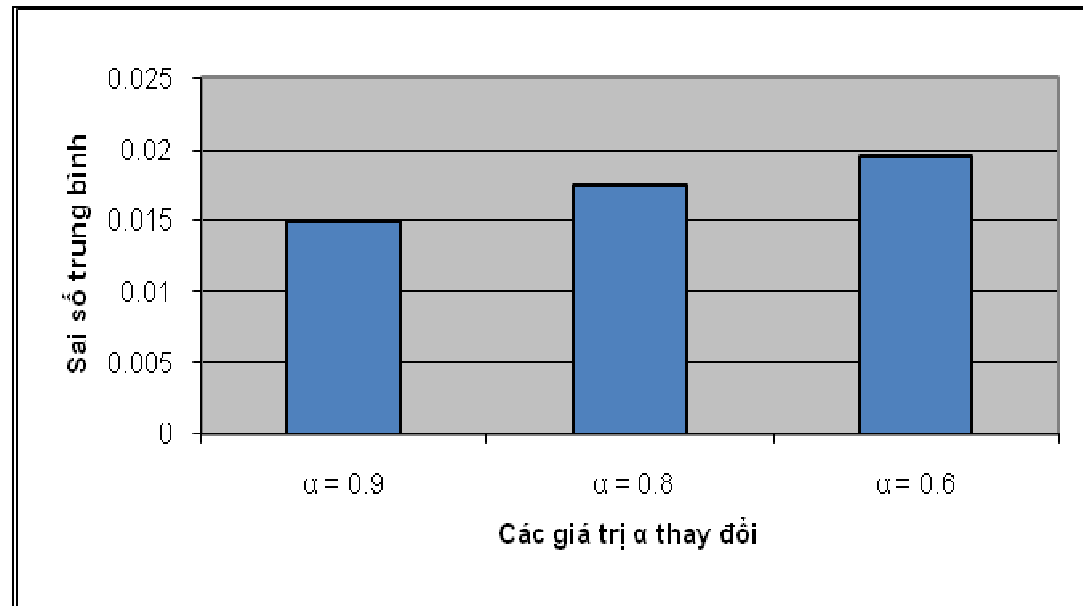


Hình 3.5 Đồ thị thời gian huấn luyện khi các tham số  $q$  và  $\alpha$  thay đổi

Nhận xét: Thời gian huấn luyện ngắn (thậm chí chỉ vài phút cho khoảng 3000 mốt) so với các thuật toán khác. Thời gian huấn luyện tăng khi  $q$  hoặc  $\alpha$  tăng.

### 3.3.2 Tính tổng quát

Lấy 400 mốt và huấn luyện mạng với các giá trị khác nhau của tham số  $q$  và  $\alpha$ , với tham số dừng  $\epsilon = 10^{-6}$ . Sau khi huấn luyện xong, lấy các điểm ngẫu nhiên không nằm trong tập mẫu huấn luyện để kiểm tra.



Hình 3.6 Đồ thị kiểm tra sai số khi  $\alpha$  thay đổi

Từ kết quả thực nghiệm trên hình 3.6 và các bảng cũng như các hình trong luận án chỉ ra rằng tính tổng quát của mạng tăng nhanh khi  $q$ ,  $\alpha$  tăng.

### 4.3. Thực nghiệm

Thực nghiệm so sánh thời gian, sai số huấn luyện và tính tổng quát của thuật toán một pha QHDH với thuật toán HDH và các thuật toán một pha Gradient QTL, QTH. Trong đó QTL là thuật toán do Looney giới thiệu với tham số độ rộng bán kính  $\sigma = \frac{1}{(2N)^{1/n}}$  và QTH là thuật toán theo gợi ý

$$\sigma = \frac{D_{\max}}{\sqrt{2N}}$$

của Haykin ;  $D_{\max}$  là khoảng cách lớn nhất giữa các mốt;  $N$  là số lượng các mốt nội suy,  $n$  là số chiều của mốt nội suy.

