ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Lê Xuân Minh Hoàng

HUẤN LUYỆN MẠNG NƠRON RBF VỚI MỐC CÁCH ĐỀU VÀ ỨNG DỤNG

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI - 2010

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Lê Xuân Minh Hoàng

HUẤN LUYỆN MẠNG NƠRON RBF VỚI MỐC CÁCH ĐỀU VÀ ỨNG DỤNG

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: PGS.TS Hoàng Xuân Huấn

HÀ NỘI – 2010

LỜI CẢM ƠN

Tôi muốn bày tỏ sự cảm ơn sâu sắc của mình tới thầy Hoàng Xuân Huấn, thuộc bộ môn Khoa học máy tính, khoa Công nghệ thông tin, trường Đại học Công nghệ, ĐHQGHN đã nhận hướng dẫn và tin tưởng để giao cho tôi một đề tài thú vị như thế này. Trong thời gian thực hiện khóa luận, thầy đã rất kiên nhẫn, nhiệt tình hướng dẫn và giúp đỡ tôi rất nhiều. Chính những hiểu biết sâu rộng và kinh nghiệm nghiên cứu khoa học của thầy đã hiều lần định hướng giúp tôi tránh khỏi đi những sai lầm và giúp tôi vượt qua mỗi khi gặp những bế tắc khi thực hiện khóa luận này.

Tôi cũng muốn bày tỏ sự cảm ơn của mình tới các các thầy, các cô trong bộ môn, cũng như các thầy, các cô trong khoa, trường đã tạo điều kiện và giúp đỡ để tôi có thể thực hiện và hoàn thành được khóa luận này. Nếu không có những kiến thức được đào tạo trong các năm vừa qua, tôi đã không thể hoàn thành khóa luận này.

TÓM TẮT NỘI DUNG

Mặc dù đã được nghiên cứu từ rất lâu, nhưng đến nay bài toán nội suy và xấp xỉ hàm nhiều biến vẫn còn có rất ít công cụ toán học để giải quyết. Mạng Nơron nhân tạo là một phương pháp hay để giải quyết bài toán nội suy, xấp xỉ hàm nhiều biến. Năm 1987 M.J.D. Powell đã đưa ra một cách tiếp cận mới để giải quyết bài toán nội suy hàm nhiều biến sử dụng kỹ thuật hàm cơ sở bán kính (Radial Basis Function - RBF), năm 1988 D.S. Bromhead và D. Lowe đề xuất kiến trúc mạng Nơron RBF và đã trở một công cụ hữu hiệu để giải quyết bài toán nội suy và xấp xỉ hàm nhiều biến(xem [11]).

Năm 2006 Hoàng Xuân Huấn và các cộng sự (xem [1]) đã đưa ra thuật toán lặp hai pha để huấn luyện mạng nơron RBF và đã cho ra kết quả tốt tuy nhiên nhược điểm của nó là sai số lớn hơn khi dữ liệu phân bố không đều. Khi áp dụng phương pháp này trên bộ dữ liệu cách đều đã cho ta thuật toán lặp một pha HDH mới với thời gian và tính tổng quát tốt hơn rất nhiều. (xem [2])

Nội dung của khóa luận này là ứng dụng thuật toán huấn luyện mạng nơron RBF với mốc cách đều để đưa ra một phương pháp nội suy xấp xỉ hàm nhiều biến với bộ dữ liệu có nhiễu trắng và chứng minh hiệu quả thông qua việc xây dựng phần mềm nội suy hàm số.

MỤC LỤC

MỞ ĐẦU 1

CHƯƠNG 1 BÀI TOÁN NỘI SUY, XẤP XỈ HÀM SỐ VÀ MẠNG NƠRON RBF 13

1.1 BÀI TOÁN NỘI SUY VÀ XẤP XỈ HÀM SỐ 13

1.1.1 Bài toán nội suy. 13

1.1.1.1 Nội suy hàm một biến. 13

1.1.1.2 Bài toán nội suy hàm nhiều biến. 14

1.1.2 Bài toán xấp xỉ 14

1.1.3 Các phương pháp giải bài toán nội suy và xấp xỉ hàm số 14

1.2 MẠNG NƠRON NHÂN TẠO 15

1.2.1 Mạng nơron sinh học : 15

1.2.2 Mạng Nơron nhân tạo 16

1.3 MẠNG NƠRON RBF 20

1.3.1 Kỹ thuật hàm cơ sở bán kính và mạng nơron RBF 20

1.3.2 Kiến trúc mạng Nơron RBF 22

1.3.3 Đặc điểm huấn luyện của mạng Nơron RBF 23

CHƯƠNG 2 THUẬT TOÁN LẶP HDH HUẤN LUYỆN MẠNG RBF 24

2.1 THUẬT TOÁN LẶP HDH HAI PHA HUẤN LUYỆN MẠNG RBF 24

2.1.1 Phương pháp lặp đơn giải hệ phương trình tuyến tính. 24

2.1.2 Thuật toán lặp hai pha huấn luyện mạng RBF 24

2.1.3 Mô tả thuật toán. 25

2.1.4 Nhận xét. 26

2.2 THUẬT TOÁN LẶP HDH MỘT PHA HUẤN LUYỆN MẠNG RBF VỚI BỘ DỮ LIỆU CÁCH ĐỀU 27

2.2.1 Biểu diễn các mốc nội suy 27

2.2.2 Mô tả thuật toán : 27

2.2.3 Nhận xét. 28

CHƯƠNG 3 : ỨNG DỤNG THUẬT TOÁN LẶP MỘT PHA HUẤN LUYỆN MẠNG RBF VÀO VIỆC GIẢI QUYẾT BÀI TOÁN NỘI SUY XẤP XỈ VỚI DỮ LIỆU NHIỄU TRẮNG 29

3.1 NHIỄU TRẮNG VÀ BÀI TOÁN XẤP XỈ NỘI SUY VỚI DỮ LIỆU NHIỄU 29

3.1.1 Bản chất của nhiễu trắng 29

3.1.2 Phân phối chuẩn 30

3.1.3 Bài toán nội suy xấp xỉ hàm với dữ liệu nhiễu trắng 31

3.2 PHƯƠNG PHÁP HỒI QUY TUYẾN TÍNH K HÀNG XÓM GẦN NHẤT 32

3.2.1 Phát biểu bài toán hồi quy. 32

3.2.2 Mô tả phương pháp kNN 32

3.3. Ý TƯỞNG VÀ PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN NỘI SUY XẤP XỈ VỚI DỮ NHIỆU NHIỄU 33

CHƯƠNG 4 XÂY DỰNG PHẦN MỀM MÔ PHỎNG 35

4.1 LẬP TRÌNH SINH NHIỄU TRẮNG THEO PHÂN PHỐI CHUẨN 35

4.1.1 Phương pháp Box-Muller 35

4.1.2 Sinh nhiễu trắng từ hàm rand() trong C++ 36

4.2 LẬP TRÌNH GIẢI HỆ PHƯƠNG TRÌNH CỦA BÀI TOÁN HỒI QUY TUYẾN TÍNH KNN 36

4.3 GIỚI THIỆU PHẦN MỀM XẤP XỈ NỘI SUY VỚI DỮ LIỆU NHIỄU 37

4.3.1 Tổng quan phần mềm 37

4.3.2 Tổ chức dữ liệu 38

4.3.3 Giao diện và chức năng 39

4.3.3.1 Tab “Nhập dữ liệu theo file” 39

4.3.3.2 Tab “Tự nhập” 41

CHƯƠNG 5 KẾT QUẢ THÍ NGHIỆM 43

5.1 THÍ NGHIỆM VỀ VIỆC THAY ĐỔI KÍCH THƯỚC LƯỚI 43

5.2 THÍ NGHIỆM VỀ VIỆC CHỌN K 47

5.3 THÍ NGHIỆM KHI TĂNG SỐ CHIỀU 49
5.4 SO SÁNH HIỆU QUẢ VỚI PHƯƠNG PHÁP KHÁC 50
CHƯƠNG 6 TỔNG KẾT VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN..... 52
6.1 Tổng kết 52
6.2 Phương hướng phát triển của đề tài..... 53
TÀI LIỆU THAM KHẢO 54

BẢNG DANH MỤC CÁC HÌNH MINH HỌA

Hình 1 : Minh họa bài toán nội suy hàm một biến 13
Hình 2: Minh họa một Neuron thần kinh sinh học 16
Hình 3: Cấu tạo một Nơron nhân tạo 17
Hình 4: Đồ thị hàm ngưỡng 18
Hình 5: Đồ thị hàm tuyến tính 18
Hình 6: Đồ thị hàm sigmoid 18
Hình 7: Đồ thị hàm tank 19
Hình 8: Đồ thị hàm Gauss 19
Hình 9: Kiến trúc mạng Nơron truyền tới 20
Hình 10: Minh họa sự ảnh hưởng của hàm bán kính 22
Hình 11: Kiến trúc của mạng RBF 23
Hình 12: Thuật toán HDH huấn luyện mạng RBF 26
Hình 13 Dữ liệu có nhiễu trắng và hàm số chuẩn 30
Hình 14 Hàm mật độ xác suất của phân phối chuẩn với phương sai kỳ vọng khác nhau 31
Hình 15 Thể hiện lưới cách trên cơ sở miền giá trị của các mốc ban đầu 34
Hình 16 Giao diện nhập dữ liệu theo file 40
Hình 17 Giao diện nhập dữ liệu thủ công 41
Hình 18 Sai số khi chọn các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu 100 mốc ngẫu nhiên, không áp dụng heuristic "ăn gian" 44
Hình 19 Sai số khi chọn các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu 200 mốc ngẫu nhiên, không áp dụng heuristic "ăn gian" 45
Hình 20 Sai số khi áp dụng các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu ngẫu nhiên 100 mốc, có heuristic "ăn gian" 45
Hình 21 Sai số khi chọn các kích cỡ khác của lưới dữ liệu cho bộ dữ liệu 200 mốc ngẫu nhiên, có áp dụng heuristic "ăn gian" 46
Hình 22 Bảng so sánh sai số của phương pháp kNN-HDH khi áp dụng cho hàm y_1 với các cách chọn k khác nhau 47
Hình 23 Bảng so sánh sai số của phương pháp kNN-HDH khi áp dụng cho hàm y_2 với các cách chọn k khác nhau 48
Hình 24 : Bảng so sánh sai số của phương pháp kNN-HDH khi dùng và không dùng Heuristic, với số chiều tăng dần 50
Hình 25: Bảng so sánh kết quả với phương pháp GIC 50

MỞ ĐẦU

Nội suy và xấp xỉ hàm số là một bài toán quen thuộc và rất quan trọng trong các lĩnh vực khoa học đời sống từ xưa đến nay. Trường hợp hàm số một biến đã được nhà toán học Lagrange nghiên cứu và giải quyết khá tốt bằng việc dùng hàm nội suy đa thức từ thế kỷ 18. Trường hợp hàm nhiều biến vì những khó khăn trong xử lý toán học cũng như tính ứng dụng trước đây chưa nhiều nên các công cụ giải quyết bài toán hàm nhiều biến vẫn còn rất hạn chế. Ngày nay, cùng với sự phát triển mạnh mẽ của máy vi tính mà bài toán nội suy và xấp xỉ hàm nhiền biến đã trở thành một vấn đề thời sự vì tính ứng dụng lớn của nó để giải quyết các vấn đề thực tiễn như phân lớp, nhận dạng mẫu...

Mạng nơron nhân tạo được biết đến như một giải pháp tốt cho vấn đề này. Ban đầu, khái niệm "Nơron nhân tạo" được biết đến lần đầu vào khoảng đầu thế kỷ 20 trong nỗ lực của con người nhằm chế tạo ra các bộ máy có khả năng suy nghĩ và học hỏi như loài người bằng việc mô phỏng mạng nơron sinh học trong bộ não của chúng ta. Trải qua nhiều năm phát triển và nghiên cứu, cơ sở lý thuyết và thực nghiệm về mạng nơron nhân tạo đã có nhiều bước tiến đáng kể. Trong khoảng 30 năm trở lại đây, với việc có thêm khả năng tính toán mạnh mẽ từ máy vi tính mà mạng nơron nhân tạo được coi là một trong những công cụ có thể giải quyết tốt bài toán nội suy hàm nhiều biến và trong thực tế hiện nay, mạng nơron nhân tạo đã được ứng dụng rất nhiều trong các ứng dụng nội suy hàm nhiều biến như phân lớp, nhận dạng mẫu …. Mạng nơron nhân tạo có nhiều loại, trong đó có mạng nơron RBF - sau này được gọi tắt là mạng RBF - được coi là một trong những loại nơron nhân tạo tốt nhất để giải quyết bài toán nội suy hàm nhiều biến. Mạng RBF đã được chú trọng nghiên cứu và đã có khá nhiều thuật toán huấn luyện mạng RBF được áp dụng nhiều trong các ứng dụng cho thấy kết quả rất khả quan. Cùng với nhu cầu huấn luyện mạng RBF một nghiên cứu mới đây được thực hiện bởi Hoàng Xuân Huấn và các cộng sự (xem [1]) để xây dựng thuật toán huấn luyện nhanh mạng RBF đã cho ra đời một thuật toán lặp được đặt tên là là thuật toán HDH. Kết quả thực nghiệm cho thấy thuật toán lặp HDH gồm có hai pha, khi nội suy hàm nhiều biến cho sai số và tốc độ tính toán rất tốt so với các thuật toán hiện hành khác. Đặc biệt khi huấn luyện trên bộ dữ liệu cách đều thì thuật toán này chỉ cần dùng một pha và giảm tiếp phần lớn thời gian tính toán. (xem [2])

Ngoài ra trong các ứng dụng thực tế với các bài toán nội suy người ta còn thấy nổi lên một vấn đề quan trọng khác, đó là do các các yếu tố khách quan, bất khả kháng mà nảy sinh sai số tại kết quả đo tại các mốc nội suy. Việc tiến hành xây dựng hệ thống nội suy xấp xỉ dựa trên các dữ liệu sai lệch làm cho hiệu quả bị thấp. Đây là một bài toán được đặt ra từ lâu nhưng vẫn còn thu hút nhiều nghiên cứu, cải tiến cho đến tận bây giờ. Nhiều nghiên cứu đã được tiến hành để vừa nội suy xấp xỉ tốt vừa khử được nhiễu, một phương pháp được biết đến là phương pháp hồi quy tuyến tính k hàng xóm gần nhất, (từ giờ xin gọi tắt là phương pháp kNN) bằng việc xây dựng hàm tuyến tính bậc 1 để cực tiểu hóa sai số tại k điểm gần nhất so với điểm cần tìm giá trị nội suy. Nhược điểm của phương pháp này là chỉ có thể tính được giá trị hồi quy tại 1 điểm được chỉ định trước, với mỗi điểm cần tính toán lại phải hồi quy lại từ đầu, không thể xây dựng nên 1 hệ thống cho phép đưa ra ngay kết quả nội suy hàm số tại điểm tùy ý.

Với bài toán nội suy xấp xỉ trên dữ liệu nhiễu này, Hoàng Xuấn Huấn đã nảy ra ý tưởng ứng dụng thuật toán lặp HDH một pha để giải quyết, cụ thể là trên miền giá trị các mốc nội suy ban đầu, ta xây dựng nên 1 bộ các mốc nội suy mới cách đều nhau (từ giờ xin được gọi là lưới nội suy cho gọn), sau đó dùng phương pháp hồi quy tuyến tính kNN để tính giá trị tại mỗi nút của lưới nội suy mới, cuối cùng dùng thuật toán lặp HDH một pha để huấn luyện mạng nơron RBF trên bộ dữ liệu cách đều mới này, ta sẽ được một mạng nơron RBF vừa khử được nhiễu vừa nội suy xấp xỉ tốt. Phương pháp này có thể kết hợp ưu điểm khử nhiễu của phương pháp kNN với ưu điểm về tốc độ và tính tổng quát của thuật toán lặp HDH một pha đồng thời loại bỏ tính bất tiện của phương pháp kNN như đã nêu trên và hạn chế của thuật toán HDH một pha rằng dữ liệu đầu vào phải có các mốc nội suy cách đều.

