|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



BÙI MINH ANH

Sử dụng mạng noron kết hợp với giải thuật di truyền trong phân tích chi phí Bệnh viện Đại học Y Hà Nội

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2022

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

BÙI MINH ANH

**Sử dụng mạng noron kết hợp với giải thuật di truyền trong phân tích chi phí Bệnh viện Đại học Y Hà Nội**

|  |  |
| --- | --- |
| Ngành : | Công nghệ thông tin |
| Mã số : | 175A071292 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | TS. Trần Mạnh Tuấn |
|  |  |

HÀ NỘI, NĂM 2022

**GÁY BÌA ĐỒ ÁN TỐT NGHIỆP, KHÓA LUẬN TỐT NGHIỆP**

**LÊ THỊ DIỆU HƯỜNG ĐỒ ÁN/KL TỐT NGHIỆP HÀ NỘI, NĂM 2021**

|  |  |
| --- | --- |
|  | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

**Họ tên sinh viên: Bùi Minh Anh Hệ đào tạo : Đại học chính quy**

**Lớp: 59TH2 Ngành: Công nghệ thông tin**

**Khoa: Công nghệ thông tin**

1. **TÊN ĐỀ TÀI:**

***Sử dụng mạng noron kết hợp với giải thuật di truyền trong phân tích chi phí Bệnh viện Đại học Y Hà Nội***

**2- CÁC TÀI LIỆU CƠ BẢN:**

1. Machine Learning cơ bản (machinelearningcoban.com)
2. Deep Learning cơ bản | Nguyễn Thanh Tuấn (nttuan8.com)
3. Dive into Deep Learning (d2l.ai)
4. Machine Learning | Coursera (coursera.com)
5. Deep Learning Specialization | Coursera (coursera.com)
6. Visualization with Python | Matplotlib (matplotlib.org)
7. Genetic Algorithms Tutorial (tutorialspoint.com/genetic\_algorithms)

**3. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

|  |  |
| --- | --- |
| **Phần** | **Họ tên giáo viên hướng dẫn** |
| Chương 1: Cơ sở lý thuyết sử dụng trong đồ án | TS. Trần Mạnh Tuấn |
| Chương 2: Ứng dụng mạng nơron nhân tạo truyền thẳng và giải thuật di truyền vào xây dựng mô hình bài toán phân tích chi phí Bệnh viện Đại học Y Hà Nội |
| Chương 3: kết quả thực nghiệm |

**4. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày ............ tháng ......... năm 2022

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . . .tháng. . . . .năm 2022 **Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày... tháng... năm 2022

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  **BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP** |

TÊN ĐỀ TÀI: SỬ DỤNG MẠNG NORON KẾT HỢP VỚI GIẢI THUẬT DI TRUYỀN TRONG PHÂN TÍCH CHI PHÍ BỆNH VIỆN ĐẠI HỌC Y HÀ NỘI

*Sinh viên thực hiện*: Bùi Minh Anh

*Lớp*: 59TH2.

*Giáo viên hướng dẫn*: TS.Trần Mạnh Tuấn

**TÓM TẮT ĐỀ TÀI**

Lý do chọn dể tài:

Trong thế giới hiện đại, do chi phí cho các dịch vụ y tế ngày càng cao, các bệnh viện cần phải đánh giá kỹ lưỡng mối quan hệ giữa số lượng bệnh nhân và tổng chi phí bệnh viện. Dựa trên những kết quả đạt được, các nhà quản lý bệnh viện có thể xây dựng các kế hoạch khác nhau để nâng cao chất lượng dịch vụ chăm sóc sức khỏe và điều chỉnh đầu tư cho phù hợp với các chiến lược tiềm năng. Phân tích chi phí bệnh viện (HCA) đã trở thành một chủ đề quan trọng và hàng đầu trong các diễn đàn chính trị, phúc lợi xã hội và y tế .

Tại Hoa Kỳ, doanh thu và chi phí bệnh viện mỗi lần nhập viện tăng khoảng 5% mỗi năm từ năm 2000 đến năm 2009 . Bảo hiểm y tế chi trả một phần hoặc toàn bộ chi phí chăm sóc cho hơn 60% trường hợp nhập viện. Số ngày nhập viện trung bình giảm do chi phí nằm viện khá cao trong khi chi phí nằm viện của bệnh nhân ngoại khoa tăng từ 7,9% (2011) lên 29,1% (2012) . Ngoài ra, các dịch vụ chăm sóc sức khỏe và y tế là rất cần thiết và được quản lý đầy đủ vì lợi ích của người dân ở cả các nước đang phát triển và đã phát triển.

Lấy cảm hứng từ tầm quan trọng của vấn đề HCA trong thực tế, đồ án đề xuất một phương pháp học máy mới cho HCA bằng cách kết hợp thuật toán di truyền và mạng nơ-ron nhân tạo. Mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GAs) là hai trong số những kỹ thuật phổ biến và có cơ sở đã được sử dụng rộng rãi trong khoa học và công nghệ với các ứng dụng trong hóa học, vật lý, tài chính y tế và sinh học .

So với các phương pháp khác phương pháp này có nhiều ưu điểm khi áp dụng cho bài toán HCA . Thứ nhất, phương pháp đề xuất khắc phục được vấn đề dữ liệu thưa thớt trong các phương pháp chuỗi thời gian như ARIMA. Thứ hai, kiến trúc của ANN được đề xuất là động cho cả công trình mạng hoàn chỉnh và không hoàn chỉnh để thích ứng với những thay đổi của đầu vào và đầu ra dữ liệu cũng như sự kết nối giữa chúng. Điều này làm cho mạng đủ linh hoạt để xử lý các loại dữ liệu lớn từ HCA. Thứ ba, phương pháp đề xuất sử dụng GA để tăng cường tìm kiếm trọng số tối ưu trong ANN, do đó tăng độ chính xác. Sự lai tạo giữa ANN và GA được tạo thành từ các mạng động cho vấn đề HCA để nó khác với các kết hợp tương tự khác có thể tìm thấy trong tài liệu. Cho đến nay, không có phương pháp kết hợp nào như vậy cho HCA, mà thay vào đó, các công trình hiện có sử dụng phương pháp thống kê để suy ra mối tương quan giữa các biến trong HCA. Cuối cùng, mạng được đề xuất dễ hiểu và dễ triển khai để sử dụng rộng rãi.

Đối tượng và phạm vi nghiên cứu :

Trong nghiên cứu, đồ án sử dụng cơ sở dữ liệu thực của Bệnh viện Đại học Y Hà Nội (HMUS) bao gồm các thuộc tính: thông tin chung của bệnh nhân (giới tính, năm sinh, nơi sinh); thông tin điều trị (mã ICD, thời gian vào / ra viện, bảo hiểm); Tổng chi phí nằm viện (phẫu thuật, dịch vụ, thuốc). Dữ liệu có bốn bệnh phổ biến và thu thập 500 hồ sơ cho mỗi bệnh.

Phương pháp nghiên cứu :

Phương pháp nghiên cứu tài liệu: nghiên cứu lý thuyết và ứng dụng mạng nơron nhân tạo và thuật toán di truyền trong bài toán dự đoán .

 Phương pháp thực nghiệm: đi sâu nghiên cứu ứng dụng mạng nơ-ron nhân tạo bắt đầu từ bước chuẩn bị dữ liệu, bao gồm các kỹ thuật cho việc trích chọn đặc trưng, làm sạch dữ liệu, tiền xử lý, kiến trúc mạng, cách huấn luyện và kiểm tra mạng.

Xây dựng mô hình học máy dựa trên mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA) : Lập trình mô hình bằng ngôn ngữ lập trình Python.

**CÁC MỤC TIÊU CHÍNH**

* Tìm hiểu các đặc trưng của mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA), khả năng và các nguyên tắc để ứng dụng thành công mạng nơ-ron nhân tạo và thuật toán di truyền trong bài toán dự báo.
* Thu thập tập dữ liệu thực tế cho các vấn đề phân tích chi phí bệnh viện (HCA) từ Bệnh viện Đại học Y Hà Nội (HMUH).
* Xây dựng mô hình học máy mới dựa trên mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA) cho bài toán phân tích chi phí bệnh viện (HCA).

**KẾT QUẢ DỰ KIẾN**

* Demo chương trình phân tích chi phí
* Báo cáo đồ án tốt nghiệp

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào.Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  **Bùi Minh Anh** |

**LỜI CẢM ƠN**

Sau hơn 4 năm học tập và nghiên cứu tại Khoa Công nghệ thông tin - Trường Đại học Thủy Lợi, em đã được trải nghiệm trong môi trường đào tạo tốt và nhận được sự chỉ dạy tận tình của các thầy, các cô trong khoa.

Đầu tiên, em xin gửi lời cảm ơn chân thành các thầy,các cô trong khoa Công nghệ thông tin nói riêng và trường Đại học Thủy Lợi nói chung đã trang bị cho em những kiến thức quý báu trong suốt những năm học vừa qua.Đây sẽ là nền tảng để em có thể hoàn thành đồ án tốt hơn.

Đặc biệt, em xin được bày tỏ lòng biết ơn và gửi lời cảm ơn chân thành tới **TS. Trần Mạnh Tuấn** , người thầyđã truyền dạy cho em những kiến thức vững chắc, đã tận tình chỉ bảo, nhắc nhở và hướng dẫn em trong suốt quá trình hoàn thành đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn sâu sắc đến: Bố mẹ và những người thân trong gia đình đã chăm sóc, nuôi dạy, hỗ trợ, động viên và tạo mọi điều kiện thuận lợi nhất cho em trong suốt thời gian qua và đặc biệt trong thời gian em làm đồ án tốt nghiệp.

Ngoài ra, em cũng chân thành cảm ơn các bạn, các anh, các chị đã ủng hộ, giúp đỡ, trao đổi kiến thức, kinh nghiệm và động viên em trong thời gian học tập và nghiên cứu.

Trong quá trình nghiên cứu của mình, mặc dù được sự hướng dẫn và chỉ dạy rất nhiệt tình, nghiêm túc của **TS. Trần Mạnh Tuấn** cùng với sự nỗ lực của bản thân nhưng cũng không thể tránh được những thiếu sót. Em rất mong nhận được sự cảm thông và những góp ý từ quý Thầy, Cô và các bạn để đề tài được hoàn thiện hơn.

*Em xin chân thành cảm ơn!*

Hà Nội, ngày … tháng … năm 2022

Sinh viên thực hiện

**Bùi Minh Anh**

MỤC LỤC

[MỤC LỤC iii](#_Toc106547339)

[DANH MỤC CÁC HÌNH ẢNH v](#_Toc106547340)

[DANH MỤC BẢNG BIỂU vi](#_Toc106547341)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ vii](#_Toc106547342)

[MỞ ĐẦU 1](#_Toc106547343)

[CHƯƠNG 1 CƠ SỞ LÝ THUYẾT 6](#_Toc106547344)

[1.1 Tổng quan về mạng noron sinh học 6](#_Toc106547345)

[1.1.1 Giới thiệu 6](#_Toc106547346)

[1.1.2 Xử lý thông tin trong bộ não 7](#_Toc106547347)

[1.1.3 Cấu trúc và hoạt động của mạng nơron sinh học 8](#_Toc106547348)

[1.1.4 Khả năng của mạng nơron sinh học (bộ não) 9](#_Toc106547349)

[1.1.5 Quá trình học của bộ não 9](#_Toc106547350)

[1.2 Tổng quan về mạng nơron nhân tạo 10](#_Toc106547351)

[1.2.1 Lịch sử phát triển của mạng nơron nhân tạo 10](#_Toc106547354)

[1.2.2 Khái niệm 12](#_Toc106547355)

[1.2.3 Cấu trúc mạng nơron 12](#_Toc106547356)

[1.2.4 Các tính chất của mạng nơron 17](#_Toc106547357)

[1.2.5 Các kiểu mô hình mạng nơron nhân tạo 17](#_Toc106547358)

[1.2.6 Kiến thúc mạng nơron truyền thẳng 18](#_Toc106547359)

[1.2.7 Kiến trúc mạng nơron hồi quy 18](#_Toc106547360)

[1.3 Nguyên tắc hoạt động của mạng nơron nhân tạo 19](#_Toc106547361)

[1.4 Phương pháp huấn luyện mạng 20](#_Toc106547362)

[1.4.1 Học có giám sát 21](#_Toc106547363)

[1.4.2 Học tăng cường 22](#_Toc106547364)

[1.4.3 Học không có giám sát 22](#_Toc106547365)

[1.5 Ngôn ngữ sử dụng 23](#_Toc106547366)

[1.6 Ứng dụng của mạng nơron nhân tạo 27](#_Toc106547367)

[CHƯƠNG 2 XÂY DỰNG MÔ HÌNH PHÂN TÍCH CHI PHÍ BỆNH VIỆN 29](#_Toc106547368)

[2.1 Mạng nơron nhân tạo truyền thẳng 29](#_Toc106547369)

[2.1.1 Mạng perceptron một lớp 29](#_Toc106547370)

[2.1.2 Mạng perceptron nhiều lớp 31](#_Toc106547371)

[2.1.3 Một số vấn đề cần chú ý khi sử dụng mạng MLP 32](#_Toc106547372)

[2.2 Vấn đề chuẩn hoá số liệu đầu vào 33](#_Toc106547373)

[2.3 Vấn đề học chưa đủ và học quá thuộc của mạng 33](#_Toc106547374)

[2.4 Lựa chọn kích thước mạng 35](#_Toc106547375)

[2.5 Giải thuật di truyền 36](#_Toc106547376)

[2.5.1 Tìm hiểu chung về Gas 36](#_Toc106547381)

[2.5.2 Các toán tử của giải thuật di truyền 40](#_Toc106547382)

[2.5.3 Các tham số của giải thuật di truyền. 41](#_Toc106547383)

[2.5.4 Các thành phần của thuật giải di truyền 41](#_Toc106547384)

[2.6 Xây dựng mô phân tích chi phí bệnh viện 43](#_Toc106547390)

[2.6.1 Giới thiệu 43](#_Toc106547391)

[2.6.2 Giới thiệu bài toán 44](#_Toc106547392)

[2.6.3 Mô hình 45](#_Toc106547393)

[2.6.4 Các bước xây dựng mô hình 45](#_Toc106547394)

[CHƯƠNG 3 KẾT QUẢ THỰC NGHIỆM 48](#_Toc106547401)

[3.1 Môi trường thực nghiệm 48](#_Toc106547402)

[3.2 Cách thu thập, phân tích và xử lý dữ liệu 49](#_Toc106547403)

[3.2.1 Thu thập dữ liệu 50](#_Toc106547404)

[3.2.2 Phân tích dữ liệu 52](#_Toc106547405)

[3.2.3 Xử lý dữ liệu 52](#_Toc106547406)

[3.3 Mô tả dữ liệu bài toán 53](#_Toc106547407)

[3.4 Tiền xử lý dữ liệu 54](#_Toc106547408)

[3.5 Cách đánh giá mô hình 55](#_Toc106547409)

[3.6 Kết quả thực nghiệm 56](#_Toc106547410)

[KẾT LUẬN 61](#_Toc106547411)

[TÀI LIỆU THAM KHẢO 63](#_Toc106547412)

DANH MỤC CÁC HÌNH ẢNH

[Hình 1 Tế bào thần kinh 6](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532977)

[Hình 2 Mô hình tế bào thần kinh 8](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532978)

[Hình 3 Cấu trúc nơron nhân tạo 13](#_Toc106532979)

[Hình 4 Mạng nơron truyền thẳng nhiều lớp (Feed-forward neural network) 18](#_Toc106532980)

[Hình 5 Cấu trúc huấn luyện mạng nơron 20](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532981)

[Hình 6 Mô hình học có giám sát 22](#_Toc106532982)

[Hình 7 Mô hình học không có giám sát 22](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532983)

[Hình 8 Ảnh minh họa Python 23](#_Toc106532984)

[Hình 9 Guido Van Rossum 24](#_Toc106532985)

[Hình 10 Mạng perceptron một lớp 30](#_Toc106532986)