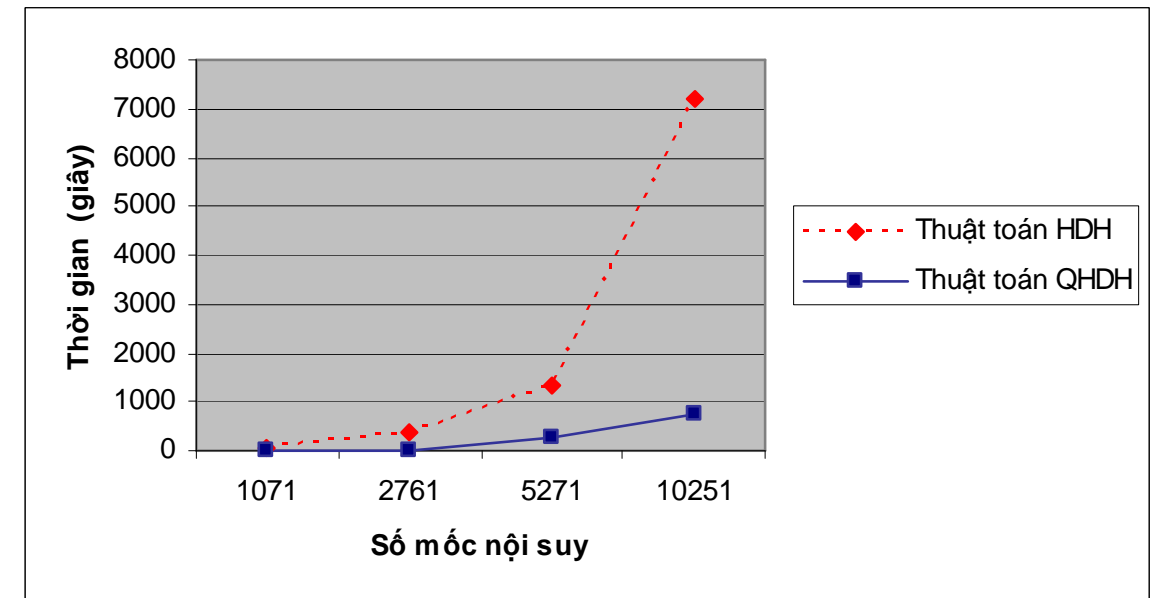
Thực nghiệm trên các hàm 2 biến và 3 biến.

Thực nghiệm trên các hàm 2 biến và 3 biến.

$$y = \frac{x_1^2 + x_1 x_2}{5} + \frac{x_2}{3} + \frac{1}{2} \quad x_1 \in [0,10], x_2 \in [0,20] \quad (4.22)$$

$$y = x_1^2 x_2 + \sin(x_2 + x_3 + 1) + 1, \quad x_1 \in [0,3], x_2 \in [0,4], x_3 \in [0,5] \quad (4.23)$$

#### 4.3.1 So sánh thời gian huấn luyện



Hình 4.2: Đồ thị so sánh thời gian huấn luyện giữa thuật toán HDH và QHDH

Thực nghiệm cho thấy thời gian huấn luyện của thuật toán QHDH giảm đáng kể so với thuật toán 2 pha HDH.

$$\Psi_{i_1, \dots, i_n}^{j_1, \dots, j_n} = \begin{cases} 0; & \text{khi: } j_1, \dots, j_n = i_1, \dots, i_n \\ -e^{-\frac{\|x^{j_1, \dots, j_n} - x^{i_1, \dots, i_n}\|_A^2}{\sigma_{i_1, \dots, i_n}^2}} & \end{cases} \quad (4.5)$$

Các bán kính  $\sigma_{i_1, \dots, i_n}$  được xác định sao cho ma trận  $\Psi$  thỏa mãn điều kiện của ánh xạ co để thực hiện pha hai của thuật toán HDH.

Tức là với số dương  $q < 1$ , chọn các  $\sigma_{i_1, \dots, i_n}$  để cho

$$q_{i_1, \dots, i_n} = \sum_{j_1, \dots, j_n} |\Psi_{i_1, \dots, i_n}^{j_1, \dots, j_n}| \leq q < 1 \quad (4.6)$$

## 4.2. Định lý cơ sở và mô tả thuật toán

### 4.2.1 Định lý cơ sở

**Định lý 4.1.** Với mọi  $q \in (0, 1)$ , nếu  $\sigma_{i_1, \dots, i_n} = \sqrt{\frac{1}{\ln\left(\frac{\sqrt[n]{1+q}-1}{8}\right)^{-1}}}$  thì  $q_{i_1, i_2, \dots, i_n} < q$ .