Từ ý tưởng ban đầu này đến thực tế, với vô số câu hỏi cần lời đáp, như chia lưới cách đều thế nào là đủ ? Nếu quá thưa thì sai số có quá lớn không ? Nếu quá dày thì liệu thời gian huấn luyện có đạt yêu cầu không ? Các yếu tố nào ảnh hưởng đến hiệu quả huấn luyện để từ đó điều chỉnh làm tăng chất lượng mạng ? …. là một đề tài hết sức thú vị để tìm hiểu. Dưới sự giúp đỡ, chỉ bảo tận tình của thầy Hoàng Xuân Huấn, tôi đã tiến hành thực hiện khóa luận tốt nghiệp, nội dung là nghiên cứu thực nghiệm để cụ thể hóa và kiểm chứng hiệu quả của phương pháp mới này, lấy tên đề tài là : "***Huấn luyện mạng nơron RBF với mốc cách đều và ứng dụng***".

Nội dung của khóa luận sẽ đi sâu nghiên cứu những vấn đề sau :

- Khảo cứu mạng nơron RBF.
- Khảo cứu nghiên cứu thuật toán lặp HDH một pha với bộ dữ liệu cách đều.
- Tìm hiểu nhiễu trắng phân phối chuẩn và cách xây dựng.
- Khảo cứu phương pháp hồi quy tuyến tính kNN.
- Xây dựng phần mềm mô phỏng hệ thống nội suy hàm nhiều biến với dữ liệu có nhiễu dựa trên việc kết hợp phương pháp kNN và thuật toán lặp HDH một pha.
- Thông qua lý thuyết lẫn thực nghiệm, nghiên cứu đặc điểm, cải tiến hiệu quả phương pháp này, chỉ ra ưu điểm so với các phương pháp khác.

Để trình bày các nội dung nghiên cứu một cách logic, nội dung khóa luận được chia làm 4 phần chương chính :

- Chương 1 : Bài toán nội suy xấp xỉ hàm số và mạng nơron RBF :

  Chương này sẽ cung cấp cái nhìn tổng thể về những khái niệm xuyên suốt trong khóa luận, bao gồm : bài toán nội suy xấp xỉ hàm nhiều biến, mạng RBF.

- Chương 2 : Thuật toán lặp HDH huấn luyện mạng nơron RBF.

  Chương này sẽ mô tả phương pháp huấn luyện mạng RBF bằng thuật toán HDH hai pha với dữ liệu ngẫu nhiên và đặc biệt là thuật toán HDH một pha với dữ liệu cách đều làm nền tảng cho phương pháp mới.

Chương 3 : Ứng dụng thuật toán lặp một pha huấn luyện mạng RBF vào việc giải quyết bài toán nội suy xấp xỉ với dữ liệu nhiễu trắng.

Chương này sẽ khảo cứu về nhiễu trắng và phương pháp hồi quy tuyến tính kNN. Từ đó trình bày ý tưởng mới để áp dụng thuật toán HDH một pha trên bộ dữ liệu không cách đều và có nhiễu bằng cách thay bộ dữ liệu đầu vào ban đầu bằng bộ dữ liệu mới với các mốc nội suy cách đều và đã kết quả đo đã được khử nhiễu thông qua phương pháp kNN. Nó cùng với chương 5 thực nghiệm là hai chương trọng tâm của khóa luận này.

- Chương 4 : Xây dựng phần mềm mô phỏng.

  Chương này tôi trình bày về phương pháp giải quyết các bài toán nhỏ như sinh nhiễu trắng theo phân phối chuẩn, hồi quy tuyến tính kNN để

đưa ra phương hướng lập trình cho chúng. Đồng thời trình bày tổng quan và giao diện, các chức năng của phần mềm

- Chương 5 : Kết quả thí nghiệm

  Chương này tôi trình bày quá trình và kết quả nghiên cứu thực nghiệm, bao gồm việc xây dựng phần mềm mô phỏng, nghiên cứu tính tổng quát với các hàm, các bộ dữ liệu với nhau. Rút ra kết luận về đặc điểm, cách chọn lưới dữ liệu, chọn k … để hoàn thiện phương pháp này. Đồng thời so sánh sai số của phương pháp này với sai số một phương pháp khác đã được công bố tại một tạp chí khoa học quốc tế có uy tín.

- Chương 6: Tổng kết và phương hướng phát triển đề tài

  Chương này tôi tổng kết lại những gì làm được trong khóa luận này và phương hướng phát triển cho đề tài.

CHƯƠNG 1
BÀI TOÁN NỘI SUY, XẤP XỈ HÀM SỐ VÀ MẠNG NƠRON RBF

Nội dung chương này bao gồm :

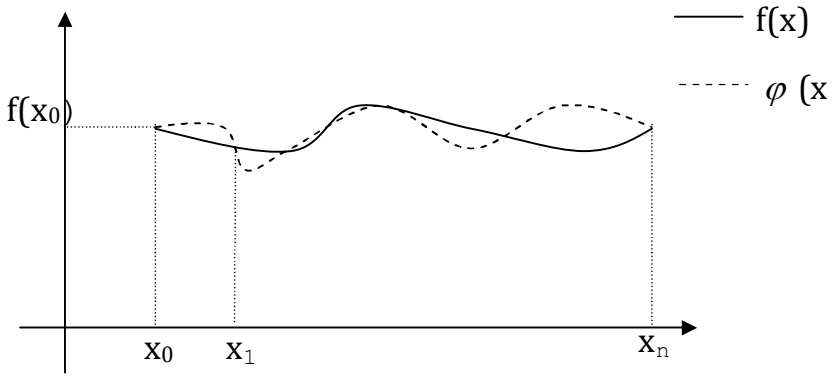
- Phát biểu bài toán nội suy và xấp xỉ hàm số
- Mạng Nơron nhân tạo
- Mạng Nơron RBF
- Bài toán nội suy xấp xỉ với dữ liệu có nhiễu trắng

1.1 BÀI TOÁN NỘI SUY VÀ XẤP XỈ HÀM SỐ

1.1.1 Bài toán nội suy.

1.1.1.1 Nội suy hàm một biến.

Bài toán nội suy hàm một biến tổng quát được đặt ra như sau: Một hàm số $y=f(x)$ ta chưa xác định được mà chỉ biết được các điểm $x_0 = a < x_1 < x_2 < ... < x_{n-1} < x_n = b$ với các giá trị $y_i = f(x_i)$. Ta cần tìm một biểu thức giải tích φ (x) để xác định gần đúng giá trị $y \approx \varphi(x)$ tại các điểm $x \in [a, b]$ của hàm $f(x)$ sao cho tại các điểm x_i thì hàm số trùng với giá trị y_i đã biết . Về phương diện hình học, ta cần tìm hàm φ (x) có dạng đã biết sao cho đồ thị của nó đi qua các điểm(x_i,y_i) với mọi $i=0,1,...,n$.



Hình 1 : Minh họa bài toán nội suy hàm một biến

Trong các ứng dụng thực tế hàm f(x) thường là hàm thực nghiệm hoặc khó tính nên các giá trị y_i chỉ lấy được bằng cách đo tại các điểm cố định x_i . Các điểm $\{x_i\}_{i=0}^{N}$ được gọi là các mốc nội suy.

1.1.1.2 Bài toán nội suy hàm nhiều biến.

Tương tự bài toán nội suy hàm một biến. Xét một hàm chưa biết $f: D(\subset R^n) \to R^m$ và một tập huấn luyện $\left\{x^k, y^k\right\}_{k=1}^{N}; \left(x^k \in R^n, y^k \in R^m\right)$ sao cho $f(x^k) = y^k; \forall k = \overline{1,n}$. Chúng ta cần tìm một hàm số φ ở một dạng đã biết để thỏa mãn điều kiện nội suy đó là : $\varphi(x^k) = y^k; \forall k = \overline{1,n}$

Với trường hợp m>1, bài toán tương đương với m bài toán nội suy m hàm nhiều biến giá trị thực, nên để đơn giản người ta thường xét bài toán có m=1.

1.1.2 Bài toán xấp xỉ

Hàm $y = f(x)$ đo được tại *n* điểm thuộc đoạn $[a,b]$

$x_1 < x_2 < \cdots < x_n;\ y_i = f(x_i)$

Với $k \le n-1$, ta tìm hàm

$$\varphi(x) = \theta(c_1, \ldots, c_k, x) \qquad (1)$$

Trong đó θ là dạng hàm cho trước, $c_{1\ldots}c_k$ là các tham số cần tìm sao cho sai số trung bình phương $\sum = \frac{1}{n}\sum_{i=1}^{n}\left(\varphi(x_i) - y_i\right)^2$ nhỏ nhất. Khi đó ta nói $\varphi(x)$ là hàm xấp xỉ tốt nhất của *y* trong lớp hàm có dạng (1) theo nghĩa tổng bình phương tối thiểu.

1.1.3 Các phương pháp giải bài toán nội suy và xấp xỉ hàm số

Bài toán nội suy hàm một biến đã được nghiên cứu nhiều từ thế kỷ 18. Ban đầu nó được giải quyết bằng phương pháp sử dụng đa thức nội suy: đa thức Lagrange, đa thức Chebysept... tuy nhiên khi số mốc nội suy lớn thì nội suy bằng đa thức thường xãy ra hiện tượng phù hợp trội(over-fitting) do bậc của đa thức thường tăng theo số mốc nội suy. Để giải quyết hiện tượng phù hợp trội, thay vì tìm đa thức nội suy người ta chỉ tìm đa thức xấp xỉ, thường được giải quyết bằng phương pháp xấp xỉ bình phương tối thiểu của Gauss. Một phương pháp khác được đề xuất vào đầu thế kỷ 20 đó là phương pháp nội suy Spline. Trong đó hàm nội suy được xác định nhờ ghép trơn các hàm nội suy dạng đơn giản (thường dùng đa thức bậc thấp) trên từng đoạn con. Phương pháp này hay được áp dụng nhiều trong kỹ thuật.

Tuy nhiên, như đã trình bày ở trên, các ứng dụng mạnh mẽ nhất của nội suy hàm nhiều biến trong thực tế ngày nay đòi hỏi phải giải quyết được bài toán nội suy hàm nhiều biến. Cùng với sự phát triển mạnh mẽ của ngành Công Nghệ

Thông Tin, bài toán nội suy xấp xỉ hàm nhiều biến được quan tâm và có những nghiên cứu đột phá trong khoảng 30 năm trở lại đây, với các cách tiếp cận chủ yếu như :

- *Học dựa trên mẫu* : Thuật ngữ này được T.Mitchell dùng để chỉ các phương pháp k-láng giêngf agần nhất, phương pháp hồi quy trọng số địa phương
- *Mạng nơron MLP*
- *Mạng nơron RBF*

Để hiểu rõ hơn, xin xem thêm trong [3]

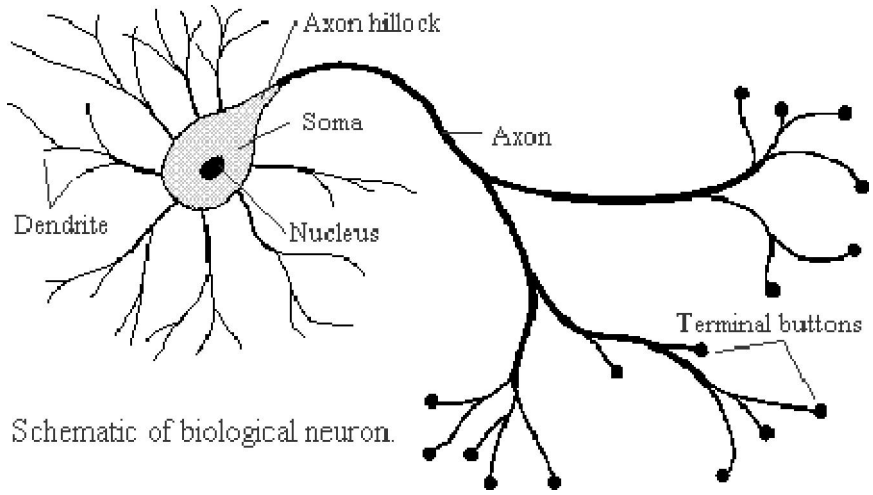
1.2 MẠNG NƠRON NHÂN TẠO

Loài người tiến hóa được đến ngày hôm nay là do có bộ não vượt trội so với các loài khác. Mặc dù vậy, bộ não người cho đến nay vẫn chứa đựng nhiều bí mật mà con người chưa giải đáp hết được. Đã có nhiền nghiên cứu về bộ não người, bao gồm những nỗ lực mô phỏng não người để tạo ra trí thông minh nhân tạo mà cấu trúc mạng nơron sinh học là một kết quả quan trọng. Mạng nơron sinh học là một mạng lưới chằng chịt các nơron có kết nối với nhau nằm trong não người.

Lấy ý tưởng từ mạng nơron sinh học, khái niệm mạng nơron nhân tạo đã ra đời, đó là một mạng gồm có các nút được thiết kế để mô hình một số tính chất của mạng nơron sinh học. Về mặt toán học thì mạng nơron nhân tạo như là một công cụ để xấp xỉ một hàm số trong không gian đa chiều. Ngoài ra, điểm giống nhau giữa mạng nơron nhân tạo và mạng nơron sinh học, đó là khả năng có thể huấn luyện hay khả năng học, đây chính là ưu điểm quan trọng nhất của mạng nơron nhân tạo, chính vì điều này mà mạng nơron nhân tạo có thể thực hiện tốt một công việc khác khi được huấn luyện và đến khi môi trường thay đổi mang nơron nhân tạo lại có thể được huấn luyện lại để thích nghi với điều kiện mới..

1.2.1 Mạng nơron sinh học :

Mạng Nơron sinh học là một mạng lưới (plexus) các Neuron có kết nối hoặc có liên quan về mặt chức năng trực thuộc hệ thần kinh ngoại biên (peripheral nervous system) hay hệ thần kinh trung ương (central nervous system).



Hình 2: Minh họa một Neuron thần kinh sinh học

Trên đây là hình ảnh của một tế bào thần kinh(Nơron thần kinh), ta chú ý thấy rằng một tế bào thần kinh có ba phần quan trọng:

-Phần đầu cũng có nhiều xúc tu (Dendrite) là nơi tiếp xúc với các với các điểm kết nối(Axon Terminal) của các tế bào thần kinh khác

-Nhân của tế bào thần kinh (Nucleus) là nơi tiếp nhận các tín hiệu điện truyền từ xúc tu. Sau khi tổng hợp và xử lý các tín hiệu nhận được nó truyền tín hiệu kết quả qua trục cảm ứng (Axon) đến các điểm kết nối (Axon Terminal) ở đuôi.

-Phần đuôi có nhiều điểm kết nối (Axon Terminal) để kết nối với các tế bào thần kinh khác.

Khi tín hiệu vào ở xúc tu kích hoạt nhân nhân Neuron có tín hiệu ra ở trục cảm ứng thì Neuron được gọi là cháy. Mặc dù W. Mculloch và W.Pitts (1940) đề xuất mô hình mạng nơron nhân tạo khá sớm nhưng định đề Heb (1949) mới là nền tảng lý luận cho mạng nơron nhân tạo.

Định đề Heb: Khi một neuron(thần kinh) A ở gần neuron B, kích hoạt thường xuyên hoặc lặp lại việc làm cháy nó thì phát triển một quá trình sinh hoá ở các neuron làm tăng tác động này.

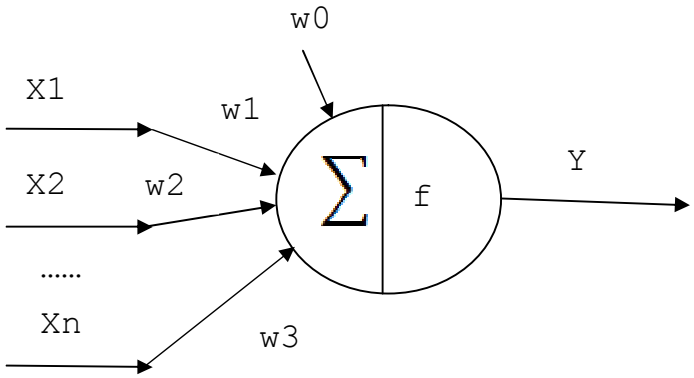
1.2.2 Mạng Nơron nhân tạo

Mạng Nơron nhân tạo được thiết kế để mô phỏng một số tính chất của mạng Nơron sinh học, tuy nhiên, ứng dụng của nó phần lớn lại có bản chất kỹ thuật. Mạng Nơron nhân tạo (Artificial Neural Network) là một máy mô phỏng cách bộ não hoạt động và thực hiên các nhiệm vụ, nó giống mạng nơron sinh học ở hai điểm :

-Tri thức được nắm bắt bởi Nơron thông qua quá trình học.

-Độ lớn của trọng số kết nối Nơron đóng vai trò khớp nối cất giữ thông tin.

a) Cấu tạo một Nơron trong mạng Nơron nhân tạo



Hình 3: Cấu tạo một Nơron nhân tạo

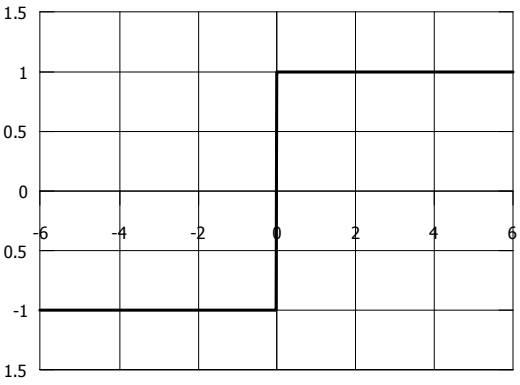
Một neuron bao gồm các liên kết nhận tín hiệu vào bao gồm các số thực x_i cùng các trọng số kết nối w_i tương ứng với nó, hàm F gọi là hàm kích hoạt để tạo tín hiệu ra dựa trên giá trị hàm tổng có trọng số của các giá trị đầu vào, Y là giá trị đầu ra của Nơron. Ta có thể biểu diễn một Nơron nhân tạo theo công thức toán học như sau:

$$Y = F\left(w_0 + \sum_{i=1}^{n} x_i w_i \right)$$

Tùy vào thực tế bài toán hàm F là một hàm cụ thể nào đấy, trong quá trình huấn luyện(học) thì các tham số w_i được xác định. Trên thực thế F thường được chọn trong những hàm sau:

1) Hàm ngưỡng

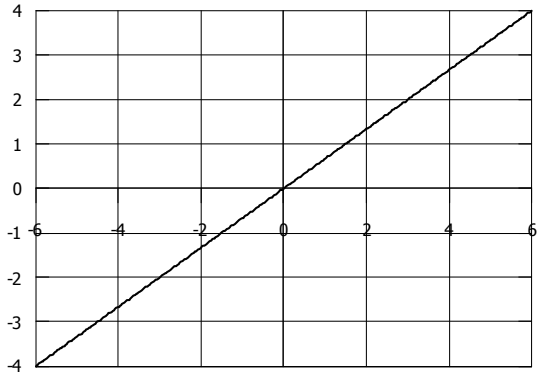
$$F(x) = \varphi(x) = \begin{cases} -1; \forall x < 0 \\ 1; \forall x \geq 0 \end{cases}$$



Hình 4: Đồ thị hàm ngưỡng

2) Hàm tuyến tính

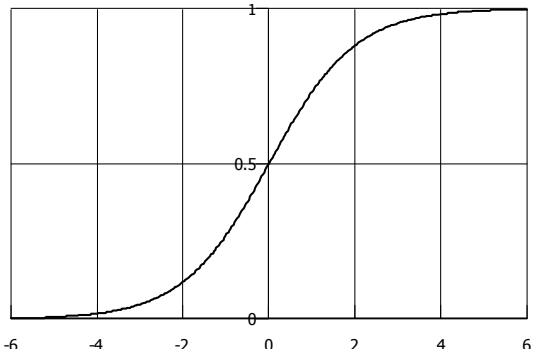
$$F(x) = ax$$



Hình 5: Đồ thị hàm tuyến tính

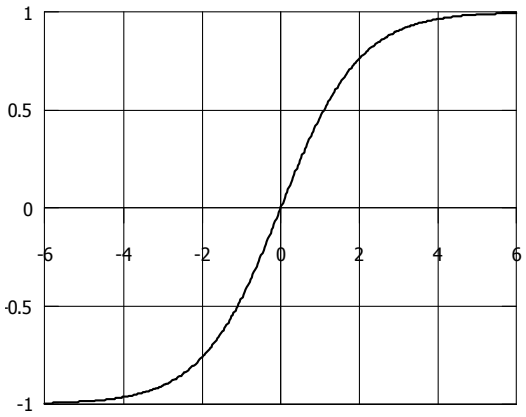
3) Hàm sigmoid

$$F(x) = \frac{1}{1+e^{-x}}$$



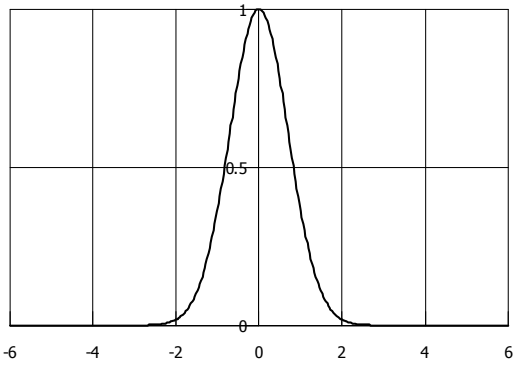
Hình 6: Đồ thị hàm sigmoid

4) Hàm tank $F(x)=\frac{1-e^{-x}}{1+e^{-x}}$



Hình 7: Đồ thị hàm tank

5) Hàm bán kính (Gauss) $F(x)=e^{-x^2}$

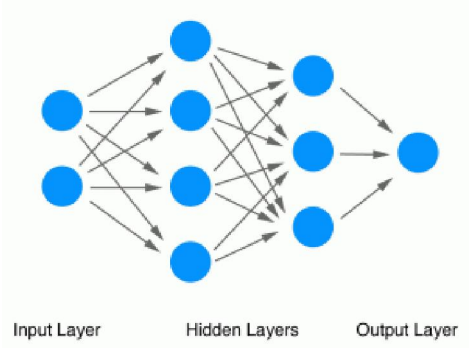


Hình 8: Đồ thị hàm Gauss

Trên thực tế thì các họ hàm sigmoid thường dùng cho mạng Nơron truyền thẳng nhiều tầng MLP vì các hàm này dễ tính đạo hàm: $f'(x)=f(x)(1-f(x))$, trong khi đó mạng Nơron RBF lại dùng hàm kích hoạt là hàm bán kính vì tính địa phương – một ưu điểm của mạng RBF sẽ được trình bày rõ hơn trong phần sau..

b) Kiến trúc của mạng Nơron nhân tạo

Kiến trúc của mạng Nơron nhân tạo lấy ý tưởng của mạng Nơron sinh học đó là sự kết nối của các Nơron. Tuy nhiên, mạng Nơron nhân tạo có kiến trúc đơn giản hơn nhiều, về cả số lượng Neuron và cả kiến trúc mạng, trong khi ở mạng Nơron tự nhiên một Neuron có thể kết nối với một Neuron khác bất kỳ ở trong mạng thì ở mạng Nơron nhân tạo các Neuron được kết nối sao cho nó có thể dễ dàng được biểu diễn bởi một mô hình toán học nào đấy. Ví dụ là trong mạng nơron truyền tới hay mạng nơron RBF các Neuron được phân thành nhiều lớp, các Neuron chỉ được kết nối với các neuron ở lớp liền trước hoặc liền sau lớp của nó



Hình 9: Kiến trúc mạng Nơron truyền tới

c) Quá trình học

Như đã nói ở trên mạng Nơron nhân tạo có khả năng huấn luyện được (học), quá trình huấn luyện là quá trình mà mạng Nơron nhân tạo tự thay đổi mình theo môi trường - ở đây là bộ dữ liệu huấn luyện - để cho ra kết quả phù hợp nhất với điều kiện của môi trường. Điều kiện để quá trình huấn luyện có thể được thực hiện là khi mạng Nơron nhân tạo đã xác định được kiến trúc cụ thể (thường là theo kinh nghiệm) trong đó bao gồm hàm kích hoạt F. Về bản chất quá trình học là quá trình xác định các tham số w_i của các Neuron trong mạng Nơron và tùy theo các thuật toán huấn luyện cụ thể, có thể bao gồm việc xác định các tham số còn chưa biết trong hàm kích hoạt. Có ba kiểu học chính, mỗi kiểu mẫu tương ứng với một nhiệm vụ học trừu tượng. Đó là học có giám sát, học không có giám sát và học tăng cường. Dưới đây xin nêu ra phương pháp học có giám sát, là phương pháp được dùng trong khóa luận này. Các phương pháp khác xem thêm [4] – chapter 4.

Học có giám sát

Trong học có giám sát, ta được cho trước một tập ví dụ gồm các cặp $(x^i, y^i, i = \overline{1..n}), x \in X, y \in Y$ và mục tiêu là tìm một hàm $f : X \to Y$ (trong lớp các hàm được phép) khớp với các ví dụ. Trên thực tế người ta thường tìm hàm f sao cho tổng bình phương sai số đạt giá trị nhỏ nhất trên tập ví dụ: $E = \sum_{i=1}^{n} \left(f(x^i) - y^i \right)^2$

1.3 MẠNG NƠRON RBF

1.3.1 Kỹ thuật hàm cơ sở bán kính và mạng nơron RBF

Hàm cơ sở bán kính được giới thiệu bởi M.J.D. Powell để giải quyết bài toán nội suy hàm nhiều biến năm 1987. Trong lĩnh vực mạng Nơron, mạng Nơron RBF

được đề xuất bởi D.S. Bromhead và D. Lowe năm 1988 cho bài toán nội suy và xấp xỉ hàm nhiều biến(xem [5]).

Dưới đây sẽ trình bày sơ lược kỹ thuật sử dụng hàm cơ sở bán kính để giải quyết bài toán nội suy hàm nhiều biến.

Kỹ thuật hàm cơ sở bán kính

Không mất tính tổng quát giả sử *m=1* khi đó hàm nội suy φ có dạng như sau :

$$\varphi(x) = \sum_{k-1}^{n} w_k \varphi_k(x) + w_0 \qquad (1)$$

ở đây φ_k là hàm cơ sở bán kính thứ *k*. Thông thường φ_k có những dạng sau:

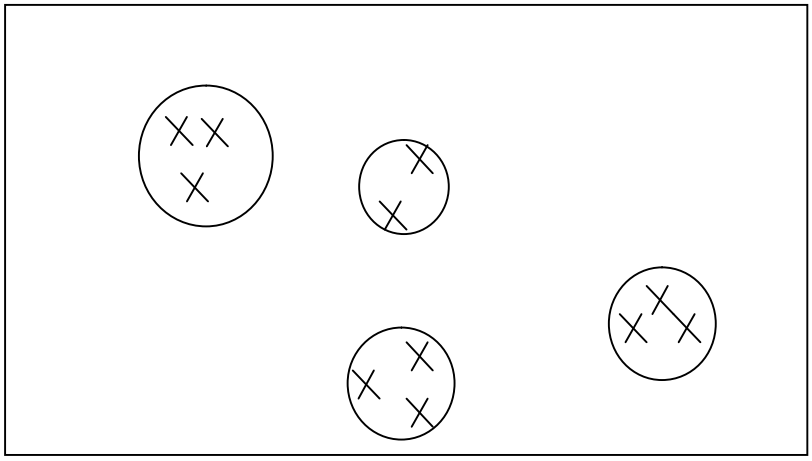
- $\varphi_k(x) - e^{\frac{-||x-v^k||^2}{\sigma_k^2}}$ (2)
- $\varphi_k(x) = \sqrt{r_k^2 + \sigma_k^2}$ (3)
- $\varphi_k(x) = \frac{1}{\sqrt{r_k^2+\sigma_k^2}}$ (4)

Trên thực tế thì người ta thường cho φ_k ở dạng (2) và trong khuôn khổ khóa luận này chỉ xét φ_k ở dạng (2).

$$\varphi_k(x) = e^{\frac{-||x-v^k||^2}{\sigma_k^2}}$$

chú ý rằng ở đây người ta ta dùng chuẩn ||.|| là chuẩn Euclidean $\|u\| = \sqrt{\sum_{i=1}^{N} u_i^2}$; v^k là tâm của mỗi hàm cở sở bán kính φ_k; σ_k là bán kính hay còn gọi là tham số độ rộng của φ_k.

Ta thấy với dạng hàm bán kính đã chọn ở trên thì khoảng cách giữa vecto input x và tâm v^k càng lớn thì giá trị của hàm bán kính càng nhỏ. Với mỗi *k* thì giá trị của bán kính σ_kđược dùng để điều khiển miền ảnh hưởng của hàm bán kính φ_k. Theo đó, nếu $\|x - v^k\| > 3\sigma_k$ thì giá hàm $\varphi_k(x) < e^{-9}$ là rất nhỏ, không có ý nghĩa.



Hình 10: Minh họa sự ảnh hưởng của hàm bán kính

Ví dụ như ở hình trên một vòng tròn to tượng trưng cho một hàm cơ sở bán kính, các hàm này chỉ ảnh hưởng đến các điểm bên trong nó bán kính của nó.

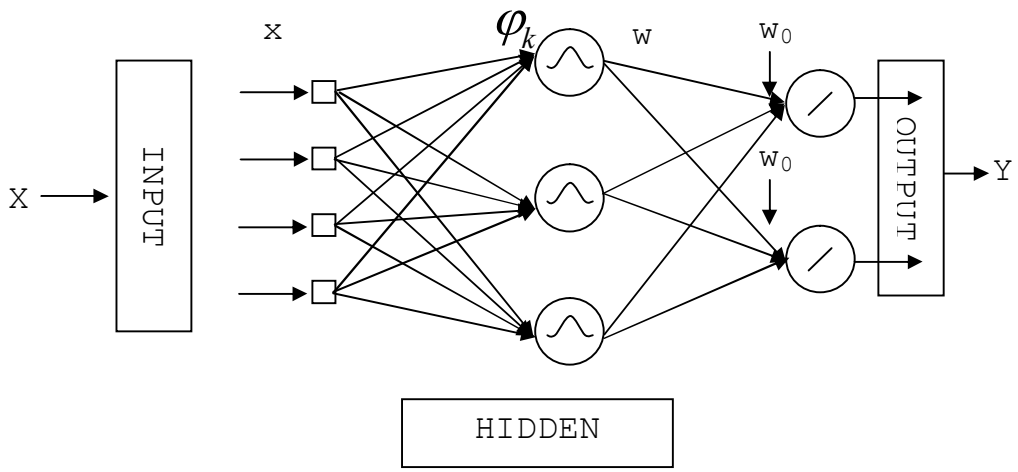
Thay công thức (2) vào (1) ta được biểu diễn toán học của kỹ thuật hàm cơ sở bán kính như sau:

$$\varphi(x^j) = \sum_{k=1}^{N} w_k \varphi_k(x^j) + w_0 = \sum_{k=1}^{N} w_k e^{\frac{-\left\|x^j - v^k\right\|^2}{\sigma_k^2}} + w_0 = y^j \quad (5)$$

Một đặc điểm rất lợi thế khi sử dụng hàm bán kính để giải quyết bài toán nội suy hàm nhiều biến, đó là khi xét giá bình phương sai số $E = \sum_{i=1}^{n} \left(\varphi\left(x^i\right) - y^i\right)^2$ thì người ta đã chứng minh được rằng *E* chỉ có một cực trị duy nhất. Do vậy việc tìm các tham số của các hàm cơ sở bán kính(w_k, v^k, σ_k) để cho *E* đạt cực tiểu sẽ được giải quyết rất nhanh và hiệu quả.

1.3.2 Kiến trúc mạng Nơron RBF

Mạng RBF là một loại mạng Nơron nhân tạo truyền thẳng gồm có ba lớp. Nó bao gồm n nút của lớp đầu vào cho vector đầu vào $x \in R^n$, N nơron ẩn (giá trị của Nơron ẩn thứ *k* chính là giá trị trả về của hàm cơ sở bán kính φ_k) và m Nơron đầu ra.



Hình 11: Kiến trúc của mạng RBF

Dĩ nhiên, như đã nói ở trên, không mất tính tổng quát, nội dung khóa luận này chỉ xét trường hợp m=1.

1.3.3 *Đặc điểm huấn luyện của mạng Nơron RBF*

Ưu điểm của mạng RBF là thời gian huấn luyện ngắn, việc thiết lập rất nhanh và đơn giản . Ngày nay mạng Nơron RBF được sử dụng trong rất nhiều lĩnh vực:

- Xử lý ảnh
- Nhận dạng tiếng nói
- Xử lý tín hiệu số
- Xác định mục tiêu cho Radar
- Chuẩn đoán y học
- Quá trình phát hiện lỗi
- Nhận dạng mẫu
- ….

CHƯƠNG 2 :
THUẬT TOÁN LẶP HDH HUẤN LUYỆN MẠNG RBF

Nội dung chương này bao gồm :

2.1 Mô tả thuật toán lặp HDH hai pha dùng cho dữ liệu huấn luyện bất kỳ

2.2 Mô tả thuật toán lặp HDH một pha dùng cho dữ liệu huấn luyện cách đều

(chi tiết về 2 thuật toán này xin xem thêm tại [6])

2.1. THUẬT TOÁN LẶP HDH HAI PHA HUẤN LUYỆN MẠNG RBF

Trong chương này, trước khi đề cập đến thuật toán lặp, tôi xin được trình bày cơ sở lý thuyết được dùng để xây dựng thuật toán

2.1.1 Phương pháp lặp đơn giải hệ phương trình tuyến tính

Giả sử ta cần giải hệ phương trình

$$Ax=B$$

Trước hết ta đưa về hệ tương đương

$$X=Bx+d$$

Trong đó B là ma trận vuông cấp n thỏa mãn$_j$

$$||B||=\max\{\sum_1^n | b_{i,j}| \ / \ i=1..n\}$$

Phương pháp lặp đơn

Với vecto $x^0=\begin{bmatrix} x^{01} \\ x^{02} \\ \dots \\ x^{0n} \end{bmatrix}$ bất kỳ, dãy nghiệm của phương trình được xây dựng bởi công thức lặp

$$x^{k+1} = Bx^k + d; \ (x^{k+1}_i = \sum_{j=1}^n b_{i,j}x^k_j + d)$$

thỏa mãn $\lim_{k\to\infty} x^k = x^*$ trong đó x* là nghiệm đúng duy nhất của

Với ước lượng sai số

$$||x^*_i - x^k_i|| <= \frac{q}{1-q} \max\{| x^k_j - x^{k-1}_j |\} \quad \forall\, i \le n$$

Thông thường ta có thể chọn x^0=d, khi đó coi như ta đã tính xấp xỉ ban đầu với $x^0 = 0$ và x^0=d là bước tiếp theo.

2.1.2 Thuật toán lặp hai pha huấn luyện mạng RBF

Xét tập huấn luyện $\{x^k, y^k\}_{k=1}^N ; (x^k \in R^n, y^k \in R^m)$, không mất tính tổng quát, ở đây ta xét mạng RBF có một Nơron output (*m=1*), khi đó biểu diễn toán học của mạng RBF là:

$$\phi(x^i) = \sum_{k=1}^{N} w_k \phi_k(x^i) + w_0 = y^i \quad (1)$$

Xét ma trận $\Phi = \left(\varphi_{k,i}\right)_{N\times N}$ trong đó $\varphi_{k,i} = \varphi_k(x^i) = e^{-\|x^i - x^k\|^2/\sigma_k^2}$, chú ý rằng ở đây ta chọn tâm của các hàm cơ sở bán kính chính là tất cả các điểm thuộc tập dữ liệu input *X*.

Ta ký hiệu *I* là ma trận đơn vị cấp *N*; $W=\begin{bmatrix} w_1 \\ ... \\ w_N \end{bmatrix}$, $Z=\begin{bmatrix} z_1 \\ ... \\ z_N \end{bmatrix}$ là các véc tơ trong không gian *N*-chiều R^N trong đó:

$$z_k = y_k - w_0, \qquad \forall k \le N \quad (2)$$

và đặt

$$\Psi = I - \Phi = \left|\psi_{k,j}\right|_{NxN} \quad (3)$$

thì

$$\psi_{k,j} = \begin{cases} 0; khi: k = j \\ -e^{-\|x^j - x^k\|^2/\sigma_k^2}; khi: k \neq j \end{cases} \quad (4)$$

Khi đó hệ phương trình (1) tương đương với hệ :

$$W = \Psi W + Z \quad (5)$$

Như đã nói ở 1.3.1, với các tham số σ_k đã chọn và w_0 tùy ý, hệ (1) và do đó hệ (5) luôn có duy nhất nghiệm W. Về sau giá tri w_0 được chọn là trung bình cộng của các giá trị y^k:

$$w_0 = \frac{1}{N}\sum_{k=1}^{N} y^k \quad (6)$$

Với mỗi k≤N, ta có hàm q_k của σ_k xác định như sau:

$$q_k = \sum_{j=1}^{N} \left|\psi_{k,j}\right| \quad (7)$$

Hàm q_k là đơn điệu tăng và với mọi số dương q <1 luôn tồn tại giá trị σ_k sao cho $q_k(\sigma_k)$=q.

2.1.3. Mô tả thuật toán.

Với sai số ε và các hằng số dương q, α <1 cho trước, thuật toán bao gồm 2 pha để xác định các tham số σ_k và W^*. Trong pha thứ nhất, ta sẽ xác định các σ_k để q_k ≤q và gần với q nhất (nghĩa là nếu thay $\sigma_k=\sigma_k/\alpha$ thì q_k>q). Pha sau tìm nghiệm gần

đúng W^* của (5) bằng phương pháp lặp đơn giản. Thuật toán được đặc tả trong hình 12.

Proceduce Thuật toán 2 pha huấn luyện mạng RBF

for k=1 to N do

Xác định các σ_k để $q_k \le q$, và nếu thay $\sigma_k=\sigma_k/\alpha$ thì $q_k>q$; // Pha 1

Tìm W^* bằng phương pháp lặp đơn(hoặc phương pháp lặp Seidel); //Pha 2

End

Hình 12: Thuật toán HDH huấn luyện mạng RBF

Để tìm nghiệm W^* của hệ (5) ta thực hiện thủ tục lặp như sau.

Khởi tạo $W^0=Z$;

Tính

$W^{k+1}= \Psi W^k + Z$; (8)

Nếu điều kiện kết thúc chưa thỏa mãn thì gán $W^0 := W^1$ và trở lại bước 2 ;

Với mỗi vectơ N-chiều u, ta ký hiệu chuẩn $\|u\|_* = \sum_{j=1}^{N} |u_j|$, điều kiện kết thúc có thể chọn một trong biểu thức sau.

a)

$$\frac{q}{1-q} \| W^1 - W^0 \|_* \le \varepsilon \quad (9)$$

b)

$$t \ge \frac{\ln \frac{\varepsilon(1-q)}{\|Z\|_*}}{\ln q} = \frac{\ln \varepsilon - \ln \|Z\|_* + \ln(1-q)}{\ln q} \quad (10)$$

, với t là số lần lặp.

2.1.4. Nhận xét

Thuật toán này có ưu điểm là cài đặt rất đơn giản và tốc độ hội tụ rất nhanh và ta có thể điều chỉnh giá trị sai số nội suy nhỏ tùy ý. Song do kiến trúc mạng phức tạp nên thường xãy ra hiện tượng phù hợp trội(over-fitting) cho tập dữ liệu huấn luyện. Để hiểu chi tiết hơn về thuật toán HDH (xem thêm tại [6]). Tại đó, tác giả, với các kết quả nghiên cứu thực nghiệm đã cho thấy tốc độ tính toán và tính tổng quát của thuật toán lặp hai pha HDH tốt hơn nhiều so với các thuật toán kinh điểm khác như phương pháp lặp Gradient hay thuật toán QTL ..... Để cho gọn và phân biệt với thuật toán lặp một pha sắp trình bày ngay sau đây, ta gọi thuật toán lặp HDH hai pha này là thuật toán HDH-2

2.2 THUẬT TOÁN LẶP HDH MỘT PHA HUẤN LUYỆN MẠNG RBF VỚI BỘ DỮ LIỆU CÁCH ĐỀU

Thuật toán lặp hai pha trên có đặc điểm thời gian huấn luyện của pha một chiếm phần lớn. Với trường hợp các mốc huấn luyện là mốc cách đều, thuật toán lặp hai pha có thể bỏ đi pha thứ nhất này, trở thành thuật toán một pha. Thuật toán này huấn luyện trên các mốc cách đều thường áp dụng với các ứng dụng ở lĩnh vực đồ họa máy tính, nhận dạng mẫu, các bài toán kỹ thuật …. và là cơ sở để giải quyết bài toán nội suy với bộ dữ liệu huấn luyện có nhiễu sắp trình bày trong chương tiếp theo.

2.2.1 Biểu diễn các mốc nội suy

Các mốc nội suy là các mốc cách đều, có thể được biểu diễn dưới dạng

$$x^{i1,i2\ldots,in} = (x_1^{i1},\ldots, x_n^{in})$$

trong đó $x_k^{ik} = x_k^0 + ik.h_k$. Với k đặc trưng cho chiều, h_k (k=1,2,..,n) là hằng số biểu diễn khoảng cách giữa 2 mốc cách đều của 1 chiều, biểu diễn sự thay đổi của chiều x_k ; ik nhận giá trị từ 0 đến n_k ; với n_k+1 là số mốc chia mỗi chiều

2.2.2. *Mô tả thuật toán* :

Thay cho chuẩn Euclide, ta xét chuẩn Mahalanobis : $||x|| = x^T Ax$, với A là ma trận có dạng

$$\begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \cdots & \cdots & \cdot & \cdots \\ 0 & 0 & \cdots & \alpha_n \end{bmatrix}$$

Các tham số α_k sẽ được chọn = h_k . Khi đó, biểu thức (1) tại mục 2.1.2 có thể được viết lại như công thức sau :

$$\varphi(x) = \sum_{i1\ldots in=1}^{N1..Nn} w_{i1..in}\varphi_{i1..in}(x)+w_0$$

Ma trận Ψ trở thành :

$$\Psi_{i1..in}^{j1..jn} = \begin{cases} 0 \; khi \; i1..in = j1..jn \\ -e^{-\sum_{p=1}^{n}\frac{(jp-ip)^2}{\sigma^2}} \end{cases}$$

$q_{i1..in}$ có thể viết lại là

$$q_{i1..in} = \sum_{j_1\ldots j_n \neq i_1\ldots i_n} -e^{-\sum_{p=1}^{n}\frac{(jp-ip)^2}{\sigma^2}}$$

Áp dụng một vài biến đổi toán học, xem chi tiết tại [6], tác giả đã chứng minh được rằng để bán kính $\sigma_{i1..in}$ thỏa mãn điều kiện $q_{i1..in} < q$ thì :

$$\sigma_{i1..in} \leq \sqrt{\frac{1}{\ln\left(\frac{6}{\sqrt[n]{1+q}-1}\right)}}$$

Như vậy, ta có thể chọn

$$\sigma_{i1..in=} \sqrt{\frac{1}{\ln\left(\frac{6}{\sqrt[n]{1+q}-1}\right)}}$$

cho mọi hàm bán kính để đảm bảo điều kiện dừng luôn xảy ra. Như vậy, pha ban đầu tính tham số độ rộng cho từng hàm bán kính, chiếm phần lớn thời gian huấn luyện đã được giải quyết tức thì, bài toán lặp hai pha trở thành thuật toán lặp một pha huấn luyện trên các mốc cách đều.

2.2.3. Nhận xét

Theo các kết quả thực nghiệm tại [6], cùng với việc cho thấy thuật toán lặp hai pha HDH đã cho thấy tính tổng quát tốt và thời gian huấn luyện nhanh hơn nhiều so với các thuật toán khác, cũng tại [6], bằng các kết quả thực nghiệm tác giả cũng chỉ ra rằng thuật toán lặp một pha HDH này với việc giảm đi phần lớn thời gian huấn luyện đã cho thấy ưu điểm rất lớn ở tốc độ tính toán, ngoài ra còn cho thấy tính tổng quát của thuật toán lặp HDH một pha còn tốt hơn so với thuật toán lặp hai pha HDH.

Thuật toán này có đặc điểm là cùng với 1 miền giá trị, nếu các mốc cách đều được chia càng dày đặc thì tính tổng quát càng tốt.

Để cho gọn, từ đây thuật toán này sẽ được gọi là HDH-1 để phân biệt với thuật toán HDH-2, và khi gọi thế này nghiễm nhiên ta coi bộ dữ liệu huấn luyện là bộ dữ liệu bao gồm các mốc nội suy cách đều.

CHƯƠNG 3 :
ỨNG DỤNG THUẬT TOÁN LẶP MỘT PHA HUẤN LUYỆN MẠNG RBF VÀO VIỆC GIẢI QUYẾT BÀI TOÁN NỘI SUY XẤP XỈ VỚI DỮ LIỆU NHIỄU TRẮNG

Nội dung chương này bao gồm :

- Nhiễu trắng và bài toán nội suy xấp xỉ có nhiễu trắng
- Phương pháp hồi quy tuyến tính kNN
- Trình bày ý tưởng và phương pháp giải quyết bài toán

4.1. NHIỄU TRẮNG VÀ BÀI TOÁN XẤP XỈ NỘI SUY VỚI DỮ LIỆU NHIỄU

4.1.1. Bản chất của nhiễu trắng

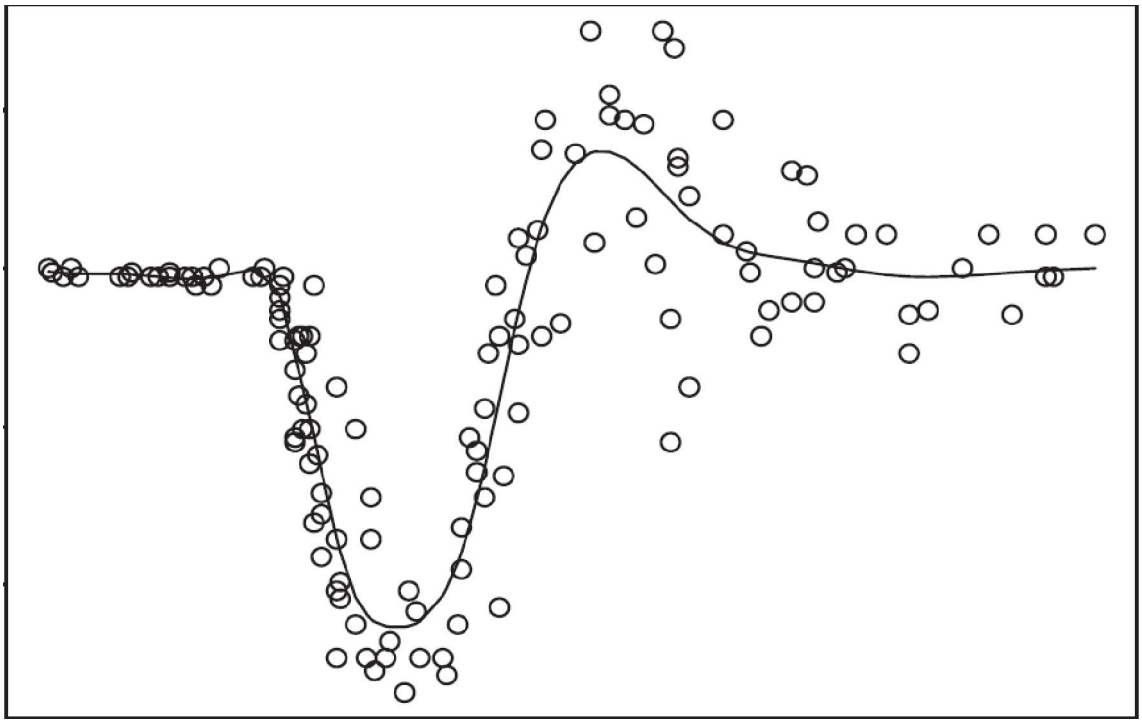
Nhiễu trắng là một vấn đề được phát sinh trong các ứng dụng thực tiễn, nó là sai số hay lỗi trong khi đo các mốc nội suy. Thông thường, nếu không muốn nói là hầu hết các phép đo đều có lỗi, kết quả đo được cho bởi công thức

$$X=T+E$$

Trong đó X là kết quả đo, T là giá trị chính xác và E là lỗi trong quá trình đo. Trong đó, lỗi E được là tổng hợp của E_r (lỗi ngẫu nhiên) và E_s (lỗi hệ thống).