[Hình 11 Đồ thị hàm XOR 32](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532987)

[Hình 12 Thực hiện hàm XOR bằng mạng MLP 32](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532988)

[Hình 13 Hàm sigmoid g(x) = 1/(1+e-x) 33](file:///C:\Users\dell\Documents\Đồ%20án\DATN_BuiMinhAnh_175A071292.docx#_Toc106532989)

[Hình 14 Nội suy hàm y = sin(x/3) + v, 0≤ x ≤ 20 sử dụng MLP 35](#_Toc106532990)

[Hình 15 Cấu trúc giải thuật di truyền 39](#_Toc106532991)

[Hình 16 Mô hình bài toán 45](#_Toc106532992)

[Hình 17 Mô hình cấu trúc mạng nơron truyền thẳng ứng dụng trong bài toán 46](#_Toc106532993)

[Hình 18 Ảnh minh họa Colab 48](#_Toc106532994)

[Hình 19 Cấu hình 49](#_Toc106532995)

[Hình 20 Màn hình làm việc với Google Colab 49](#_Toc106532996)

[Hình 21 Mô hình xử lý dữ liệu 50](#_Toc106532997)

[Hình 22 Hình ảnh dữ liệu thô được thu thập chưa qua xử lý 54](#_Toc106532998)

[Hình 23 Dữ liệu sau khi qua xử lý excel 55](#_Toc106532999)

[Hình 24 Ma trận chưa chuyển hóa đánh giá kết quả dự đoán bài toán 59](#_Toc106533000)

[Hình 25 Ma trận chuyển hóa đánh giá kết quả dự đoán bài toán 59](#_Toc106533001)

DANH MỤC BẢNG BIỂU

[Bảng 1 Một số hàm kích hoạt thông dụng 17](#_Toc106459923)

[Bảng 2 Một số thư viện của python dùng trong học máy 27](#_Toc106459924)

[Bảng 3 Thống kê so sánh về độ chính xác khi thay đổi xác suất lai ghép và xác suất đột biến. 58](#_Toc106459925)

[Bảng 4 Thống kê so sánh về độ chính xác khi thay đổi một số phương pháp chọn lọc, lai ghép và đột biến trong GA 58](#_Toc106459926)

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

|  |  |
| --- | --- |
| **ANN**  **HCA**  **MLP**  **GA**  **HMUH** | Artificial Neural Network  Hospital cost analysis  Multilayer Perceptron  Genetic Algorithm  Hanoi Medical University Hospital |

MỞ ĐẦU

**1. Tổng quan nghiên cứu**

Trong bối cảnh ứng dụng công nghệ thông tin ngày càng tăng, dữ liệu phát sinh  
hoạt động quản lý, kinh doanh, sản xuất của các công ty, tổ chức ngày càng nhiều.  
Các công ty, tổ chức cần nhanh chóng đưa ra các quyết định bằng cách xử lý nhiều  
yếu tố với quy mô và tính phức tạp ngày càng tăng. Để có quyết định chính xác nhất,người quản lý thường thực hiện việc dự đoán hay phân loại vấn đề cần giải quyết trước khi đưa ra quyết định. Ngoài việc dựa trên các yếu tố liên quan trực tiếp đến vấn đề, người ra quyết định còn dựa trên kinh nghiệm bản thân và thông tin có được từ các hoạt động trước đó.

Trong thực tế, chúng ta bắt gặp nhiều bài toán như đánh giá cảm quan thực phẩm,  
dự đoán thị trường chứng khoán, định giá tài sản, đánh giá nhân viên, … Đó là các  
bài toán thuộc lớp bài toán dự đoán và phân loại. Đã có nhiều phương pháp đưa ra để giải các bài toán trên như phương pháp thống kê, hội quy, cây quyết định, mạng  
nơron nhân tạo, …

Trong đó, mạng nơron nhân tạo nhờ khả năng đọc, nhớ lại và khái quát hóa từ các  
mẫu dữ liệu huấn luyện, đã trở thành một trong hướng nghiên cứu chính của lĩnh vực trí tuệ nhân tạo. Xét về mặt lý thuyết, mạng noron nhân tạo tương đối độc lập với bản chất của các quá trình vật lý cần phân loại, dự đoán.

Ở mỗi lớp bài toán trên, đều có đặc điểm chung là khi giải bằng mạng nơron nhân  
tạo như: thu thập dữ liệu mẫu, tiền xử lý dữ liệu, xác định thông số mạng và huấn  
luyện. Do đó việc nghiên cứu để tổng quát hóa các bài toán và xây dựng phần mềm  
ứng dụng mạng noron nhân tạo có thể dùng cho nhiều bài toán cùng một lớp là hoàn toàn khả thi.

Mô hình mạng noron và giải thuật di truyền được sử dụng rộng rãi bởi vì có khả năng giải được nhiều bài toán với các lĩnh vục khác nhau: dự đoán, phân loại, mô  
hình hóa. Mạng nơron thích hợp để giải quyết bài toán thể hiện mối quan hệ giữa  
một tập hợp đầu vào và đầu ra biết trước.

**2. Lý do chọn đề tài**

Trong thế giới hiện đại, do chi phí cho các dịch vụ y tế ngày càng cao, các bệnh viện cần phải đánh giá kỹ lưỡng mối quan hệ giữa số lượng bệnh nhân và tổng chi phí bệnh viện. Dựa trên những kết quả đạt được, các nhà quản lý bệnh viện có thể xây dựng các kế hoạch khác nhau để nâng cao chất lượng dịch vụ chăm sóc sức khỏe và điều chỉnh đầu tư cho phù hợp với các chiến lược tiềm năng. Phân tích chi phí bệnh viện (HCA) đã trở thành một chủ đề quan trọng và hàng đầu trong các diễn đàn chính trị, phúc lợi xã hội và y tế .

Lấy cảm hứng từ tầm quan trọng của vấn đề HCA trong thực tế, đồ án đề xuất một phương pháp học máy mới cho HCA bằng cách kết hợp thuật toán di truyền và mạng nơ-ron nhân tạo. Mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GAs) là hai trong số những kỹ thuật phổ biến và có cơ sở đã được sử dụng rộng rãi trong khoa học và công nghệ với các ứng dụng trong hóa học, vật lý, tài chính y tế và sinh học .

So với các phương pháp khác phương pháp này có nhiều ưu điểm khi áp dụng cho bài toán HCA . Thứ nhất, phương pháp đề xuất khắc phục được vấn đề dữ liệu thưa thớt trong các phương pháp chuỗi thời gian như ARIMA. Thứ hai, kiến trúc của ANN được đề xuất là động cho cả công trình mạng hoàn chỉnh và không hoàn chỉnh để thích ứng với những thay đổi của đầu vào và đầu ra dữ liệu cũng như sự kết nối giữa chúng. Điều này làm cho mạng đủ linh hoạt để xử lý các loại dữ liệu lớn từ HCA. Thứ ba, phương pháp đề xuất sử dụng GA để tăng cường tìm kiếm trọng số tối ưu trong ANN, do đó tăng độ chính xác. Sự lai tạo giữa ANN và GA được tạo thành từ các mạng động cho vấn đề HCA để nó khác với các kết hợp tương tự khác có thể tìm thấy trong tài liệu. Cho đến nay, không có phương pháp kết hợp nào như vậy cho HCA, mà thay vào đó, các công trình hiện có sử dụng phương pháp thống kê để suy ra mối tương quan giữa các biến trong HCA. Cuối cùng, mạng được đề xuất dễ hiểu và dễ triển khai để sử dụng rộng rãi.

**3. Đối tượng nghiên cứu**

* Lý thuyết về mạng nơron nhân tạo, cấu trúc, mô hình xây dựng mạng nơron nhân tạo.
* Lý thuyết về phương pháp huấn luyện mạng nơron nhân tạo.
* Lý thuyết về giải thuật di truyền.
* Ứng dụng giải thuật di truyền vào huấn luyện mạng nơron nhân tạo truyền thẳng.
* Lý thuyết về bài toán Phân tích chi phí bệnh viện (HCA).
* Ứng dụng mạng nơron nhân tạo truyền thẳng vào bài toán Phân tích chi phí bệnh viện (HCA).

**4. Phạm vi nghiên cứu**

* Mạng noron.
* Thuật toán di truyền.
* Mạng noron truyền thẳng.
* Ngôn ngữ Python.

**5. Mục tiêu nghiên cứu**

* Tìm hiểu các đặc trưng của mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA), khả năng và các nguyên tắc để ứng dụng thành công mạng nơ-ron nhân tạo và thuật toán di truyền trong bài toán dự báo.
* Thu thập tập dữ liệu thực tế cho các vấn đề phân tích chi phí bệnh viện (HCA) từ Bệnh viện Đại học Y Hà Nội (HMUH).
* Xây dựng mô hình học máy mới dựa trên mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA) cho bài toán phân tích chi phí bệnh viện (HCA).

**6. Phương pháp nghiên cứu**

* Phương pháp nghiên cứu tài liệu: nghiên cứu lý thuyết và ứng dụng mạng nơron nhân tạo và thuật toán di truyền trong bài toán dự đoán .
* Phương pháp thực nghiệm: đi sâu nghiên cứu ứng dụng mạng nơ-ron nhân tạo bắt đầu từ bước chuẩn bị dữ liệu, bao gồm các kỹ thuật cho việc trích chọn đặc trưng, làm sạch dữ liệu, tiền xử lý, kiến trúc mạng, cách huấn luyện và kiểm tra mạng.
* Xây dựng mô hình học máy dựa trên mạng nơ-ron nhân tạo (ANN) và thuật toán di truyền (GA) : Lập trình mô hình bằng ngôn ngữ lập trình Python.

**7. Cấu trúc của** **đồ án tốt nghiệp.**

Bố cục của đồ án gồm các nội dung chính sau:

* Nội dung của đề tài nghiên cứu về lý thuyết mạng nơrơn sinh học, lý thuyết mạng nơrơn nhân tạo, lý thuyết mạng nơrơn truyền thằng.
* Nghiên cứu về lý thuyết thuật toán di truyền.
* Nghiên cứu lý thuyết ứng dụng thuật toán di truyền vào đển huấn luyện mạng nơrơn nhân tạo truyền thẳng.
* Nghiên cứu lý thuyết bài toán dự đoán độ tin cậy và tính hợp lệ của phân tích chi phí bệnh viện.
* Nghiên cứu ứng dụng mạng nơrơn nhân tạo vào bài toán phân tích chi phí bệnh viện.
* Lập trình cài đặt mạng nơrơn nhân tạo truyền thẳng nhiều lớp.
* Lập trình cài đặt thuật toán di truyền ứng dụng huấn luyện mạng nơrơn nhận tạo truyền thẳng.

Ngoài phần mở đầu, tổng quan lĩnh vực nghiên cứu và kết luận, bố cục của luận văn gồm ba chương chính như sau:

* Chương 1: Cơ sở lý thuyết.
* Chương 2: Xây dựng mô hình phân tích chi phí bệnh viện.
* Chương 3: Kết quả kiểm thử.

# CƠ SỞ LÝ THUYẾT

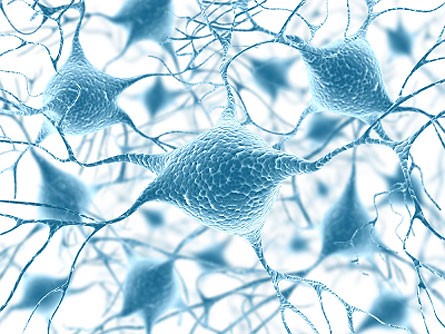
## Tổng quan về mạng noron sinh học

### Giới thiệu

Bộ não người có chức năng hết sức quan trọng trong đời sống của con người. Nó gần như kiểm soát hầu hết mọi hành vi của con người từ các hoạt động cơ bắp đơn giản đến những hoạt động phức tạp như học tập, nhớ, suy luận, tư duy, sáng tạo,…

Bộ não người có mạng lưới gồm khoảng 1011 phần tử (gọi là tế bào) liên kết phức tạp với nhau. Trong đó:

* Khoảng 1010 phần tử là nơron.
* Khoảng 9\*1010 phần tử là các tế bào thần kinh đệm và chúng có nhiệm vụ phục vụ cũng như hỗ trợ cho các nơron.



Hình 1 Tế bào thần kinh

*Hình 1* cho thấy hàng tỷ tế bào thần kinh trong bộ não của bạn được kết nối bởi một mạng lưới kết nối. Tế bào thần kinh tương tác thông qua các kết nối điện tương tự như trong một máy tính.

Cho đến nay người ta vẫn chưa thực sự biết rõ về cấu tạo chi tiết của bộ não.Các đặc tính của não người

*Tính phân lớp*: Các vùng trong bộ não được phân thành các lớp, thông tin được xử lý theo các tầng

*Tính module*: Các vùng của bộ nhớ được phân thành các môđun được mã hóa bằng các định nghĩa mối quan hệ tích hợp giữa các tín hiệu vào qua các giác quan và các tín hiệu ra.

*Mối liên kết*: Liên kết giữa các lớp dẫn đến các dữ liệu dùng chung xem như các liên hệ phản hồi khi truyền tín hiệu.

*Xử lý phân tán các tín hiện vào*: Các tín hiệu vào được truyền qua nhiều kênh thông tin khác nhau, được xử lý bằng các phương pháp đặc biệt.

### Xử lý thông tin trong bộ não

Thông tin được tiếp nhận từ các giác quan và chuyển vào các tế bào thần kinh vận động và các tế bào cơ. Mỗi tế bào thần kinh tiếp nhận thông tin, điện thế sẽ tăng trong thần kinh cảm giác, nếu điện thế này vượt ngưỡng nó tạo ra dòng điện trong tế bào thần kinh cảm giác, ý nghĩa dòng điện đó được giải mã và lưu ở thần kinh trung ương, kết quả xử lý thông tin được gửi đến các tế bào cơ.

Các tế bào thần kinh đưa các tín hiệu giống nhau, do đó không thể phân biệt được đó là của loài động vật nguyên thủy hay của một giáo sư. Các khớp thần kinh chỉ cho các tín hiệu phù hợp qua chúng, còn lại các tín hiệu khác bị cản lại. Lượng tín hiện được biến đổi được gọi là cường độ khớp thần kinh đó chính là trọng số của nơron trong mạng nơron nhân tạo.

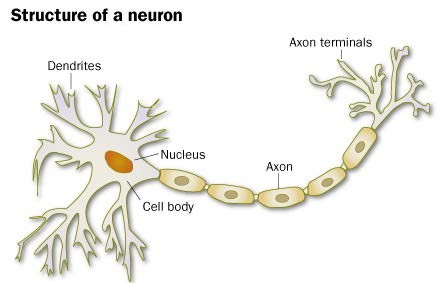
Hoạt động của cả hệ thống thần kinh bao gồm não bộ và các giác quan như sau:

* Trước hết con người bị kích thích bởi các giác quan từ bên ngoài hoặc trong cơ thể. Sự kích thích đó được biến thành các xung điện bởi chính các giác quan tiếp nhận kích thích. Những tín hiệu này được chuyển về trung ương thần kinh là não bộ để xử lý. Trong thực tế não bộ liên tục nhận thông tin xử lý, đánh giá và so sánh với những thông tin lưu trữ để đưa ra các quyết định.
* Những mệnh lệnh cần thiết được phát sinh và gửi đến những bộ phận thi hành thích hợp như các cơ tay, chân,…. Những bộ phận thi hành biến những xung điện thành dữ liệu của hệ thống.

### Cấu trúc và hoạt động của mạng nơron sinh học

Mỗi tế bào thần kinh gồm 3 thành phần chính:

* Thân tế bào thần kinh (*cell body* còn gọi là *soma*)
* Hệ thống các dây thần kinh tiếp nhận (*dendrites*)
* Một sợi trục thần kinh (*axon*).