### 4.2.2 Mô tả thuật toán một pha

Mô tả thuật toán một pha (gọi tắt là thuật toán QHDH)

Với hằng số dương  $q < 1$  ta có thể chọn trước các  $\sigma_{i_1, \dots, i_n}$  theo định lý 4.1 (hoặc công thức (4.21)) để thỏa mãn điều kiện của ánh xạ co trong công thức (4.6). Sau đó áp dụng phương pháp lặp đơn trong pha thứ hai của thuật toán HDH để xác định trọng số tầng ra.

#### Proceduce Thuật toán 1 pha huấn luyện mạng

##### Bước 1: Xác định bán kính

Xác định các  $\sigma_{i_1, \dots, i_n}$  theo công thức (4.21) ;

##### Bước 2: Tính trọng số tầng ra

Tìm  $W^*$  bằng phương pháp lặp đơn; // pha 2 của thuật toán HDH;

**End**

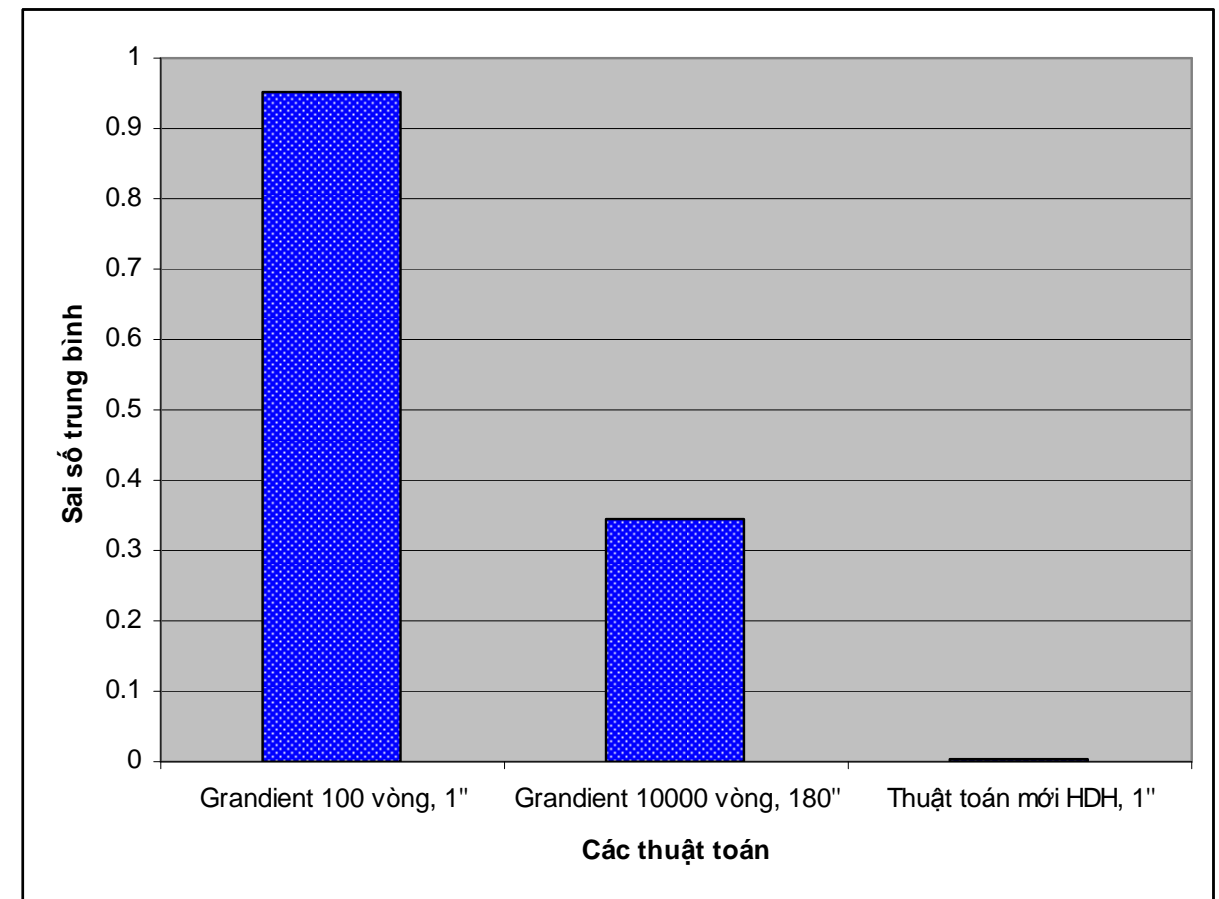
Hình 4.1: Thuật toán 1 pha huấn luyện mạng RBF với mốc cách đều.