$$\mathbf{E=E_r+E_s}$$

Lỗi hệ thống là lỗi do các yếu tố chủ quan về công cụ đo lường hoặc do các tác động ngoại cảnh có thể khắc phục được, vì vậy với bài toán tổng quát nội suy hàm nhiền biến tron khóa luận này ta chỉ xét lỗi ngẫu nhiên. Lỗi ngẫu nhiên được gây ra bởi bất kỳ yếu tố ngẫu nhiên nào có ảnh hưởng đến sự đo lường trên mẫu. Theo lý thuyết về sai số đo lường, sai số ngẫu nhiên không có bất kỳ tác dụng nhất quán nào trên toàn bộ mẫu. Thay vào đó, nó làm cho các giá trị đo được tăng hoặc giảm một cách ngẫu nhiên so với giá trị chính xác. Điều này có nghĩa là nếu chúng ta coi tất cả các lỗi ngẫu nhiên nằm trên một phân bố thì chúng sẽ có tổng bằng 0. Đặc điểm quan trọng của lỗi ngẫu nhiên là nó có thể biến đổi dữ liệu nhưng không làm ảnh hưởng đến giá trị trung bình của nhóm. Cho nên, các lỗi ngẫu nhiên này được gọi là nhiễu trắng. Để rõ hơn chi tiết xin xem thêm trong [7]



Hình 13 Dữ liệu có nhiễu trắng và hàm số chuẩn

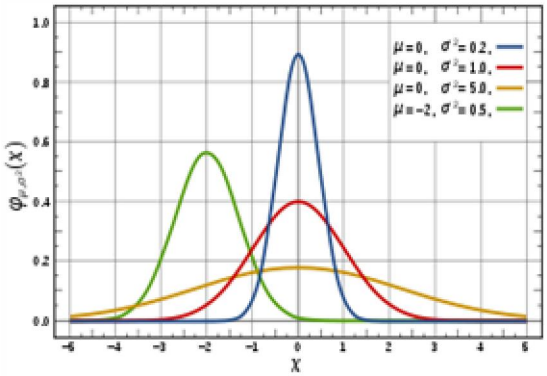
Để thể hiện sai số mà ở đây là sai số ngẫu nhiên, tức nhiễu trắng, người ta tìm cách biểu diễn chúng trong một phân phối, ở đây ta sử dụng phân phối chuẩn.

3.1.2 Phân phối chuẩn

Phân phối chuẩn, hay còn gọi là phân phối Gauss (xem chi tiết ở [8]), là một phân phối xác suất cực kỳ quan trọng trong nhiều lĩnh vực. Nó là họ phân phối có dạng tổng quát giống nhau, chỉ khác tham số trung vị và phương sai**.** Cách dễ thấy nhất để thể hiện đặc tính của phân phối này là thông qua hàm mật độ xác suất với công thức :

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Trong đó μ được gọi là trung vị, là giá trị các mật độ xác suất cao nhất và là trung bình cộng của phân phối. Hàm Gauss đối xứng qua μ. σ^2 được gọi là phương sai, nó thể hiện mức độ tập trung của phân phối xung quanh trung vị. σ được gọi là độ lệch chuẩn và chính là độ rộng của hàm mật độ. Phân phối chuẩn được thể hiện dưới dạng $N(\mu, \sigma^2)$



Hình 14 Hàm mật độ xác suất của phân phối chuẩn với phương sai kỳ vọng khác nhau

Hình trên thể hiện hàm mật độ với 4 tập tham số khác nhau, ta thấy đường có μ=-2 đối xứng với đường x=-2 nằm lệch hẳn ra khỏi 3 đường còn lại đối xứng qua đường thẳng x=0. Các hàm có phương sai khác nhau thể hiện sự tập trung xung quanh trị trung bình khác nhau, các hàm có phương sai nhỏ thể hiện sự tập trung dày đặc xung quanh trị trung bình và ngược lại.

Một số tính chất với phân phối chuẩn:

- Hàm mật độ là đối xứng qua giá trị trung bình.
- 68.26894921371% của diện tích dưới đường cong là nằm trong độ lệch chuẩn 1 tính từ trị trung bình.
- 95.4499736103% của diện tích dưới đường cong là nằm trong độ lệch chuẩn 2.
- 99.7300203936% của diện tích dưới đường cong là nằm trong độ lệch chuẩn 3.
- 99.9936657516% của diện tích dưới đường cong là nằm trong độ lệch chuẩn 4.
- ….

Trong thực nghiệm, ta thường giả thiết rằng dữ liệu lấy từ tổng thể có dang phân phối xấp xỉ chuẩn. Nếu giả thiết này được kiểm chứng thì có khoảng 68% số giá trị nằm trong khoảng 1 độ lệch chuẩn so với trị trung bình, khoảng 95% số giá trị trong khoảng hai lần độ lệch chuẩn và khoảng 99.7% nằm trong khoảng 3 lần độ lệch chuẩn. Đó là "quy luật 68-95-99.7" hoặc quy tắc kinh nghiệm.

3.1.3 Bài toán nội suy xấp xỉ hàm với dữ liệu nhiễu trắng

Bài toán này cũng tương tự như bài toán nội suy xấp xỉ hàm nhiều biến như đã nêu trên, điểm khác biệt là ở bài toán này, bộ dữ liệu huấn luyện mà ta có bao gồm các mốc nội suy và các giá trị bằng giá trị đo được tại các mốc đó cộng với sai số và dãy các sai số là một dãy nhiễu trắng phân bố chuẩn. Cụ thể là :

Hàm *f(x)* đo được tại *n* điểm $x_1,x_2,..,x_n$ thuộc miền D được các kết quả :

$$y_i = f(x_i) + \varepsilon_i \; \forall i = \overline{1,n}$$

Với $\varepsilon \sim N(0, \sigma^2)$

Ta cần tìm hàm $\varphi(x)$ sao cho trung bình tổng bình phương sai số

$$\sum = \frac{1}{n}\sum_{i=1}^{n}\left(\varphi(x_i) - f(x_i)\right)^2$$

nhỏ nhất.

3.2 PHƯƠNG PHÁP HỒI QUY TUYẾN TÍNH K HÀNG XÓM GẦN NHẤT

Đây là một phương pháp phổ biến để hồi quy hàm số, ưu điểm của nó là cài đặt đơn giản và có thể khử nhiễu, nhược điểm lớn nhất của nó là chỉ có thể hồi quy tại những điểm định trước, với mỗi giá trị cần biết thì phải hồi quy lại từ đầu, không thể xây dựng nên một hệ thống cho ra kết quả xấp xỉ ngay tại mỗi điểm bất kỳ. Tuy nhiên nó được coi là một công cụ hữu hiệu được áp dụng trong nhiều phương pháp, trong đó có phương pháp sắp được giới thiệu sau đây đây. Nhưng trước hết, tôi xin được mô tả phương pháp hồi quy tuyến tính k hàng xóm gần nhất (từ sau xin được gọi tắt là phương pháp kNN)

3.2.1 Phát biểu bài toán hồi quy.

Xét miền giới nội D trong R^n và $f: D\ (\subset R^n) \to R^m$ là một hàm liên tục xác định trên D. Người ta chỉ mới xác định được tại tập T gồm N điểm $x^1, x^2 .... x^N$ trong D là $f(x^i) = y^i$ với mọi $i=1,2...,N$ và cần tính giá trị của $f(x)$ tại các điểm x khác trong D ($x = x_1,\ldots,x_n$).

Ta tìm một hàm $\varphi(x)$ xác định trên D có dạng đã biết sao cho:

$$\varphi(x^i) \approx y^i \ , \ \forall i=1,...N. \qquad (1.)$$

và dùng $\varphi(x)$ thay cho f(x). Khi m >1, bài toán nội suy tương đương với m bài toán nội suy m hàm nhiều biến giá trị thực, nên để đơn giản ta chỉ cần xét với m=1.

3.2.2 Mô tả phương pháp kNN

Trong phương pháp này, người ta chọn trước số tự nhiên k. Với mỗi $x \in D$, $x = x_1,\ldots,x_n$ ta xác định giá trị $\varphi(x)$ qua giá trị của f tại k mốc nội suy gần nó nhất như sau.

Ký hiệu $z_1,\ldots,z_k$ là k điểm trong T gần x nhất (với $d(u,v)$ là khoảng cách của hai điểm u,v bất kỳ trong D đã cho), khi đó $\varphi(x)$ xác định như sau:

$$\varphi(x) = \sum_{j=1}^{n} \rho_j x_j + \rho_0 \qquad (2)$$

Trong đó ρ_i được xác định để tổng bình phương sai số trên tập điểm $z_1,\ldots,z_k$ đạt cực tiểu.

Tức là: $\Sigma = \frac{1}{2}\sum_{i=1}^{k}\left(\varphi(z^i) - f(z^i)\right)^2 = \sum_{i=1}^{k}\frac{1}{2}\left(\sum_{j=1}^{n}\rho_j z_j^i + \rho_0 - f(z^i)\right)^2$ nhỏ nhất, với . Ta tìm các hệ số ρ_i (phụ thuộc x) nhờ hệ phương trình:

$$\frac{\partial \Sigma}{\partial p_t} = 0 \quad \forall t = 0,..,n$$

Tức là hệ

$$\sum_{i=1}^{k}\left(\sum_{j=1}^{n} p_j z_j^i + \rho_0 - f(z^i)\right) z_t^i = 0 \qquad (3)$$

Và

$$\sum_{i=1}^{k}\left(\sum_{j=1}^{n} p_j z_j^i + \rho_0 - f(z^i)\right) = 0 \qquad (4)$$

Giải hệ (3,4), với mỗi x ta xác định được bộ p_t, t = tương ứng để xác định $\varphi(x)$ theo (2).

3.3 Ý TƯỞNG VÀ PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN NỘI SUY XẤP XỈ VỚI DỮ NHIỆU NHIỄU

Với việc huấn luyện trên các mốc cách đều, thuật toán lặp một pha HDH hứa hẹn có thể áp dụng nhiều vào các ứng dụng cụ thể, đòi hỏi thời gian huấn luyện nhanh trong các lĩnh vực như đồ họa máy tính, nhận dạng mẫu …

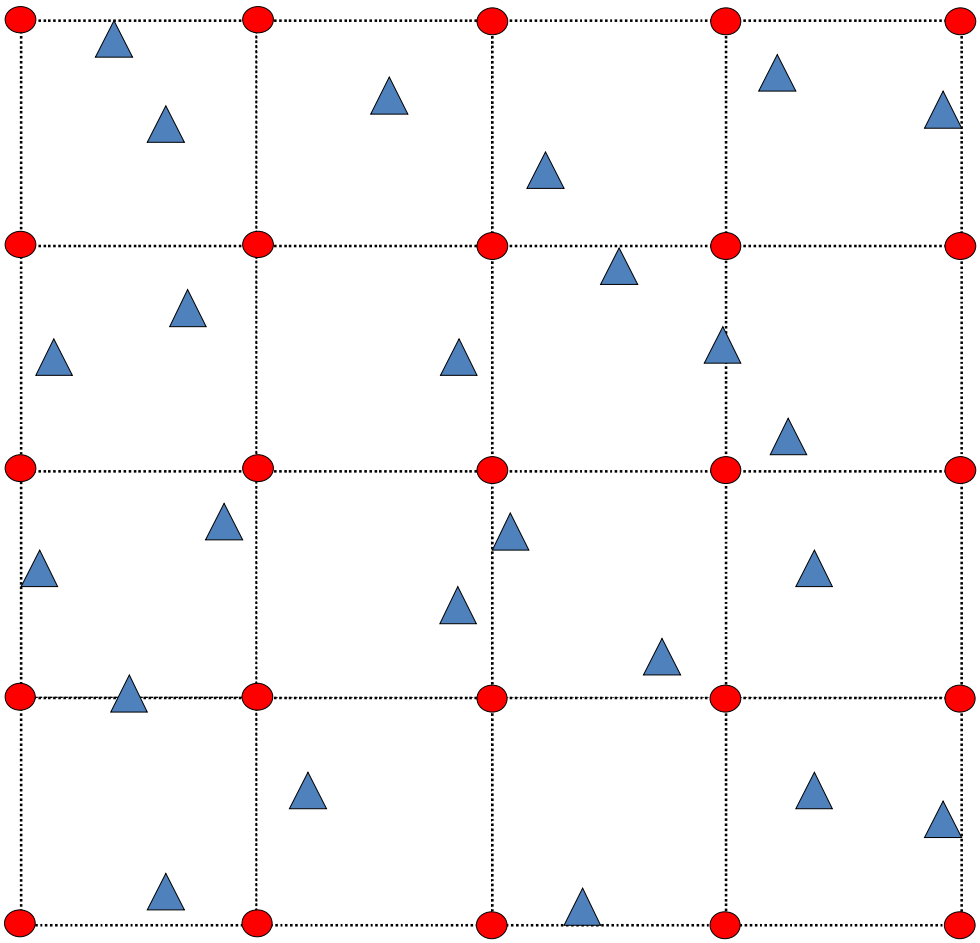
Để tận dụng tối đa các ưu điểm và tăng phạm vi áp dụng, Hoàng Xuân Huấn đã đưa ra ý tưởng để ứng dụng thuật toán HDH-1 trong việc quyết bài toán nội suy hàm nhiều biến có nhiễu trắng, và các mốc nội suy không cách đều.

Bản chất của phương pháp này là :

Bước 1 : Dựa trên bộ dữ liệu ban đầu với các mốc nội suy không cách đều và giá trị đo được tại mỗi mốc bị nhiễu trắng, bằng phương pháp hồi quy, ta tạo ra một bộ dữ liệu mới với các mốc nội suy là các nút cách đều trên 1 lưới dữ liệu xác định trước trong miền giá trị của các mốc nội suy ban đầu. Giá trị đo được tại mỗi mốc cách đều mới đã được khử nhiễu.

Bước 2: Sau khi có bộ dữ liệu mới gồm các mốc nội suy cách đều và giá trị đầu ra đã được khử nhiễu, dùng thuật toán lặp HDH một pha huấn luyện mạng RBF trên

bộ dữ liệu mới này, ta được 1 mạng vừa có khả năng nội suy xấp xỉ hàm, vừa khử được nhiễu.



Hình 15 Thể hiện lưới cách trên cơ sở miền giá trị của các mốc ban đầu

Hình trên thể hiện trong trường hợp dữ liệu nội suy có 2 chiều, lưới các mốc nội suy mới (các hình tròn) được xây dựng dựa trên miền giá trị của các mốc nội suy cũ (hình tam giác. Giá trị tại mỗi mốc nội suy cách đều (hình tròn) sẽ được tính bằng cách hồi quy dựa trên các giá trị đo được tại k mốc cũ (hình tam giác) gần nó nhất. Mạng RBF sẽ được huấn luyện bằng thuật toán HDH-1 dựa trên bộ dữ liệu gồm đầu vào là các mốc nội suy mới, cách đều (các hình tròn) và giá trị đã được khử nhiễu tại mỗi mốc.

CHƯƠNG 4
XÂY DỰNG PHẦN MỀM MÔ PHỎNG

Nội dung chương này bao gồm :

- Lập trình sinh nhiễu trắng theo phân phối chuẩn
- Lập trình giải bài toán hồi quy tuyến tính kNN
- Tổng quan phần mềm

Các mô tả lập trình trong chương này sẽ nêu ra các phương án lập trình để giải quyết các bài toán nhỏ đã đề cập ở trên, cụ thể là cách sinh nhiễu trắng theo phân phối chuẩn và lập trình giải bài toán hồi quy tuyến tính kNN.