Hình 2 Mô hình tế bào thần kinh

Hệ thống dây thần kinh tiếp nhận là một lưới dày đặc các dây thần kinh dạng cây bao bọc xung quanh thân tế bào, chúng dẫn các tín hiệu đến phần thân tế bào. Thân tế bào sẽ tổng hợp các tín hiệu đầu vào này, làm thay đổi điện thế của nó và khi vượt qua một mức ngưỡng thì sẽ cho một xung điện trên sợi trục thần kinh ra (*axon*). Các dây thần kinh *axon* có thể rẽ ra nhiều nhánh để nối đến các dây thần kinh vào hoặc nối trực tiếp với phần thân của các tế bào thần kinh khác thông qua các khớp thần kinh (*synapse*).

Khi một tế bào thần kinh hoạt động, nó được kích thích tạo ra một tín hiệu điện hóa chạy dọc theo sợi axon và dẫn đến các khớp thần kinh. Khớp thần kinh được chia làm 2 loại: khớp nối kích thích (*excitalory*) và khớp nối ức chế (*inhibitory*). Tại các khớp thần kinh này xảy ra các quá trình phản ứng và giải phóng các chất hữu cơ tạo nên các tín hiệu điện kích thích tế bào thần kinh. Cường độ tín hiệu mà một tế bào thần kinh nhận được phụ thuộc chủ yếu vào mức độ liên kết của các khớp nối. Những nghiên cứu hoạt động của hệ thần kinh đã chỉ ra rằng quá trình “học” của bộ não chính là việc hình thành hoặc thay đổi mức độ liên kết của các khớp nối.

### Khả năng của mạng nơron sinh học (bộ não)

Bộ não người có chức năng hết sức quan trọng đối với đời sống của con người. Cấu tạo của nó rất phức tạp, tinh vi hết sức quan trọng bởi được tạo thành từ mạng nơron có hàng chục tỉ tế bào với mức bộ liên kết giữa các nơron là rất cao. Hơn nữa, nó còn được chia thành các vùng và các lớp khác nhau.

Bộ não hoạt động dựa trên cơ chế hoạt động song song của các nơron tạo nên nó.

Bộ nhớ được tổ chức theo các bó thông tin và truy cập theo nội dung (có thể truy xuất thông tin dựa theo các giá trị thuộc tính của đối tượng).

Bộ não có khả năng tổng quát hóa, có thể truy xuất các tri thức hay các mối liên kết chung của đối tượng tương ứng với một khái niệm chung nào đó.

Bộ não có khả năng học.

### Quá trình học của bộ não

Khi các xung tín hiệu từ các “dây thần kinh vào” tới các khớp nối, khớp nối sẽ cho tín hiệu đi qua hoặc không kích thích nơron tiếp theo. Do vậy hình thành một con đường truyền xung nhất định.

Học là làm sao cho con đường này được lặp đi lặp lại nhiều lần, nên sức cản của các khớp nối sẽ nhỏ dần, tạo điều kiện cho những lần lặp lại dễ dàng hơn. Có thể nói: Toàn bộ những kiến thức, kinh nghiệm của một người tích lũy được và lưu giữ trong đầu chính là hệ thống sức cản của các khớp nối.

Như vậy, nơron sinh học hoạt động theo cách thức sau: nhận tín hiệu đầu vào, xử lý các tín hiệu này và cho ra một tín hiệu đầu ra. Tín hiệu đầu ra này sau đó được truyền đi làm tín hiệu đầu vào cho các nơron khác.

Dựa trên những hiểu biết về nơron sinh học, con người xây dựng nơron nhân tạo hy vọng tạo nên một mô hình có sức mạnh như bộ não.

## Tổng quan về mạng nơron nhân tạo

**Mạng nơron nhân tạo** (*Artificial Neural Networks – ANN*) [1] là hệ thống được xây dựng dựa trên nguyên tắc cấu tạo của bộ não người. Nó cho chúng ta một hướng mới trong nghiên cứu hệ thống thông tin. Mạng nơron nhân tạo có thể thực hiện các bài toán: Tính toán gần đúng các hàm số, thực hiện các bài toán tối ưu, nhận mẫu, nhận dạng và điều khiển đối tượng hiệu quả hơn so với các phương pháp truyền thống.

Mạng nơron nhân tạo có một số lượng lớn mối liên kết của các phần tử biến đổi có liên kết song song. Nó có hành vi tương tự như bộ não người với khả năng học và tổng hợp thông tin từ sự luyện tập của các tập mẫu dữ liệu. Các phần tử biến đổi của mạng nơron nhân tạo được gọi là các nơron nhân tạo hoặc gọi tắt là nơron.



### Lịch sử phát triển của mạng nơron nhân tạo

Cuối thế kỷ XIX – đầu thế kỷ XX, một số nghiên cứu về vật lý, tâm lý và hệ thần kinh của nhà khoa học Herman, Emst Mach và Ivan Ivalov đã đưa ra các lý thuyết về quá trình học, sự tưởng tượng, sự quyết định,…của hệ thần kinh nhưng chưa có sự mô tả toàn học cho hoạt động của mạng nơron.

Năm 1943, mô hình đơn giản mạng nơron bằng mạch điện tử lần đầu tiên được đưa ra bởi Warren McCulloch và Walter Pits cùng với sự khẳng định mạng nơron nhân tạo về nguyên lý có thể thực hiện được trong phạm vi tính toán các hàm số học và logic. Đây là điểm khởi đầu của lĩnh vực mạng nơron.

Sau đó Donal Nebb đưa ra một cơ chế giải thích cho quá trình học (*Learning*) diễn ra trong các nơron sinh học.

Cuối thập niên 50, ứng dụng thực tế đầu tiên của mạng nơron nhân tạo do Frank Rosenblatt đưa ra. Mạng của ông đưa ra là mạng Perceptron có kết hợp luật học (*Learning rule*) dùng để nhận dạng mẫu (*pattern recognition*). Cùng thời gian đó, Bernard Widrow và Ted Hoff giới thiệu một thuật toán học (*Learning algorithm*) và sử dụng nó để huấn luyện (Training) các mạng nơron tiếp hợp tuyến tính (tương tự mạng Rosenblatt).

Năm 1969, Minskey và Pappert là hai nhà toán học nổi tiếng thời đó đã chỉ ra những hạn chế của mạng Perceptron của Rosenblatt và mạng Widrow-Hoff làm nhiều người nghĩ rằng nghiên cứu về mạng nơron sẽ vào ngõ cụt. Hơn nữa vào thời gian này chưa có những máy tính số mạnh để thực nghiệm mạng nơron nên các nghiên cứu về mạng nơron bị trì hoãn gần một thập kỷ.

Năm 1972, Teuvo Kohonen và James Anderson độc lập phát triển các mạng nơron mới với năng lực nhớ (*memory*) và khả năng tự tổ chức (*self-organizing*). Cũng trong giai đoạn này, Stephen Grossberg cũng nghiên cứu tích cực về mạng tự tổ chức. Sang thập kỷ 80, khi ngành công nghiệp máy tính phát triển mạnh mẽ thì những nghiên cứu về mạng nơron tăng lên một cách đột ngột. Có hai phát kiến quan trọng nhất là:

* Sử dụng cơ học thống kê để giải thích hoạt động của mạng hồi quy một lớp (*recurrent network*), loại mạng được sử dụng như một bộ nhớ kết hợp, được nhà vật lý John Hopfield mô tả.
* Sử dụng thuật toán lan truyền ngược (*back-propagation algorithm*) để huấn luyện các mạng Perceptron đa lớp (*mutilayer perceptron network*). David Rumelhalt và James McClrlland là những người trình bày thuật toán lan truyền ngược có ảnh hưởng nhất (1968).

Ngày nay, lĩnh vực mạng nơron được nghiên cứu, phát triển mạnh mẽ và ứng dụng rất nhiều trong thực tế.

### Khái niệm

Mạng nơron nhân tạo là sự kết hợp giữa các nơron nhân tạo với nhau. Mỗi liên kết kèm theo một trọng số nào đó đặc trưng cho đặc tính kích hoạt ức chế giữa các nơron. Các nơron còn gọi là các nút (node) được sắp xếp trong mạng theo các lớp, bao gồm lớp ra (output layer) và các lớp ẩn (hidden layer).

Các đặc điểm của mạng nơron nhân tạo:

* Mạng được xây dựng bằng các nơron liên kết lại với nhau.
* Chức năng của mạng được xác định bởi: cấu trúc mạng, quá trình xử lý bên trong của từng nơron, và mức độ liên kết giữa các nơron.
* Mức độ liên kết giữa các nơron được xác định thông qua quá trình học của mạng (quá trình huấn luyện mạng). Có thể xem các trọng số là các phương tiện để lưu trữ thông tin dài hạn trong mạng nơron. Nhiệm vụ của quá trình huấn luyện mạng là cập nhật các trọng số khi có thông tin về các mẫu học.

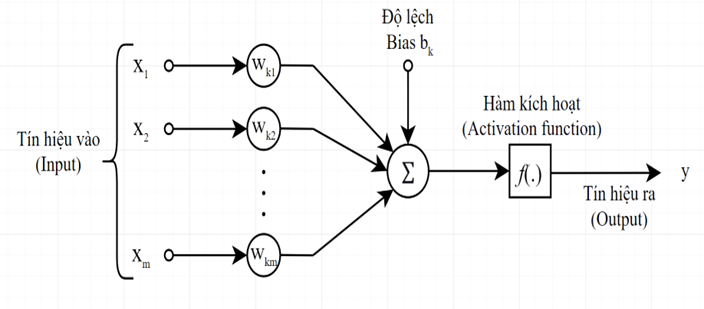
Mạng nơron dựa trên việc mô phỏng cấp thấp hệ thống nơron sinh học. Trong tương lai với sự phát triển mô phỏng nơron sinh học, chúng ta có thể có loại máy tính thông minh thật sự.

### Cấu trúc mạng nơron

Mạng nơron nhân tạo gồm hai thành phần:

* Các nút (đơn vị xử lý, nơron)
* Các liên kết giữa chúng được gán một trọng số nào đó đặc trưng cho cường độ liên kết.

Cấu trúc của một noron được mô tả:



Hình 3 Cấu trúc nơron nhân tạo

Các thành phần cơ bản của một nơron nhân tạo gồm:

* *Tập các đầu vào*: là các tín hiệu vào (*input signals*) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.
* *Tập các liên kết*: Mỗi liên kết được thể hiện bởi một trọng số (gọi là trọng số liên kết – *Synaptic weight*). Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là wkn. Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.
* *Hàm tổng* (*Summing function*): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.
* *Ngưỡng* (còn gọi là một độ lệch - *bias*): Ngưỡng này thường được đưa vào như một thành phần của hàm kích hoạt.
* *Hàm kích hoạt(Activation function)*: Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng đã cho. Thông thường, phạm vi đầu ra của mỗi nơron được giới hạn trong đoạn [0,1] hoặc [–1,1]. Các hàm kích hoạt rất đa dạng, có thể là các hàm tuyến tính hoặc phi tuyến. Việc lựa chọn hàm kích hoạt nào là tùy thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng.
* Đầu ra: là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.

Về mặt toán học, cấu trúc của một nơron i được mô tả bằng cặp biểu thức sau:

trong đó: x1, x2, …xm là các tín hiệu đầu vào

wi1, wi2,…wim là các trọng số kết nối của nơron thứ i

neti là hàm tổng

f là hàm kích hoạt.

* i* là một ngưỡng.

yi là tín hiệu đầu ra của nơron.

Như vậy, tương tự như nơron sinh học, nơron nhân tạo cũng nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả đến hàm kích hoạt), và cho một tín hiệu đầu ra (là kết quả của hàm kích hoạt).

| **STT** | **Tên hàm** | **Dạng đồ thị** | **Phương trình** |
| --- | --- | --- | --- |
| 1 | Hàm đồng nhất (Linear function, Identity function ) |  |  |
| 2 | Hàm bước nhị phân(Binary step) |  |  |
| 3 | Hàm sigmoid (Sigmoid function (logsig)) |  |  |
| 4 | Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig)) |  |  |
| 5 | Hàm ArcTan |  |  |
| 6 | Hàm ReLU |  |  |
| 7 | Hàm Leaky ReLU |  |  |
| 8 | Hàm Sinusoid |  |  |
| 9 | HàmSinc |  |  |
| 10 | Hàm Gaussian |  |  |
| 11 | SoftPlus |  |  |
| 12 | Bent identity |  |  |

Bảng 1 Một số hàm kích hoạt thông dụng

### Các tính chất của mạng nơron

*Là hệ phi tuyến*: Mạng nơron có khả năng to lớn trong lĩnh vực nhận dạng và điều khiển các đối tượng phi tuyến

*Là hệ xử lý song song*: Mạng nơron có cấu trúc song song, do đó có độ tính toán rất cao, rất phù hợp với lĩnh vực nhận dạng và điều khiển.

*Là hệ học và thích nghi*: Mạng được luyện từ các số liệu quá khứ và có khả năng tự chỉnh khi số liệu đầu vào bị mất, có thể điều khiển online.

*Là hệ nhiều biến, nhiều đầu vào, nhiều đầu ra (MIMO****)***, rất tiện dùng khi đối tượng điều khiển có nhiều biến số.

### Các kiểu mô hình mạng nơron nhân tạo

Mạng nơron nhân tạo là sự liên kết của các nơron nhân tạo. Sự sắp xếp bố trí của các nơron và cách thức liên hệ giữa chúng tạo nên kiến trúc mạng nơron.

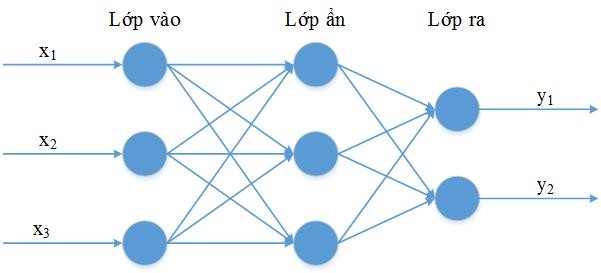
Theo cách sắp xếp nơron thì có kiến trúc mạng 1 lớp (*single-layer*) là mạng chỉ có 1 lớp ra và kiến trúc mạng nhiều lớp (*multiple-layer*) là mạng có các lớp ẩn.

Theo cách liên kết giữa các nơron thì kiến trúc mạng truyển thẳng (*feedforward networks*) và kiến trúc mạng hồi quy (*recurrent networks*).

### Kiến thúc mạng nơron truyền thẳng

Dòng dữ liệu từ đơn vị đầu vào đến đơn vị đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết phản hồi. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép [2].

Mô hình mạng nơron truyền thẳng như sau:



Hình 4 Mạng nơron truyền thẳng nhiều lớp (Feed-forward neural network)

Mạng nơron truyền thẳng và các vấn đề liên quan sẽ được trình bày rõ ở **Chương 2**.

### Kiến trúc mạng nơron hồi quy

Mạng hồi quy còn được gọi là mạng phản hồi là mạng tự liên kết thành các vòng và liên kết hồi quy giữa các nơron.

* Mạng nơron hồi quy có trọng số liên kết đối xứng như mạng Hopfield luôn hội tụ về trạng thái ổn định.
* Mạng liên kết hai chiều (BAM) là mạng thuộc nhóm mạng nơron hồi quy liên kết tay đôi, trong đó bảo đảm nơron của cùng một lớp không liên kết với nhau, cũng hội tụ về trạnh thái ổn định.