## 3.4. So sánh với phương pháp Gradient

Thực nghiệm tương tự để so sánh với mạng huấn luyện bằng phương pháp Gradient cho thấy mạng huấn luyện bằng thuật toán mới huấn luyện nhanh hơn nhiều và tính tổng quát cũng tốt hơn.

### 3.4.1 So sánh thời gian và độ chính xác của những điểm huấn luyện.

Huấn luyện bằng thuật toán HDH với  $\epsilon = 10^{-6}$ ,  $q = 0.8$ ,  $\alpha = 0.9$  và thuật toán Gradient trong hai trường hợp 100 vòng lặp và 10000 vòng lặp.

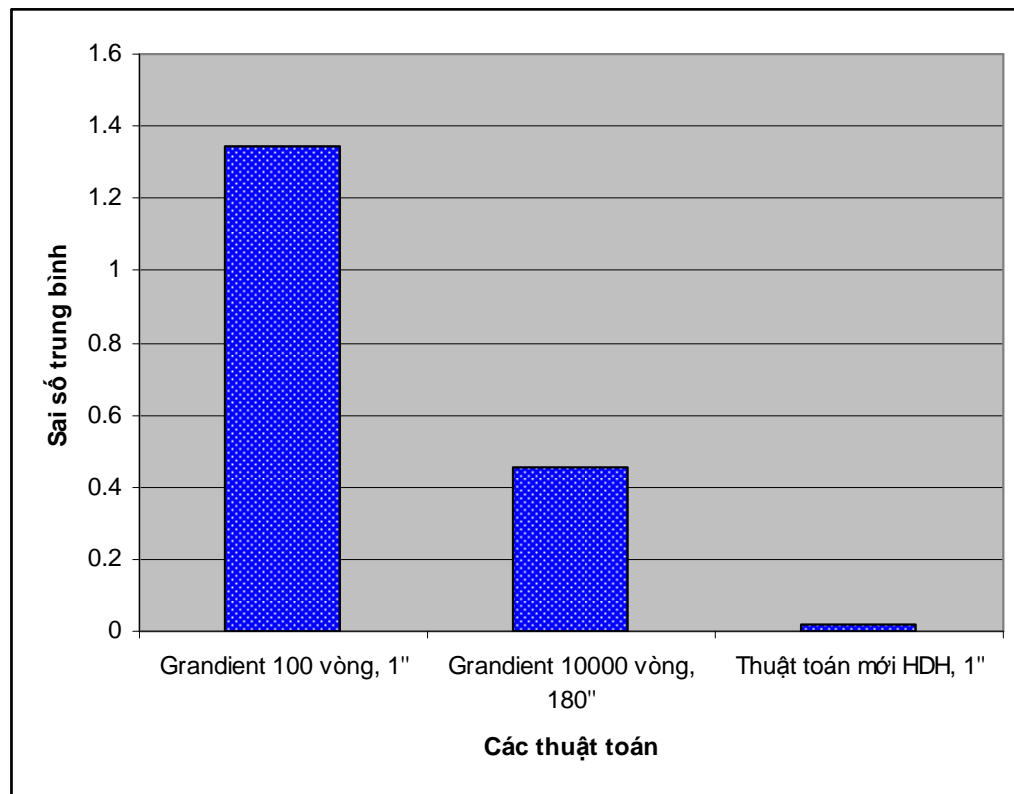


Hình 3.8 So sánh độ chính xác và thời gian của thuật toán mới và thuật toán Gradient

Theo thực nghiệm ta thấy thuật toán mới HDH tốt hơn so với thuật toán Gradient trong cả hai trường hợp về thời gian huấn luyện và tính chính xác.

### 3.4.2 So sánh tính tổng quát

Lấy ra các mốc để làm tập điểm kiểm tra. Sau khi huấn luyện bằng hai thuật toán với các tham số như mục 3.4.1. So sánh sai số ở tập kiểm tra để đánh giá tính tổng quát.