4.1 LẬP TRÌNH SINH NHIỄU TRẮNG THEO PHÂN PHỐI CHUẨN

Để xây dựng phân phối chuẩn từ hàm phân phối đều rand() của C++, tôi đã dựa theo phương pháp Box Muller (xem chi tiết tại [9]) được trình bày dưới đây :

4.1.1 Phương pháp Box-Muller

Từ một tính chất của phân phối Gauss : "*Nếu* $X \sim N(\mu, \sigma^2)$ *và a,b là các số thực thì* $aX + b \sim N(a + b, (a\sigma)^2)$*"*

Ta có thể tìm ra dãy $X \sim N(\mu, \sigma^2)$ với μ, σ bất kỳ từ dãy $Y \sim N(0,1)$ bởi công thức : $X \sim \sigma Y + \mu$

Phương pháp Box-Muller cho phép ta sinh ra 1 dãy phân phối chuẩn $\sim N(0,1)$ để từ đó có thể chuẩn qua dãy phân phối chuẩn $\sim N(\mu, \sigma^2)$ bất kỳ. Phương pháp này được trình bày như sau :

Phân phối $N(0,1)$, theo định nghĩa, được biểu diễn dưới dạng hàm phân phối xác suất:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Như vậy, để lấy ra 1 cặp X=x, Y=y ở 2 dãy phân phối chuẩn tương ứng hàm mật độ xác suất sẽ là :

$$f(x,y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}}$$

Ta đặt

$$X = r \cos\theta$$

$$Y = r \sin\theta$$

Ta sẽ tìm cách sinh dãy R và θ để từ đó sinh dãy X, Y

Từ …

$\Pr(X \le r) = 1$

do đó θ sẽ được sinh theo phân phối đều trong miền $[0; 2\pi]$ hay có thể viết dưới dạng

$\theta = 2\pi U_1$

với U_1 được phân phối đều trong miền $[0; 1]$.

Để sinh r, từ …. Ta định nghĩa hàm U(R) tính xác suất để sinh cặp *(x,y)* sao cho $x^2 + y^2 \le R^2$ $(r \le R)$là :

$$U(R) = \frac{1}{2\pi}\int_0^{2\pi} d\theta \int_0^R e^{-\frac{r^2}{2}} r dr$$

$$U(R) = \int_0^{\frac{R^2}{2}} e^{-u} du = 1 - e^{-\frac{R^2}{2}}$$

Đặt $p = U(R)$, ta có : $R = \sqrt{-2\ln(1-p)}$; với mỗi p ta sẽ có 1 bán kính R để X,Y là tọa độ của các điểm nằm trong hình tròn bán kính R

Khi này p sẽ có giá trị phân bố đều, đặt , $U_2 = 1 - p$; U_2 cũng có giá trị phân bố đều, từ đây ta có thể xây dựng được 2 dãy phân bố chuẩn độc lập :

$x = \sqrt{-2\ln U_2}\sin 2\pi U_1$

$y = \sqrt{-2\ln U_2}\sin 2\pi U_1$

4.1.2 Sinh nhiễu trắng từ hàm rand() trong C++

Như vậy, với việc dùng hàm rand() trong C++ tạo ra 2 dãy phân phối đều, ta có thể tính được 2 dãy phân phối chuẩn N(0,1), mỗi phần tử của dãy nhân với tham số phương sai rồi trừ đi một khoảng bằng sai số trung bình giữa tổng của chúng với kỳ vọng, ta được dãy số thể hiện nhiễu trắng với kỳ vọng bằng 0 và phương sai theo thiệt lập ban đầu.

4.2 LẬP TRÌNH GIẢI HỆ PHƯƠNG TRÌNH CỦA BÀI TOÁN HỒI QUY TUYẾN TÍNH KNN

Từ hệ (3),(4) của 3.2.2 :

$$\sum_{i=1}^{k}\left(\sum_{j=1}^{n} p_j z_j^i + \rho_0 - f(z^i)\right) z_t^i = 0 \qquad (3)$$

Và

$$\sum_{i=1}^{k}\left(\sum_{j=1}^{n} p_j z_j^i + \rho_0 - f(z^i)\right) = (\qquad (4)$$

Để giải hệ này, ta đưa chúng về dưới dạng phép nhân ma trận.

Đặt P là ma trận vecto 1 x (n+1) : $[p_0 \quad p_1$

Z là ma trận $\begin{bmatrix} 1 \\ z_1^1 \\ \vdots \\ z_n^1 \end{bmatrix}$; coi = 1 1

Y là ma trận

Khi này, (3) và (4) tương đương với :

(Z.P-Y) =

Tương đương với $Z^T.Z.P = Z^T.Y$

Đặt $A=Z^T.Z$; $B = Z^T.Y$ ta có :

A.P=B

Đây chính là hệ phương trình tuyến tính với P là ma trận vecto cần tìm, vì A là ma trận vuông, ta chỉ việc dùng phương pháp Crammer để giải :

$$p_i = \frac{\det(Ai)}{\det(A)}$$

Với Ai là ma trận A với cột thứ i được thay bởi ma trận vecto B.

4.3 GIỚI THIỆU PHẦN MỀM XẤP XỈ NỘI SUY VỚI DỮ LIỆU NHIỄU

4.3.1 Tổng quan phần mềm

Đây là phần mềm xây dựng và huấn luyện mạng nơron RBF nội suy xấp xỉ hàm nhiều biến từ dữ liệu nhiễu. Tôi chọn lập trình bằng ngôn ngữ C++, trên IDE Visual C++ 2010 Release Candidate, Framework .NET . Sản phẩm được dịch ra dưới dạng Windows Form, chạy trên hệ điều hành Windows với điều kiện cài đặt Microsoft .NET Framework version 2.0 Redistributable Package, tên file là dotnetfx.exe, dung lượng 22MB ; có thể tải miễn phí ở địa chỉ:

http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en

4.3.2 Tổ chức dữ liệu

Các mốc nội suy x^k được thể hiện dưới dạng các mảng số thực. Các giá trị y^k, vì trong khóa luận này chỉ xét trường hợp đầu ra 1 chiều, nên được cho dưới dạng 1 số thực.

Tôi lập trình theo cách hướng đối tượng, các đối tượng quan trọng được viết thành từng lớp đặt trong các file header để dễ dàng chỉnh sửa hoặc trao đổi với những người quan tâm, gồm :

- Class mangnoron (mô phỏng mạng nơron RBF)
- Class bosinhphanphoichuan (mô phỏng máy sinh phân phối chuẩn Gauss)
- Class hambk (mô phỏng hàm bán kính, các class này được dùng trong class mangnoron)
- Class matran (mô phỏng ma trận, dùng cho việc tính định thức)
- Class maytinh (mô phỏng hàm số từ 1 xâu nhập vào)

Phương pháp kNN-HDH và các thuật toán cấu thành nên nó là HDH-1 và kNN đều được viết dưới dạng phương thức của class mangnoron.

Để giảm bớt yêu cầu bộ nhớ của chương trình, 1 số bước có tính đệ quy hay phải khai báo biến nhiều lần được đơn giản hóa, ví dụ như việc tính chuẩn Mahalanobis tại thuật toán HDH-1. Thay vì khởi tạo ma trận A

$$\begin{bmatrix} \mathbf{h_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h_2} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdot & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h_n} \end{bmatrix}$$

rồi tính $\|x\| = x^T A x$

ta chỉ việc tính $\|x\| - \sum_{i=1}^{n} \left(\frac{x_i}{h_i}\right)^2$.

4.3.3 Giao diện và chức năng

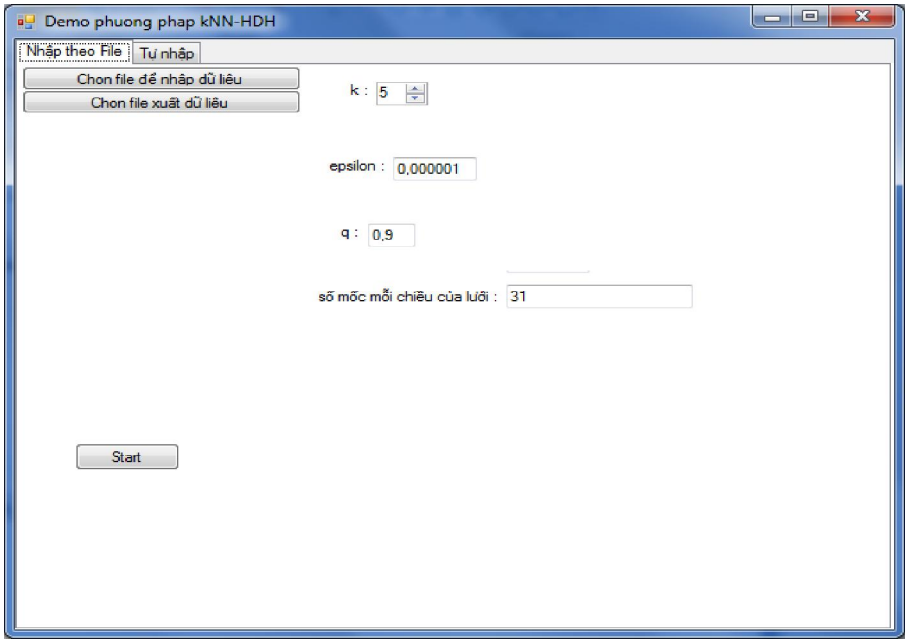
Mặc dù là bản Demo, phần mềm này được thiết kế để tiện cho cả việc nghiên cứu lẫn ứng dụng thực tế. Phần mềm có chức năng chính

- Nhập dữ liệu (có nhiễu trắng) theo 2 cách
  - Thủ công
  - Nhập từ file input
- Xuất các dữ liệu mô tả mạng nơron RBF đã huấn luyện ra file output
- Đưa ra sai số huấn luyện trên giao diện

Giao diện của chương trình gồm 2 Tab : Tab 'Nhập theo file' và Tab 'Tự nhập'; mỗi Tab thể hiện một cách nhập dữ liệu. Người dùng tùy theo việc muốn nhập dữ liệu theo kiểu nào mà chọn 1 trong 2 Tab. Sau đây tôi xin được giới thiệu giao diện và chức năng của phần mềm theo 2 Tab này.

4.3.3.1 Tab "Nhập dữ liệu theo file"

Để nhập dữ liệu theo file, ta chọn Tab 1 'Nhập theo file', và có giao diện dưới đây



Hình 16 Giao diện nhập dữ liệu theo file

Giao diện này đơn giản, ngoài các TextBox,Combo Box để nhập các tham số huấn luyện của thuật toán kNN-HDH như hình trên, phần nhập dữ liệu gồm có 3 button, 2 button để chọn file input, output như đã ghi trên nhãn, 1 button "Start" để bắt đầu việc huấn luyện.

File input là 1 file txt gồm các số thực được sắp xếp theo quy ước :

- Dòng đầu tiên là n – số chiều của các mốc nội suy
- Dòng thứ hai là m – số các mốc nội suy
- Dòng thứ i+2, ($i = \overline{1,m}$) có (n+1) số thực tương ứng với các số $x_1^i, x_2^i \dots x_n^i, y^i$ (tại dòng thứ i+2) để thể hiện 1 mốc nội suy n chiều và giá trị đo được tại mốc đó

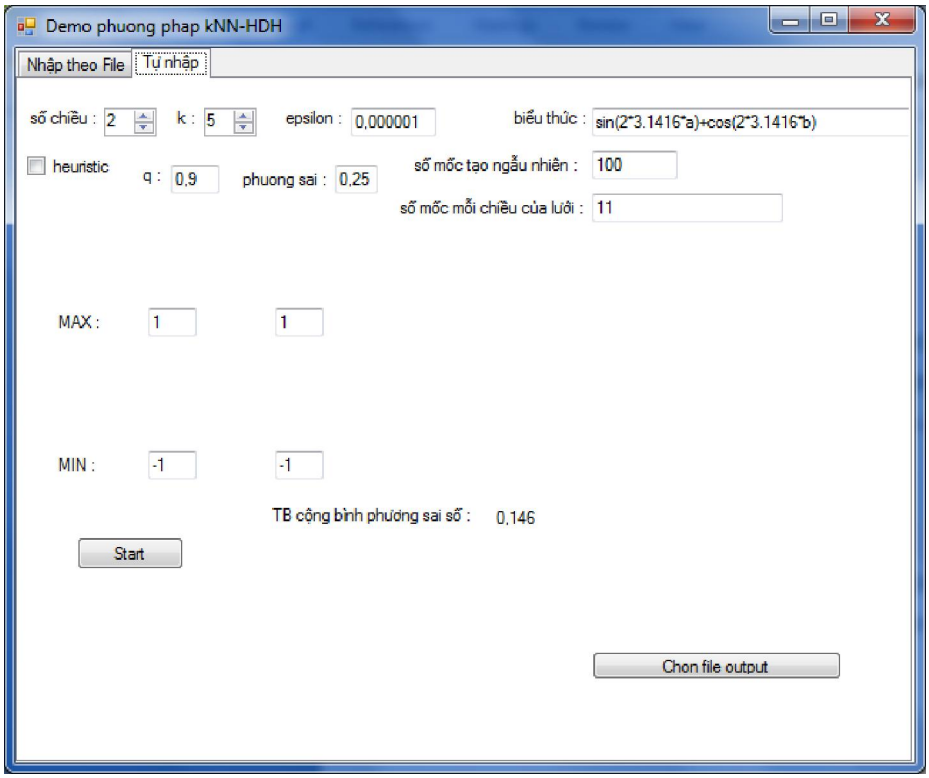
File output cũng là 1 file txt bao gồm các dữ liệu mô tả mạng RBF sau khi huấn luyện, được sắp xếp như sau:

- Dòng thứ nhất là w_0
- Các dòng tiếp theo, cứ 3 dòng một được dùng để mô tả 1 hàm bán kính. Cụ thể là với $k = \overline{0,m-1}$
  - Dòng 3*k+2 gồm n số thực tương ứng với các số $x_1^k, x_2^k \dots x_n^k$ với x^k là tâm của hàm bán kính thứ k
  - Dòng 3*k+3 là tham số độ rộng của hàm bán kính thứ k
  - Dòng 3*k+4 là hệ số w_k

Mỗi khi nhấn button 'Start', phần mềm sẽ lấy dữ liệu từ file input làm bộ dữ liệu huấn luyện rồi huấn luyện mạng nơron RBF theo bộ dữ liệu này, sau đó truyền các dữ liệu số mô tả mạng RBF ra file output.

4.3.3.2 Tab "Tự nhập"

Để nhập dữ liệu theo cách thủ công, ta chọn Tab 'Tự nhập', giao diện như dưới đây



Hình 17 Giao diện nhập dữ liệu thủ công

Vì số lượng các mốc nội suy lớn, cho nên ở Tab này thay vì nhập từng mốc, người dùng sẽ chọn miền giá trị cho các mốc nội suy. (Nếu muốn nhập chi tiết các mốc nội suy, người dùng có thể chọn cách nhập theo file sẽ được trình bày như trên).

Cụ thể là người dùng sẽ chọn số chiều n. Vơi mỗi n được chọn thì các label "chiều 1", "chiều 2" … hiện dần ra khi n tăng và ẩn bớt khi n giảm. Cùng với đó là các textbox để người dùng nhập giá trị max và min của từng chiều cũng hiện và ẩn ra theo, người dùng sẽ có thể tạo miền giá trị cho các mốc nội suy bằng cách này. Chương trình sẽ tạo ra các mốc nội suy ngẫu nhiên nằm trong miền đó. Số mốc tạo ngẫu nhiên mặc định là 100, người dùng có thể tự nhập vào tại TextBox "số mốc ngẫu nhiên".

Sau khi có các mốc nội suy rồi, giá trị đo được tại các mốc nội suy sẽ bằng giá trị hàm số cần nội suy xấp xỉ (nhập ở TextBox "biểu thức") cộng với 1 sai số được sinh từ dãy phân phối chuẩn (có kỳ vọng mặc định =0 vì là nhiễu trắng) và phương sai được điền ở TextBox "phương sai" (mặc định là 0,25).

Sau khi xây dựng xong bộ dữ liệu huấn luyện, phần mềm sẽ huấn luyện mạng RBF theo thuật toán kNN-HDH với các tham số đã được người dùng điền vào giao diện như trên.