Nghiên cứu mạng nơron hồi quy có trọng số liên kết không đối xứng sẽ gặp nhiều phức tạp hơn so với mạng truyền thẳng và mạng hồi quy đối xứng. Mạng nơron hồi quy có khả năng nhận dạng mẫu, nhận dạng các hàm phi tuyến, dự báo … Một ưu điểm khác của mạng nơron hồi quy là chỉ cần mạng nhỏ hơn về cấu trúc cũng có khả năng như mạng truyền thẳng có cấu trúc lớn hơn. Nó khắc phục được giả thuyết truyền thống của mạng nơron là coi mạng có số nơron đủ lớn. Gồm có 2 loại:

* Mạng nơron hồi quy không hoàn toàn
* Mạng nơron hồi quy hoàn toàn

## Nguyên tắc hoạt động của mạng nơron nhân tạo

Ta thấy rằng các nơron trong cùng 1 lớp thì nhận tín hiệu đầu vào cùng một lúc. Do đó, về nguyên tắc chúng có thể xử lý song song. Hoạt động của mạng nơron có thể xem như hoạt động của một hệ thống xử lý thông tin được cấu thành từ nhiều phần tử hoạt động song song. Khi mạng nơron hoạt động, các thành phần của vectơ tín hiệu vào **x** = ( x1, x2, ... , xN) được đưa vào mạng, tiếp đó các nơron ở lớp ẩn và lớp ra sẽ được kích hoạt dần dần. Sau một quá trình tính toán tại các nơron mạng sẽ được kích hoạt hoàn toàn và cho ra vectơ tín hiệu đầu ra **y** = (y1 , y2 , ... , yk) tại S nơron lớp ra, Ta có thể coi mạng nơron như một bảng tra cứu giữa **y** và **x** mà không cần biết hàm quan hệ tường minh của **y** theo **x**.

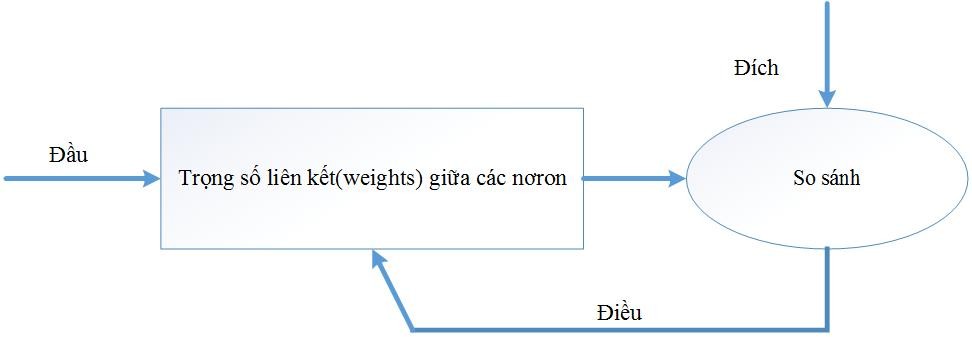
Sự khác biệt giữa mạng nơron và hệ thống xử lý thông thường là khả năng thích nghi với dữ liệu vào. Đó là ma trận trọng số và độ lệch (hệ số *bias)* của mạng có thể hiệu chỉnh để mạng thích nghi được với bài toán đặt ra. Quá trình hiệu chỉnh các trọng số và hệ số *bias* của mạng gọi là quá trình huấn luyện mạng (*training*) bằng một số phương pháp học.

## Phương pháp huấn luyện mạng

Thông thường mạng nơron được điều chỉnh hoặc được huấn luyện để hướng các đầu vào riêng biệt đến đích ở đầu ra.

Cấu trúc huấn luyện mạng

* Trọng số liên kết của mạng được điều chỉnh trên cơ sở so sánh đầu ra với đích mong muốn (target) cho tới khi đầu ra mạng phù hợp với đích.
* Những cặp vào/đích (input/taget) được dùng đề giám sát cho sự huấn luyện mạng.



Hình 5 Cấu trúc huấn luyện mạng nơron

Để có được một số cặp vào/ra, ở đó mỗi giá trị vào được gửi đến mạng và giá trị ra tương ứng được thực hiện bằng mạng là sự xem xét và so sánh với giá trị mong muốn. Bình thường tồn tại một sai số bởi lẽ giá trị mong muốn không hoàn toàn phù hợp với giá trị thực. Sau mỗi lần chạy, ta có tổng bình phương của tất cả các sai số. Sai số này được sử dụng để xác định các trọng số liên kết mới.

Sau mỗi lần chạy, trọng số liên kết của mạng được sửa đổi với đặc tính tốt hơn tương ứng với đặc tính mong muốn. Từng cặp giá trị vào/ra phải được kiểm tra và trọng số được điều chỉnh một vài lần. Sự thay đổi các trọng số liên kết của mạng được dừng lại nếu tổng các bình phương sai số nhỏ hơn một giá trị đặt trước hoặc đã chạy đủ một số lần chạy xác định (trong trường hợp này mạng có thể không thỏa mãn yêu cầu đặt ra do sai lệch còn cao). Có hai kiểu học:

* Học thông số (*Paramater Learning*): Tìm ra biểu thức cập nhật các thông số về trọng số, cập nhật kết nối giữa các nơron.
* Học cấu trúc (*Structure Learning*): Trọng tâm là sự biến đổi cấu trúc của mạng nơron gồm số lượng nút (node) và các mẫu liên kết.

Nhiệm vụ của việc học thông số là bằng cách nào đó, tìm được ma trận chính xác mong muốn từ ma trận giả thuyết ban đầu với cấu trúc của mạng nơron có sẵn.

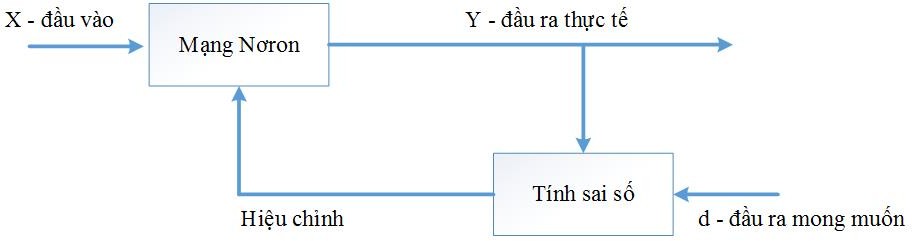
Để làm được việc đó, mạng nơron sử dụng các trọng số điều chỉnh, với nhiều phương pháp học khác nhau có thể tính toán gần đúng ma trận W cần tìm đặc trưng cho mạng. Có ba phương pháp học, mỗi phương pháp học tương ứng với một nhiệm vụ học trừu tượng:

* Học có giám sát (Supervised Learning)
* Học không giám sát (Unsupervised Learning)
* Học tăng cường (Reinforcement Learning)

### Học có giám sát

Là quá trình học ở mỗi thời điểm thứ i khi đưa tín hiệu xi vào mạng nơron, tương ứng sẽ có các đáp ứng mong muốn di của đầu ra cho trước ở thời điểm đó. Hay nói cách khác, trong quá trình học có giám sát, mạng nơron được cung cấp liên tục các cặp số liệu mong muốn vào/ra ở từng thời điểm (x1, d1), (x2, d2), …., (xN, dk) khi cho tín hiệu vào thực là xN sẽ tương ứng có tín hiệu đầu ra cũng được lặp lại là dk giống như mong muốn. Kết quả của quá trình học có giám sát là tạo được một hộp đen có đầu vào là vectơ tín hiệu vào X sẽ đưa ra được câu trả lời đúng d.

Để đạt được kết quả mong muốn trên, khi đưa vào tín hiệu xn, thông thường sẽ có sai lệch ek giữa tín hiệu đầu ra thực yk với tín hiệu đầu ra mong muốn dk. Sai lệch đó sẽ được truyền ngược tới đầu vào để điều chỉnh thông số mạng nơron là ma trận trọng số W. Quá trình cứ thế tiếp diễn sao cho sai lệnh giữa tín hiện ra mong muốn và tín hiệu ra thực tế nằm trong phạm vi cho phép, kết quả nhận được ma trận trọng số với các phần tử wkN đã được điều chỉnh phù hợp với đặc điểm của đối tượng hay hàm số mạng nơron cần học.



Hình 6 Mô hình học có giám sát

### Học tăng cường

Là phương pháp học trong đó tín hiệu z được đưa từ bên ngoài nhưng không được đầy đủ mà có thể chỉ đưa đại diện 1 bit để có tính chất kiểm tra quá trình đúng hay sai. Tín hiệu đó được gọi là tín hiệu tăng cường (*Reinforcement Signal*). Phương pháp học tăng cường chỉ là một phương pháp học riêng của phương pháp học có giám sát, bởi vì nó cũng nhận tín hiệu chỉ đạo từ bên ngoài.

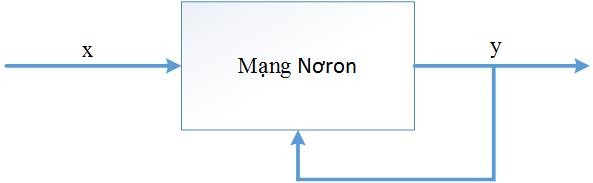
Tín hiệu giám sát bên ngoài z thường được tiến hành bởi các tín hiệu ước lượng để tạo thông tin ước lượng cho mạng nơron điều chỉnh trọng số với hy vọng sự ước lượng đó mang lại sự chính xác trong quá trình tính toán.

Mô hình học tăng cường tương tự như mô hình học có giám sát.

### Học không có giám sát

Trong trường hợp này, hoàn toàn không có tín hiệu ở bên ngoài. Giá trị mục tiêu điều khiển không được cung cấp và không được tăng cường. Mạng phải khám phá các mẫu, các nét đặc trưng, tính cân đối tính tương quan…. Trong khi khám phá các đặc trưng khác, mạng nơron đã phải qua việc tự thay đổi thông số, vấn đề đó được gọi tự tổ chức (*Self – Organizing*).

Mô hình học không có giám sát được minh họa:



Hình 7 Mô hình học không có giám sát

## Ngôn ngữ sử dụng

Thực nghiệm trong bài được cài đặt trên ngôn ngữ python.

* Ngôn ngữ lập trình python là gì

Python [3] là một ngôn ngữ lập trình bậc cao được sáng tạo bởi Guido Van Rossum vào cuối những năm 1980s và được giới thiệu lần đầu tiên năm 1991.Python là một ngôn ngữ được coi là sáng sủa, cấu trúc rõ ràng sử dụng các câu lệnh thân thiện với người dùng.



Hình 8 Ảnh minh họa Python

* Lịch sử của Python

Python là một ngôn ngữ khá cũ được tạo ra bởi Guido Van Rossum. Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991.



Hình 9 Guido Van Rossum

* Tính năng chính của Python
* Ngôn ngữ lập trình đơn giản, dễ học: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.
* Miễn phí, mã nguồn mở: Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.
* Khả năng di chuyển: Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.
* Khả năng mở rộng và có thể nhúng: Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.
* Ngôn ngữ thông dịch cấp cao: Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.
* Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.
* Hướng đối tượng: Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.
* Ứng dụng thực tế của ngôn ngữ Python

Python được ứng dụng trong nhiều ngành nghề bởi sự đơn giản của nó cùng với bộ sưu tập thư viện khổng lồ của mình python đã được ứng dụng trong một số ngành như:

* Lập trình ứng dụng web: Bạn có thể tạo web app có khả năng mở rộng (scalable) được bằng cách sử dụng framework và CMS (Hệ thống quản trị nội dung) được tích hợp trong Python. Vài nền tảng phổ biến để tạo web app là: Django, Flask, Pyramid, Plone, Django CMS. Các trang như Mozilla, Reddit, Instagram và PBS đều được viết bằng Python.
* Khoa học và tính toán: Có nhiều thư viện trong Python cho khoa học và tính toán số liệu, như SciPy và NumPy, được sử dụng cho những mục đích chung chung trong tính toán. Và, có những thư viện cụ thể như: EarthPy cho khoa học trái đất, AstroPy cho Thiên văn học,... Ngoài ra, Python còn được sử dụng nhiều trong machine learning, khai thác dữ liệu và deep learning.
* Tạo nguyên mẫu phần mềm: Python chậm hơn khi so sánh với các ngôn ngữ được biên dịch như C++ và Java. Nó có thể không phải là lựa chọn tốt nếu nguồn lực bị giới hạn và yêu cầu về hiệu quả là bắt buộc. Tuy nhiên, Python là ngôn ngữ tuyệt vời để tạo những nguyên mẫu (bản chạy thử - prototype). Ví dụ, bạn có thể sử dụng Pygame (thư viện viết game) để tạo nguyên mẫu game trước. Nếu thích nguyên mẫu đó có thể dùng C++ để viết game thực sự.
* Ngôn ngữ tốt để dạy lập trình: Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình. Bên cạnh những tính năng và khả năng tuyệt vời thì cú pháp đơn giản và dễ sử dụng của nó là lý do chính cho việc này.Ứng dụng không thể không nói đến của python là ứng dụng trong khoa học dữ liệu, học máy(machine learning), trí tuệ nhân tạo. Hiện nay python đang là một ngôn ngữ lập trình phổ biến nhất cho nghiên cứu và phát triển trong học máy.Một trong những lý do mà Python trở thành ngôn ngữ phổ biến nhất được sử dụng trong machine learning là bộ sưu tập thư viện khổng lồ.

Một số thư viện trong Python được sử dụng trong học máy được trình bày ở bảng sau:

| **Tên thư viện** | **Mô tả** |
| --- | --- |
| Numpy | Thư viện hỗ trợ cho việc tính toán, xử lý các mảng, ma trận lớn đa chiều thực hiện các phép tính như nhân ma trận, tính ma trận chuyển vị. |
| Pandas | Thư viện để phân tích dữ liệu, nó được phát triển đặc biệt để khai thác và chuẩn bị dữ liệu. |
| Scikit-learn | Thư viện dựa trên ngôn ngữ python để xây dựng các mô hình học máy. Scikit-learn cung cấp nhiều thuật toán học tập cho phép hồi quy, phân cụm, phân loại. |
| TensorFlow | Thư viện rất mạnh trong ML, hỗ trợ cài đặt mô hình, huấn luyện và kiểm thử mô hình. |
| Keras | Thư viện mạng noron |
| Matplotlib | Thư viện giúp biểu diễn dữ liệu dưới dạng đồ thị hai chiều và 3 chiều. |
| Seaborn | Thư viện dựa trên thư viện Matplotlib, nó như 1 interface, wapper của matplotlib để trực quan hóa dữ liệu. |
| Scipy | Thư viện SciPy được xây dựng dựa trên thư viện Numpy, cung cấp thao tác mảng N chiều thuận tiện và nhanh chóng. SciPy gồm các gói con (submodule) cho đại số tuyến tính, tối ưu hóa, tích hợp và thống kê. |
| Scikit-Fuzzy | Thư viện Scikit-Fuzzy là một thư viện Python 3 mới được xây dựng dựa trên Scipy cho phép thực hiện các thuật toán logic mờ. |
| IPython | Một trình bao mạnh mẽ để dễ dàng chỉnh sửa và ghi lại các phiên làm việc. Nó cũng hỗ trợ trực quan hóa và tính toán song song. |

Bảng 2 Một số thư viện của python dùng trong học máy

## Ứng dụng của mạng nơron nhân tạo

Với những đặc điểm của mạng nơron nhân tạo, mạng nơron đã được sử dụng để giải quyết nhiều bài toán thuộc nhiều lĩnh vực của các ngành khác nhau. Các nhóm ứng dụng mà mạng nơron nhân tạo đã được áp dụng rất có hiệu quả là:

*Bài toán phân lớp*: Loại bài toán này đòi hỏi giải quyết vấn đề phân loại các đối tượng quan sát được thành các nhóm dựa trên các đặc điểm của các nhóm đối tượng đó. Đây là dạng bài toán cơ sở của rất nhiều bài toán thực tế như

* Nhận dạng chữ viết, ký tự quang học
* Nhân dạng tiếng nói
* Phân loại gen
* Phân loại chất lượng sản phẩm

*Bài toán dự báo*: Mạng nơron nhân tạo đã được ứng dụng thành công trong việc xây dựng các mô hình dự báo sử dụng tập dữ liệu trong quá khứ để dự đoán số liệu trong tương lai. Đây là nhóm bài toán khó và rất quan trọng trong nhiều ngành khoa học.

* Dự báo lượng sử dụng khí gas
* Dự báo thời tiết

*Bài toán điều khiển và tối ưu hóa*: Nhờ khả năng học và xấp xỉ hàm mà mạng nơron nhân tạo đã được sử dụng trong nhiều hệ thống điều khiển tự động cũng như góp phần giải quyết những bài toán tối ưu trong thực tế

* Điều khiển tốc độ cơ điện điện một chiều

Tóm lại, mạng nơron nhân tạo được xem như là một cách tiếp cận đầy tiềm năng để giải quyết các bài toán có tính phi tuyến, phức tạp và đặc biệt là trong tình huống mối quan hệ bản chất vật lý của quá trình cần nghiên cứu không dễ thiết lập tường minh.