Hình 3.9 So sánh tính tổng quát của thuật toán mới HDH và thuật toán Gradient

Qua thực nghiệm ta thấy thuật toán mới HDH có thời gian huấn luyện và sai số nhỏ hơn nhiều cũng như tính tổng quát tốt hơn so với thuật toán Gradient.

### 3.5 Nhận xét chung về thuật toán lặp hai pha HDH

Qua thực nghiệm ta thấy thuật toán mới HDH có thời gian huấn luyện mạng nhanh kể cả khi số mốc lớn. Với thuật toán cực tiểu sai số tổng các bình phương, Looney cho rằng chỉ nên chạy với số mốc nhỏ hơn 200 còn thuật toán này vẫn thực hiện được khi số mốc gần 5000.

Để dàng điều khiển cân bằng giữa tốc độ hội tụ và tính tổng quát của mạng bằng việc điều chỉnh các tham số, có khả năng song song hoá thuật toán.

Tóm lại, thuật toán mới đề xuất HDH dễ thực hiện và có hiệu quả cao. Thực nghiệm cho thấy nó giải quyết tốt bài toán nội suy với số mốc lớn và có nhiều ưu điểm nổi trội.

## CHƯƠNG 4. BÀI TOÁN NỘI SUY VỚI MỐC CÁCH ĐỀU

Trong nhiều ứng dụng ở các lĩnh vực đồ họa máy tính, phương trình toán lý, các bài toán kỹ thuật, và nhận dạng mẫu thường gặp bài toán nội suy với mốc cách đều. Chương này đề xuất thuật toán một pha để giải quyết cho những bài toán này.

Kết quả của chương đã được đăng trong kỷ yếu hội thảo quốc gia các vấn đề chọn lọc của CNTT lần thứ X.

### 4.1 Biểu diễn bài toán

*Bài toán mốc cách đều:* Khi các mốc nội suy là mốc cách đều thì chúng có thể biểu diễn các mốc nội suy theo bộ chỉ số dạng  $(i_1, i_2, \dots, i_n)$  với  $x^{i_1, i_2, \dots, i_n} = (x_1^{i_1}, \dots, x_n^{i_n})$ , trong đó:  $x_k^{ik} = x_k^0 + ik \cdot h_k$ ;  $h_k$  ( $k = 1, \dots, n$ ) là các hằng số bước cho trước (bước thay đổi của biến  $x_k$ ),  $n$  là số chiều, và  $ik$  nhận giá trị từ 0 đến  $N_k$  ( $N_k + 1$  là số mốc chia của từng chiều).

Thay cho chuẩn Euclide, ta xét chuẩn Mahalanobis:  $\|x\|_A = \sqrt{x^T A x}$ ; với  $A$  là ma trận đường chéo có dạng:

$$A = \begin{pmatrix} \frac{1}{a_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{a_2^2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{a_n^2} \end{pmatrix}$$

Các tham số  $a_k$  sẽ được chọn thích hợp với các bước  $h_k$  (dưới đây sẽ cho thấy là  $a_k = h_k \forall k$ ). Biểu thức (3.10) và (3.11) được viết lại là:

$$\varphi(x) = \sum_{i_1, \dots, i_n=1}^{N_1, \dots, N_n} w_{i_1, \dots, i_n} \varphi_{i_1, \dots, i_n}(x) + w_0 \quad (4.2)$$

$$\text{trong đó, } \varphi_{i_1, \dots, i_n}(x) = e^{-\|x - x^{i_1, \dots, i_n}\|_A^2 / \sigma_{i_1, \dots, i_n}^2} \quad (4.3)$$

Ma trận  $\Phi = (\varphi_{i_1, \dots, i_n}^{j_1, \dots, j_n})_{N \times N}$  là ma trận vuông cấp  $N = N_1 \dots N_n$ . Với

$$\varphi_{i_1, \dots, i_n}^{j_1, \dots, j_n} = \varphi_{i_1, \dots, i_n}(x^{j_1, \dots, j_n}) = e^{-\|x^{j_1, \dots, j_n} - x^{i_1, \dots, i_n}\|_A^2 / \sigma_{i_1, \dots, i_n}^2} \quad (4.4)$$

$\Psi = I - \Phi$  sẽ là: