**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Đào Tuấn Anh**

**NHẬN DẠNG KHUÔN MẶT SỬ DỤNG MẠNG NƠ RON NHÂN TẠO**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**HÀ NỘI - 2017**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Đào Tuấn Anh**

**NHẬN DẠNG KHUÔN MẶT SỬ DỤNG MẠNG NƠ RON NHÂN TẠO**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: TS Lê Quang Hiếu**

**Cán bộ đồng hướng dẫn: Anh Trịnh Đức Bình**

**HÀ NỘI - 2017**

**LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn tới các thầy cô và các cán bộ của trường đại học Công Nghệ - Đại học Quốc gia Hà Nội đã tận tình dạy bảo và luôn tạo điều kiện thuận lợi cho em được học tập và nghiên cứu trong suốt 4 năm đại học.

Em xin gửi lời cảm ơn sâu sắc đến thầy giáo TS Lê Quang Hiếu người đã tận tình hướng dẫn và đóng góp ý kiến quý báu trong suốt quá trình thực hiện khóa luận tốt nghiệp. Đồng thời, em xin cảm ơn anh Trịnh Đức Bình, nhân viên Trung tâm Nghiên cứu và Phát triển di động của Samsung Việt Nam, đã cùng đồng hành và giúp đỡ em trong suốt khóa luận này.

Cuối cùng, em muốn gửi lời cảm ơn tới gia đình, bạn bè, tập thể lớp K58CD đặc biệt là bố mẹ những người đã không quản khó khăn, vất vả nuôi em ăn học để trở thành người như ngày hôm nay.

Em xin chân thành cảm ơn.

**NHẬN DẠNG KHUÔN MẶT SỬ DỤNG MẠNG NƠ RON NHÂN TẠO**

**Đào Tuấn Anh**

*QH-2013/I-CQ - Công nghệ thông tin*

**Tóm tắt Khóa luận tốt nghiệp:** Hiện nay, với sự phát triển nhanh chóng của các thiết bị công nghệ, lần lượt máy ảnh kĩ thuật số, máy quay phim, điện thoại thông minh ra đời. Do vậy, lượng dữ liệu chúng ta nhận được bằng hình ảnh là rất lớn. Từ thực tế đó, nhu cầu về xử lý ảnh cũng trở thành một nhu cầu thiết yếu của xã hội. Không chỉ dừng lại ở việc xử lý những tấm ảnh bị nhòe hay phục chế ảnh cũ, người ta đã sử dụng xử lý ảnh để giải quyết được những bài toán lớn hơn và có tính ứng dụng cao hơn trong thực tế đời sống. Hệ thống nhận diện khuôn mặt là một trong số đó. Nhưng để đem lại một hiệu suất cao hơn so với xử lý ảnh thông thường, người ta đã kết hợp trí tuệ nhân tạo vào trong việc nhận diện và cụ thể ở đây chính là sử dụng nơ ron nhân tạo, một thứ mà con người đã tìm ra từ 60 năm trước nhưng đang dần trở lại một cách mạnh mẽ, một thứ công cụ chuyên về dự đoán và nhận dạng. Do vậy, trong đề tài khóa luận này em muốn giới thiệu về mạng nơ ron và thực nghiệm sử dụng mạng nơ ron. Nội dung của khóa luận sẽ tập trung trình bày những đặc điểm cơ bản của mạng nơ ron, các mô hình, cấu trúc, luật học của mạng nơ ron, các bước tiến hành xây dựng hệ thống nhận diện khuôn mặt sử dụng mạng nơ ron và phần thực nghiệm.

***Từ khóa:*** *Mạng nơ ron nhân tạo, Hệ thống nhận diện khuôn mặt*

**LỜI CAM ĐOAN**

Tôi xin cam đoan rằng mọi kết quả trình bày trong khóa luận đều do tôi thực hiện dưới sự hướng dẫn của TS. Lê Quang Hiếu.

Tất cả các tham khảo nghiên cứu liên quan đều nêu rõ nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong khóa luận. Khóa luận không sao chép tài liệu, công trình nghiên cứu từ người khác mà không chỉ rõ về mặt tài liệu tham khảo.

Các thống kê, các kết quả trình bày khóa luận đều lấy từ thực nghiệm khi chạy chương trình. Nếu tôi sai tôi hoàn toàn chịu trách nhiệm theo quy định của trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội.

Hà Nội, tháng 4 năm 2017

Đào Tuấn Anh

**MỤC LỤC**

[Chương 1. Mở đầu 1](#_Toc481906971)

[Chương 2. Tổng quan về mạng nơ ron nhân tạo 3](#_Toc481906972)

[2.1. Mạng nơ ron sinh học 3](#_Toc481906973)

[2.2. Giới thiệu về mạng nơ ron nhân tạo 5](#_Toc481906974)

[2.2.1. Định nghĩa về nơ ron nhân tạo 5](#_Toc481906975)

[2.2.2. Mô hình nơ ron nhân tạo 5](#_Toc481906976)

[2.2.3. Định nghĩa mạng nơ ron nhân tạo 9](#_Toc481906977)

[2.2.4. Các đặc trưng của mạng nơ ron nhân tạo 10](#_Toc481906978)

[2.2.5. Chức năng của mạng nơ ron nhân tạo 11](#_Toc481906979)

[2.2.6. Các kiến trúc của mạng nơ ron nhân tạo 11](#_Toc481906980)

[2.2.7. Các phương pháp của mạng nơ ron nhân tạo 14](#_Toc481906981)

[2.3. Giới thiệu mạng nơ ron đa lớp truyền thẳng 16](#_Toc481906982)

[2.3.1. Tổng quan về mạng nơ ron đa lớp truyền thẳng 16](#_Toc481906983)

[2.3.2. Kiến trúc mạng 16](#_Toc481906984)

[2.3.3. Giải thuật lan truyền ngược (Backpropagation Algorithm) 17](#_Toc481906985)

[2.3.4. Các kỹ thuật cải tiến thuật toán lan truyền ngược 20](#_Toc481906986)

[2.3.5. Overfitting trong mạng nơ ron nhân tạo 26](#_Toc481906987)

[Chương 3. Phương pháp phân tích thành phần chính 28](#_Toc481906988)

[3.1. Giới thiệu 28](#_Toc481906989)

[3.2. Cơ sở toán học 28](#_Toc481906990)

[3.2.1. Thống kê 28](#_Toc481906991)

[3.2.2. Ma trận 30](#_Toc481906992)

[3.3. Các bước thực hiện phương pháp 32](#_Toc481906993)

[3.3.1. Tạo tập dữ liệu 32](#_Toc481906994)

[3.3.2. Trừ cho giá trị trung bình 33](#_Toc481906995)

[3.3.3. Tính toán ma trận hiệp phương sai 33](#_Toc481906996)

[3.3.4. Tính toán véc tơ riêng và giá trị riêng của ma trận hiệp phương sai 33](#_Toc481906997)

[3.3.5. Chọn các thành phần và tạo ra một véc tơ đặc trưng 34](#_Toc481906998)

[3.3.6. Lấy ra bộ dữ liệu mới 35](#_Toc481906999)

[3.4. Lấy lại giá trị cũ 35](#_Toc481907000)

[Chương 4. Ứng dụng mạng nơ ron để xây dựng hệ thống nhận diện khuôn mặt 36](#_Toc481907001)

[4.1. Phát biểu bài toán 36](#_Toc481907002)

[4.2. Giải quyết bài toán 36](#_Toc481907003)

[4.2.1. Tập dữ liệu sử dụng 36](#_Toc481907004)

[4.2.2. Sơ đồ luồng bước huấn luyện 38](#_Toc481907005)

[4.2.3. Sơ đồ luồng bước nhận diện 43](#_Toc481907006)

[4.3. Mô tả về hệ thống 44](#_Toc481907007)

[4.3.1. Giao diện chương trình 45](#_Toc481907008)

[4.3.2. Phần nhận diện 45](#_Toc481907009)

[4.3.3. Phần huấn luyện 48](#_Toc481907010)

[4.4. Kết quả thực nghiệm 48](#_Toc481907011)

[Chương 5. Kết luận và hướng phát triển 54](#_Toc481907012)

[5.1. Kết luận 54](#_Toc481907013)

[5.2. Hướng phát triển 54](#_Toc481907014)

[Tài liệu tham khảo 56](#_Toc481907015)

**DANH MỤC CÁC BẢNG**

[Bảng 3.1. Bảng dữ liệu mẫu phương pháp phân tích thành phần chính 32](#_Toc481907044)

[Bảng 3.2. Bảng dữ liệu sau khi trừ đi giá trị trung bình 33](#_Toc481907045)

[Bảng 4.1. Độ chính xác của nhận dạng dựa trên số lượng eigenfaces 49](#_Toc481907046)

[Bảng 4.2. Kết quả độ chính xác 50 lần huấn luyện đầu tiên 49](#_Toc481907047)

[Bảng 4.3. Kết quả độ chính xác từ lần huấn luyện 60 đến 100 49](#_Toc481907048)

[Bảng 4.4. Các lần lặp của phương pháp cross-validation 51](#_Toc481907049)

[Bảng 4.5. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 1 51](#_Toc481907050)

[Bảng 4.6. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 2 52](#_Toc481907051)

[Bảng 4.7. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 3 52](#_Toc481907052)

[Bảng 4.8. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 4 52](#_Toc481907053)

**DANH MỤC CÁC HÌNH VẼ**

[Hình 2.1. Mô hình mạng nơ ron sinh học 3](#_Toc481907056)

[Hình 2.2. Mô hình nơ ron nhân tạo 6](#_Toc481907057)

[Hình 2.3. Hàm đồng nhất 7](#_Toc481907058)

[Hình 2.4. Hàm bước nhị phân với q = 1 8](#_Toc481907059)

[Hình 2.5. Hàm Sigmoid 8](#_Toc481907060)

[Hình 2.6. Hàm sigmoid lưỡng cực 9](#_Toc481907061)

[Hình 2.7. Kiến trúc mạng truyền thẳng 12](#_Toc481907062)

[Hình 2.8. Hình ảnh xe bus 13](#_Toc481907063)

[Hình 2.9. Hình ảnh xe bus đã khoanh vùng 13](#_Toc481907064)

[Hình 2.10. Kiến trúc mạng phản hồi 14](#_Toc481907065)

[Hình 2.11. Sơ đồ mô tả luật học giám sát 15](#_Toc481907066)

[Hình 2.12. Biểu đồ diễn tả overfitting. 27](#_Toc481907067)

[Hình 4.4.1. Hình ảnh khuôn mặt có trong cơ sở dữ liệu mẫu 37](#_Toc481907068)

[Hình 4.2. Sơ đồ luồng bước huấn luyện hệ thống nhận diện khuôn mặt 38](#_Toc481907069)

[Hình 4.3. Sơ đồ khối trích rút đặc trưng 39](#_Toc481907070)

[Hình 4.4. Ảnh trung bình của tập dữ liệu học 40](#_Toc481907071)

[Hình 4.5. Danh sách các eigenfaces 42](#_Toc481907072)

[Hình 4.6. Sơ đồ luồng bước nhận diện hệ thống nhận diện khuôn mặt 43](#_Toc481907073)

[Hình 4.7. Sơ đồ luồng tìm tập ảnh mới tại tập kiểm tra 43](#_Toc481907074)

[Hình 4.8. Giao diện giới thiệu chương trình 45](#_Toc481907075)

[Hình 4.9. Hình ảnh phần nhận diện 46](#_Toc481907076)

[Hình 4.10. Chọn file ảnh để nhận diện 46](#_Toc481907077)

[Hình 4.11. Sau khi đã chọn ảnh để nhận diện 47](#_Toc481907078)

[Hình 4.12. Thông báo hiển thị người được nhận diện 47](#_Toc481907079)

[Hình 4.13. Ảnh người được nhận diện hiện ra 47](#_Toc481907080)

[Hình 4.14. Phần huấn luyện mạng nơ ron 48](#_Toc481907081)

[Hình 4.15. Thông báo hiển thị độ chính xác 48](#_Toc481907082)

[Hình 4.16. Biểu đồ hiện thị kết quả độ chính xác 50](#_Toc481907083)

[Hình 5.1. Các bức ảnh bị nhận diện sai 54](#_Toc481907084)

# Mở đầu

Mạng nơ ron nhân tạo, gọi tắt là mạng nơ ron, là một mô hình xử lý thông tin dựa trên hoạt động của mạng nơ ron trong não người. Nó được ra đời vào khoảng 60 năm trước, nhưng cho đến một vài năm trở lại đây, chúng ta đã chứng kiến sự hồi sinh mạnh mẽ của nó. Mạng nơ ron nhân tạo được cấu thành bởi một số lượng lớn các nơ ron, mỗi nơ ron sẽ gắn với một trọng số cùng các trọng số liên kết trong mạng, chúng làm việc thống nhất để giải quyết một vấn đề nào đó. Nhưng tùy vào từng ứng dụng, mạng nơ ron có thể mang các kiến trúc khác nhau, cho phép thông tin giữa các nơ ron trong mạng lan truyền theo nhiều phương pháp và định hướng thích hợp.

Hiện nay, mô hình này được ứng dụng rộng rãi và đạt được nhiều kết quả tốt trong hầu hết mọi lĩnh vực của trí tuệ nhân tạo, bao gồm thị giác máy tính, xử lý ngôn ngữ tự nhiên, nhận diện giọng nói, nhận diện hình ảnh, nhận diện khuôn mặt. Trong đó, nhận diện khuôn mặt đang trở thành một vấn đề lớn do nhu cầu bức thiết của xã hội bởi hệ thống này có rất nhiều ứng dụng trong thực tế. Người ta có thể tra cứu thông tin tội phạm thông qua nhận diện khuôn mặt, giám sát bằng camera để phát hiện tội phạm tại các khu vực công cộng, tìm trẻ lạc dựa vào các camera đặt ở các nơi công cộng hay phát hiện các nhân vật VIP khi đặt chân vào khách sạn. Và với mạng nơ ron, công nghệ nhận diện khuôn mặt đã đạt được những kết quả nhận diện tốt hơn rất nhiều so với các phương pháp tiền nhiệm. Đó là lý do vì sao tôi lựa chọn đề tài “Nhận diện khuôn mặt sử dụng mạng nơ ron nhân tạo” để làm khóa luận tốt nghiệp. Khóa luận này sẽ tập trung vào giới thiệu mạng nơ ron nhân tạo, phương pháp trích rút dữ liệu để giúp tăng tốc độ duyệt cho mạng nơ ron và bài toán nhận diện khuôn mặt.

Phần còn lại của khóa luận sẽ chia thành năm chương trong đó có một phần kết luận:

Chương 2: Giới thiệu tổng quan về mạng nơ ron nhân tạo. Trong chương này trình bày về cơ sở xây dựng mạng nơ ron nhân tạo dựa trên mạng nơ ron sinh học và đưa ra các cấu trúc của mạng nơ ron nhân tạo trong thực tế. Tìm hiểu sâu hơn về mạng nơ ron đa lớp truyền thẳng và giải thuật lan truyền ngược.

Chương 3: Giới thiệu về phương pháp phân tích thành phần chính. Trong chương này, trình bày về phương pháp phân tích thành phần chính và ứng dụng của phương pháp thành phần chính trong bài toán nhận diện khuôn mặt với mạng nơ ron.

Chương 4: Ứng dụng mạng nơ ron xây dựng hệ thống hệ thống nhận diện khuôn mặt người. Ở chương này, khóa luận sẽ trình bày về hệ thống nhận diện khuôn mặt và kết quả đạt được của phương pháp nhận diện khuôn mặt bằng mạng nơ ron nhân tạo.

Chương 5: Kết luận và hướng phát triển

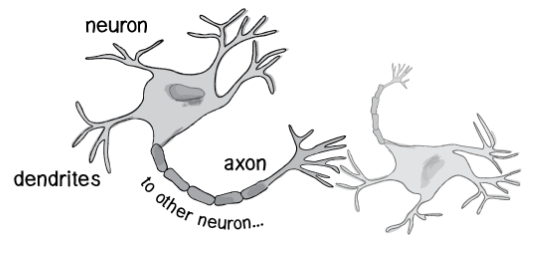
# Tổng quan về mạng nơ ron nhân tạo

## Mạng nơ ron sinh học

Não người không chỉ là cơ quan quan trọng nhất trong cơ thể con người mà còn là cơ quan phức tạp nhất. Bên trong não chứa một hệ thống mạng nơ ron phức tạp nhưng được liên kết chặt chẽ và có sự linh hoạt về xử lý thông tin trong môi trường bất đinh.

Trong bộ não người có khoảng 1011 - 1012 tế bào thần kinh được gọi là các nơ ron và mỗi nơ ron có thể liên kết với 104 nơ ron khác thông qua các khớp nối thần kinh (synapse). Một nơ ron sẽ có cấu tạo chính gồm 2 thành phần cơ bản:

* Thân nơron được giới hạn trong một màng membran và trong cùng là nhân. Nó là nơi phát sinh xung động thần kinh và cũng có thể là nơi tiếp nhận xung động thần kinh từ nơi khác. Thân nơ ron có các đường rẽ nhánh có nhiệm vụ phát hoặc nhận xung động thần kinh.
* Giữa các nơ ron có một liên kết gọi là axon, axon có nhiệm vụ kết nối đầu ra của nơ ron này với đầu vào của nơ ron khác. Mỗi mối liên kết của nơ ron và axon được gọi là synapse. Synapse có một đặc trưng đó là trọng số của nó, và mỗi xung động thần kinh đi qua nó để vào hoặc ra khỏi nơ ron đều nhận được trọng số này. Các trọng số ấy chính là các tham số tự do có thể thay đổi để phù hợp với môi trường xung quanh.



Hình 2.1. Mô hình mạng nơ ron sinh học

Dựa trên chức năng mà rễ của nơ ron được thành hai loại khác nhau: Rễ nhận xung động thần kinh từ các nơ ron khác thông qua axon được gọi là rễ đầu vào, rễ phát xung động thần kinh tới các nơ ron khác thông qua axon được gọi là rễ đầu ra. Và chúng đều nối với axon thông qua khớp nối gọi là synapse. Trong thực tế, nơ ron chỉ có một đầu ra nhưng lại có thể có nhiều đầu vào. Bởi thế mỗi nơ ron có thể tiếp nhận thông tin từ nhiều nơ ron khác nhau nhằm nâng cao sự liên kết giữa các nơ ron.

Mỗi nơ ron ở trong não người được hoạt động theo quá trình điện hóa tự nhiên và đó là lý do vì sao mà mỗi nơ ron có thể phát và nhận được xung động thần kinh từ nơ ron khác. Nơ ron sẽ vào trạng thái kích thích khi có một tác động từ bên ngoài, tác động này chính là lúc mà xung thần kinh được đưa đến nơ ron này, xung thần kinh đó sẽ tạo cho nơ ron một tín hiệu tác động vượt quá ngưỡng cân bằng của nó. Khi ấy, trong tế bào nơ ron sẽ sinh ra rất nhiều các phản ứng hóa học từ đó làm cho nơ ron bị đưa vào một trạng thái kích thích mới gọi là trạng thái kích thích hoàn toàn. Ở trạng thái này, nơ ron sẽ sinh ra một thế năng, thế năng này sẽ tồn tại ở nơ ron trong một thời gian ngắn sau đó nó được chuyển vào mạng nơ ron thông qua axon và từ đó đẩy xung thần kinh của nơ ron này tới các nơ ron khác ở trong mạng.

Nhờ có sự xuất hiện của synapse, mạng nơ ron sinh học có thể thay đổi các đáp ứng kích thích theo thời gian. Các đáp ứng này có thể tăng hay giảm tùy theo môi trường xung quanh nhằm thích nghi với môi trường đó. Do sự liên kết của các nơ ron với nhau nên khi một nơ ron thay đổi đáp ứng kích thích thì sẽ dẫn đến sự thay đổi trạng thái của toàn bộ các nơ ron khác liên kết với nó, từ đó, thay đổi toàn mạng nơ ron. Việc thay đổi này chính là quá trình “học” của nơ ron trong não bộ con người.

Hoạt động tư duy của con người phần nhiều mang tính chủ quan định tính, dựa trên những bài học, kinh nghiệm học được từ khi còn nhỏ. Càng lớn dần thì những bài học hay kinh nghiệm của con người ngày càng nhiều, giúp con người giải quyết các vấn đề trong tự nhiên, xã hội tốt hơn. Bộ não có khả năng tự thay đổi cấu trúc để thích nghi dần với môi trường, làm cho cấu trúc bộ não ngày càng trở nên phức tạp sau mỗi lần học. Cấu trúc của mạng nơ ron sinh học nhờ đó mà cũng thay đổi nhằm thích nghi và để trở nên phù hợp hơn. Một số cấu trúc của nơron được xác định trước, một số sau này mới được hình thành và một số thì bị huỷ bỏ qua quá trình chọn lọc tự nhiên, học và thích nghi.

Qua quá trình tìm hiểu, nghiên cứu cấu trúc và hoạt động của bộ não người, các nhà khoa học đã và đang xây dựng và phát triển các mô hình xử lý thông tin mô phỏng hoạt động của bộ não người. Đó chính là mô hình mạng nơron nhân tạo.

## Giới thiệu về mạng nơ ron nhân tạo

### Định nghĩa về nơ ron nhân tạo

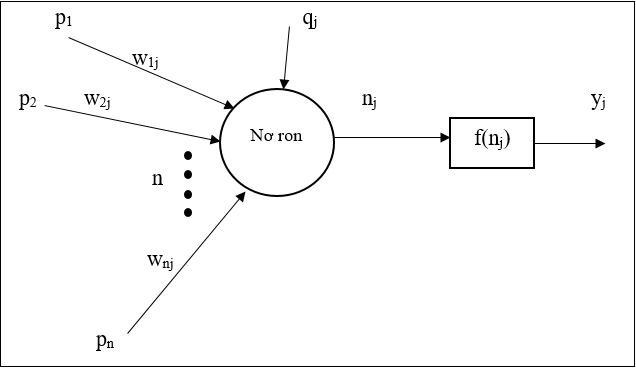
Nơ ron nhân tạo là sự tái hiện đơn giản nhất của nơ ron sinh học. Mỗi nơ ron sẽ có nhiệm vụ đó chính tổng hợp các thông tin từ các đầu vào và sau đó đưa ra đã xử lý tại một đầu ra duy nhất.

Dựa trên cấu trúc thực tế của nơ ron sinh học, mỗi nơ ron nhân tạo sẽ có thể có nhiều đầu vào nhận tin nhưng chỉ có duy nhất một đầu ra. Mỗi đầu vào được gắn với một hệ số bất kì nào đó được gọi là trọng số (weight). Các trọng số này có vai trò cực kì quan trọng trong mạng nơ ron, bởi nó ứng với trọng số liên kết của các khớp nối trong nơ ron sinh học. Do các trọng số liên kết của các khớp nối kích thích và khớp nối ức chế trong nơ ron sinh học có thể âm hoặc dương, vậy nên trọng số trong nơ ron nhân tạo cũng mang tính chất này.

Mỗi nơ ron nhân tạo có một giá trị gọi là giá trị ngưỡng (threshhold). Khi nơ ron nhận được thông tin từ các đầu vào, nó sẽ tính tổng các tích giá trị đầu vào và các trọng số liên kết của mỗi đầu vào, đồng thời so sánh với giá trị ngưỡng để tạo ra một giá trị nhất định, giá trị này sẽ tiếp tục được đưa đến một nơ ron khác có sự liên kết với nơ ron này. Ngoài ra, giá trị đầu ra còn phụ thuộc vào hàm truyền (transfer function), hàm này có nhiệm vụ nhận vào tín hiệu và tính toán để đưa ra kết quả đầu ra.

### Mô hình nơ ron nhân tạo

Một nơ ron nhân tạo đã thể hiện gần như đầy đủ các tính chất cơ bản mà nơ ron sinh học có được. Chúng nhận và xử lý thông tin để sử dụng cho hoạt động của toàn mạng nơ ron. Nó sẽ nhận các tín hiệu đầu vào lấy từ các nơ ron khác truyền đến và xử lý để đưa ra tín hiệu đầu ra. Sau đây là mô hình của nơ ron nhân tạo:



Hình 2.2. Mô hình nơ ron nhân tạo

Trong đó:

* : Các tín hiệu đầu vào với i từ 1 đến n
* : Các trọng số tương ứng với các tín hiệu đầu vào với i từ 1 đến n của nơ ron j.
* : Giá trị ngưỡng của nơ ron j
* : Tín hiệu tổng hợp đầu vào của nơ ron j
* : Hàm kích hoạt
* : Tín hiệu đầu ra của nơ ron j

Đầu vào của nơ ron nhân tạo gồm n tín hiệu với i từ 1 đến n. Mỗi tín hiệu đầu vào tương ứng với một trong số , nó thể hiện mức độ ảnh hưởng của tín hiệu đến nơ ron j. Tín hiệu đầu vào của một nơ ron có thể là dữ liệu từ bên ngoài mạng hoặc từ kết quả đầu ra của chính nó hoặc một nơ ron khác.

Nhằm làm tăng khả năng thích của mạng nơ ron trong toàn bộ quá trình học, người ta gắn thêm một giá trị ngưỡng cho mỗi nơ ron, giá trị này được coi như một đầu vào của mỗi nơ ron trong mạng. Giá trị ngưỡng này được gọi là bias và trong mô hình trên ta kí hiệu giá trị này bằng kí hiệu nghĩa là giá trị ngưỡng của nơ ron thứ j.

Các tín hiệu đầu vào của mỗi nơ ron được tổng hợp bằng một bộ cộng, kết quả cho quá trình này được gọi là của nơ ron thứ j. Ta có thể tính toán được giá trị của bằng công thức sau:

[1]

Hoặc

Sau khi tổng hợp được tín hiệu đầu vào , ta sẽ sử dụng hàm kích hoạt (transfer function) biến đổi để thu được tín hiệu đầu ra . Ta có công thức sau:

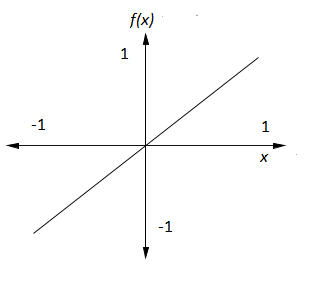
Tóm lại, ta có thể xem một nơ ron là một hàm phi tuyến có nhiều đầu và và chỉ có một đầu ra duy nhất.

Hàm kích hoạt ở mỗi nơ ron phải thỏa mãn các điều kiện sau:

* Tín hiệu đầu ra phải không âm với mọi giá trị của với j từ 1 đến n.
* Hàm kích hoạt phải có giá trị nằm trong khoảng

Hàm kích hoạt hay còn được gọi là hàm nén vì chúng nén tín hiệu đầu ra vào một khoảng nhỏ. Sau đây là một số hàm kích hoạt hay được sử dụng:

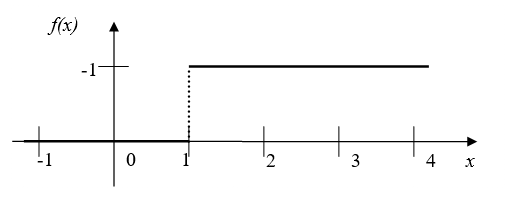
* Hàm đồng nhất (Linear function, Identity function)



Hình 2.3. Hàm đồng nhất

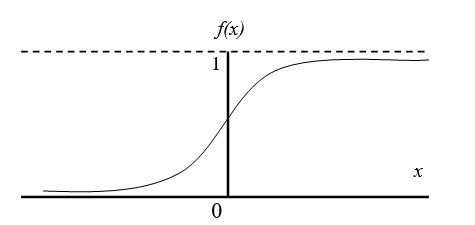
* Hàm bước nhị phân (Binary step function, Hard limit function)

Hàm này được gọi là hàm ngưỡng (Threshhold function hay Heaviside function):



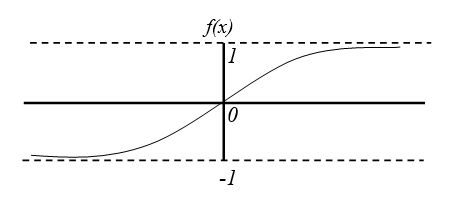
Hình 2.4. Hàm bước nhị phân với q = 1

* Hàm sigmoid (Sigmoid function (Logistic))



Hình 2.5. Hàm Sigmoid

* Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))



Hình 2.6. Hàm sigmoid lưỡng cực

### Định nghĩa mạng nơ ron nhân tạo

Mạng nơ ron nhân tạo là sự kết hợp của một tập các nơ ron nhân tạo với nhau, chúng được liên kết với nhau và cùng hoạt động song song. Mạng nơ ron sẽ hoạt động dựa trên cấu trúc đã cấu thành nó, các trọng số liên kết của các nơ ron và việc chọn hàm truyền để xử lý ở bên trong của mỗi nơ ron. Ngoài chức năng xử lý, hệ thống mạng nơ ron còn có khả năng học số liệu và tổng quát hóa từ các số liệu đã học. Các nơ ron trọng mạng được gọi là các nút, các nút này sẽ được sắp xếp theo một thứ tự cụ thể tùy theo cấu trúc mạng. Mỗi mạng nơ ron thường sẽ bao gồm ba lớp chứa các nơ ron: Lớp đầu vào (chứa các nơ ron nhận input đầu vào xử lý rồi truyền đến các nơ ron trọng mạng), lớp đầu ra (chứa các nơ ron là kết quả của quá trình xử lý trong toàn mạng), lớp ẩn (chưa các nơ ron xử lý trung gian để tăng hiệu quả cho quá trình duyệt mạng).

Có ba yếu tố quan trọng sẽ quyết định chức năng của mạng:

* Cấu trúc mạng
* Quá trình xử lý bên trong nơ ron (Các hàm truyền)
* Mức độ liên kết giữa các nơ ron

Mức độ liên kết của các nơ ron sẽ được xác định dần dần thông qua quá trình “học” của toàn mạng hay còn gọi là quá trình huấn luyện mạng. Mức độ liên kết này được biểu hiện bởi các trọng số, các trọng số này có vai trò rất quan trọng bởi nó chính là phương tiện để lưu trữ thông tin của mạng nơ ron. Và việc cập nhật trọng số được coi là nhiệm vụ chính, nhiệm vụ quan trọng nhất của quá trình huấn luyện mạng nơ ron.

Ta có thể thấy rằng một mạng nơ ron là một bộ xử lý song song, có xu hướng hoạt động tự nhiên lưu trữ các thông tin và dựa trên quá trình học để có thể tạo ra các tri thức mới. Điều này đã được ứng dụng rất thành công trong các bài toán về dự đoán và nhận dạng. Từ đó, ta thấy được sự giống nhau giữa mạng nơ ron nhân tạo và bộ não giống nhau ở hai khía cạnh:

* Tri thức có được thông qua quá trình học
* Các trọng số liên kết giữa các nơ ron được dùng như phương tiện lưu trữ.

Vậy, từ khái niệm khái quát về mạng nơ ron ta có thể thấy được mạng nơ ron nhân tạo trong thực tế là một mô hình có thể giải quyết được đa số các bài toán về dự đoán và nhận dạng trong thực tế. Bởi, nó mô phỏng não người một cách trực quan nhất, từ đó, có thể thu thập, lưu trữ và sử dụng tri thức đã có để đưa ra các dự đoán tốt nhất.

### Các đặc trưng của mạng nơ ron nhân tạo

* Tính chất phi tuyến: Đây là một tính chất rất quan trọng. Mạng nơ ron nhân tạo có thể tính toán phi tuyến hoặc tuyến tính. Nó cho phép mạng nơ ron tạo ra các mặt cắt trong không gian dữ liệu, không phải là siêu phẳng mà là lồi lõm không đều. Như vậy, không gian mẫu có thể chia thành nhiều miền. Mỗi miền được gọi là một lớp.
* Khả năng tự thích nghi và tự tổ chức: Các mạng nơ ron có khả năng xử lý thích nghi và điều chỉnh bền vững dựa vào các thuật toán học và quy tắc tự tổ chức. Khả năng xử lý thích nghi thể hiện ở khả năng biến đổi các trọng số trong mạng tùy theo sự thay đổi của môi trường. Một mạng nơ ron đã được huấn luyện trong môi trường xác định, vẫn có thể thích nghi trong môi trường khác bằng cách tự thay đổi trọng số các liên kết. Vì vậy dù mẫu không ổn định hay bị nhiễu thì nó vẫn có thể đáp ứng được thích hợp. Tuy nhiên, đôi khi sự tự thích nghi ảnh hưởng xấu đến kết quả cuối cùng của mạng.
* Tính chịu lỗi: Não người có khả năng chấp nhận lỗi, tức là đối với những thông tin thiếu chính xác và không đầy đủ mà vẫn có thể xử lý, giải quyết được vấn đề đặt ra. Mạng nơ ron bắt chước khả năng này của bộ não. Hệ thống mạng nơ ron vẫn có thể tiếp tục hoạt động và điều chỉnh, khi nhận tín hiệu vào có một thông tin bị sai lệch hay bị nhiễu. Khi một số nơ ron không thực hiện tính toán, hay một vài liên kết bị hỏng thì khả năng của mạng chỉ bị giảm đi chứ không đổ vỡ hoàn toàn. Mạng nơ ron luôn đưa ra kết quả hợp lý trong mọi tình huống.
* Tính đồng dạng trong thiết kế: Các mô hình mạng nơ ron đều có chung đặc điểm là được cấu thành từ các nơ ron riêng biệt liên kết với nhau. Nghĩa là mô hình càng lớn, càng phức tạp thì mạng nơ ron sẽ càng cho kết quả chính xác. Các tổ hợp mạng có thể xây dựng bằng cách tích hợp nhiều mạng khác nhau.

### Chức năng của mạng nơ ron nhân tạo

#### Chức năng phân loại mẫu

Phân loại mẫu là sự phân chia các mẫu ra thành các nhóm khác nhau, các nhóm này sẽ có chung một số đặc điểm nhất định nào đó. Mạng nơ ron có thể thể tự phân loại các mẫu đầu vào và tạo ra các mẫu đầu ra có phân loại. Khi cho một mẫu đầu vào thì mạng nơ ron sẽ tạo ra được một mẫu đầu ra đúng với phân loại của chúng. Ta có thể nói mạng nơ ron là một bộ phân loại mẫu. Điểm làm cho mạng nơ ron tạo nên sự khác biệt với các bộ phân loại mẫu khác trong học máy chính là khả năng tự học và tổng quát hóa dựa trên các mẫu đã có.

#### Học và tổng quát hóa

Học là việc mạng nơ ron thu thập được các thông tin, các mẫu, các tri thức ở đầu vào và các thông tin, các mẫu, các tri thức tương ứng ở đầu ra và mạng có thể học để phân loại được chúng cho thật chính xác.

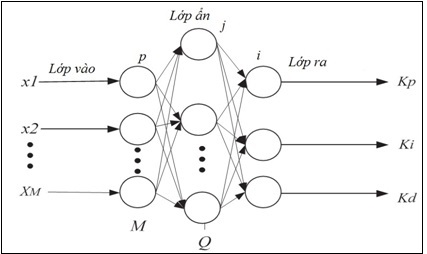
Tổng quát hóa là mạng không chỉ nhận dạng chính xác được mẫu mà mạng đã được học, nó còn có thể rút ra được những đặc tính riêng của từng mẫu và tạo ra một tri thức mới. Từ đó, mạng có thể tự nhận biết và phân loại các mẫu mới mà nó chưa từng được học.

Chức năng này của mạng cho thấy tính linh động của mạng nơ ron, nó có thể giải quyết một số vấn đề tính toán hoặc suy luận khi thiếu thông tin, hoặc có rất ít thông tin, hay thông tin không đầy đủ. Quan trọng nhất là phải xây dựng được một mô hình mạng thật tốt và phải có phương pháp học phù hợp với bài toán đưa ra.

### Các kiến trúc của mạng nơ ron nhân tạo

#### Kiến trúc truyền thẳng (feedforward architechture)

Đây là dạng kiến trúc mà các nơ ron sẽ liên tiếp truyền tin từ lớp này sang lớp kế tiếp mà không hề có truyền tin ngược lại từ nơ ron lớp sau về nơ ron lớp trước. Do vậy các lớp bên trong của mạng nơ ron ngoài lớp đầu vào và đầu ra không có kết nối với bên ngoài, do vậy chúng được gọi là các lớp ẩn. Mỗi nút hay nơ ron trong lớp này sẽ được nối với mỗi nơ ron ở trong lớp tiếp theo. Do đó, các thông tin sẽ liên tục được truyền tải từ lớp này sang lớp kia. Và không có sự liên kết giữa các nơ ron trong cùng một lớp.



Hình 2.7. Kiến trúc mạng truyền thẳng

Với kiến trúc này, mạng nơ ron được cấu thành từ nhiều lớp liên kết với nhau, bao gồm một lớp đầu vào, một lớp đầu ra, lớp ẩn (có thể có một, nhiều hoặc không có lớp ẩn nào). Trong đó tín hiệu đầu vào sẽ được đưa vào lớp vào của mạng, tín hiệu đầu ra là kết quả của quá trình xử lý toàn mạng sẽ xuất ở lớp đầu ra.

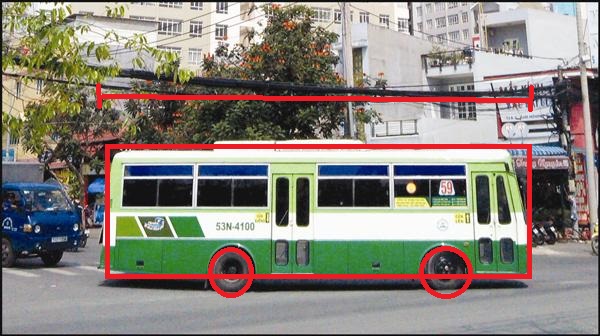
Lớp ẩn là thành phần nội tại của mạng, nó không có bất kì tiếp xúc nào với môi trường bên ngoài. Số lượng lớp ẩn có thể dao động từ 0 đến một vài lớp, trong trường hợp mạng nơ ron có n lớp ẩn trong đó n > 1 thì mạng nơ non đó gọi là mạng nơ ron sâu (Deep neural networks). Công việc của lớp ẩn đó là chuyển đổi đầu các đầu vào của lớp trước đó thành một thứ gì đó mà lớp đầu ra có thể sử dụng.

Ví dụ, ta cần sử dụng mạng nơ ron để xác định trong bức ảnh sau có chiếc xe bus hay không?



Hình 2.8. Hình ảnh xe bus

Ta sẽ giải quyết vấn đề này bằng cách sử dụng một lớp để dò bánh xe (nhận diện xem trong hình ảnh có bánh xe hay không), một lớp để nhận dạng khối hộp (do đa phần xe bus của các hãng hiện nay đều có dạng hình hộp từ đó để nhận ra xem nó có phải hình dáng chiếc xe bus hay không), một lớp để dò kích thước của chiếc xe so với khung hình (để cho ta biết nó có đủ lớp để là một chiếc xe bus hay không). Các yếu tố này được khoanh vùng bởi nét màu đỏ ở hình sau.



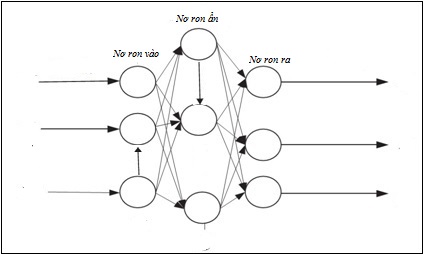
Hình 2.9. Hình ảnh xe bus đã khoanh vùng

Các yếu tố được nêu ở trên chính là ví dụ về các lớp ẩn trong mạng nơ ron, chúng không phải một phần của hình ảnh thô, nhưng đó là công cụ được thiết kế để giúp ta xác định được chiếc xe bus. Nếu như tất cả các lớp này được sử dụng ở trong mạng thì việc nhận diện chiếc xe bus sẽ có độ chính xác rất lớn.

Các lớp ẩn sẽ phát huy hết tác dụng của nó khi giải quyết một vấn đề phức tạp theo ngữ cảnh hoặc các vấn đề không quá rõ ràng như nhận diện hình ảnh và cụ thể vấn đề mà khóa luận đang nhắc đến đó là nhận diện khuôn mặt.

#### Kiến trúc mạng phản hồi (Feedback architecture)

Đây là dạng kiến trúc khác với kiến trúc truyền thẳng, mạng có kiến trúc hồi quy có chứa các liên kết ngược. Nghĩa là trọng nơ ron với kiến trúc này, các tín hiệu của nơ ron ở trong mạng có thể đi theo cả hai hướng bằng cách tạo liên kết vòng ở trong mạng. Kiến trúc mạng phản hồi rất mạnh mẽ và có thể trở nên cực kì phức tạp. Các tính toán thu được từ đầu vào trước đó sẽ được lưu lại sau đó được nạp lại vào mạng, đây là điểm khác biệt hoàn toàn với mạng có cấu trúc truyền thẳng. Mạng có kiến trúc này cực kì linh động trong việc thay đổi trạng thái, nó sẽ thay đổi liên tục cho đến khi đạt một điểm cân bằng. Điểm cân bằng này sẽ được tồn tại cho đến khi có những thay đổi đầu vào và lúc ấy điểm cân bằng sẽ thay đổi, mạng sẽ tiếp tục quá trình học của mình. Đặc điểm này của mạng phản hồi có ý nghĩa đặc biệt trong xử lý ngôn ngữ tự nhiên, nhất là trong việc dự đoán văn bản (tóm tắt một bài báo), xác định ngôn ngữ, nhận diện giọng nói, phân loại các tín hiệu.



Hình 2.10. Kiến trúc mạng phản hồi

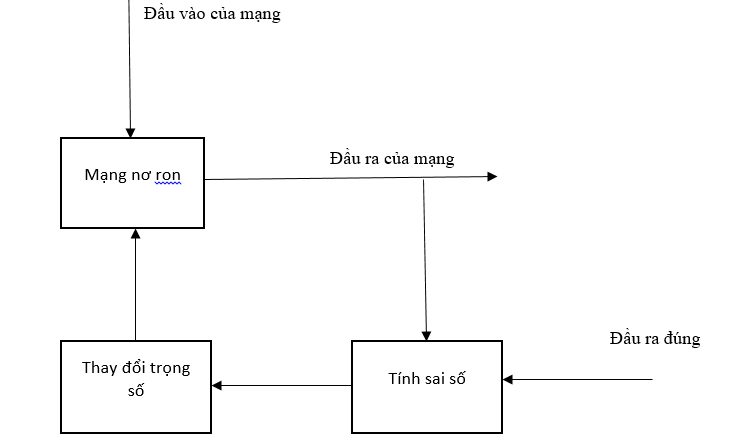
### Các phương pháp của mạng nơ ron nhân tạo

Mạng nơ ron nhân tạo có khả năng xử lý thông tin khá giống với bộ não người, vậy nên mạng nơ ron có khả năng học, khả năng tìm ra các tri thức mới dựa vào những tri thức đã có. Trong thái thái học, thông tin trong mạng được lan truyền theo các chiều cụ thể, tùy vào từng loại cấu trúc mạng để làm thay đổi trọng số nhiều lần, từ đó đạt được trọng số tốt nhất có thể cho ra các mẫu dữ liệu tương đương đã học. Do bản chất của việc duyệt mạng nơ ron chính là học máy, vậy nên mạng nơ ron cũng sẽ có hai kiểu học ứng với các cấu trúc mạng khác nhau, các kiểu học đó là học có giám sát và học không giám sát.

#### Học có giám sát

Học có giám sát là một kỹ thuật máy học từ một tập dữ liệu gán nhãn cho trước. Tập dữ liệu cho trước sẽ chứa nhiều bộ dữ liệu khác nhau. Nhiệm vụ của học có giám sát đó chính là dự đoán đầu ra mong muốn của giá trị đầu vào. Hay nói cách khác, học có giám sát tức là máy học dựa vào sự trợ giúp của con người, con người sẽ dạy cho máy học và giá trị đầu ra mong được định trước bởi con người. Vậy, khi học có giám sát với mạng nơ ron, ta sẽ cung cấp cho mạng nơ ron, tập hợp các giá trị đầu vào và đầu ra mong muốn tương ứng. Sau đó, đưa dần từng cặp giá trị đầu vào và ra tương ứng vào đầu vào và đầu ra của mạng nơ ron. Mỗi cặp dữ liệu ra đưa vào trong mạng xử lý tới khi thu được đầu ra ta sẽ so sánh với đầu ra của lần duyệt mạng trước đó. Sau mỗi lần duyệt mạng, sai số giữa đầu ra của mạng lần này với lần trước đó sẽ được sinh ra, đây chính là thành phần sử dụng để điều chỉnh trọng số và hệ số ngưỡng của các nơ ron trong mạng sao cho kết quả đầu ra của mạng là gần với kết quả đầu ra mong muốn nhất.

Trong mạng nơ ron ta sẽ có hai cách để sử dụng tập dữ liệu mẫu đó là học toàn bộ tất cả cùng 1 lúc cho đến khi thành thạo mẫu hoặc là học lần lượt từng mẫu cho đến khi có kết quả tốt nhất rồi mới chuyển sang mẫu tiếp theo.



Hình 2.11. Sơ đồ mô tả luật học giám sát

Để có thể so sánh được mức độ chính xác của đầu ra mạng nơ ron so với đầu ra đúng ta cần dùng hàm để đánh giá sai số. Hàm tổng bình phương sai số là hàm sai số phổ thông thường được dùng nhất, hàm này được sử dụng bằng cách tổng của các bình phương hiệu giữa kết quả đầu ra chính xác và kết quả đầu ra của mạng. Từ đó dựa vào thuật toán duyệt mạng để thay đổi trọng số. Mô hình học này thường được ứng dụng trong các bài toán về nhận dạng.

#### Học không giám sát

Đây là một kĩ thuật nhằm tìm ra một mô hình hay cấu trúc bị ẩn bởi tập dữ liệu không được gán nhãn cho trước. Phương pháp này khác với học không giám sát là không thể xác định trước đầu ra từ tập dữ liệu huấn luyện được. Tùy thuộc vào tập huấn luyện thì kết quả đầu ra sẽ khác nhau. Đây cũng là cách học phù hợp với kiểu mạng nơ ron nhân tạo có cấu trúc phản hồi được sử dụng trong các vấn đề về xử lý ngôn ngữ tự nhiên.

## Giới thiệu mạng nơ ron đa lớp truyền thẳng

### Tổng quan về mạng nơ ron đa lớp truyền thẳng

Có nhiều mô hình mạng nơ ron đã được nghiên cứu, trong đó, mạng nơ ron đa lớp truyền thẳng với thuật toán lan truyền ngược là mô hình đơn giản và hay được sử dụng nhất. Trong khóa luận này, ta sẽ chỉ tìm hiểu và áp dụng mạng nơ ron đa lớp truyền thẳng cho bài toán nhận diện khuôn mặt bởi mức độ phù hợp của nó.

Mạng nơ ron nhân tạo là mô hình học từ dữ liệu có trước dựa trên cơ chế tổ chức và xử lý thông tin giống não bộ của con người. Thông qua quá trình học các mẫu dữ liệu trong tập dữ liệu cho trước, mạng nơ ron nhân tạo sẽ tự thay đổi chính mình và càng lúc càng tích lũy thêm nhiều kinh nghiệm để có thể giải quyết được tất cả các vấn đề.

Mạng nơ ron nhân tạo có thể ví như một đồ thị định hướng có trọng số đặc biệt. Trong đó, các nút chưa các trọng số được phân vào các lớp. Lớp nhập có nhiệm vụ nhận thông tin đầu vào, các lớp ẩn có nhiệm vụ thực hiện các biến đổi (tổng hợp và nén) thông tin lan truyền qua các nút ẩn và kết quả đạt được của mạng nơ ron nhân tạo sẽ được cho ra ở lớp xuất. Bằng cách hiệu chỉnh trọng số ở các nút (giảm sai số giữa kết quả đầu ra của mạng nơ ron với kết quả đúng), mạng nơ ron có thể học và đưa ra các quy luật từ các tập dữ liệu thu thập có trước (sai số trên tập dữ liệu có thể rất nhỏ).

### Kiến trúc mạng

Đây là một loại mạng nơ ron thuộc loại kiến trúc truyền thẳng và đa lớp, trong mạng có thể có một hoặc nhiều lớp ẩn. Mỗi lớp ở trong mạng có một ma trận trọng số , vector bias và véc tơ đầu ra. Để có thể phân biệt giữa các lớp trong mạng với nhau người ta dùng thêm các chỉ số phụ cho mỗi biến. Nếu coi lớp đang xét là lớp thứ trong mạng nơ ron thì sẽ là ma trận trọng số của lớp , sẽ là véc tơ bias cho lớp thứ i. Các nơ ron trong cùng một lớp thường sẽ có chung một hàm truyền, có thể là hàm sigmoid, hàm bước nhị phân, hàm đồng nhất hay hàm sigmoid lưỡng cực.

Nếu xét, một mạng nơ ron truyền thẳng có đầu vào, có nơ ron ở lớp thứ nhất, nơ ron ở lớp thứ hai. Đầu ra của lớp thứ nhất là đầu vào của lớp thứ hai. Vậy nên, đầu vào của lớp thứ 2 chính là trong véc tơ đầu ra của lớp thứ nhất và nó sẽ có ma trận trọng số có kích thước là x . Lớp cuối cùng đưa ra kết quả được gọi là lớp đầu ra. Các lớp còn lại trừ lớp đầu vào và lớp đưa ra kết quả cuối cùng của mạng gọi là các lớp ẩn.

### Giải thuật lan truyền ngược (Backpropagation Algorithm)

Giải thuật lan truyền ngược đã được giới thiệu vào những năm 1970, nhưng tầm quan trọng của nó đã không được đánh giá cao cho đến khi bài báo nổi tiếng về nó của ba nhà nghiên cứu David Rumelhart, Geoffrey Hinton và Ronald Williams năm 1986 ra đời. Bài báo này miêu tả một số mạng nơ ron được hoạt động bởi giải thuật lan truyền ngược và nó đã đem lại tốc độ học nhanh hơn cho mạng nơ ron, phương pháp này đã giải quyết các vấn đề về duyệt mạng nơ ron trước đó. Giải thuật lan truyền ngược thường được dùng trong mạng nơ ron nhân tạo có dạng: một lớp nhập – một lớp ẩn – một lớp xuất. Giả sử cho một tập luyện K mẫu gồm có đầu vào và các nhãn như sau

trong đó

Thuật toán lan truyền ngược cung cấp kĩ thuật hiệu chỉnh các trọng số trong mạng nhằm mục đích phân lớp các giá trị đầu vào một cách chính xác, hay nói cách khác: mạng có thể học và hiểu được dữ liệu đầu vào.

Để đo lường sự trùng khớp giữa ánh xạ cần xây dựng với hàm đích cho trước (hay nói cách khác là để đánh giá sự chính xác khả năng học của mạng) ta sử dụng sai số trung bình bình phương E.

Cho một tập mẫu gồm N mẫu

Gọi là giá trị ước tính qua mạng nơ ron, thì sai số sẽ trung bình sẽ được tính: [2]

Trong trường hợp mô hình mạng nơ ron là hàm phi tuyến, việc xác định được tập trọng số để mô hình đạt được sai số ít nhất là rất khó. Chính vì thế phương pháp giảm gradient (gradient descent) thường được sử dụng trong các trường hợp phức tạp này.

Phương pháp giảm gradient bao gồm các bước cơ bản sau:

* Chọn ngẫn nhiên một điểm trong không gian trọng số
* Tính độ dốc của mặt lỗi tại
* Cập nhật các trọng số theo hướng dốc nhất của mặt lỗi
* Xem điểm này như điểm mới

Phương pháp này thực hiện lặp đi lặp lại từ bước 2 đến bước 4 cho đến khi các giá trị trong bộ trọng số tiếp cận được điểm thấp nhất trong mặt lỗi.

Biến thiên trọng số ở mỗi bước (bước thứ i) sẽ được tính theo công thức sau:

Với là hai hằng số dương đặc trưng cho tốc độ học (learning rate) và mô men của quá trình học. Tuy nhiên có một số cải tiến cho phép thay đổi các hệ số trong từng giai đoạn học, cũng như theo từng mẫu học để việc luyện mạng đạt hiệu quả cao nhất.

Nói tóm lại, thuật toán lan truyền ngược có thể tóm tắt được thành hai quá trình: lan truyền tiến (forward) và lan truyền ngược (backward)

*Lan truyền tiến*: là quá trình tính giá trị đầu ra dựa trên những trọng số có sẵn. Giả sử tại với một lớp j của mạng có đầu vào là tương ứng là các giá trị trọng số . Ta sẽ có giá trị :

Hàm f() được gọi là hàm truyền mà phần 2.2.2 đã nhắc đến. Hàm truyền trong mạng tiến đa mức cũng có các tính chất sau:

* Bị chặn
* Đơn điệu tăng
* Hàm liên tục trơn

Mọi hàm đều thỏa mãn tính chất trên đều có thể sử dụng trong mạng tiến đa mức. Một số hàm sử dụng trong mạng lan truyền ngược đó là:

* Hàm sigmoid
* Hàm sigmoid lưỡng cực
* Hàm đồng nhất
* Hàm tang-hyperbol

Trong bốn hàm trên, hàm logistic được sử dụng phổ biến nhất. Khi các giá trị được tính, giá trị sai số trung bình phương E sẽ được xác định. Khi đó quá trình lan truyền ngược sẽ được tiến hành.

*Lan truyền ngược:* Là quá trình cập nhập lại các trọng số mạng trong ma trận trọng số dựa trên sai số trung bình phương. Gọi là trọng số tại bước thứ , công thức cập nhật trọng số sẽ là:

được tính theo công thức sau:

Trong đó:

Với là giá trị đưa vào hàm truyền tương ứng ở mỗi nút.

Với mạng đang xét là mạng ba lớp, tại nút xuất theo quy tắc chuỗi ta có:

Đạo hàm của sai số trung bình bình phương và hàm logistic được tính như sau:

Vậy giá trị tại nút xuất được tính với công thức sau:

Do đó, giá trị tại nút xuất sẽ được tính:

Tương tự cho nút ẩn thứ j:

với là trọng số cung nối từ nút ẩn thứ đến nút xuất.

Nói tóm lại, khi quá trình lan truyền ngược thực hiện, các trọng số sẽ được cập nhật lại sao cho sai số trung bình bình phương có xu hướng giảm dần. Quá trình lan truyền tiến và lan truyền ngược được thực hiện xen kẽ nhiều lần cho đến khi đạt được giá trị sai số trung bình bình phương E mong muốn. Ta càng lấy giá trị tốc độ học càng nhỏ bao nhiêu thì sự thay đổi trọng số càng nhỏ bấy nhiêu và từ đó sẽ tăng độ chính xác cho mô hình. Nhưng lại vướng phải một vấn đề đó là tốc độ học sẽ bị ảnh hưởng, cụ thể là sẽ bị chậm đi. Ngược lại, nếu chọn tốc độ học lớn thì sự thay đổi về trọng số sẽ không ổn định, dẫn đến sai lệch trong kết quả đầu ra của mạng nơ ron.

### Các kỹ thuật cải tiến thuật toán lan truyền ngược

#### Khởi tạo và cố định tốc độ học

Theo đề nghị của Harry A.C.Eaton và Tracy L.Oliver, tốc độ học nên được khởi tạo theo công thức [3]:

Và đặt mô men học:

Với phương pháp này, người ta đã chia mẫu đã luyện thành m nhóm con theo các đặc tính riêng (những mẫu tương tự gom vào 1 nhóm con), với là số mẫu luyện trong nhóm con thứ .

#### Giảm tốc độ học theo quá trình học

Trong quá trình huấn luyện dữ liệu, để tăng tốc độ học Christian Darken và John Moody đã giới thiệu mỗi kỹ thuật có tên “Search – Then – Converge” nghĩa là Tìm và Hội Tụ [3]. Giả sử tốc độ học khởi đầu là tại lần lặp thứ tốc độ học sẽ được cập nhật bởi công thức sau:

với r là hằng số điều chỉnh tốc độ học.

Việc dùng kĩ thuật này cho phép đẩy nhanh tốc độ học lúc ban đầu, càng về sau, tốc độ học sẽ càng giảm tức quá trình huấn luyện sẽ chậm đi. Việc lựa chọn hằng số r tương ứng với dữ liệu luyện dựa trên phương pháp thử và sai số.

#### Điều chỉnh tốc độ học theo mỗi mẫu luyện

Năm 1989, J. Schmidhuber đã giới thiệu kĩ thuật điều chỉnh tốc độ học theo mẫu luyện, kỹ thuật này không dùng hệ số mô men [3]. Chính vì thế, tiếp tuyến của mặt lỗi trong quá trình luyện sẽ được sử dụng. Những giá trị trọng số được tìm bằng cách tính điểm 0 là giao điểm giữa mặt phẳng lỗi với mặt phẳng 0. Trong thực tế sẽ tồn tại một số mặt phẵng lỗi không bao giờ đạt đến mặt phẳng 0, vì vậy một giá trị hằng số (có giá trị nhỏ) được thêm vào nhằm đảm bảo tồn tại điểm 0 trên mặt phẳng lỗi.

, theo J. Schmidhuber kỹ thuật này đảm bảo sẽ không mắc phải lỗi cực tiểu địa phương (Khi và ). Tuy nhiên, việc cập nhật như vậy là rất lớn, điều đó có thể làm ảnh hưởng đến toàn bộ mạng.

#### Tiến hóa tốc độ học

R. Salomon sử dụng kỹ thuật tiến hóa đơn giản để điều chỉnh tốc độ học [3]. Với một tập các giá trị có tốc độ học (được tạo bằng cách tăng hay giảm giá trị ở bước trước) chọn giá trị tốt nhất trong tập để dùng cho bước kế tiếp. Kỹ thuật này được tiến hành theo các bước sau.

* Tạo hai mạng như nhau và khởi tạo tốc độ học
* Điều chỉnh trọng số của 2 mạng theo công thức:
* Tinh tổng lỗi cho hai mạng, chọn và khởi tạo hai mạng khác tương tự nhau, nếu tổng lỗi tăng lên với tốc độ học được khởi tạo lại. Trong trường hợp lỗi giảm, tiếp tục luyện mạng với tốc độ luyện tương ứng theo công thức trên.

Kỹ thuật này thích hợp cho mạng lan truyền ngược trong việc điều chỉnh tốc độ học, song kỹ thuật này vẫn còn tồn tại một số hạn chế đó là: Thời gian luyện sẽ tăng gấp đôi, không phù hợp với mạng đã được luyện.

#### Điều chỉnh tốc độ học bằng thay đổi tín hiệu gradient

Các kỹ thuật cải tiến giới thiệu ở trên được phân vào nhóm điều chỉnh tốc độ học toàn phần. Kỹ thuật này điều chỉnh tốc độ học bằng thay đổi tín hiệu được xem là kỹ thuật cải tiến tốt nhất [3] trong nhóm điều chỉnh tốc độ học địa phương. Kỹ thuật này dựa trên ý tưởng, mỗi trọng số trong mạng sẽ được cập nhật với tốc độ học khác nhau. Kỹ thuật này sẽ khảo sát hai giá trị cuối trong mức giảm gradient, tốc độ học sẽ thay đổi theo tín hiệu này.

Thuật toán được mô tả với các bước sau:

* Khởi tạo với một giá trị đủ nhỏ
* Tốc độ học được điều chỉnh

(n-1) nếu

(n-1) nếu

với

* Cập nhật trọng số

Các đặc trưng của kỹ thuật cải tiến nhóm thay đổi hệ số tốc độ học:

* Điều chỉnh tốc độ học thay đổi tín hiệu
* SuperSAB
* Quickprop
* Backprop và kỹ thuật giảm tốc độ học
* Delta – Bar – Delta

#### Ngăn lỗi bão hòa

Mặc dù thuật toán lan truyền ngược và phương pháp giảm gradient có rất nhiều ưu điểm trong việc giải bài toán về phân lớp và nhận dạng song bên cạnh đó, phương pháp này vẫn còn tồn tại một vài khuyết điểm sau:

* Qua trình học thường rơi vào những cực tiểu địa phương
* Tốc độ học thường chậm

Trong các nghiên cứu, để tăng tính hiệu quả của việc học trọng mạng nơ ron, người ta đã phát hiện ra rằng để tăng hiệu quả việc học thì cần phải ngăn chặn hiện tượng bão hòa sớm trong quá trình học. Hiện tượng bão hòa sớm là hiện tượng kết quả xuất luôn có giá trị lỗi cao trong giai đoạn đầu quá trình học (đây cũng là quá trình hội tụ trong việc học trở nên chậm lại). Lỗi bão hòa là một trong những nguyên nhân gây nên hiện tượng bão hòa sớm. Lỗi bão hòa là hiện tượng các nút ở trọng mạng sẽ có các kết xuất gần tiến đến giá trị 0 hay 1 nhưng lại không đúng với kết quả mong muốn thật sự. Dựa vào phần 2.3.3 ta có được hai công thức:

Biến thiên trọng số ở mỗi bước (bước thứ t):

với là giá trị đầu ra mẫu luyện và z là giá trị dự đoán.

Với công thức trên, khi z càng tiến đến 0 hay 1 thì dù cho sai số có đạt được giá trị nào đó đi chăng nữa thì cũng sẽ tiến tới 0 (trạng thái bão hòa, khi ). Như vậy khi có lỗi bão hòa xuất hiện thì giá trị trọng số sẽ được cập nhật một lượng rất ít làm cho quá trình hội tụ chậm đi.

Vì thế, việc giải quyết lỗi bão hòa sẽ làm cho tốc độ hội tụ của quá trình học tăng lên, cũng như không rơi vào trạng thái cực tiểu địa phương.

Để ngăn chặn lỗi bão hòa xảy ra, năm 2001 một nhóm tác giả ở Đài Loan đã đề nghị một công thức [4]

với a là hệ số tỉ lệ, n là một hằng số

Từ đó ta được giá trị tại nút xuất như sau:

Ảnh hưởng của ESP lên sai số trung bình bình phương E được tính:

Công thức tính E ban đầu:

Biến thiên trọng số tại bước thứ t:

Vậy khi đó:

Hay:

Với

Lấy tích phân (đơn giản hóa không xét trọng ngưỡng)

Giả sử hàm ESP như sau:

Khi đó sai số trung bình bình phương được tính như sau:

Dựa vào công thức biến thiên hàm số tại bước thứ t và hàm ESP ta có:

= OFFSET – Distance – Entropy

trong công thức trên có thể chia làm 2 thành phần:

Distance\_Entropy: phần sau của công thức có thể xem như khoảng Entropy vì là khoảng cách Entropy được định nghĩa

Đây chính là sai số giữa giá trị mong muốn và giá trị ước tính thông qua mạng nơ ron. Với cách tính như trên, ví dụ, khi t = 1, khoảng cách Entropy sẽ là giá trị này lớn hơn khi chưa tiến đến 1, khi z = 1, khoảng cách Entropy sẽ bằng 0. Vì vậy, khoảng cách Entropy sẽ ảnh hưởng lên quá trình học của mạng nơ ron (điều kiện khi lỗi bão hòa xuất hiện).

Giá trị OFFSET góp phần làm cho việc học mạng tốt hơn cũng như góp phần làm tăng khả năng ngăn chặn lỗi bão hòa xảy ra.

Để tính toán giá trị cho toàn bộ các nút của lớp ta có công thức sau:

Tương tự tại nút xuất của mạng, các nút ẩn hàm ESP cũng được đưa vào, khi đó ta có giá trị tại các nút xuất như sau:

với c là hằng số dương nhỏ hơn 1 (thường là 0.01).

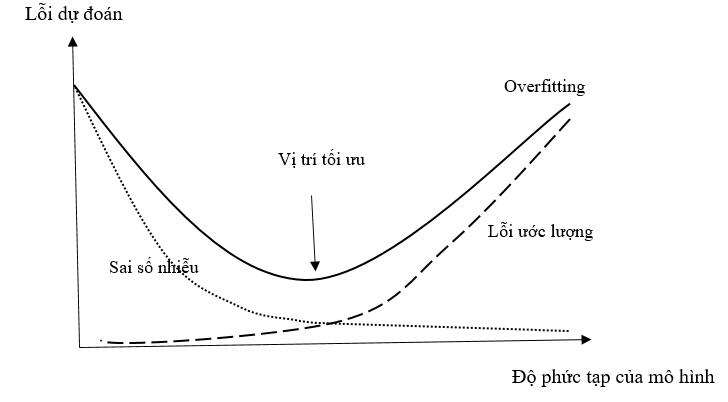
Phần quan trọng trong công thức của phương pháp ngăn lỗi bão hòa là việc lựa chọn các giá trị hằng số . Sau khi phân tích sự phân tán của dữ liệu thực nghiệm thì thường có giá trị trong khoảng từ (0,1). Việc xác định các giá trị được tổng kết như sau:

* Khi có sự hội tụ của việc học không rõ ràng, thì cần giảm giá trị n xuống và tăng lên.
* Khi sự hội tụ tốt nhưng thiếu khả năng tổng quát thì cần giảm giá trị để làm giảm ảnh hưởng hàm ESP, nếu vẫn còn hiện tượng trên thì tăng giá trị .

### Overfitting trong mạng nơ ron nhân tạo

Trong thống kê và học máy, một trong những nhiệm vụ phổ biến nhất đó chính là đưa ra một mô hình phù hợp với dữ liệu huấn luyện, từ đó có thể đưa ra các dự đoán đáng tin cậy hơn đối với các dữ liệu chưa được huấn luyện. Trong overfitting, một mô hình thống kê mô tả lỗi ngẫu nhiên hoặc độ nhiễu thay vì mối quan hệ cơ bản. Overfitting xảy ra khi mô hình quá phức tạp, chẳng hạn như có quá nhiều tham số so với số lần huấn luyện. Mô mô hình bị overfit có hiệu suất dự đoán kém, vì nó phản ứng mạnh với các biến động nhỏ trong dữ liệu huấn luyện.

Hình sau đây mô tả vì sao overfitting lại xảy ra. Trong lỗi dự đoán, có 2 thành phần chính đấy là sai số nhiễu và lỗi ước lượng. Sai số nhiễu là sai số hệ thống do sự can thiệp không định dạng trong dữ liệu vì mô hình hiệu chuẩn không đủ độ phức tạp để nắm bắt tất cả các nhiễu của mối quan hệ giữa kết quả phản hồi từ mô hình và kết quả mong muốn. Sai số ước lượng là do mô hình hóa các loại nhiễu ngẫu nhiên được đo lường khác nhau. Dự đoán tối ưu thu được khi lỗi nhiễu và sai số ước lượng cân bằng nhau, đó chính là điểm tối ưu được đánh dấu ở trong hình vẽ. Lỗi dự đoán tăng lên do một mô hình quá phức tạp được chính là overfitting.



Hình 2.12. Biểu đồ diễn tả overfitting.

Overfitting là một vấn đề lớn trong mạng nơ ron. Điều này đặc biệt đúng trong các mạng hiện đại, thường có số lượng rất lớn các trọng số và ngưỡng của mỗi nơ ron. Để huấn luyện có hiệu quả, ta cần phải có một cách để phát hiện khi overfitting đang diễn ra.

Cách rõ ràng để phát hiện overfitting đó chính là sử dụng phương pháp theo dõi dữ liệu thử nghiệm trên mạng nơ ron mà ta đã xây dựng. Nếu thấy tính chính xác dữ liệu không còn được cải thiện, thì ta nên ngừng huấn luyện. Tất nhiên, đây không hoàn toàn là một dấu hiệu của overfitting. Có thể tính chính xác dữ liệu của dữ liệu thử nghiệm và dữ liệu huấn luyện đều ngừng cùng một thời điểm. Tuy nhiên, việc áp dụng chiến lược này sẽ ngăn chặn việc overfitting. Vậy sau khi nhận diện được overfitting, điều ta cần đó là giải quyết vấn đề overfitting và có một giải pháp để có thể giải quyết được vấn đề đó trong mạng nơ ron đó chính là regulazation.

# Phương pháp phân tích thành phần chính

## Giới thiệu

Trong quá trình học máy hay cụ thể hơn là trong việc huấn luyện của mạng nơ ron, chúng ta sẽ gặp phải một trường hợp đó là số chiều của dữ liệu đầu vào quá lớn. Điều này ảnh hưởng không nhỏ đến quá trình học, bởi khi gặp vấn đề này, quá trình học sẽ bị chậm, dẫn đến việc để có kết quả cuối thì sẽ mất khá nhiều thời gian để tính toán. Vậy vấn đề nảy sinh đó là khi gặp vấn đề quá lớn như vậy làm cách nào để tìm ra không gian có số chiều ít hơn. Chính vì lý do đó phương pháp phân tích thành phần chính đã ra đời.

Mục tiêu của phương pháp phân tích thành phần chính đó chính là giảm số chiều của một tập véc tơ sao cho vẫn đảm bảo được tối đa những thông tin quan trọng nhất. Quá trình này trong học máy người ta gọi là trích rút dữ liệu (feature extraction).

## Cơ sở toán học

### Thống kê

#### Độ lệch chuẩn

Trong thống kê, độ lệch chuẩn là một đại lượng thống kê mô tả dùng để đo mức độ phân tán của một tập dữ liệu đã được lập thành bảng tần số. Có thể tính ra độ lệch chuẩn bằng cách tính tổng các bình phương của khoảng cách mỗi điểm dữ liệu đến giá trị trung bình, sau đó chia cho số lượng các phần tử và cuối cùng là lấy căn bậc hai của giá trị vừa tìm được. Khi hai tập dữ liệu có cùng giá trị trung bình cộng, tập nào có độ lệch chuẩn lớn hơn là tập có dữ liệu biến thiên nhiều hơn. Trong trường hợp hai tập dữ liệu có giá trị trung bình cộng không bằng nhau, thì việc so sánh độ lệch chuẩn của chúng không có ý nghĩa. Độ lệch chuẩn còn được sử dụng khi tính sai số chuẩn. Khi lấy độ lệch chuẩn chia cho căn bậc hai của số lượng quan sát trong tập dữ liệu, sẽ có giá trị của sai số chuẩn.

Công thức tính độ lệch chuẩn:

trong đó: là tần số của dữ liệu thứ của tập dữ liệu, là giá trị trung bình của tập dữ liệu, n là số lượng phần tử của tập dữ liệu.

#### Phương sai

Phương sai là một phương pháp khác trong thống kê để đo mức độ phân tán của tập dữ liệu. Trong thực tế, nó gần như giống với độ lệch chuẩn:

Công thức tính phương sai:

Ta nhận thấy rằng, phương sai chỉ đơn giản là bình phương của độ lệch chuẩn. Cả hai đại lượng đều là thước đo cho sự phân tán dữ liệu. Độ lệch chuẩn là thước đo phổ biến nhất, nhưng phương sai cũng được sử dụng. Lý do là vì phương sai cung cấp một nền tảng cho phần tiếp theo, hiệp phương sai.

#### Hiệp phương sai

Hai đại lượng ở trên đều xem xét ở dữ liệu một chiều. Các bộ dữ liệu này có thể là chiều cao tất cả mọi người trong phòng hay cân nặng của các sinh viên trong một lớp. Tuy nhiên, nhiều bộ dữ liệu có nhiều hơn một chiều, và mục đích của việc phân tích thống kê tập dữ liệu này xem liệu có bất kỳ quan hệ nào giữa các kích thước. Ví dụ: chúng ta có cả chiều cao của tất cả sinh viên và cân nặng của họ. Sau đó, ta sử dụng phân tích thống kê để xem chiều cao có ảnh hưởng đến điểm thi môn lập trình hay không. Độ lệch chuẩn và phương sai chỉ sử dụng cho trường hợp có một chiều. do đó chỉ có thể tính từng chiều dữ liệu độc lập chứ không thể kết hợp cả hai lại được. Vậy nên để giải quyết vấn đề này, người ta sử dụng hiệp phương sai. Hiệp phương sai là một phép đo giữa 2 chiều dữ liệu. Nếu ta tính hiệp phương sai của một chiều và chính nó thì ta được kết quả đó chính là phương sai.

Nếu ta xét 2 chiều dữ liệu có ký hiệu là X và Y thì phép đo hiệp phương sai có thể được viết như sau:

Trở lại ví dụ ta nhắc đến ở trên

* Đặt A là tập chiều cao sinh viên
* Đặt B là tập cân nặng tương ứng của sinh viên
* Vậy ta có hiệp phương sai của trường hợp này là:

Nếu thì cả hai kích thước tăng cùng nhau nghĩa là sinh viên càng cao thì có cân nặng càng lớn.

Nếu thì cả hai kích thước đối nghịch nhau nghĩa là sinh viên càng cao thì có cân nặng càng thấp.

Nếu thì cả hai kích thước là độc lập, không phụ thuộc lẫn nhau.

Từ đó ta thấy được tác dụng của hiệp phương sai đó là tìm ra sự liên kết giữa các chiều của tập dữ liệu.

#### Ma trận hiệp phương sai

Hiệp phương sai là một đại lượng luôn được xét trên 2 chiều dữ liệu. Nếu như ta có một tập dữ liệu có trên hai chiều dữ liệu, chúng ta sẽ có nhiều hơn một hiệp phương sai được đưa ra. Ví dụ, từ 3 chiều của tập dữ liệu (chiều x, y, z) ta có thể tính toán ra được , , . Trong thực tế, với n chiều dữ liệu thì ta có thể tính ra được hiệp phương sai khác nhau.

Cách hữu dụng nhất để lấy toàn bộ các giá trị hiệp phương sai khác nhau về chiều đó là tính toán toàn bộ các giá trị này và đưa chúng vào trong một ma trận. Và ma trận đó được gọi là ma trận hiệp phương sai. Ta có công thức cho ma trận hiệp phương sai với dữ liệu có n chiều như sau:

với là ma trận với n dòng và n cột, là tập dữ liệu chiều thứ i.

Ví dụ với 3 chiều dữ liệu thì ma trận hiệp phương sai của ta sẽ là:

### Ma trận

Ở phần này sẽ cung cấp những kiến thức cơ bản về ma trận được sử dụng trong thuật toán phân tích thành phần chính. Đặc biệt, đó là hai khái niệm véc tơ riêng (eigenvectors) và giá trị riêng (eigenvalues) của ma trận cho trước.

#### Véc tơ riêng (Eigenvectors)

Như đã biết, ta có thể nhân được hai ma trận với nhau, miễn là chúng có thể tương thích về kích cỡ với nhau. Véc tơ riêng là một trường hợp đặc biệt của phép nhân ma trận. Xét ví dụ sau:

Ở phép toán thứ nhất, véc tơ kết quả không phải là tích của một số nguyên với véc tơ gốc, còn ở phép toán thứ 2, véc tơ kết quả là tích của 4 lần véc tơ gốc. Phép toán thứ 2 có kết quả như vậy bởi véc tơ là véc tơ có hai chiều, véc tơ này được biểu diễn bởi một đường mũi tên từ điểm (0,0) đến điểm (3,2). Nếu nhân ma trận này với một ma trận chuyển đổi, ta sẽ nhận được một véc tơ khác được biến đổi từ vị trí gốc của nó.

Đó là bản chất của sự chuyển đổi mà các véc tơ riêng sinh ra. Hãy tưởng tượng một ma trận chuyển đổi, khi nhân với nhân với một véc tơ bất kì. Sau đó, ta thấy một véc tơ khác nằm cùng trên đường thẳng của véc tơ là hình ảnh phản chiếu của véc tơ . Véc tơ này sẽ là véc tơ riêng của ma trận biến đổi.

Một đặc trưng của véc tơ riêng đó là véc tơ riêng chỉ có thể tìm thấy ở các ma trận vuông, không phải mọi ma trận vuông đều có các véc tơ riêng. Ví dụ, ta xét một ma trận biến đổi có véc tơ riêng thì véc tơ riêng của ma trận này sẽ có véc tơ riêng.

Xét ví dụ sau đây:

Một đặc trưng khác của véc tơ riêng đó là nếu trước khi nhân véc tơ riêng với ma trận biến đổi, ta tăng kích cỡ của véc tơ đó lên một lượng cụ thể. Ta vẫn nhận được kết quả là véc tơ đó khi tăng kích cỡ. Bởi ta chỉ tăng kích cỡ của véc tơ còn về hướng thì không thay đổi.

Đặc trưng cuối cùng của véc tơ riêng đó là tất cả các véc tơ riêng của ma trận đều vuông góc với nhau.

Một điều quan trọng trong việc tìm véc tơ riêng, đó chính là khi tìm véc tơ riêng, người ta luôn muốn tìm các véc tơ có độ dài chính xác là 1. Điều này là do, chiều dài của một véc tơ không ảnh hưởng đến nó có phải là một véc tơ riêng hay không. Vì vây, để giữa các véc riêng chuẩn, ta thường quy về để véc tơ có chiều dài là 1, để cho tất cả các véc tơ riêng có cùng chiều dài. Sau đây là một ví dụ cụ thể:

Như đã xét ở trên là một véc tơ riêng và độ dài của véc tơ là

Vậy để véc tơ riêng có độ dài là 1 ta chỉ cần chia cả hai chiều của véc tơ với độ dài của véc tơ

#### Giá trị riêng (Eigenvalues)

Giá trị riêng có mối quan hệ rất chặt chẽ với các véc tơ riêng.

Xét ví dụ sau đây:

Như đã xét ở phần 3.1.2.1, chính là một véc tơ riêng còn số 2 chính là giá trị riêng của véc tơ riêng này.

Vì vậy, ta có thể thấy rằng, các véc tơ riêng và các giá trị riêng luôn đi theo một cặp. Trong một bài toán yêu cầu về tìm véc tơ riêng, ta sẽ luôn tìm được giá trị riêng đi kèm với véc tơ đó.

## Các bước thực hiện phương pháp

### Tạo tập dữ liệu

Ở đây ta sẽ sử dụng một tập ví dụ, từ đó thể hiện các bước thực hiện một cách rõ ràng nhất. Dữ liệu ví dụ này sẽ chỉ có 2 chiều để thuận tiện cho việc tính toán

Dữ liệu sẽ được gồm có 2 chiều dữ liệu như sau:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 2.5 | 0.5 | 2.2 | 1.9 | 3.1 | 2.3 | 2 | 1 | 1.5 | 1.1 |
| y | 2.4 | 0.7 | 2.9 | 2.2 | 3.0 | 2.7 | 1.6 | 1.1 | 1.6 | 0.9 |

Bảng 3.1. Bảng dữ liệu mẫu phương pháp phân tích thành phần chính

### Trừ cho giá trị trung bình

Để phương pháp phân tích thành phần chính hoạt động bình thường ta phải lấy mỗi giá trị của từng chiều dữ liệu trừ giá trị trung bình của chiều tương ứng. Ở vi dụ này ta sẽ được kết quả như sau

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0.69 | -1.31 | 0.39 | 0.09 | 1.29 | 0.49 | 0.19 | -0.81 | -0.31 | -0.71 |
| y | 0.49 | -1.21 | 0.99 | 0.29 | 1.09 | 0.79 | -0.31 | -0.81 | -0.31 | -1.01 |

Bảng 3.2. Bảng dữ liệu sau khi trừ đi giá trị trung bình

### Tính toán ma trận hiệp phương sai

Ta sử dụng trực tiếp công thức về ma trận hiệp phương sai như đã trình bày ở phần 3.1.1.4. Bởi dữ liệu có 2 chiều vậy nên ma trận hiệp phương sai sẽ có số chiều là 2 x 2. Sau các bước tính toán ta có được kết quả sau:

Do yếu tố chéo nhau của ma trận này là tích cực, vậy nên 2 biến của ma trận này là đồng biến (cùng tăng).

### Tính toán véc tơ riêng và giá trị riêng của ma trận hiệp phương sai

Do ma trận hiệp phương sai là có số chiều bằng số cột nên chúng ta có thể tính toán được véc tơ riêng và giá trị riêng cho ma trận này.

Lập phương trình tính đặc trưng của ma trận . Đặt giá trị riêng của ma trận này là V

Ta tính được giá trị giá trị riêng và véc tơ riêng của ma trận hiệp phương sai là:

Véc tơ riêng này đã đạt được yêu cầu bởi nó đã có độ dài là 1.

Bằng quá trình lấy các véc tơ riêng của ma trận hiệp phương sai, ta có thể trích xuất được dòng đặc trưng của dữ liệu. Các bước còn lại chủ yếu liên quan đến việc chuyển đổi dữ liệu sao cho nó được biểu hiện dưới dạng đường thẳng.

### Chọn các thành phần và tạo ra một véc tơ đặc trưng

Bước này của quá trình phân tích thành phần chính chính là bước để nén dữ liệu và giảm chiều của tập dữ liệu. Với giá trị riêng và véc tơ riêng tính toán được ở trên, ta thấy mỗi véc tơ riêng lại có một giá trị riêng khác nhau. Trong thực tế, các đặc trưng riêng của tập dữ liệu chính là các véc tơ riêng có giá trị riêng là cao nhất, và nó được gọi là thành phần chính của bộ dữ liệu.

Nói chung, một khi các véc tơ riêng được tìm thấy ma trận hiệp phương sai, bước tiếp theo là sắp xếp chúng theo thứ tự của giá trị riêng từ cao đến thấp. Điều này cho ta thấy các thành phần chính quan trọng của bộ dữ liệu. Bây giờ, ta có thể bỏ đi các thành phần có giá trị riêng nhỏ hơn. Đương nhiên khi ta loại bỏ các thành phần này, ta sẽ tạo ra sai số nhưng do giá trị nhỏ nên mức sai số không quá lớn. Sau khi loại bỏ, ta sẽ có một tập dữ liệu có kích thước nhỏ hơn so với tập gốc. Nếu ban đầu có n chiều trong tập dữ liệu, thì ta sẽ tính được n véc tơ riêng và n giá trị riêng tương ứng, sau đó ta chỉ chọn p véc tơ riêng đầu tiên, thì tập dữ liệu cuối cùng sẽ có kích thước là p. Tập p lúc này sẽ được gọi là véc tơ đặc trưng.

Trở lại với ví dụ ta đang xét, trong thực tế chúng ta có 2 véc tơ riêng tương ứng với 2 sự lựa chọn. Sau đây là 2 véc tơ riêng là kết quả của các bước trước đó, nhưng đã được sắp xếp theo thứ tự giảm dần của giá trị riêng

Ta sẽ loại bỏ véc tơ riêng có giá trị riêng nhỏ, và ta sẽ thu được một thành phần đấy chính là véc tơ đặc trưng của tập dữ liệu:

### Lấy ra bộ dữ liệu mới

Bước cuối cùng của phương pháp phân tích thành phần chính, là bước dễ nhất. Khi chúng ta đã tìm được véc tơ đặc trưng, chúng ta có thể dễ dàng chuyển vị ma trận chứa véc tơ này và nhân chúng với dữ liệu từ tập gốc cũng đã được chuyển vị.

với là ma trận véc tơ đặc trưng đã được chuyển vị, là ma trận thể hiện dữ liệu sau khi trừ đi giá trị trung bình và cũng đã được chuyển vị, là ma trận thể hiện dữ liệu mà ta cần tìm kiếm, dữ liệu đã được giảm chiều.

## Lấy lại giá trị cũ

Đây cũng là một vấn đề được những người sử dụng phương pháp phân tích thành phần chính khá quan tâm. Ta chỉ có thể lấy lại giá trị ban đầu nếu như ta có đầy đủ số lượng véc tơ riêng ban đầu. Bởi vì ta giảm số lượng của véc tơ riêng ở lần chuyển đổi cuối cùng nên dữ liệu ban đầu sẽ mất đi một số thông tin.

Ở phần 3.2.1.6 ta đã nhận được công thức tính được giá trị cuối cùng sau khi giảm chiều

Vậy để lấy lại được dữ liệu ban đầu, ta sẽ có công thức sau:

Nhưng là ma trận thể hiện dữ liệu sau khi trừ đi giá trị trung bình đã được chuyển vị. Vậy nên để lấy lại dữ liệu ban đầu ta cần phải cộng thêm một thêm giá trị trung bình ban đầu đã trừ đi

với là dữ liệu gốc, là giá trị trung bình gốc.

Công thức này được áp dụng khi không có đầy đủ véc tơ riêng ban đầu mà chỉ có ma trận đặc trưng chứa các véc tơ riêng có giá trị riêng cao nhất. Vì vậy, ngay cả khi không có đầy đủ các véc tơ riêng ban đầu, phương trình trên vẫn đem lại cho ta một kết quả chính xác của dữ liệu gốc.

# Ứng dụng mạng nơ ron để xây dựng hệ thống nhận diện khuôn mặt

## Phát biểu bài toán

Khuôn mặt là trọng tâm chính của sự chú ý trong xã hội. Nó đóng một vai trò quan trọng trong việc truyền đạt cảm xúc. Mỗi một con người có thể nhận dạng được khuôn mặt của hàng ngàn người khác nhau trong suốt cuộc đời và xác định khuôn mặt quen thuộc ngay cả sau nhiều năm không gặp gỡ. Khả năng này trong não bộ con người là khá mạnh, mặc dù có sự thay đổi lớn về kích thích thị giác do điều kiện nhìn, biểu hiện, độ lão hóa và những thay đổi về mặt hình thức của người được nhận dạng như thay đổi kiểu tóc, để râu hoặc đeo kính. Nhận dạng khuôn mặt đã trở thành một vấn đề quan trọng trong nhiều ứng dụng của hệ thống an ninh, xác minh thẻ tín dụng, nhận diện hình sự… Ngay cả khả năng chỉ nhận dạng khuôn mặt, trái với việc nhận ra chúng cũng đã là một vấn đề khá quan trọng. Nhận dạng khuôn mặt đã được nghiên cứu trong suốt 20 năm trở lại đây. Phát triển một mô hình tính toán nhận dạng ra khuôn mặt khá là khó khăn bởi vì khuôn mặt phức tạp, kích thích thị giác đa chiều. Do đó, nhận dạng khuôn mặt là một tác vụ khó dành cho máy tính.

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, các công nghệ mới dần dần ra đời trong đó có mạng nơ ron. Ta có thể xây dựng được được một công cụ cho phép nhận diện khuôn mặt của con người dựa trên tập đầu vào đó là tập các ảnh đã có sẵn với định danh đầy đủ của mỗi cá nhân là tên của thư mục chứa danh sách ảnh.

## Giải quyết bài toán

### Tập dữ liệu sử dụng

Bài toán nhận dạng khuôn mặt thông thường sẽ có hai bước chính: Dò tìm khuôn mặt và sau đó là nhận diện khuôn mặt. Nhưng ở phạm vi của khóa luận này, ta chỉ xét đến bước thứ hai đó là nhận diện khuôn mặt. Vậy nên, việc lựa chọn tập dữ liệu như nào cũng là một vấn đề cần phải quan tâm. Tập dữ liệu sử dụng cần được cắt sẵn khuôn mặt người, ảnh đứng và có chung kích cỡ với nhau.

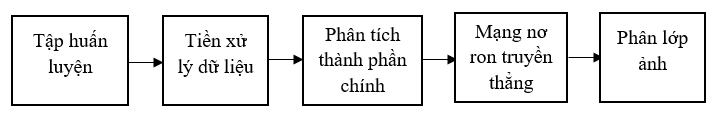
Tôi đã lựa chọn một tập ảnh có tên “The ORL Database of Faces” [5] chứa khuôn mặt của 40 người bởi nó đáp ứng đủ các yêu cầu đặt ra. Cơ sở dữ liệu này đã từng được sử dụng để làm tập dữ liệu đầu vào cho một án nhận diện khuôn mặt của Đại học Cambridge. Có 10 hình ảnh khuôn mặt khác nhau cho mỗi đối tượng. Đối với mỗi đối tượng, hình ảnh được chụp vào những thời điểm khác nhau, thay đổi ánh sáng, biểu hiện khuôn mặt, các chi tiết trên khuôn mặt. Tất cả các hình ảnh được chụp thẳng đứng và chuyển sẵn về dạng màu tối. Sau đây là hình ảnh về tập ảnh này:



Hình 4.4.1. Hình ảnh khuôn mặt có trong cơ sở dữ liệu mẫu

Với mỗi đối tượng, tôi sẽ sử dụng 8 ảnh để đưa vào tập dữ liệu huấn luyện và 2 ảnh vào tập dữ liệu kiểm tra. Cụ thể, ta sẽ chọn 8 ảnh đầu tiên của mỗi người đưa vào tập huấn luyện và 2 ảnh cuối cùng sẽ đưa vào tập kiểm tra. Vậy mục tiêu hướng đến của ứng dụng chính là xác định được một ảnh trong tập dữ liệu kiểm tra chứa mặt của người nào.

### Sơ đồ luồng bước huấn luyện



Hình 4.2. Sơ đồ luồng bước huấn luyện hệ thống nhận diện khuôn mặt

#### Tiền xử lý dữ liệu

Quá trình tiền xử lý đối với khuôn mặt nhằm chuẩn hóa dữ liệu và kích thước ảnh. Các ảnh ở trong cơ sở dữ liệu của ORL đã là các ảnh có kích thước ảnh bằng nhau và chuẩn hóa sẵn ảnh theo kiểu greyscale – một hệ thống màu có mô hình đơn giản nhất với 256 cấp độ xám biến thiên từ màu đen đến màu trắng.

Ta sử dụng ảnh greyscale trong bước tiền xử lý dữ liệu bởi một bức ảnh dạng greyscale có thể biểu diễn dưới dạng một ma trận hai chiều, bởi mỗi điểm ảnh chỉ được biểu diễn bởi một kênh màu đơn.

Nếu như tập dữ liệu ban đầu là tập các ảnh dạng RGB – một hệ thống màu với 3 màu sắc cơ bản đỏ, xanh dương và xanh lục, mỗi điểm ảnh sẽ được biểu diễn bằng 3 kênh màu này với các giá trị từ 0 đến 255, thì ta cần chuyển về dạng greyscale thông qua công thức:

trong đó:

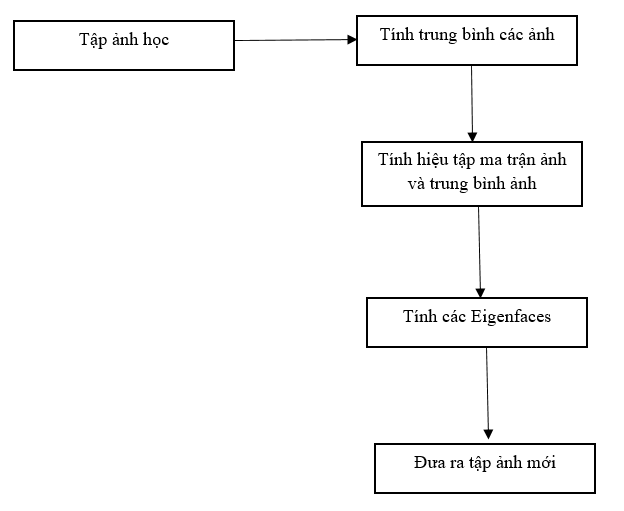
* **:** Là cường độ sáng tại điểm ảnh (x, y) của ảnh greyscale.
* : Là giá trị của kênh màu Red(Đỏ) tại điểm ảnh (x, y) của ảnh màu(RGB).
* **:** Là giá trị của kênh màu Green(Xanh lá cây) tại điểm ảnh (x, y) của ảnh màu(RGB).
* **:**Là giá trị của kênh màu Blue(Xanh lơ) tại điểm ảnh (x, y) của ảnh màu(RGB).

#### Phân tích thành phần chính

Mỗi bức ảnh trong cơ sở dữ liệu ban đầu đều có kích cỡ là 92 x 112 pixels. Nghĩa là khi đưa vào trong mạng nơ ron, ta cần số lượng nơ ron đầu vào là 10.304. Đây là một số lượng nơ ron trong một lớp khá lớn, điều này sẽ làm ảnh hưởng đến thời gian quá trình huấn luyện mạng nơ ron. Do đó mỗi hệ thống sử dụng mạng nơ ron làm trung tâm đều phải có một cơ chế để trích rút đặc trưng của dữ liệu.

Trong bài toán này, cơ chế trích rút dữ liệu sẽ được sử dụng để trích rút ra những điểm chính trên khuôn mặt của mỗi người. Từ đó, giảm chiều dữ liệu đầu vào của mạng nơ ron, giúp cho quá trình huấn luyện mạng trở nên nhanh chóng hơn. Và phương pháp trích rút dữ liệu được sử dụng ở đây đó chính là phương pháp phân tích thành phần chính bởi nó đáp ứng đủ yêu cầu của quá trình trích rút dữ liệu đó là giảm chiều tập dữ liệu mẫu.

Quá trình trích rút đặc trưng sẽ được thể hiện qua biểu đồ sau:



Hình 4.3. Sơ đồ khối trích rút đặc trưng

* Tính trung bình các ảnh

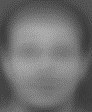
Mỗi ảnh trong tập ảnh học sẽ được biểu diễu dưới dạng véc tơ với i là ảnh thứ i trong tập ảnh học. Véc tơ có số chiều là N.

Tập ảnh ban đầu sẽ được biểu diễn dưới dạng:

với M là số lượng ảnh trong tập ảnh học.

Ảnh trung bình sẽ được tính bằng trung bình cộng của tất cả các ảnh đã được ma trận hóa. Bước ảnh trung bình này sẽ thể hiện ra được sự giống nhau nhất giữa các khuôn mặt.

Kết quả thực hiện:



Hình 4.4. Ảnh trung bình của tập dữ liệu học

* Tính hiệu tập ma trận ảnh và trung bình ảnh

Gọi là hiệu của ảnh thứ i trong tập huấn luyện và ma trận trung bình ảnh. Ta có:

* Tính toán eigenfaces

Eigenface chính là các véc tơ riêng của ma trận hiệp phương sai được lập ra từ tập dữ liệu các ảnh ứng với các giá trị riêng có giá trị lớn nhất. Ta có công thức tính ma trận hiệp phương sai như sau: [6]

trong đó với (Số chiều của A là )

Ma trận hiệp phương sai C sẽ có số chiều là . Từ đây ta có thể lấy ra được N véc tơ riêng và giá trị riêng tương ứng. Ví dụ, đối với hình ảnh kích thước , ta phải tính toán ma trận hiệp phương sai có kích thước và tìm ra véc tơ riêng. Đây là cách tính toán không hiệu quả bởi, ta không cần hầu hết tất cả các véc tơ đó. Số lượng véc tơ riêng được giới hạn bởi số lượng hình ảnh trong tập học, vì vậy nếu chúng ta có M hình ảnh thì số lượng véc tơ riêng sẽ M-1 tương ứng với các giá trị riêng. Vậy để tính toán được giá trị riêng cho chuẩn xác, ta cần chú ý đến công thức sau:

Nếu và là véc tơ riêng và giá trị riêng của ma trận , ta có

Nhân cả hai vế với ma trận A ta có:

Ta lại có:

Vậy nên, để tính toán được véc tơ riêng của C, ta chỉ cần tính véc tơ riêng của A rồi nhân ma trận A

Sau khi tính toán được các véc tơ riêng của C, ta sẽ chọn ra K véc tơ có giá trị riêng lớn nhất. Đó chính là các eigenface.

Ví dụ: K=80. Kết quả thực hiện của phương pháp



Hình 4.5. Danh sách các eigenfaces

* Đưa ra tập ảnh mới

Ứng với mục 3.2.1.6 ta sẽ đưa ra được danh sách các ảnh đã được giảm số chiều. Công thức sau sẽ mô tả công thức để tính ma trận hình ảnh i mới.

Tập ảnh này sẽ được sử dụng để làm tập huấn luyện cho mạng nơ ron ở bước tiếp theo.

#### Xây dựng mạng nơ ron truyền thẳng

Đầu tiên, dựa vào bước trích rút dữ liệu trước đó, tập ảnh mới lúc này với mỗi ảnh được biểu diễn bằng một véc tơ 80 chiều. Vậy nên ta sẽ xây dựng một mạng nơ ron với K đầu vào và 40 đầu ra tương ứng với 40 người trong tập dữ liệu. Mỗi đầu ra tương ứng là một ma trận 1 chiều có giá trị bằng 0 hoặc 1 để biểu diễn, nếu đầu ra là người thứ i thì hàng thứ i trong ma trận sẽ bằng 1 còn lại là bằng 0. Ví dụ với đầu ra là người thứ nhất thì ma trận sẽ có phần tử đầu tiên là 1, các phần tử còn lại là 0.

Tiếp đó, ta xây dựng mạng nơ ron truyền thẳng gồm có 3 lớp. Một lớp input có 80 đầu vào. Một lớp đầu ra có 40 nơ ron.

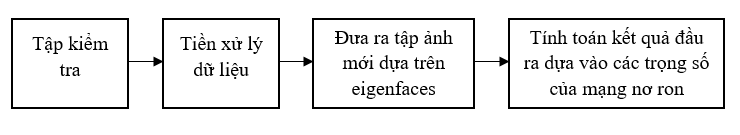
Có một điều khá khó khăn trong quá trình huấn luyện mạng đó chính là việc chọn số nơ ron lớp ẩn. Nếu số nơ ron lớp ẩn quá lớn hoặc quá nhỏ thì sẽ không đạt được kết quả như mong muốn. Sau khi chạy thử nghiệm nhiều lần với dữ liệu huấn luyện thì kết quả đạt được tốt nhất khi lựa chọn số nơ ron lớp ẩn là 180.

Một số thông số khác của mạng nơ ron:

* Tốc độ học: 0.5
* Số lần học (epochs): 100

Mạng nơ ron nhân tạo trong bài toán này sẽ sử dụng thuật toán lan truyền ngược để huấn luyện.

### Sơ đồ luồng bước nhận diện

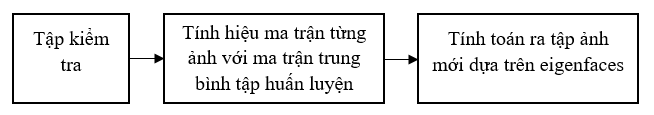


Hình 4.6. Sơ đồ luồng bước nhận diện hệ thống nhận diện khuôn mặt

#### Tiền xử lý dữ liệu

Ta thực hiện tương tự như ở tập huấn luyện tại mục 4.2.2.1

#### Đưa ra tập ảnh mới



Hình 4.7. Sơ đồ luồng tìm tập ảnh mới tại tập kiểm tra

* Tính hiệu ma trân từng ảnh với ma trận trung bình tập huấn luyện

Gọi tập ảnh kiểm tra là trong đó n là số ảnh trong tập kiểm tra, ma trận trung bình tập huấn luyện là và là hiệu của ảnh thứ i trong tập huấn luyện và ma trận trung bình ảnh.

Ta có:

Trong đó: đã được tính toán ở mục 4.2.2.2

* Tính toán ra tập ảnh mới dựa trên eigenfaces

Tương tự với bước đưa ra tập ảnh mới ở mục 4.2.2.2, gọi là ma trận tổng hợp các véc tơ riêng ta đã tính toán được.

Vậy giá trị mới của tập ảnh kiểm tra sẽ được tính toán như sau:

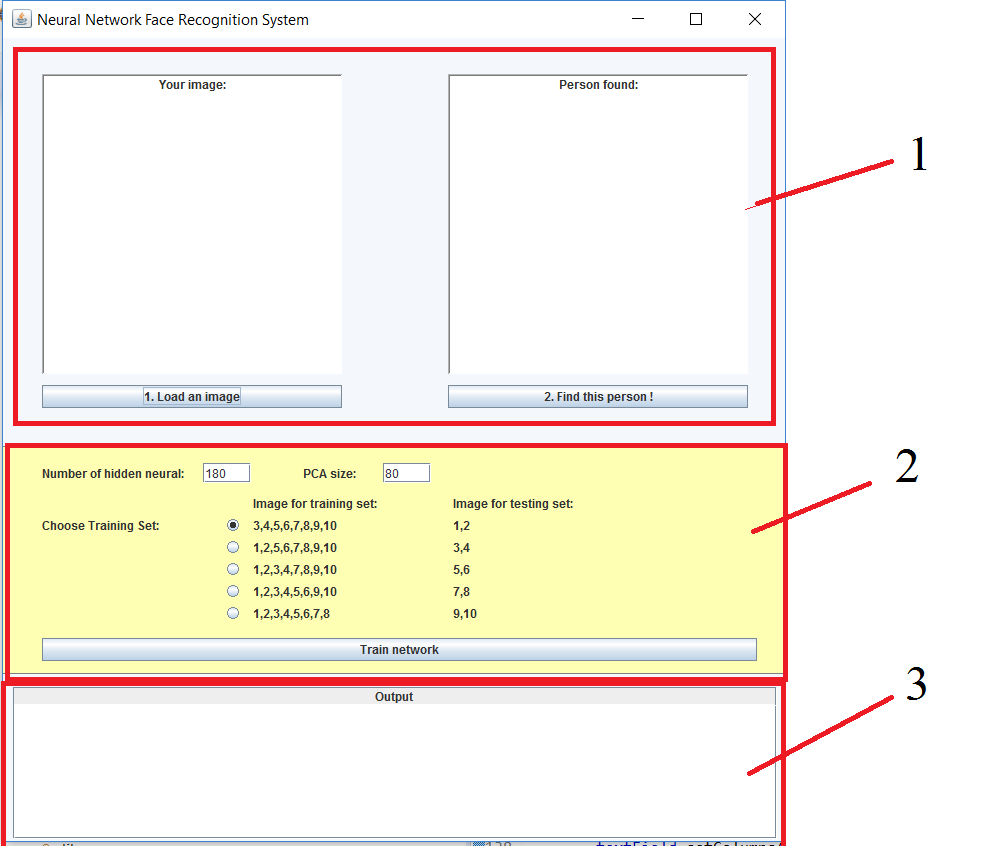
Trong đó, là ảnh thứ i của tập ảnh kiểm tra mới

#### Tính toán kết quả đầu ra dựa trên các trọng số của mạng nơ ron

Các ảnh trong tập kiểm tra mới được đưa trở lại mạng nơ ron truyền thẳng để tính toán. Nhưng ở lần này, ta chỉ thực hiện một nửa đầu của thuật toán lan truyền ngược đó là bước lan truyền tiến. Bởi sau quá trình huấn luyện mạng nơ ron, các trọng số đã được điều chỉnh ở mức ổn định và trong bước này ta sẽ giữ ổn định các trọng số rồi từ đó tính toán để đưa ra kết quả nhận dạng cuối cùng dựa vào đầu ra của mạng nơ ron.

## Mô tả về hệ thống

### Giao diện chương trình



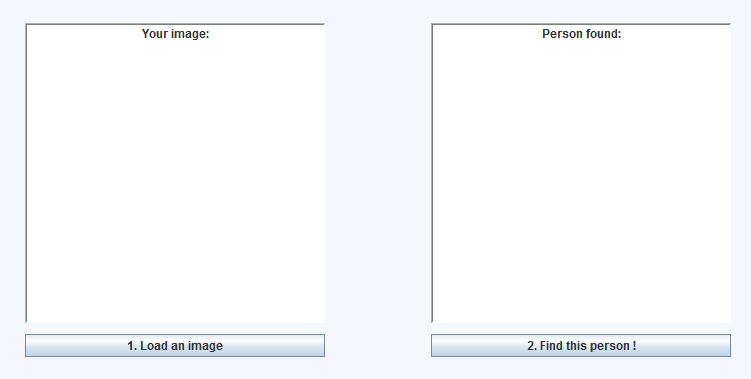
Hình 4.8. Giao diện giới thiệu chương trình

Giao diện chương trình được chia làm ba phần:

* Phần nhận diện (1)
* Phần huấn luyện (2)
* Phần hiển thị output, thành phần này là nơi hiển thị các kết quả khi huấn luyện của mạng nơ ron. (3)

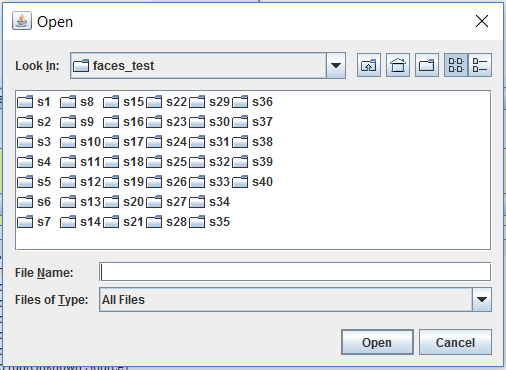
### Phần nhận diện

Phần này bao gồm 2 khung chứa hình ảnh và tương ứng là 2 button ở bên dưới biểu thị tác dụng hiển thị của mỗi khung chứa ảnh. Button thứ nhất “1. Load an image” để tải một bức ảnh từ trong tập kiểm tra để phục vụ cho nhận diện, khi chọn một hình ảnh, hình ảnh đó sẽ hiển thị lên bên trung khung “your image:”. Button thứ hai “2. Person this found !” có tác dụng nhận diện xem khuôn mặt đã được chọn ra ở phần “your image:” là khuôn mặt của ai và sau đó tải ảnh khuôn mặt của người đó và hiển thị ở khung “person found:”. Button thứ hai này chỉ có thể sử dụng trong trường hợp đã chọn một bức ảnh ở mục thứ nhất và đã huấn luyện dữ liệu với mạng nơ ron.



Hình 4.9. Hình ảnh phần nhận diện

Khi lựa chọn vào button “1. Load an image” cửa sổ chọn ảnh sẽ hiện ra, đường dẫn mặc định được trỏ đến chính là danh sách các ảnh trong tập dữ liệu:



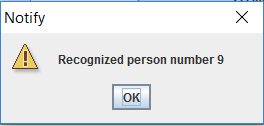
Hình 4.10. Chọn file ảnh để nhận diện

Sau khi ta chọn một bức ảnh, giao diện hình 4.7 thay đổi như sau:



Hình 4.11. Sau khi đã chọn ảnh để nhận diện

Khi huấn luyện mạng và chọn ảnh xong, button “2. Find this person!” sẽ hoạt động. Khi nhấn vào button này, một thông báo về người được nhận diện sẽ được hiện ra



Hình 4.12. Thông báo hiển thị người được nhận diện

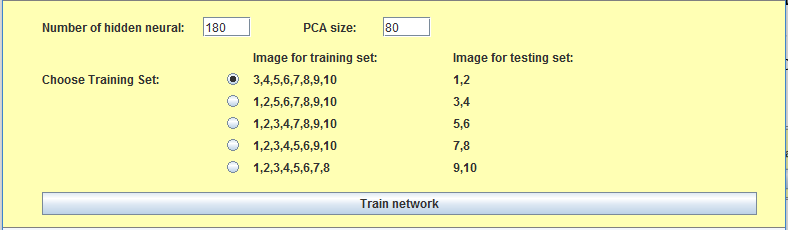
Cùng lúc đó, ảnh của người được nhận diện sẽ được hiển thị ở phần “Person found: ” , bức ảnh này là bức ảnh đầu tiên trong tập huấn luyện của người được nhận diện.



Hình 4.13. Ảnh người được nhận diện hiện ra

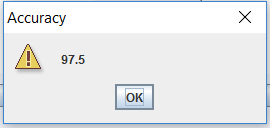
### Phần huấn luyện

Phần này bao gồm bốn thành phần chủ đạo. Thành phần thứ nhất là “Number of hidden neural: ”, đây là nơi để nhập số lượng của nút tại lớp ẩn của mạng nơ ron. Thành phần thứ hai là “PCA size: ”, đây là nơi để nhập số thành phần chính cần giữ lại sau phương pháp phân tích thành phần chính, đây cũng chính là số lượng nơ ron đầu vào cho mạng nơ ron. Thành phần thứ ba là “Choose Training Set: ”, ta sẽ lựa chọn ảnh cho tập huấn luyện và tập kiểm tra, trong đó các số từ 1 đến 10 biểu thị mỗi bức ảnh của một người. Thành phần thứ 4 là button “Train Network”, button này có tác dụng huấn luyện mạng dựa vào các thông số đã đưa ra ở phần 4.2.4. Ngoài ra, ta có thể thay đổi thông số của số nơ ron lớp ẩn và số lượng nơ ron đầu vào bằng cách thay đổi số mặc định đã cho cho trước ở 2 thành phần trước đó.



Hình 4.14. Phần huấn luyện mạng nơ ron

Khi kết thúc quá trình huấn luyện mạng nơ ron, một thông báo về đánh giá độ chính xác trên tập test được đưa ra. Độ chính xác này được tính toán dựa trên tỉ lệ giữa số ảnh nhận diện đúng bởi mạng nơ ron và tập kết quả đúng.



Hình 4.15. Thông báo hiển thị độ chính xác

## Kết quả thực nghiệm

Phần đầu tiên của kết quả thực nghiệm, tôi sẽ giải thích lý do vì sao tôi lại chọn số lượng thành phần chính được giữ lại trong phương pháp phân tích thành phần chính là 80, các thông số còn lại của mạng nơ ron giữ nguyên. Các kết quả độ chính xác sẽ được tính bằng tỉ lệ số khuôn mặt nhận diện đúng so với tổng số lượng ảnh tập kiểm tra. Ta sẽ có bảng thống kê sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Số lượng eigenfaces** | **Đánh giá nhận dạng chính xác (%)** | | | |
| **Lần 1** | **Lần 2** | **Lần 3** | **Trung bình** |
| 20 | 93.75 | 93.75 | 95 | 94.167 |
| 30 | 96.25 | 95 | 93.75 | 95.000 |
| 40 | 93.75 | 95 | 95 | 94.583 |
| 50 | 97.5 | 93.75 | 96.25 | 95.833 |
| 60 | 95 | 95 | 96.25 | 95.417 |
| 70 | 96.25 | 93.75 | 97.5 | 95.833 |
| 80 | 97.5 | 96.25 | 97.5 | 97.083 |
| 90 | 95 | 96.25 | 95 | 95.417 |

Bảng 4.1. Độ chính xác của nhận dạng dựa trên số lượng eigenfaces

Theo như bảng trên, ta thấy được, tại lần huấn luyện thứ 100, kết quả độ chính xác đo được là lớn nhất, 97.083% nhận diện chính xác khuôn mặt trong tập kiểm tra. Bởi vậy, ta lựa chọn số lượng thành phần chính là 80.

Sau đây là kết quả đánh giá nhận dạng khi sau 100 lần huấn, kết quả sẽ được ghi lại sau mỗi 10 bước huấn luyện toàn bộ cơ sở dữ liệu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lần huấn luyện** | 10 | 20 | 30 | 40 | 50 |
| **Kết quả lần 1 (%)** | 85 | 88.75 | 93.75 | 92.5 | 93.75 |
| **Kết quả lần 2 (%)** | 83.75 | 90 | 87.5 | 91.25 | 96.25 |
| **Kết quả lần 3 (%)** | 85 | 83.75 | 95 | 96.25 | 93.75 |
| **Trung bình (%)** | 84.583 | 87.5 | 92.083 | 93.333 | 94.583 |

Bảng 4.2. Kết quả độ chính xác 50 lần huấn luyện đầu tiên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lần huấn luyện** | 60 | 70 | 80 | 90 | 100 |
| **Kết quả lần 1 (%)** | 95 | 93.75 | 97.5 | 96.25 | 97.5 |
| **Kết quả lần 2 (%)** | 95 | 96.25 | 95 | 95 | 97.5 |
| **Kết quả lần 3 (%)** | 93.75 | 95 | 96.25 | 97.5 | 96.25 |
| **Trung bình (%)** | 94.583 | 95 | 96.25 | 96.25 | 97.083 |

Bảng 4.3. Kết quả độ chính xác từ lần huấn luyện 60 đến 100

Hình 4.16. Biểu đồ hiện thị kết quả độ chính xác

Với kết quả này, ta thấy, mô hình mạng nơ ron truyền thẳng được sử dụng trong bài toán nhận diện khuôn mặt có các thông số sau:

* Số lượng lớp: 3
* Số lượng nơ ron đầu vào: 80
* Số lượng nơ ron lớp ẩn: 180
* Số lượng nơ ron đâu ra: 40
* Số lần huấn luyện: 100
* Tốc độ học: 0.5
* Thuật toán học: Lan truyền ngược
* Thuật toán đánh giá: Đánh giá dựa trên tỉ lệ nhận dạng đúng và tập mẫu.

Mô hình này không quá phức tạp, nhưng lại đạt hiệu suất khá tốt trong quá trình học và nhận diện khuôn mặt ở tập kiểm tra. Mô hình này tốt trong quá trình học bởi kết quả độ chính xác tăng dần lên sau mỗi lần học và đã tăng từ 84.584% ở 10 lần đầu tiên để đạt đến 97.083% ở lần huấn luyện thứ 100.

Bên cạnh kết quả đã đạt được này, ta cần sử dụng một phương pháp đánh giá để tăng tính thuyết phục cho mô hình, thay vì hiện tại ta đang sử dụng cố định là 8 ảnh đầu tiên của mỗi người sẽ đưa vào tập huấn luyện còn 2 ảnh cuối sẽ đưa vào tập kiểm tra. Ta sẽ sử dụng một phương pháp có tên là cross-validation, đây là phương pháp phổ biến trong học máy để đánh giá tính chính xác của mô hình được sử dụng. Phương pháp này phân chia dữ liệu thành k tập con có cùng kích thước. Tại mỗi vòng lặp sử dụng một tập con là tập thử nghiệm và các tập còn lại là tập huấn luyện. Như vậy, thay vì sử dụng một phần dữ liệu làm tập dữ liệu học thì phương pháp này dùng toàn bộ dữ liệu để dạy cho máy. Sau khi đánh giá kết quả của từng vòng lặp kết thúc, kết quả đánh giá cuối cùng sẽ trung bình cộng của các lần đánh giá trong vòng lặp.

Áp dụng phương pháp đó vào tập ảnh đang xét, ta đang sử dụng 2 ảnh để đưa vào tập kiểm tra, vậy nên ta sẽ có 5 lần lặp như sau:

|  |  |  |
| --- | --- | --- |
| Lần lặp | Ảnh đưa vào tập huấn luyện | Ảnh đưa vào tập kiểm tra |
| 1 | 3, 4, 5, 6, 7, 8, 9, 10 | 1, 2 |
| 2 | 1, 2, 5, 6, 7, 8, 9, 10 | 3, 4 |
| 3 | 1, 2, 3, 4, 7, 8, 9, 10 | 5, 6 |
| 4 | 1, 2, 3, 4, 5, 6, 9, 10 | 7, 8 |
| 5 | 1, 2, 3, 4, 5, 6, 7, 8 | 9, 10 |

Bảng 4.4. Các lần lặp của phương pháp cross-validation

Sau đây là kết quả tính toán của từng lần lặp một, ta thực hiện đầy đủ các bước từ tiền xử lý, trích rút dữ liệu và huấn luyện với mạng nơ ron như đã trình bày ở trên.

* Lần lặp thứ nhất:

|  |  |
| --- | --- |
| **Lần huấn luyện** | 100 |
| **Kết quả lần 1 (%)** | 97.5 |
| **Kết quả lần 2 (%)** | 98.75 |
| **Kết quả lần 3 (%)** | 100 |
| **Trung bình (%)** | 98.75 |

Bảng 4.5. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 1

Lần lặp thứ nhất cho kết quả độ chính xác là 98.75%

* Lần lặp thứ hai:

|  |  |
| --- | --- |
| **Lần huấn luyện** | 100 |
| **Kết quả lần 1 (%)** | 96.25 |
| **Kết quả lần 2 (%)** | 98.75 |
| **Kết quả lần 3 (%)** | 100 |
| **Trung bình (%)** | 98.33 |

Bảng 4.6. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 2

Lần lặp thứ hai cho kết quả độ chính xác là 98.33%

* Lần lặp thứ ba:

|  |  |
| --- | --- |
| **Lần huấn luyện** | 100 |
| **Kết quả lần 1 (%)** | 100 |
| **Kết quả lần 2 (%)** | 100 |
| **Kết quả lần 3 (%)** | 98.75 |
| **Trung bình (%)** | 99.58 |

Bảng 4.7. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 3

Lần lặp thứ 3 cho kết quả độ chính xác là 99.58%

* Lần lặp thứ tư:

|  |  |
| --- | --- |
| **Lần huấn luyện** | 100 |
| **Kết quả lần 1 (%)** | 96.25 |
| **Kết quả lần 2 (%)** | 97.5 |
| **Kết quả lần 3 (%)** | 98.75 |
| **Trung bình (%)** | 97.5 |

Bảng 4.8. Kết quả độ chính xác sau 100 lần huấn luyện lần lặp 4

Lần lặp thứ 4 cho kết quả độ chính xác là 97.5%

* Lần lặp thứ năm đã được tính toán được biểu thị ở các bảng 4.3 và 4.4, kết quả độ chính xác là 97.083%.

Kết quả của cả 5 lần lặp đều đạt ở mức cao, ta có thể kết luận độ chính xác của mô hình mạng nơ ron đã sử dụng có giá trị là:

Có thể nói, mô hình mạng nơ ron đã sử dụng khá ổn định, điều đó đã thể hiện ở kết quả độ chính xác của mỗi lần lặp khác nhau, mỗi lần lặp là một tập huấn luyện và tập kiểm tra khác nhau. Nhưng kết quả không chênh lệch nhau quá nhiều.

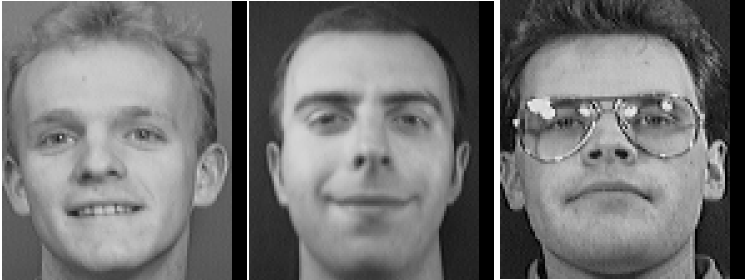
# Kết luận và hướng phát triển

## Kết luận

Hiện nay, việc áp dụng mạng nơ ron nhân tạo vào trong thực tế đời sống đang dần trở nên phổ biến bởi những hiệu quả của nó mang lại. Nhiều công ty, tổ chức cũng đang rất quan tâm đến vấn đề này bởi nó chính là tương lai của thế giới. Mạng nơ ron có thể giải quyết được rất nhiều bài toán khác nhau. Đây có thể coi là một ưu thế của mạng nơ ron mà các thuật toán khác khó lòng có thể có được. Nhận diện khuôn mặt cũng là một công nghệ đang đi theo xu hướng đó.

Hiện tại, hệ thống nhận dạng khuôn mặt sử dụng mạng nơ ron kết hợp với phương pháp trích rút dữ liệu bằng phương pháp phân tích thành phần chính vẫn còn đang khá đơn giản về mặt mô hình mạng và cả về tập dữ liệu. Nhưng phần nào đã giải quyết được bài toán đặt ra đó là nhận dạng được khuôn mặt của một người khi trước đó đã được “học” về khuôn mặt của người đó thông qua tập ảnh huấn luyện. Không những vậy, nó còn đưa về một kết quả chính xác khá cao 98.2486 %. Đó là một kết quả khá khả quan khi tiếp cận vấn đề về nhận diện khuôn mặt.

Tuy nhiên, vẫn còn một vài điểm hạn chế của hệ thống, đó là một số khuôn mặt bị nhận diện sai.



Hình 5.1. Các bức ảnh bị nhận diện sai

Những bức ảnh này bị nhận diện sai là do sắc thái biểu cảm của khuôn mặt, các phụ kiện trên khuôn mặt gây nhiễu cho việc nhận diện. Hạn chế của phương pháp sử dụng mạng nơ ron ở trên đấy là chưa xử lý được những tình huống đặc biệt này.

## Hướng phát triển

Với những hạn chế do phương pháp sử dụng để xây dựng hệ thống mang lại, trước hết cần phải tìm ra một cơ chế trích rút tốt hơn so với phương pháp phân tích thành phần chính hoặc cải tiến chính phương pháp này để có thể lấy ra được một tập dữ liệu mới thể hiện được gần như chính xác về các khuôn mặt, bởi trong thực tế các khuôn mặt sử dụng để nhận dạng sẽ còn bị ảnh hưởng đến từ môi trường hay chất lượng ảnh. Tiếp đó, xây dựng một mô hình mạng nơ ron mới dựa trên tập dữ liệu mới thu được. Từ đó hoàn thành ứng dụng để có một độ chính xác cao hơn.

Sau khi hoàn thành được công việc trên và đạt được một thành quả nào đó, tôi sẽ sử dụng mạng này để xây dựng một ứng dụng trong thực tế. Chẳng hạn như xây dựng ứng dụng mở khóa màn hình điện thoại thông minh bằng cách nhận diện khuôn mặt hay quản lý điểm danh nhân viên bằng nhận diện khuôn mặt.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | Michael Nielsen, Pavel Dudrenov, Neural Networks and Deep Learning, San Francisco, 2015. |
| [2] | D. Kriesel, A Brief Introduction to Neural Networks, University of Bonn, Germany, 2005. |
| [3] | W. Schiffmann, M. Joost, R. Werner, Optimization of the backpropagation Algorithm for Training Multilayer Perceptrons, University of Koblenz, Institute of Physics, Rheinau 156075 Koblenz, 1994. |
| [4] | Lee, H., Cher, C. and Huang, T., Learning efficiency improment of backpropagation algorithm by error saturation prevention method, National Taiwan University of Science and Technology, Department of Electronic Engineering, 2001. |
| [5] | AT&T, "The Database of Faces," AT&T Laboratories Cambridge, 2002. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. |
| [6] | Marijeta Slavković, Dubravka Jevtić, Face Recognition Using Eigenface Approach, Serbia, 2012. |