# XÂY DỰNG MÔ HÌNH PHÂN TÍCH CHI PHÍ BỆNH VIỆN

## Mạng nơron nhân tạo truyền thẳng

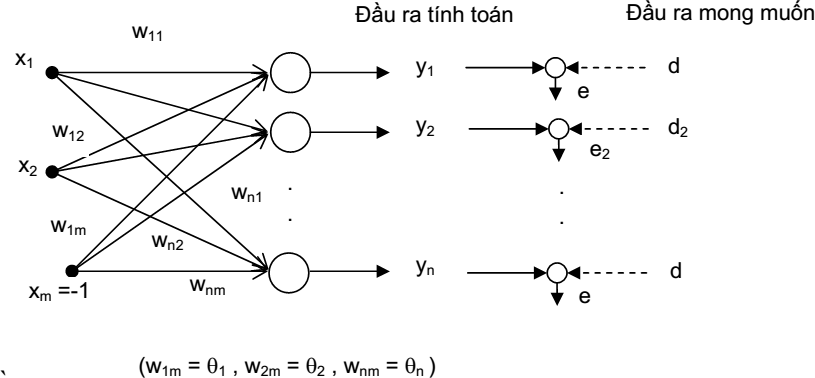
### Mạng perceptron một lớp

Mạng perceptron một lớp do F.Rosenblatt đề xuất năm 1960 là mạng truyền thẳng chỉ một lớp vào và một lớp ra không có lớp ẩn. Trên mỗi lớp này có thể có một hoặc nhiều nơron. Mô hình mạng nơron của Rosenblatt sử dụng hàm ngưỡng đóng vai trò là hàm chuyển. Do đó, tổng của các tín hiệu vào lớn hơn giá trị ngưỡng thì giá trị đầu ra của nơron sẽ là 1, còn trái lại sẽ là 0.

Ngay từ khi mạng Perceptron một lớp được đề xuất nó đã được sử dụng để giải quyết bài toán phân lớp. Một đối tượng sẽ được nơron *i* phân vào lớp A nếu

Tổng thông tin đầu vào :

Trong đó wij là trọng số liên kết từ nơron j tới nơron i, xj là đầu vào từ nơron j, và θ là ngưỡng của nơron i. Trong trường hợp trái lại đối tượng sẽ được phân vào lớp B. Việc huấn luyện mạng dựa trên phương pháp học có giám sát với tập mẫu học là {(x(k), d(k))}, k= 1,2, …, p .Trong đó d(k) = [d1(k), d2(k), …, dn(k)]T là đầu ra quan sát được tương ứng với đầu vào x(k) = [x1(k), x2(k), …, xm(k)]T (với m là số đầu vào, n là số đầu ra và p là cặp mẫu đầu vào - đầu ra dùng cho việc học). Như vậy chúng ta mong rằng sau quá trình học, đầu ra tính toán được y(k) = [y1(k), y2(k), …, yn(k)]T sẽ bằng với đầu ra của mẫu học d(k).



Hình 10 Mạng perceptron một lớp

Để bắt đầu quá trình luyện mạng, các trọng số được gán giá trị ngẫu nhiên trong khoảng [-3, 3]. Sau đó hiệu chỉnh các trọng số cho phù hợp với mẫu học để làm giảm sai số giữa y(k) và d(k)

Các bước tiến hành:

* Xác định ngẫu nhiên bộ trọng số.
* Với mỗi mẫu học (x(k), d(k)), k=1,2, …, p thực hiện các bước:
* Tính giá trị y(k) theo công thức
* Xác định sai số tại nơron i: , trong đó *di* là giá trị đầu ra quan sát được và *yi* là giá trị đầu ra tính toán tại nơron thứ *i*
* Tính là số gia của trọng số (trọng số liên kết giữa đầu vào j tới nơron
* theo công thức: trong đó là tốc độ học (0<<1)
* Hiệu chỉnh trong đó là trọng số sau khi điều chỉnh ở lần học tại thời điểm t

Rosenblatt đã chứng minh rằng quá trình học của mạng Perceptron sẽ hội tụ tới bộ trọng số W, biểu diễn đúng các mẫu học với điều kiện là các mẫu này biểu thị các điểm rời rạc của một hàm khả tách tuyến tính nào đó (f: Rn → R được gọi là khả tách tuyến tính nếu các tập {F-1(xk)}, với xk thuộc miền trị của f, có thể tách được với nhau bởi các siêu phẳng trong không gian Rn).

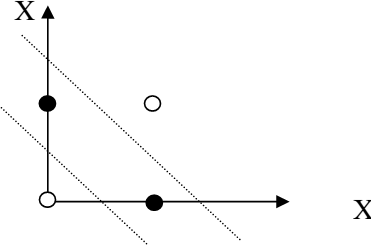
Năm 1969, Minsky và Papert đã chứng minh một cách chặt chẽ rằng lớp hàm thể hiện sự phụ thuộc giữa đầu vào và đầu ra có thể học bởi mạng Perceptron một lớp là lớp hàm khả tách tuyến tính. Khả tách tuyến tính là trường hợp tồn tại một mặt siêu phẳng để phân cách tất cả các đối tượng của một lớp này với một lớp khác, ví dụ một mặt phẳng sẽ phân chia không gian ba chiều thành hai vùng riêng biệt. Mở rộng ra, nếu có n đầu vào , n > 2 thì công thức tạo nên một siêu phẳng có n-1 chiều trong không gian n chiều, nó chia không gian đó thành hai nửa. Trong nhiều bài toán thực tế đòi hỏi chia các vùng của các điểm trong một siêu không gian thành các lớp riêng biệt. Loại bài toán này gọi là bài toán phân lớp. Bài toán phân lớp có thể giải quyết bằng cách tìm các tham số thích hợp cho một siêu phẳng để nó có thể chia không gian n chiều thành các vùng riêng biệt.

Với tính chất của như đã nêu trên, mạng perceptron một lớp có thể mô tả các hàm logic như AND, OR và NOT. Tuy nhiên nó không thể hiện được hàm XOR. Như vậy chứng tỏ mô hình perceptron một lớp không thể giải quyết bài toán này. Vấn đề này sẽ được giải quyết bằng mô hình mạng nơron perceptron nhiều lớp (Multilayer Perceptron - MLP).

### Mạng perceptron nhiều lớp

Mạng perceptron nhiều lớp (Multilayer Perceptron –MLP) còn được gọi là mạng truyền thẳng nhiều lớp là sự mở rộng của mô hình mạng perceptron với sự bổ sung thêm những lớp ẩn và các nơron trong các lớp ẩn này có hàm chuyển (hàm kích hoạt) dạng phi tuyến. Mạng MLP có một lớp ẩn là mạng nơron nhân tạo được sử dụng phổ biến nhất, nó có thể xấp xỉ các hàm liên tục được định nghĩa trên một miền có giới hạn cũng như những hàm là tập hợp hữu hạn của các điểm rời rạc.

**Giải quyết bài toán XOR với mạng MLP**



Hình 11 Đồ thị hàm XOR

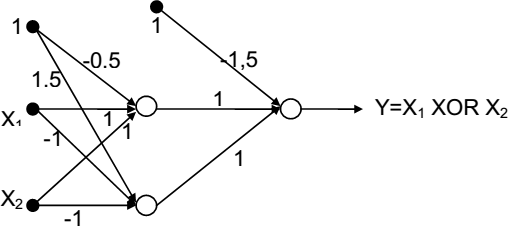
Ta thấy có thể dùng hai đường thẳng để phân tách với trường hợp hàm XOR.

và

Hay ta giải hệ bất phương trình:

=

Dễ thấy mỗi bất phương trình ở trên có thể được thực hiện bằng một nơron và đầu ra của hai nơron này (2 bất phương trình) là đầu vào của một hàm AND. Do vậy có thể sử dụng mạng MLP sau để thực hiện chức năng của hàm XOR như sau:



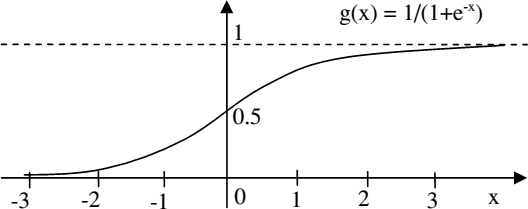
Hình 12 Thực hiện hàm XOR bằng mạng MLP

### Một số vấn đề cần chú ý khi sử dụng mạng MLP

Mạng nơron perceptron nhiều lớp là loại mạng nơron được sử dụng trong nhiều ứng dụng thực tế. Tuy nhiên, để mạng có thể đưa ra kết quả tốt, chúng ta cần quan tâm đến một số vấn đề có ảnh hưởng khá quan trọng đến hiệu quả làm việc của nó bao gồm: vấn đề chuẩn hoá số liệu đầu vào, vấn đề học chưa đủ và học quá của mạng, vấn đề lựa chọn một cấu trúc mạng phù hợp với bài toán.

## Vấn đề chuẩn hoá số liệu đầu vào

Mạng MLP thường sử dụng hàm chuyển là hàm sigmoid có dạng như sau:



Hình 13 Hàm sigmoid g(x) = 1/(1+e-x)

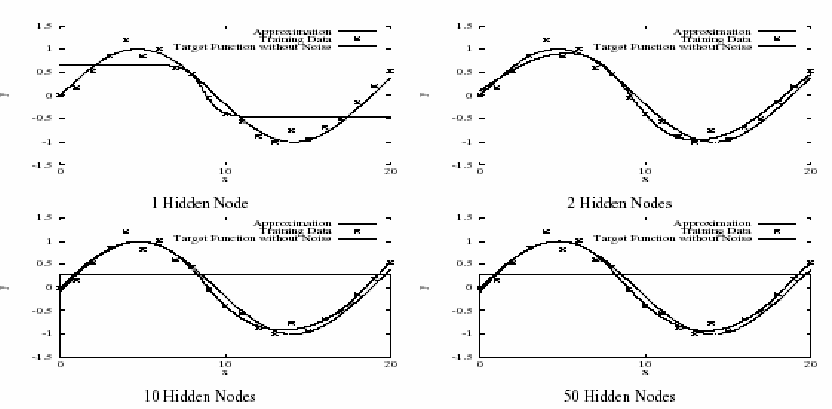
Với dạng hàm này, giá trị ở đầu ra của mỗi nơron nằm trong phạm vi khoảng (0,1) và nó đạt các giá trị bão hoà (xấp xỉ 0 hay 1) khi lớn. Do đó, khi đầu vào của mạng có giá trị tuyệt đối lớn thì ta cần chuẩn hoá nó về khoảng có giá trị nhỏ, nếu không thì các nơron tại các lớp ẩn ngay ban đầu đã có thể đạt giá trị bão hoà và quá trình học của mạng không đạt kết quả mong muốn. Với dạng hàm như trên thì giá trị đầu vào của mạng thường được chuẩn hoá về khoảng thuộc đoạn [-3, 3]. Mặt khác, do tín hiệu đầu ra của nơron nằm trong khoảng giá trị (0,1) nên các giá trị đầu ra thực tế trong các mẫu học cũng cần chuẩn hoá về khoảng giá trị này để có thể dùng cho quá trình luyện mạng. Do vậy trong quá trình tính toán, để có các giá trị thực tế ở đầu ra của mạng chúng ta cần phải chuyển các giá trị trong khoảng (0,1) về miền các giá trị thực tế.

## Vấn đề học chưa đủ và học quá thuộc của mạng

Vấn đề mấu chốt khi xây dựng một mạng nơron nhân tạo là làm thế nào mạng có khả năng tổng quát hoá cao để đưa ra kết quả tốt cả với những trường hợp đầu vào của mạng không nằm trong tập mẫu đã dùng để luyện mạng. Giống như các mô hình hồi quy phi tuyến khác, đối với mạng nơron nhân tạo ta cũng phải giải quyết hai vấn đề là ANN học chưa đủ (underfitting) và học quá (overfitting) [4]. Khi mạng có cấu trúc (số nút ẩn và liên kết) cũng như số lần học chưa đủ so với nhu cầu của bài toán thì sẽ dẫn tới tình trạng mạng không đủ khả năng mô tả gần đúng mối quan hệ tương quan giữa đầu vào và đầu ra của quá trình cần dự báo và dẫn tới học chưa đủ. Trái lại, nếu mạng quá phức tạp (quá nhiều nút ẩn và quá nhiều tham số) và được học “quá khít” đối với các mẫu dùng để luyện mạng thì có thể dẫn tới tình trạng mạng học cả thành phần nhiễu lẫn trong các mẫu đó, đây là tình trạng “học quá thuộc” của mạng. Vấn đề nêu trên có thể làm cho nhiều loại mạng nơron, đặc biệt là mạng MLP có thể có những trường hợp cho kết quả dự đoán rất sai lệch với thực tế.

Một số giải pháp cho vấn đề học quá của mạng:

* *Sử dụng tập số liệu có tính đại diện tốt để luyện mạng:*Đây được xem là một cách khá tốt để tránh hiện tượng overfitting. Khi tập mẫu dùng để luyện mạng thể hiện được nhiều trạng thái có thể xẩy ra của quá trình cần nghiên cứu thì sau khi học mạng sẽ có khả năng tổng quát hoá tương đối tốt từ tập dữ liệu đó và sẽ không chịu ảnh hưởng nhiều của hiện tượng overfitting. Ngoài ra một số biện pháp dưới đây cũng có thể góp phần quan trọng giúp khắc phục hiện tượng overfitting của mạng.
* *Lựa chọn cấu trúc mô hình phù hợp:* Việc lựa chọn mô hình của mạng (số lớp ẩn, số nơron trên mỗi lớp ẩn) có ảnh hưởng quan trọng đến hiện tượng học chưa đủ (underfitting) và học quá (overfitting) của mạng. Nghiên cứu của Steve Lawrence và C.Lee Giles về bài toán nội suy hàm y = sin(x/3) + v, 0 ≤ x ≤ 20. Với v là biến ngẫu nhiên nằm trong khoảng (- 0.25, 0.25) cho thấy nếu mạng chỉ gồm một nút ẩn thì mạng không thể học được hàm này. Mạng có hai nút ẩn có khả năng tổng quát hoá tốt nhất (tuy không chính xác hoàn toàn với các mẫu nhưng nó tạo nên giáng điệu gần với hàm cần xấp xỉ nhất và nó đã không quá khít với nhiễu của mẫu học). Các mạng có độ phức tạp hơn (10 nút ẩn, 50 nút ẩn) tuy nó có thể học khá chính xác các mẫu được sử dụng nhưng chính điều này lại làm cho nó học quá nhiều cả thành phần nhiễu nên khả năng tổng quát hoá giảm và dẫn tới hiện tượng overfitting.



Hình 14 Nội suy hàm y = sin(x/3) + v, 0≤ x ≤ 20 sử dụng MLP

* *Dừng học đúng lúc:*Năm 1991, trong một nghiên cứu về vấn đề học quá của mạng, hai tác giả Nelson và Illingworth đã đưa ra giải pháp dừng học đúng lúc để tránh hiện tượng học quá của mạng như sau:
* Tập mẫu được chia làm hai phần: một phần dùng để luyện mạng và phần còn lại để kiểm thử.
* Sử dụng các giá trị khởi tạo nhỏ
* Sử dụng hằng số tốc độ học có giá trị thấp.
* Tính toán sự thay đổi lỗi kiểm thử trong quá trình luyện mạng.
* Dừng học khi thấy lỗi kiểm thử bắt đầu tăng

## Lựa chọn kích thước mạng

Các công trình dựa trên định lý của Kolmogorov dự kiến rằng toàn bộ các ánh xạ liên tục từ [0,1]p đến [0,1]n đều có thể được xấp xỉ bằng một mạng perceptron ba lớp có lớp vào gồm p nơron , lớp ra gồm n nơron và lớp ẩn gồm (2p+1) nơron .