Button "Chọn file output" được dùng nếu người dùng muốn xuất dữ liệu mô tả mạng RBF sau huấn luyện ra file. Thứ tự dữ liệu xuất ra file giống như mô tả đã nêu ở 4.3.2.1

Vì giao diện này được làm với mục đích giúp người dùng dễ dàng kiểm chứng các kết quả thực nghiệm, nên sau khi huấn luyện mạng RBF xong, sai số trung bình tại các mốc huấn luyện sẽ được lấy trung bình cộng của tổng bình phương và kết quả được đưa ra TextBox "TB cộng bình phương sai số" như trên. Ngoài ra, checkBox Heuristic cũng sẽ được người dùng tích vào nếu muốn áp dụng heuristic "ăn gian" khi thí nghiệm.

CHƯƠNG 5:
KẾT QUẢ THÍ NGHIỆM

Nội dung chương này bao gồm:

- Thí nghiệm thay đổi kích thước lưới
- Thí nghiệm về việc chọn k
- Thí nghiệm khi tăng số chiều
- So sánh hiệu quả với thuật toán khác

Để làm nổi bật các đặc điểm của phương pháp này, tôi sẽ thiết lập một module để thực hiện 1 heuristic, tạm gọi là "ăn gian" để giả thiết rằng phương pháp kNN là hoàn hảo, vừa hồi quy vừa khử nhiễu với sai số bằng 0. Để mô phỏng heuristic này, lệnh "ăn gian" sẽ được viết trong phần mềm, và khi kích hoạt, thì phần mềm, khi tính giá trị nội suy y_i tại mỗi nút lưới x_i ,thay vì dùng phương pháp kNN để tính ra hàm rồi gán y_i :, thì ta sẽ dùng ngay hàm số cần nội suy xấp xỉ để gán giá trị y_i

5.1 THÍ NGHIỆM VỀ VIỆC THAY ĐỔI KÍCH THƯỚC LƯỚI

Vì mạng RBF sẽ được huấn luyện không phải trên dữ liệu ngẫu nhiên ban đầu mà là trên lưới dữ liệu cách đều được thiết lập sau khi hồi quy từ dữ liệu ban đầu, cho nên mặc dù thuật toán HDH-1 có tốc độ tính toán nhanh nhưng vẫn tồn tại nghi ngờ rằng sai số huấn luyện có thể lớn, dựa vào tính chất của thuật toán HDH-1 pha trình bày ở cuối chương 2 rằng lưới dữ liệu càng dày thì xấp xỉ càng tốt dẫn đến quan ngại rằng trong phương pháp này ta phải thiết lập lưới dữ liệu mới rất dày đặc mới có thể cho sai số chấp nhận được. Thí nghiệm dưới đây cho ra kết quả khá bất ngờ về kích thước hợp lý của lưới dữ liệu.

Hàm số được dùng làm thí nghiệm ở đây là hàm

$$y_1 = \sin(2\pi x_1) + \cos(2\pi x_2)$$

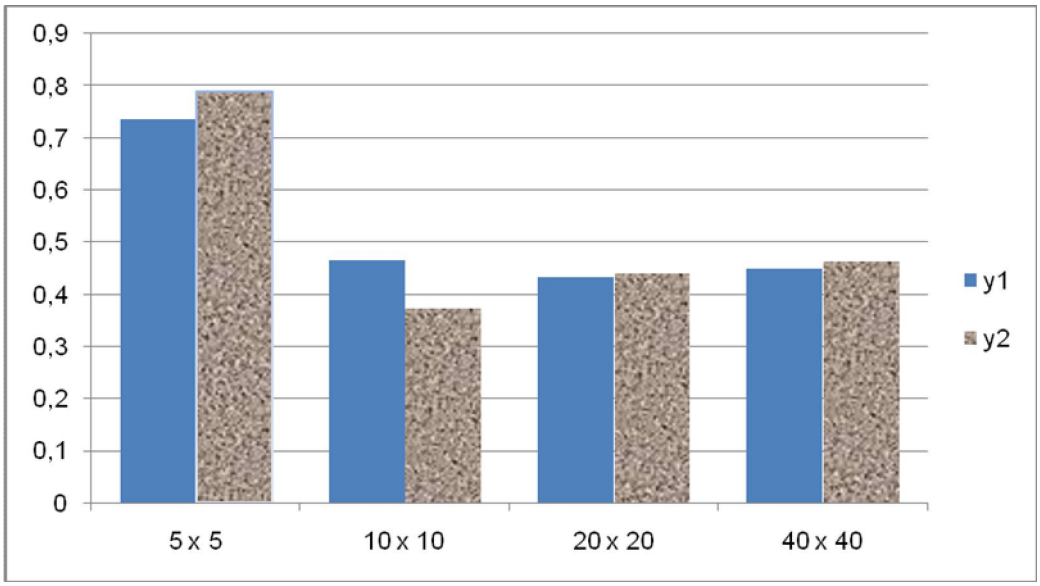
$$y_2 = 3\sin(\pi x_1 x_2) + 2\sin(\pi x_1 + x_2)$$

Các hàm này được lấy từ thí nghiệm của [10] để tiện so sánh tại phần sau.

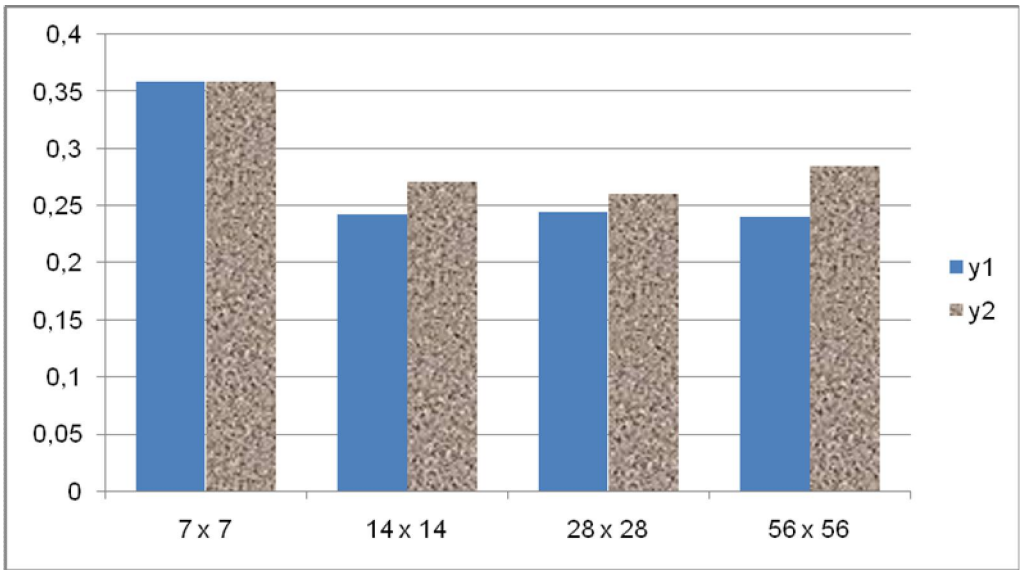
Dữ liệu ban đầu gồm có các mốc nội suy phân bố ngẫu nhiên trên miền giá trị đầu vào D : $[-1; 1] \times$; ta sẽ thử ở cả 2 trường hợp số mốc nội suy m=100 và m=200

Kết quả đo tại các mốc này bằng giá trị hàm số thực cộng với sai số (nhiễu trắng). Dãy các sai số được phân bố theo phân phối chuẩn có phương sai là 0.25.

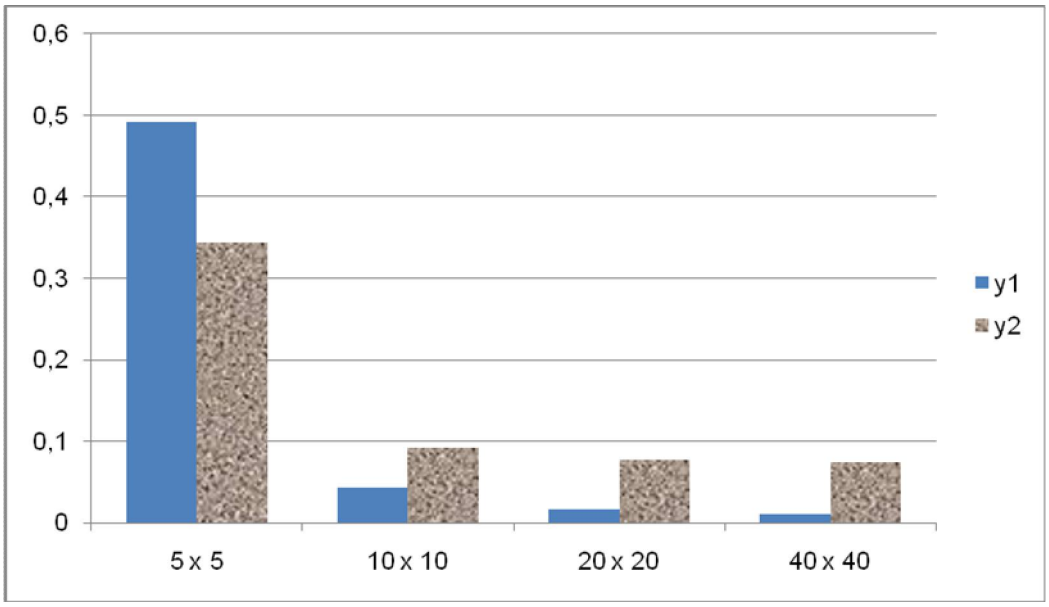
Bảng dưới đây so sánh sai số của phương pháp này khi khởi tạo lưới dữ liệu ở các kích cỡ khác nhau, và ở 2 trường hợp dùng và không dùng heuristic. Kích thước lưới dữ liệu mỗi lần tăng được tăng gấp đôi mỗi chiều, số mốc cách đều của lưới tạo sau nhiều gấp 4 lần số mốc cách đều của lưới tạo ngay trước nó.



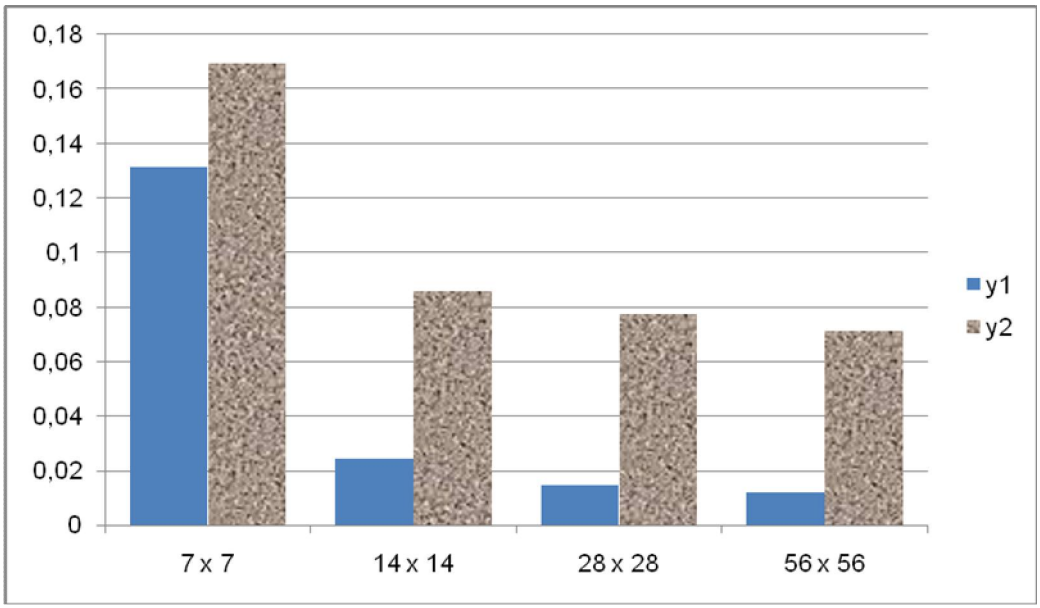
Hình 18 Sai số khi chọn các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu 100 mốc ngẫu nhiên, không áp dụng heuristic "ăn gian"



Hình 19 Sai số khi chọn các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu 200 mốc ngẫu nhiên, không áp dụng heuristic "ăn gian"



Hình 20 Sai số khi áp dụng các kích cỡ khác nhau của lưới dữ liệu cho bộ dữ liệu ngẫu nhiên 100 mốc, có heuristic "ăn gian"



Hình 21 Sai số khi chọn các kích cỡ khác của lưới dữ liệu cho bộ dữ liệu 200 mốc ngẫu nhiên, có áp dụng heuristic "ăn gian"

Ta thấy rằng dù mỗi lần thiết lập số nút lưới tăng gấp 4 lần so với lần thiết lập trước, nhưng sai số tổng quát không giảm nhiều. Thử với một số hàm số khác, ta đều thấy hiện tượng rằng mật độ lưới dữ liệu quá thưa thì sẽ cho sai số lớn, tuy nhiên khi cho mật độ đó dày đặc lên thì chỉ hiệu quả lớn ở 1 khoảng nhất định, lưới dữ liệu dày đặc đến một mức nào đó, thì khi tiếp tục làm dày đặc hơn thì sai số không giảm đi bao nhiêu so với sự gia tăng số mốc cách đều cần huấn luyện.

Đặc biệt, sai số khi đặt mật độ nút lưới ở mức thứ 2 (10 x 10 với m=100) hay (14 x 14 với m=200) tốt hơn nhiều khi đặt ở mức thứ nhất, và không tồi hơn bao nhiêu với các mức kế tiếp, mặc dù độ dày đặc tăng đều lên 4 lần mỗi mức. Chú ý rằng số nút lưới của lưới này xấp xỉ với m mốc nội suy của bộ dữ liệu ban đầu.

Nhận xét :

Thí nghiệm này đã cho thấy lưới dữ liệu mới cần khởi tạo không phải quá dày đặc như đã lo ngại ban đầu. Thực nghiệm đã cho thấy chỉ cần số nút lưới dữ liệu xấp xỉ số mốc nội suy ban đầu đã có thể cho hiệu quả huấn luyện nói chung là hợp lý. Tùy thuộc vào ứng dụng cụ thể mà có thể tùy chỉnh kích thước lưới dữ liệu, ví dụ như tăng kích thước để làm giảm đi sai số, trong khi thời gian huấn luyện vẫn nằm trong khoảng cho phép, như đặc điểm huấn luyện rất nhanh của phương pháp này.

Ta thấy rõ sự khác nhau khi làm thí nghiệm giữa việc áp dụng và không áp dụng heuristic, điều này cho thấy tầm ảnh hưởng lớn của bước hồi quy kNN.

5.2 THÍ NGHIỆM VỀ VIỆC CHỌN K

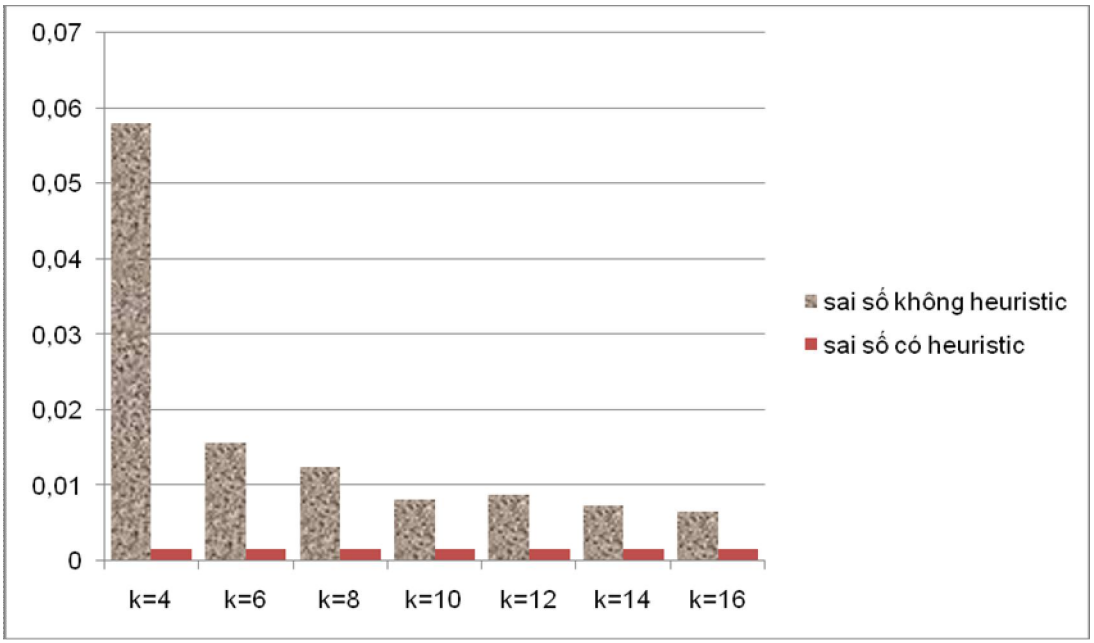
Trong phương pháp này, việc chọn k khi hồi quy tuyến tính kNN thế nào cho tốt được coi là rất quan trọng. Vì phương pháp kNN không chỉ hồi quy mà còn làm nhiệm vụ khử nhiễu. Nếu hồi quy không tốt sẽ tạo ra lưới dữ liệu mà giá trị tại mỗi nút lưới khác xa so với giá trị thực được khử nhiễu và từ đó làm cho việc nội suy xấp xỉ kém đi nhiều. Việc chọn k trong phương pháp kNN còn là một bài toán mở, người ta mới chỉ đưa ra được khuyến nghị là nên chọn k lớn hơn số chiều n.