Tuy nhiên không thể chỉ ra được chính xác số lượng nơron tối ưu trong mạng, tính chất của các nơron, tức là dạng phi tuyến cụ thể thực hiện phép xấp xỉ này.

Một số công trình nghiên cứu về chủ đề này cho rằng số nơron tối ưu ở lớp ẩn thường nhỏ hơn (2p+1).

Ngoài ra cũng cần phải nói cơ sở dữ liệu học phải có kích thước phù hợp với kiến trúc mạng. Theo Vapnik và Chervonenkis, cơ sở dữ liệu học phải có số mẫu thoả mãn: N ≈ 10.Nw, ở đó Nw là số trọng số của mạng.

Gọi số nơron thuộc lớp ẩn là L, số nơron ở lớp vào là p thì trọng số của các kết nối giữa lớp vào và lớp ẩn thứ nhất (kể cả ngưỡng) là: D = (p+1).L

Theo một số kết quả nghiên cứu, số mẫu của cơ sở dữ liệu học cần phải thoả mãn N ≈ 4.D

Khi số lượng mẫu của cơ sở dữ liệu học chưa đạt đến giới hạn cần thiết thì ta nên làm giảm số lượng các kết nối để tránh hiện tượng học thuộc lòng.

## Giải thuật di truyền



### Tìm hiểu chung về Gas

Genetic algorithms (thuật giải di truyền) là một giải thuật mô phỏng theo quá trình chọn lọc tự nhiên, là kỹ thuật chung giúp giải quyết vấn đề bài toán bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung (dựa trên thuyết tiến hóa muôn loài của Darwin) trong điều kiện qui định sẵn của môi trường. Lấy ý tưởng từ quá trình tiến hoá tự nhiên, xuất phát từ một lớp các lời giải tiềm năng ban đầu, GA tiến hành tìm kiếm trên không gian lời giải bằng cách xây dựng lớp lời giải mới tốt hơn (tối ưu hơn) lời giải cũ. Quá trình xây dựng lớp lời giải mới được tiến hành dựa trên việc chọn lọc, lai ghép, đột biến từ lớp lời giải ban đầu. Quần thể lời giải trải qua quá trình tiến hoá: ở mỗi thế hệ lại tái sinh các lời giải tương đối tốt, trong khi các lời giải “xấu” thì chết đi.

Vậy GAs làm gì?

Trong GA, một tập các biến của bài toán đưa ra được mã hóa sang một chuỗi (hay một cấu trúc mã hóa khác) tương tự như một nhiễm sắc thể trong tự nhiên. Mỗi chuỗi bao gồm một giải pháp có thể của bài toán. Giải thuật di truyền sử dụng các toán tử được sinh ra bởi sự chọc lọc tự nhiên một quần thể các chuỗi nhị phân (hoặc các cấu trúc khác), mã hóa khoảng tham số trên mỗi thế hệ, khảo sát các phạm vi khác nhau của không gian tham số, và định hướng tìm kiếm đối với khoảng mà là xác suất cao để tìm kiếm sự thực hiện tốt hơn. Thuật toán di truyền gồm có bốn quy luật cơ bản là *lai ghép, đột biến, sinh sản và chọn lọc tự nhiên*

**Quá trình lai ghép** (phép lai) quá trình này diễn ra bằng cách ghép một hay nhiều đoạn gen từ hai nhiễm sắc thể cha-mẹ để hình thành nhiễm sắc thể mới mang đặc tính của cả cha lẫn mẹ.

Phép lai này có thể mô tả như sau:

* Chọn ngẫu nhiên hai hay nhiều cá thể trong quần thể. Giả sử chuỗi nhiễm sắc thể của cha và mẹ đều có chiều dài là m. Tìm điểm lai bằng cách tạo ngẫu nhiên một con số từ 1 đến m-1. Như vậy, điểm lai này sẽ chia hai chuỗi nhiễm sắc thể cha-mẹ thành hai nhóm nhiễm sắc thể con là m1 và m2. Hai chuỗi nhiễm sắc thể con lúc này sẽ là m11+m22 và m21+m12. Đưa hai chuỗi nhiễm sắc thể con vào quần thể để tiếp tục tham gia quá trình tiến hóa

**Quá trình đột biến** (phép đột biến) quá trình tiến hóa đƣợc gọi là quá trình đột biến khi một hoặc một số tính trạng của con không được thừa hưởng từ hai chuỗi nhiễm sắc thể cha-mẹ. Phép đột biến xảy ra với xác suất thấp hơn rất nhiều lần so với xác suất xảy ra phép lai. Phép đột biến có thể mô tả như sau:

* Chọn ngẫu nhiên một số k từ khoảng 1 ≥ k ≥ m
* Thay đổi giá trị của gen thứ k
* Đưa nhiễm sắc thể con vào quần thể để tham gia quá trình tiến hóa tiếp theo

Quá trình sinh sản và chọn lọc (phép tái sinh và phép chọn)

**Phép tái sinh**: là quá trình các cá thể được sao chép dựa trên độ thích nghi của nó. Độ thích nghi là một hàm được gán các giá trị thực cho các cá thể trong quần thể của nó. Phép tái sinh có thể mô phỏng như sau:

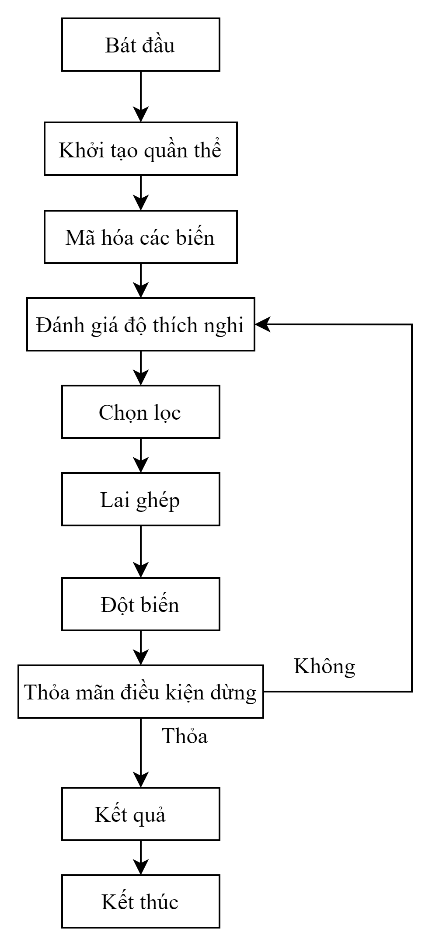
* Tính độ thích nghi của từng cá thể trong quần thể, lập bảng cộng dồn các giá trị thích nghi đó (theo thứ tự gán cho từng cá thể) ta được tổng độ thích nghi. Giả sử quần thể có n cá thể. Gọi độ thích nghi của cá thể thứ i là Fi, tổng dồn thứ i là Ft. Tổng độ thích nghi là Fm Tạo số ngẫu nhiên F có giá trị trong đoạn từ 0 đến Fm
* Chọn cá thể k đầu tiên thỏa mãn F ≥ Ft đưa vào quần thể của thế hệ mới.

**Phép chọn**: là quá trình loại bỏ các cá thể xấu và để lại những cá thể tốt.

Phép chọn được mô tả như sau:

* Sắp xếp quần thể theo thứ tự độ thích nghi giảm dần
* Loại bỏ các cá thể cuối dãy, chỉ để lại n cá thể tốt nhất.

Cấu trúc giải thuật di truyền tổng quát:



Hình 15 Cấu trúc giải thuật di truyền

Sau đây là những nguyên tắc cơ bản thực hiện giải thuật di truyền GAs:

B1: Khởi tạo và mã hóa một quần thể ngẫu nhiên của NST. Đó gọi là “quần thể hiện tại”

B2: Đánh giá độ thích nghi của mỗi NST trong quần thể hiện tại.

B3: Tạo ra thế hệ trung gian, thông qua chọn lựa suy diễn các NST trong quần thể hiện tại tuỳ theo độ thích nghi. Đó sẽ là cha mẹ của những thế hệ tiếp theo.

B4: Áp dụng toán tử lai ghép và nghịch đảo đối với những cặp hoặc NST đơn trong thế hệ trung gian, qua đó sẽ sản sinh ra một thế hệ NST mới. Đó là quần thể hiện tại.

Lặp lại các bước 2-4 cho đến khi một giải pháp phù hợp được tìm thấy.

### Các toán tử của giải thuật di truyền

* Toán tử chọn lọc
* Chọn lọc dựa trên độ thích nghi.
* Chọn lọc dựa trên sự xếp hạng
* Chọn lọc dựa trên sự cạnh tranh
* Chọn lọc hướng không gian
* Toán tử di cư
* Toán tử nghịch đảo
* Toán tử đột biến
* Toán tử lai ghép
* Lai ghép một điểm (one-point crossover)
* Lai ghép hai điểm (two-point crossover)
* Lai ghép N điểm (N-point crossover)
* Lai ghép đồng nhất (Uniform crossover)

### Các tham số của giải thuật di truyền.

**Xác suất lai ghép**: là tham số cho biết tần suất thực hiện toán tử lai ghép. Nếu không có lai ghép, cá thể con sẽ chính là bản sao của cá thể “cha mẹ”. Nếu xác suất lai ghép bằng 100%, khi đó mọi cá thể con đều được tạo ra qua quá trình lai ghép.

**Xác suất đột biến**: là tham số cho biết tần suất đột biến của nhiễm sắc thể. Nếu không có đột biến, thế hệ con được tạo ra ngay sau giai đoạn lai ghép mà không bị thay đổi. Ngược lại, một hoặc một số phần của nhiễm sắc thể sẽ bị thay đổi. Nếu xác suất đột biến là 100%, toàn bộ nhiễm sắc thể sẽ bị thay đổi. Nếu tham số này bằng 0%, không có gì bị thay đổi hết.

**Kích thước quần thể**: là tham số cho biết có bao nhiêu cá thể (NST) trong 1 thế hệ của quần thể. Nếu có quá ít cá thể, khả năng thực hiện lai ghép rất nhỏ và khi đó chỉ có một vùng tìm kiếm nhỏ mới được khảo sát. Ngược lại, việc kích thước quần thể quá lớn cũng không tốt, do nó sẽ làm chậm quá trình giải bài toán.

### Các thành phần của thuật giải di truyền



**Khởi động quần thể ban đầu**

Tạo quần thể đầu tiên trong thuật giải, là nơi xuất phát quá trình tiến hóa, bao gồm tất cả các giá trị thô ban đầu. Tùy theo vấn đề của bài toán mà có cách khởi động khác nhau. Trước một bài toán áp dụng thuật giải di truyền, ta cần phải xác định rõ nhiễm sắc thể và cá thể cho vấn đề, và thông thường đó sẽ kết quả cuối cùng. Việc phân tích sẽ dựa trên kết quả là cơ bản nhất.

**Đánh giá cá thể**

Chắc chắn rằng việc chọn cá thể sẽ thông qua kết quả, hay mục đích của vấn đề. Dựa trên mức độ thích nghi của cá thể, bao gồm những vướng mắc mà cá thể gặp phải. Thông thường, đặt mỗi vấn đề nhỏ tương ứng với một giá trị điểm thích nghi, kết quả đánh giá gồm tổng các số điểm đó. Cá thể tốt nhất sẽ có số điểm thấp nhất hoặc lớn nhất.

Theo thuyết tiến hóa của Darwin, nhiễm sắc thể tốt nhất sẽ tồn tại và tạo ra các cá thể con mới. Có nhiều phương pháp để chọn các nhiễm sắc thể tốt nhất.

* Chọn lọc Roulette (Roulette Wheel Selection)
* Chọn lọc xếp hạng (Rank Selection)
* Chọn lọc cạnh tranh (Tournament Selection)

**Toán tử lai ghép**

Lai ghép nhằm nâng cao kết quả cá thể, do đó, toán tử lai ghép sẽ tạo điều kiện cho tiến trình hội tụ nhanh hay chậm. Còn tùy thuộc vào cách tổ chức và phân bố các nhiễm sắc thể mà chúng ta có xác suất lai ghép nhanh hay chậm. Sau đây là vài phương pháp lai ghép thông dụng trong kỹ thuật di truyền:

* Lai ghép ánh xạ từng phần (PMX Partial Mapped Crossover)
* Lai ghép có trật tự (OX Order Crossover)
* Lai ghép dựa trên vị trí (Position Based Crossover)
* Lai ghép dựa trên thứ tự (Order Base Crossover)
* Lai ghép có chu trình (CX Cycle Crossover)
* Lai ghép thứ tự tuyến tính (LOX Linear Order Crossover)

**Toán tử đột biến**

Cũng giống như lai ghép, toán tử đột biến làm tăng nhanh quá trình hội tụ, nhưng tăng một cách đột ngột, cũng có khi sẽ không gây tác dụng gì một khi không thành công. Không ai có thể đánh giá được phương pháp đột biến nào tốt hơn, do đó có một vài phương pháp đơn giản, cũng có vài trường hợp khá phức tạp. Người ta thường chọn một trong những phương pháp sau :

* Đột biến đảo ngược (Inversion Mutation)
* Đột biến chèn (Insertion Mutation)
* Đột biến thay thế (Displacement Mutation)
* Đột biến tương hỗ (Reciprocal Exchange Mutation)
* Đột biến chuyển dịch (Shift Mutation)

**Điều kiện kết thúc**

Thoát ra quá trình tiến hóa quần thể, dựa vào bài toán mà có các cách kết thúc vấn đề khác nhau, một khi đã đạt đến mức yêu cầu. Một vài trường hợp thông thường như sau:

* Kết thúc theo kết quả: một khi đạt đến mức giá trị yêu cầu thì chấm dứt ngay quá trình thực hiện.
* Kết thúc dựa vào số thế hệ: chọn số thế hệ, quá trình sẽ dừng đúng ngay số thế hệ đã qui định trước, không cần biết kết quả như thế nào.
* Tính theo thời gian: không cần biết đã bao nhiêu thế hệ hay kết quả nào, chỉ dựa vào số giờ qui định mà kết thúc.
* Tổ hợp: dùng nhiều phương án khác nhau cho vấn đề, chẳng hạn như : chạy theo số thế hệ xong sau đó đánh giá cho chạy theo kết quả, hoặc ngược lại.

## Xây dựng mô phân tích chi phí bệnh viện

### Giới thiệu

Việc dự báo dữ liệu là một bài toán rất phức tạp, cả về số lượng dữ liệu cần quan tâm cũng như độ chính xác của dữ liệu dự báo. Do vậy, việc cân nhắc để có thể chọn được mô hình phù hợp cho việc dự báo dữ liệu là một việc rất khó khăn (chỉ có thể bằng phương pháp thử-sai). Tuy nhiên, thuật toán di truyền GA là thuật toán được ứng dụng rất rộng rãi trong các lĩnh vực: nhận dạng, phân lớp, dự báo… đã được thực tế chứng tỏ là một công cụ tốt áp dụng cho các bài toán trong lĩnh vực dự báo dữ liệu.

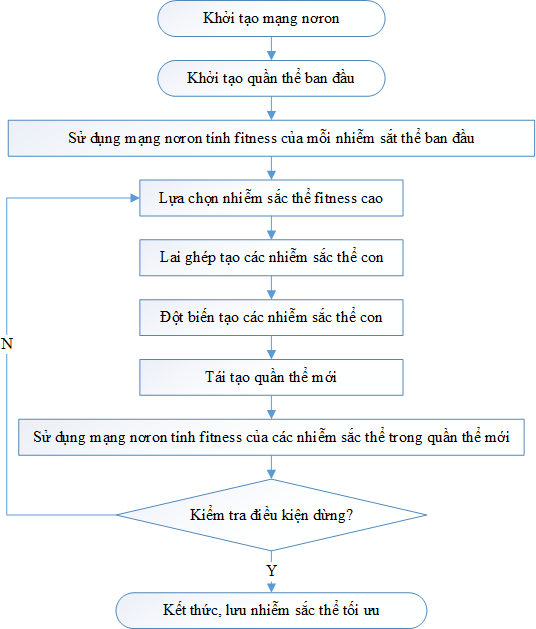
Do đặc trưng về độ phức tạp dữ liệu, các dữ liệu đầu ra thường là các con số (mảng các số) dấu phảy động cho nên việc lựa chọn cấu trúc mạng phù hợp thường là sử dụng phương pháp thử-sai (trial and errors). Đồng thời cần phải chuẩn hóa (loại bỏ các dữ liệu sai, thừa, đưa chúng về đoạn [0,1] hoặc [-1,1],...) các dữ liệu đầu vào và đầu ra để mạng có khả năng học tốt hơn từ các dữ liệu được cung cấp.

Trong việc dự báo dữ liệu, nếu dữ liệu ở nhiều khoảng thời gian khác nhau được đưa vào mạng để huấn luyện thì việc dự báo chính xác là rất khó nếu như mục đích là dự báo chính xác 100% dữ liệu trong tương lai. Ta chỉ có thể có được kết quả dự báo với một mức độ chính xác nào đó chấp nhận được.

### Giới thiệu bài toán

Chi phí cho chăm sóc y tế đang ngày càng tăng lên một cách nhanh chóng, các nhà cung cấp dịch vụ y tế, các phòng khám, bệnh viện ngày càng chịu áp lực để giảm chi phí, và áp lực còn nhiều nhiều hơn nữa trong việc gia tăng chất lượng dịch vụ và an toàn người bệnh, giảm thời gian chờ của bệnh nhân, và giảm thiểu các sai sót và các vụ kiện tụng liên quan. Tổng thời gian từ lúc bệnh nhân bước vào bệnh viện đến khi hoàn tất quá trình khám bệnh, việc đưa ra các quyết định điều trị và các cận lâm sàng phù hợp… đã trở thành thước đo chính cho sự cải tiến trong dịch vụ y tế. Vì vậy, phân tích chi phí bệnh viện (HCA) đang trở thành một chủ đề quan trọng và hàng đầu trong lĩnh vực chính trị, phúc lợi xã hội và các diễn đàn y tế [5]. Dựa trên kết quả HCA, các nhà cung cấp dịch vụ y tế, các phòng khám, bệnh viện có thể xây dựng nhiều kế hoạch để nâng cao chất lượng dịch vụ ,đầu tư vào các dịch vụ chăm sóc sức khỏe tiềm năng để cung cấp dịch vụ chất lượng tốt với chi phí thấp hơn.

### Mô hình



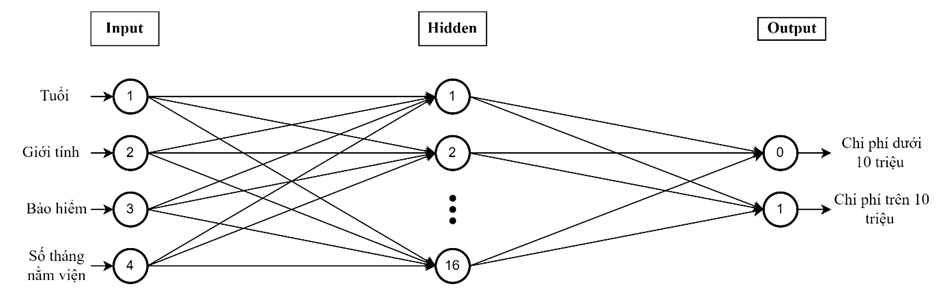
Hình 16 Mô hình bài toán

### Các bước xây dựng mô hình

**Bước 1:** Khởi tạo mạng nơron truyền thẳng

* Số lượng đầu vào là 4, số nơron tầng vào là 4.
* Số nơron tầng ẩn : 16
* Số nơron tầng ra 2 nơron.
* Hàm kích hoạt được sử dụng trong mạng là hàm ReLU.

Xây dựng được mô hình cấu trúc mạng nơron cho bài toán như sau:



Hình 17 Mô hình cấu trúc mạng nơron truyền thẳng ứng dụng trong bài toán

**Bước 2:** Khởi tạo quần thể ban đầu.

Các tham số khởi tạo quần thể:

* Số lượng quần thể ban đầu : 500 nhiễm sắc thể.
* n số nơron tầng ra (cụ thể bài toán này n = 2).
* Số thế hệ: 60

**Bước 3:** Sử dụng mạng nơron truyền thẳng tính fitness của mỗi nhiễm sắc thể được khởi tạo ở *Bước 2* (hàm fitness được tính bằng độ chính xác của mô hình phân lớp).

Trong đó:

* NumCorrectClassify là số mẫu được phân loại chính xác.
* TotalNumberSamples là tổng số mẫu học.

**Bước 4:** Lựa trọn một nhiễm sắc thể có fitness cao nhất

**Bước 5:** Lai ghép

* Lựa chọn các nhiễm sắc thể cha mẹ có fitness cao.
* Sau đó thực hiện lai ghép hai nhiễm sắc thể cha mẹ để tạo ra các nhiễm sắc thể con mới.
* Xác suất lai ghép: 0,9 đến 0,95

**Bước 6:** Đột biến

* Các nhiễm sắc thể mới được tạo ra ở Bước 5 đều được đột biến phụ thuộc vào xác suất đột biến: 0,05 đến 0,2

**Bước 7:** Thực hiện giống *Bước 3* để tính fitness trên quần thể mới

**Bước 8:** So sánh điều kiện dừng (số thế hệ = 60)

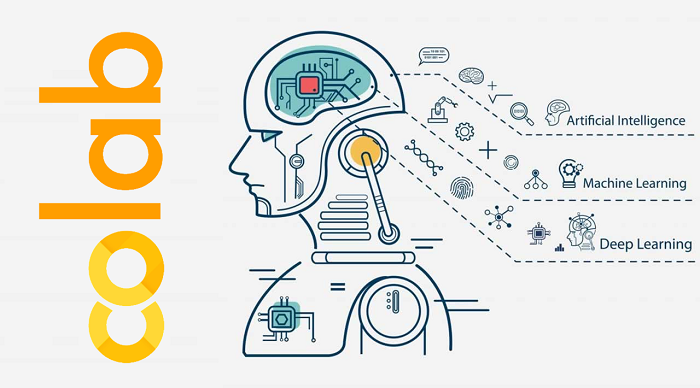
* Nếu thỏa điều kiện dừng:
* Kết thúc huấn luyện
* Lữu nhiễm sắc thể *(bộ trọng số)* tối ưu
* Nếu không thỏa điều kiện dừng:
* Quay lại *Bước 4*



# KẾT QUẢ THỰC NGHIỆM

## Môi trường thực nghiệm

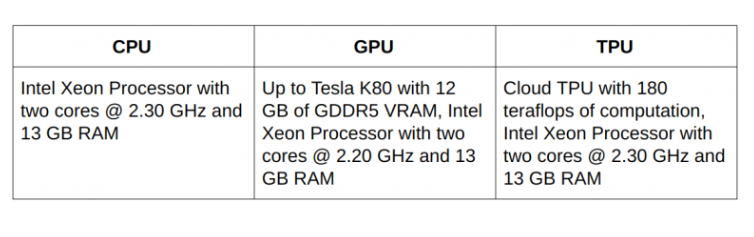
Thực nghiệm được triển khai trên Google Colaboratory. Google Colaboratory (gọi tắt là Google Colab hay Colab) là một sản phẩm của Google Research. Colab dựa trên Jupyter Notebook, người dùng có thể viết và thực thi đoạn mã python thông qua trình duyệt và đặc biệt rất phù hợp với data analysis, machine learning và giáo dục [6].



Hình 18 Ảnh minh họa Colab

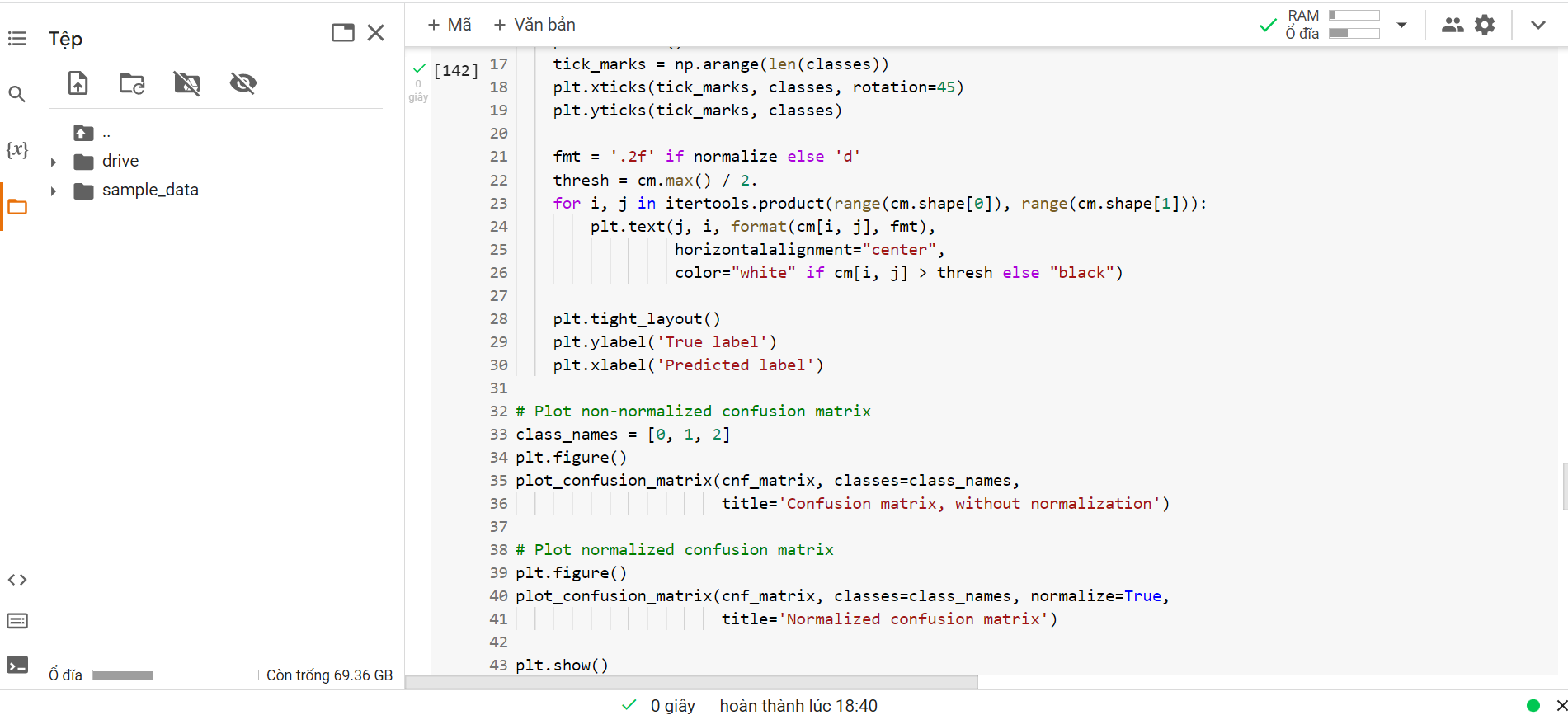
Sự phát triển mạnh mẽ của Machine learning và Deep learning trong những năm gần đây không chỉ bởi các thuật toán, các mô hình tân tiến liên tiếp ra đời mà còn bởi sự phát triển không ngừng của phần cứng, đặc biệt là GPU. Việc tính toán về toán học cho Deep Learning trên CPU có thể mất hàng tháng! Nhưng những tính toán này có thể được gán cho GPU để thực hiện nhanh hơn. Việc train Neural Network trên CPU không được khuyến nghị. GPU cần thiết cho việc tính toán ở mức độ cao.

Như chúng ta đều biết, GPU rất đắt tiền và các nền tảng đám mây như AWS và GCP cũng vậy. Vậy bạn có thể lấy GPU ở đâu? Colab xuất hiện để giúp giải quyết vấn đề này! Colab cung cấp cho bạn GPU Nvidia Tesla K80 miễn phí có giá khoảng 1,7 lakh Rupee (tương đương hơn 53 triệu đồng) và là một “con quái vật” thực sự về hiệu suất.



Hình 19 Cấu hình

Google Colab (Colaboratory) là công cụ thường dùng của các nhà khoa học dữ liệu sử dụng Python. Google Colab (Colaboratory) cài đặt sẵn cho chúng ta những thư viện rất phổ biến trong nghiên cứu Deep Learning như Scipy, Numpy, Pandas , Tensorflow và Keras…

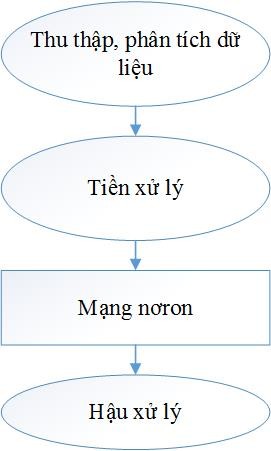


Hình 20 Màn hình làm việc với Google Colab

## Cách thu thập, phân tích và xử lý dữ liệu

Dữ liệu đóng vai trò rất quan trọng trong các giải pháp sử dụng mạng nơron. Chất lượng, độ tin cậy, tính sẵn có và phù hợp của dữ liệu được sử dụng để phát triển hệ thống giúp cho các giải pháp thành công. Các mô hình đơn giản cũng có thể đạt được những kết quả nhất định nếu như dữ liệu được xử lý tốt, bộc lộ được các thông tin quan trọng. Bên cạnh đó, các mô hình tốt có thể sẽ không cho ta các kết quả mong muốn nếu dữ liệu đưa vào quá phức tạp và rắc rối.

Việc xử lý dữ liệu bắt đầu bằng việc thu thập và phân tích dữ liệu, sau đó là bước tiền xử lý. Dữ liệu sau khi qua bước tiền xử lý được đưa vào mạng nơron. Cuối cùng, dữ liệu đầu ra của mạng nơron qua bước hậu xử lý, bước này sẽ thực hiện biến đổi kết quả trả về của mạng nơron sang dạng hiểu được yêu cầu của bài toán. Sau đây, ta xem quá trình xử lý dữ liệu.



Hình 21 Mô hình xử lý dữ liệu

### Thu thập dữ liệu

Thu thập dữ liệu bao gồm 3 bước.

#### Xác định yêu cầu dữ liệu

Đầu tiên cần thực hiện khi lập kế hoạch thu thập dữ liệu là xác định xem các dữ liệu nào là cần thiết để có thể giải quyết bài toán. Về tổng thể, ta có thể cần sự trợ giúp của các chuyên gia trong lĩnh vực của bài toán cần giải quyết.

Ta cần phải biết

* Các dữ liệu chắc chắn có liên quan đến bài toán
* Các dữ liệu nào có thể liên quan
* Các dữ liệu nào là phụ trợ

Các dữ liệu có liên quan và có thể liên quan đến bài toán cần phải được xem là các đầu vào cho hệ thống.

#### Xác định nguồn dữ liệu

Bước kế tiếp là quyết định nơi sẽ lấy dữ liệu, điều này cho phép ta xác định được các ước lượng thực tế về những khó khăn và phí tốn cho việc thu thập dữ liệu. Nếu ứng dụng yêu cầu các dữ liệu thời gian thực, những ước lượng này cần tính đến khả năng chuyển đổi các dữ liệu tương tự thành dạng số.

Trong một số trường hợp, ta có thể chọn dữ liệu mô phỏng từ các tình huống thực tế. Tuy nhiên, cần phải quan tâm đến độ chính xác và khả năng thực hiện của dữ liệu đối với các trường hợp cụ thể.

#### Xác định lượng dữ liệu

Ta cần phải ước đoán số lượng dữ liệu cần thiết để có thể sử dụng trong việc xây dựng mạng. Nếu lấy quá ít dữ liệu thì những dữ liệu này sẽ không thể phản ánh toàn bộ các thuộc tính mà mạng cần phải học và do đó mạng sẽ không có được phản ứng mong đợi đối với những dữ liệu mà nó chưa được huấn luyện. Mặt khác, cũng không nên đưa vào huấn luyện cho mạng quá nhiều dữ liệu. Về tổng thể, lượng dữ liệu cần thiết bị chi phối bởi số các trường hợp cần luyện cho mạng. Bản chất đa chiều của dữ liệu và cách giải quyết mong muốn là các nhân tố chính xác định số các trường hợp cần luyện cho mạng và kéo theo là lượng dữ liệu cần thiết.