Thí nghiệm dưới đây đưa ra một vài kết luận thú vị về việc chọn k. Tại thí nghiệm này, ta xét 2 hàm số

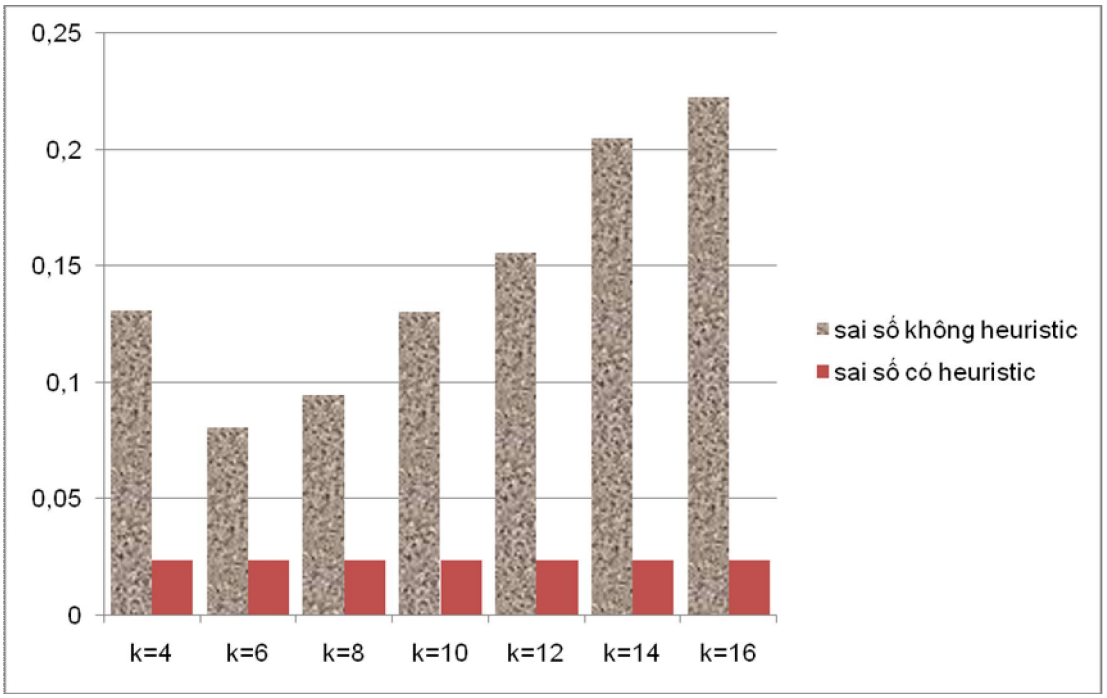
$$y_1 = \frac{\sin(2\pi x_1) + \cos(2\pi x_2)}{4}$$

$$y_2 = \sin(2\pi x_1) + \cos(2\pi x_2)$$

Dễ thấy 2 hàm này tỷ lệ với nhau, vì nội dung thí nghiệm này là xét cách chọn k dựa vào độ lớn miền giá trị của hàm. Tham số m được chọn là 200, theo như kết quả thí nghiệm ở **5.1,** ta chọn lưới dữ liệu có kích thước 14 x 14, xét lần lượt các trường hợp k=4,6,8,10,12,14,16. Kết quả mỗi lần thử nghiệm được so sánh kèm với kết quả khi kích hoạt heuristic "ăn gian"



Hình 22 Bảng so sánh sai số của phương pháp kNN-HDH khi áp dụng cho hàm y_1 với các cách chọn k khác nhau



Hình 23 Bảng so sánh sai số của phương pháp kNN-HDH khi áp dụng cho hàm y_2 với các cách chọn k khác nhau

Tại đây, ta thầy hàm y2 đạt giá trị tốt nhất với k=6 và khi tăng dần k thì sai số tồi hơn nhiều. Trong khi hàm y1 thì càng tăng k càng tốt.

Điều này có thể giải thích như sau : Sai số chênh lệch thi thử với những k khác nhau là do với những k khác nhau thì hiệu quả của phương pháp kNN khác nhau. Ta có thể thấy khi dùng heuristic thì sai số không đổi.

Đặc điểm của phương pháp hồi quy kNN là khi k càng lớn thì hiệu quả khử nhiễu càng tốt, nhưng đồng thời, vì phải hồi quy các nút ở quá xa cho nên hiệu quả hồi quy cũng bị kém đi. Ngược lại khi k càng nhỏ thì hiệu quả hồi quy càng tốt (trừ trường hợp k nhỏ quá không đủ để hồi quy thì sai số sẽ tăng vọt như đã thấy trong biểu đồ) , tuy nhiên khi giảm k thì khả năng khử nhiễu trắng lại kém đi vì phải nội suy với ít mốc sẽ không trung hòa được nhiễu trắng.

Nhận xét :

Giá trị k tốt nhất sẽ phải cân bằng giữa hiệu quả của hồi quy và hiệu quả của khử nhiễu, tùy theo các bài toán cụ thể để chọn k lớn hay k nhỏ. Xét ví dụ trên, ta thấy vì miền giá trị của y_1 nhỏ hơn của y_2 cho nên nhiễu có ảnh hưởng lớn hơn đến kết quả

hồi quy, vì thế công việc khử nhiễu phải được đặt ưu tiên hơn so với bài toán với hàm y_2, do đó ở đây k lớn hơn thì tốt. Biểu đồ trên cho thấy k càng lớn càng tốt nhưng khi xét đến k=30, vì đặt quá nặng nhiệm vụ khử nhiễu so với hồi quy cho nên sai số sẽ lại tăng lên 0,03

Ngược lại, bài toán với hàm y_2 vì tỷ lệ miền giá trị của nó với nhiễu trắng lớn hơn so với bài toán hàm y_1 cho nên ưu tiên khử nhiễu không được đặt nặng như bài toán với hàm y_1, vì thế chọn k nhỏ hơn so với trường hợp với hàm y_1 sẽ cho hiệu quả tốt nhất.

5.3 THÍ NGHIỆM KHI TĂNG SỐ CHIỀU

Một phần rất quan trọng của phương pháp này là hồi quy kNN, nó vừa có vai trò hồi quy ra các mốc để thuật toán lặp 1 pha HDH dựa vào để huấn luyện, vừa có vai trò khử nhiễu, nếu hồi quy kNN càng tốt thì sai số sẽ giảm đi rất nhiều, mức độ tốt nhất mà ta có thể mong đợi là nó hồi quy chính xác hàm số cần nội suy xấp xỉ tại những nút lưới. (Tức là khi heuristic thành hiện thực)

Xét về số chiều của các vecto đầu vào, ta thấy : Do tính chất của phương pháp kNN. Việc chọn k thường được khuyến nghị là nên chọn k>n, sau khi thỏa mãn điều kiện k>n, thì "k hợp lý" nên đủ thấp để hồi quy cho sai số nhỏ. Vì vậy, khi số chiều n tăng, thì "k hợp lý" cũng sẽ tăng theo. Mặt khác, khi k càng tăng thì việc khử nhiễu lại càng tốt. Vì vậy, với cùng 1 dải nhiễu, nếu số chiều càng tăng thì việc khử nhiễu càng tốt, qua đó nâng cao tính hiệu quả của phương pháp kNN-HDH

Trong quá trình thí nghiệm dưới đây, khi thử với các hàm số càng nhiều biến, thì sai số sau khi đã chọn k thích hợp, lại càng gần với sai số khi áp dụng heuristic "ăn gian", cụ thể kết quả như sau :

Ta thử nghiệm với 5 hàm số

- $u_1 = s$
- $u_2 = \sin(2\pi x_1) + co$
- $u_3 = 3\sin(\pi x_1 x_2) + 2\sin(\pi$
- $u_4 = \sin(2\pi x_1) + \cos(2\pi x_2) + \cos(x_1 +$
- $u_5 = \sin(2\pi x_1 + x_3 + x_4) + \cos(\pi x_2 x_3 + x_4) + \cos(x_1 + x_2 + x_3) + \sin(x_1 + x_2 + x_3 + x_4)$

Kết quả thực nghiệm như sau :

Hình 24 : Bảng so sánh sai số của phương pháp kNN-HDH khi dùng và không dùng Heuristic, với số chiều tăng dần

u_1 (1 chiều)		u_2 (2 chiều)		u_3 (2 chiều)		u_4 (3 chiều)		u_5 (4 chiều)	
K=16	Heuristic	K=6	Heuristic	K=5	Heuristic	K=6	Heuristic	K=8	Heuristic
0.0065	0.0001	0.0807	0.0232	0.1302	0.0752	0.4576	0.3333	0.8162	0.6312

Tại đây ta thấy với số chiều tăng dần, sai số khi không dùng heuristic càng lúc tiến lại lại gần sai số khi dùng heuristic, tức là việc hồi quy kNN càng hiệu quả khi ta tăng số chiều của các mốc nội suy, đúng như nhận định ở trên.

.

5.4 SO SÁNH HIỆU QUẢ VỚI PHƯƠNG PHÁP KHÁC

Chương này tôi xin được so sánh hiệu quả của phương pháp kNN-HDH với một phương pháp nội suy xấp xỉ hàm nhiều biến với dữ liệu nhiễu của tác giả Tomohiro Ando đã được công bố trên tạp chí Journal of Statistical Planning and Inference năm 2008. [10]

Tại bài báo này, tác giả dùng 3 phương pháp là GIC, NIC, MIC để thử với các hàm số u_2, u_3 như ở trên, miền giá trị cũng là $D = [-1; 1]$ x. Ta sẽ so sánh kết quả của phương pháp kNN-HDH với phương pháp tốt nhất của tác giả là phương pháp GIC, trong cả 2 trường hợp số mốc nội suy ban đầu là m=100 và m=200.

Vì thời gian có hạn, cộng với việc thuật toán GIC cài đặt rất phức tạp nên trong khóa luận này tôi chưa cài đặt thuật toán GIC mà chỉ lấy bộ dữ liệu và kết quả của tác giả để so sánh với phương pháp kNN-HDH

Hình 25: Bảng so sánh kết quả với phương pháp GIC

m=100	u_2	u_3
GIC	0.38873	1.18925
kNN-HDH	0.1175	0.4599
m=200	u_2	u_3
GIC	0.14857	0.34603
kNN-HDH	0.0892	0.2846

Nhận xét :

Với kết quả này, ta thấy phương pháp kNN-HDH cho sai số tốt hơn phương pháp GIC, mặc dù tác giả Tomohiro Ando mới chỉ thử với các hàm 2 biến, chưa thử với các hàm nhiều biến hơn vốn là ưu điểm của phương pháp kNN-HDH.

CHƯƠNG 6:
TỔNG KẾT VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

Nội dung chương này bao gồm:

- Tổng kết
- Phương hướng phát triển của đề tài

6.1 Tổng kết

Đến đây tôi đã hoàn thành khóa luận tốt nghiệp với đề tài ***"Huấn luyện mạng nơron RBF với mốc cách đều và ứng dụng"*** với mục đích mô phỏng phương pháp ứng dụng thuật toán HDH-1 vào việc xây dựng hệ thống nội suy xấp xỉ hàm nhiều biến với dữ liệu nhiễu và nghiên cứu thực nghiệm nhằm tìm ra các đặc điểm, lý giải và đưa ra các cách hoàn thiện phương pháp này.

Những công việc đã làm được :

- Tìm hiểu kiến trúc mạng RBF, đặc điểm mạng RBF, từ đó hiểu được các phương pháp huấn luyện mạng RBF, ở đây là thuật toán HDH-2 và HDH-1.
- Tìm hiểu về nhiễu trắng và phương pháp sinh nhiễu trắng
- Tìm hiểu về phương pháp hồi quy tuyến tính kNN
- Hiểu được ý tưởng về phương pháp kNN-HDH và lập trình mô phỏng thành công, phần mềm tiện cho nghiên cứu lẫn ứng dụng
- Phát hiện ra được các đặc điểm cơ bản, quan trọng của phương pháp kNN-HDH, đưa ra các cải tiến lớn, cho thấy hiệu quả cao của phương pháp này. Đó là :
    - Lưới dữ liệu chỉ cần có số nút xấp xỉ với số mốc nội suy của bộ dữ liệu ban đầu
    - K được chọn với mục đích cân bằng giữa tính khử nhiễu và tính hồi quy, cụ thể là dựa trên tỷ lệ giữa miền giá trị của hàm số và nhiễu trắng.
    - Khi thực nghiệm nên áp dụng phương pháp heuristic "ăn gian", nhằm tách riêng 2 bước hồi quy tuyến tính kNN và thuật toán HDH-1. Như thế sẽ làm nổi bật đặc điểm của phương pháp này
    - Khi số chiều càng lớn, hiệu quả càng tốt vì khử nhiễu tốt hơn trong khi vẫn tối ưu sai số hồi quy.
    - Các kết quả thí nghiệm đều được giải thích khớp với nền tảng lý thuyết.

6.2 Phương hướng phát triển của đề tài

Do đây là phương pháp hoàn toàn mới, trong khi thời gian nghiên cứu lại có hạn nên còn một số điều tôi chưa đi sâu hơn được. Đây cũng là những điều có thể dùng để làm phương hương phát triển cho đề tài. Đó là :

- Cần thêm nhiều mô phỏng các phương pháp khác để so sánh với phương pháp kNN-HDH để chứng minh ưu điểm của nó, đặc biệt là tốc độ huấn luyện và trường hợp nhiều chiều.
- Áp dụng phương pháp kNN-HDH vào ứng dụng cụ thể như : Xử lý ảnh, nhận dạng giọng nói ….

TÀI LIỆU THAM KHẢO

[1] Hoang Xuan Huan, Dang Thi Thu Hien and Huu Tue Huynh, A Novel Efficient Algorithm for Training Interpolation Radial Basis Function Networks, Signal Processing 87 ,2708 – 2717, 2007.

[2] Hoang Xuan Huan, Dang Thi Thu Hien and Huynh Huu Tue, An efficient algorithm for training interpolation RBF networks with equally spaced nodes, submitted to IEEE Transactions on Neural Networks

[3] T.M. Mitchell, Machine learning, McGraw-Hill, 1997

[4] J. Shlens, A Tutorial on Principal Component Analysis, April 22, 2009

[5] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. Complex Systems, vol. 2, 321-355, 1988.

[6] Đặng Thị Thu Hiền, Luận án tiến sỹ công nghệ thông tin, chuyên ngành Khoa học máy tính, mã số : 62.48.0101, Đại học Công nghệ, ĐHQG Hà Nội, 2009

[7]William M.K. Trochim, Measurement Error
http://www.socialresearchmethods.net/kb/measerr.php

[8]Wikipedia®, Normal distribution
http://en.wikipedia.org/wiki/Normal_distribution

[9]G.E.P Box and Mervin E. Muller, A Note on the Generation of Random Normal Deviates, Ann. Math. Statist. Volume 29, Number 2 (1958), 610-611.

[10]Tomohiro Ando, Sadanori Konishi and Seiya Imoto, Nonlinear regression modeling via regularized radial basis function network, Journal of Statical Planning and Inference, 2008, trang 16-18