Việc định lượng gần đúng lượng dữ liệu cần đưa vào luyện mạng là hết sức cần thiết. Thông thường, dữ liệu thường thiếu hoàn chỉnh, do đó nếu muốn mạng có khả năng thực hiện được những điều mà ta mong đợi thì nó cần phải được luyện với lượng dữ liệu lớn hơn. Đương nhiên, nếu có được độ chính xác và hoàn chỉnh của dữ liệu thì số các trường hợp cần thiết phải đưa vào mạng có thể giảm đi.

### Phân tích dữ liệu

Có hai kỹ thuật cơ bản phân tích dữ liệu [7].

#### Phân tích thống kê

Mạng nơron có thể được xem như là một mở rộng của các phương pháp thống kê chuẩn. Các thử nghiệm có thể cho ta biết được khả năng mà mạng có thể thực hiện. Hơn nữa, phân tích có thể cho ta các đầu mối để xác định các đặc trưng.

Ví dụ, nếu dữ liệu được chia thành các lớp, các thử nghiệm thống kê có thể xác định được khả năng phân biệt các lớp trong dữ liệu thô hoặc dữ liệu đã qua tiền xử lý.

#### Trực quan hóa dữ liệu

Trực quan hóa dữ liệu bằng cách vẽ biểu đồ trên các dữ liệu theo một dạng thích hợp sẽ cho ta thấy được đặc trưng phân biệt của dữ liệu, chẳng hạn như: các điểm lệch hay các điểm đỉnh. Điều này nếu thực hiện được, có thể áp dụng thêm các thao tác tiền xử lý để tăng cường các đặc trưng đó.

Thông thường, phân tích dữ liệu bao gồm cả các kiểm tra thống kê và trực quan hóa. Các kiểm tra này sẽ được lặp đi lặp lại. Trực quan hóa cho ta sự đánh giá về dữ liệu và các khái niệm sơ khởi về các mẫu nằm sau dữ liệu. Trong khi các phương pháp thống kê cho phép ta kiểm thử những khái niệm này.

### Xử lý dữ liệu

#### Tiền xử lý

Tiền xử lý dữ liệu là một kỹ thuật để chuyển đổi dữ liệu thô sang một định dạng dễ hiểu hơn. Dữ liệu trong thế giới thực (dữ liệu thô) luôn không hoàn hảo và bạn không thể gửi dữ liệu đó qua mô hình vì nó phát sinh một số lỗi nhất định . Do đó, trước tiên chúng ta phải xử lý trước dữ liệu để giảm khả năng xảy ra lỗi của tập dữ liệu trước khi gửi tập dữ liệu qua mô hình [8].

Chuyển đổi dữ liệu về khuôn dạng phù hợp đối với đầu vào của mạng nơron – điều này thường đơn giản hóa quá trình xử lý của mạng phải thực hiện trong thời gian ngắn. Các chuyển đồi này có thể bao gồm:

* Áp dụng một hàm toán học cho đầu vào
* Mã hóa các dữ liệu văn bản trong cơ sở dữ liệu
* Chuyển đổi dữ liệu sao cho nó có giá trị nằm trong khoảng [0, 1]

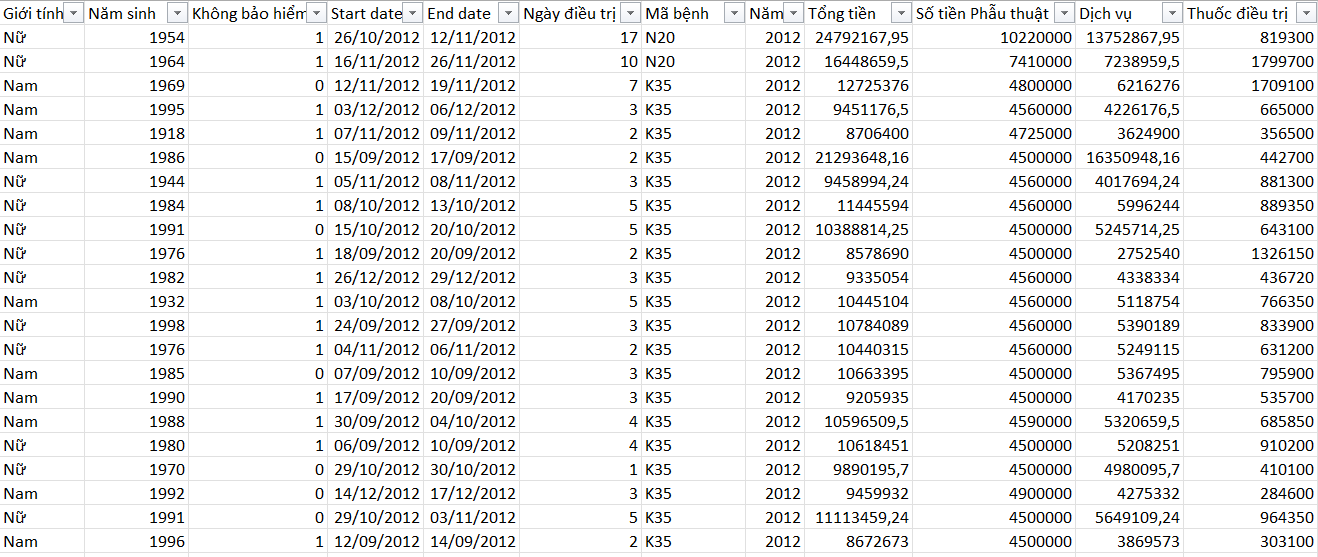
Lựa chọn các dữ liệu xác đáng nhất – việc lựa chọn này cần cẩn thận các dữ liệu phù hợp sẽ làm cho mạng dễ xây dựng và tăng cường hiệu năng của chúng đối với các dữ liệu nhiễu.

#### Hậu xử lý

Hậu xử lý bao gồm các xử lý áp dụng cho đầu ra của mạng. Hậu xử lý hoàn toàn phụ thuộc vào các ứng dụng cụ thể và có thể bao gồm cả việc phát hiện các tham số có giá trị vượt quá khoảng cho phép hoặc sử dụng đầu ra của mạng như một đầu vào của một hệ khác. Đôi khi, hậu xử lý chỉ đơn giản là quá trình ngược lại đối với quá trình tiền xử lý.

## Mô tả dữ liệu bài toán

Trong nghiên cứu, đồ án sử dụng cơ sở dữ liệu thực của Bệnh viện Đại học Y Hà Nội (HMUS) từ năm 2011 đến năm 2015 bao gồm các thuộc tính: thông tin chung của bệnh nhân (giới tính, năm sinh, nơi sinh); thông tin điều trị (mã ICD, thời gian vào / ra viện, bảo hiểm); Tổng chi phí nằm viện (phẫu thuật, dịch vụ, thuốc) [3].

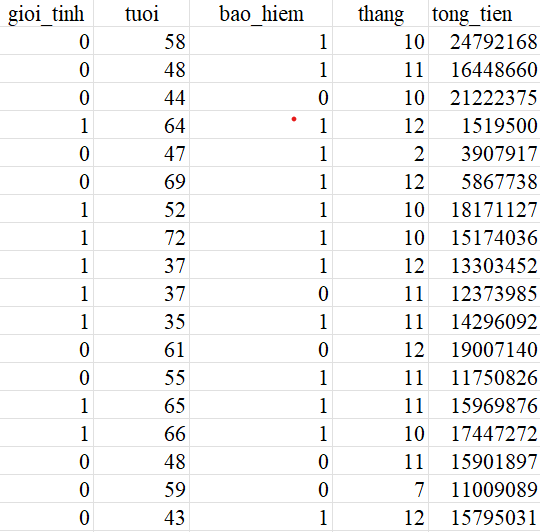


Hình 22 Hình ảnh dữ liệu thô được thu thập chưa qua xử lý

## Tiền xử lý dữ liệu

Các bước tiền xử lý dữ liệu:

* Trước tiên xử lý dữ liệu trực tiếp ở bên ngoài Excel như :
* Chuyển đổi file dữ liệu gốc ban đầu ở định dạng ‘xlsx’ sang định dạng ‘csv’ và loại bỏ một số cột không cần thiết.
* Xử lý dữ liệu bị thiếu: Dữ liệu bị thiếu có thể do nhiều nguyên nhân như dữ liệu bị mất, dữ liệu bị người dùng quên điền, …..
* Tính cột ‘tuoi’ = ‘nam’ – ‘nam\_sinh’
* Mã hóa các dữ liệu văn bản trong cơ sở dữ liệu (‘gioi\_tinh’:nam = 1, nu = 0)
* Chuyển đổi dữ liệu sao cho nó có giá trị nằm trong khoảng [0, 1]
* Phân loại cột ‘tong\_tien’ thành 2 loại :’dưới 10 triệu’ là 0, trên 10 triệu’ là 1



Hình 23 Dữ liệu sau khi qua xử lý excel

* Xử lý dữ liệu trong chương trình
* Nhập các thư viện cần thiết để xử lý dữ liệu trong chương trình.Thư viện là một tập hợp các mô-dun có thể được gọi và sử dụng như: Numpy, pandas, matplotlib,…
* Import dữ liệu: Đa số các tập dữ liệu đều được lưu dưới định dạng file csv và được import qua phương thức read\_csv(‘tên file’) trong thư viện pandas

## Cách đánh giá mô hình

Để đánh giá mức độ chính xác của mô hình ở đây ta sử dụng 4 chỉ số : Accuracy(độ chính xác), Precision, Recall, F1-score

Xét bài toán phân loại nhị phân với các chỉ số sau:

* True Positive (TP): dự đoán và thực tế đều là positive (mô hình phân loại đúng mẫu positive)
* False Positive (FP): thực tế negative là nhưng dự đoán lại là positive (mô hình bị nhầm lẫn mẫu positive là negative)
* True Negative (TN): dự đoán và thực tế đều là negative (mô hình phân loại đúng mẫu negative):
* False Negative (FN): thực tế là positive nhưng dự đoán lại là negative (mô hình bị nhầm lẫn mẫu negative là positive)
* Accuracy: độ chính xác được tính một cách đơn giản và hay được sử dụng:

(3-1)

* Precision: được tính bằng tỷ lệ số mẫu True Positive (TP) trên tổng số những mẫu được phân loại là Positive(TP+FP)

* Recall: được tính bằng tỷ lệ số mẫu True Positive (TP) trên tổng số mẫu thực sự là Positive(TP+FN)

* F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). FI-score được tính như sau:

## Kết quả thực nghiệm

**Đầu vào**: Tuổi , Giới tính, Bảo hiểm, Số tháng nằm viện.

Mỗi mẫu dữ liệu đầu vào là một vecto 4 chiều:

X=() với (i = 1, …, 840)

**Đầu ra**: Tổng chi phí bệnh viện được chia thành 2 khoảng là dưới 10 triệu và trên 10 triệu.

**Cấu hình mạng nơron truyền thẳng:**

* Số nơron tầng vào là 4
* Số nơron tầng ẩn là 16
* Số nơron tầng ra là 2

**Cấu hình thông số giải thuật di truyền:**

* Số lượng cá thể trong quần thể (Population Size) là 500
* Số thế hệ là 60
* Phương pháp chọn lọc: Chọn lọc xếp hạng (rank), chọn lọc bằng bánh xe Roulette (rws), chọn lọc cạnh tranh (tournament), chọn lọc ngẫu nhiên (random).
* Phương pháp lai ghép: lai ghép một điểm (single\_point), lai ghép 2 điểm (two\_point), lai ghép đồng nhất (uniform).
* Xác suất lai ghép là 0,9 – 0,95
* Phương pháp đột biến: Đột biến hoán đổi (swap), đột biến ngẫu nhiên (random), đột biến đảo ngược (inversion)
* Xác suất đột biến là 0,05 – 0,2

**Đánh giá mô hình cho bài toán:**

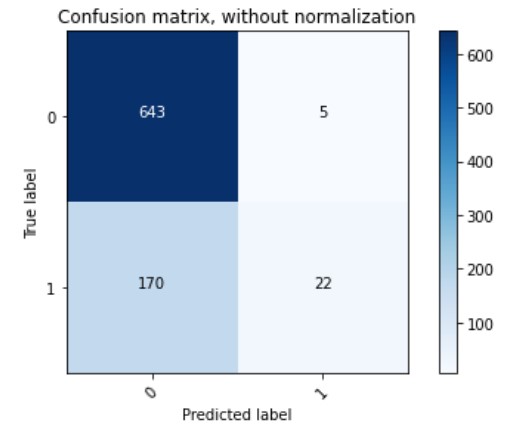
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Xác suất lai ghép | Xác suất đột biến | Accuracy | Precision | Recall | F1\_score |
| 0,9 | 0,05 | 79,0476% | 98,7654% | 79,2079% | 87,9121% |
| 0,91 | 0,05 | 79,1667% | 99.,0741% | 79,1615% | 88,0055% |
| 0,92 | 0,05 | 79,0476% | 99,0741% | 79,0640% | 87,9452% |
| 0,93 | 0,05 | 78,8095% | 99,3827% | 78,4409% | 87,6787% |
| 0,94 | 0,05 | 77,8571% | 100% | 77,6978% | 87,4494% |
| 0,95 | 0,05 | 78,4524% | 99,0741% | 78,5802% | 87,6451% |
| 0,91 | 0,1 | 78,4524% | 99,6914% | 78,3030% | 87,7122% |
| 0,91 | 0,15 | 78,4524% | 99,6914% | 78,3030% | 87,7122% |
| 0,91 | 0,2 | 78,4524% | 98,4568% | 78,2238% | 87,5772% |

Bảng 3 Thống kê so sánh về độ chính xác khi thay đổi xác suất lai ghép và xác suất đột biến.

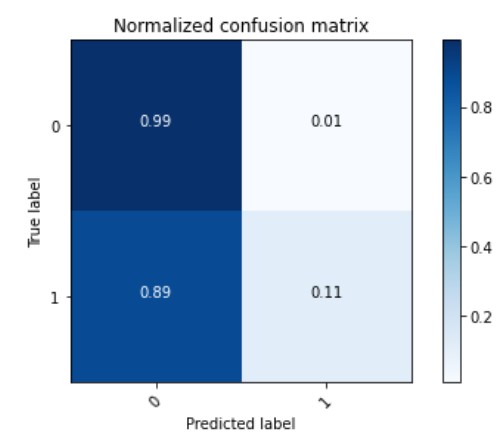
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phương pháp chọn cha mẹ | Phương pháp lai ghép | Phương pháp đột biến | Accuracy | Precision | Recall | F1\_score |
| rws | single\_point | random | 77,6190% | 97,2222% | 78,7500% | 87,0165% |
| rank | single\_point | random | 79,1667% | 99.,0741% | 79,1615% | 88,0055% |
| tournament | single\_point | random | 77,7381% | 99,3827% | 77,8718% | 87,6526% |
| random | single\_point | random | 78,3333% | 99,8457% | 78,1401% | 87,6694% |
| rank | two\_points | random | 78,8095% | 99,5370% | 78,6585% | 87,8746% |
| rank | uniform | random | 78,6904% | 99,5370% | 78,5627% | 87,8148% |
| rank | single\_point | swap | 78,0952% | 99,2284% | 78,2238% | 87,4829% |
| rank | single\_point | inversion | 77,6190% | 98,9198% | 77,9805% | 87,2108% |

Bảng 4 Thống kê so sánh về độ chính xác khi thay đổi một số phương pháp chọn lọc, lai ghép và đột biến trong GA

**Kết quả dự đoán cao nhất thông qua ma trận nhiễu:**



Hình 24 Ma trận chưa chuyển hóa đánh giá kết quả dự đoán bài toán



Hình 25 Ma trận chuyển hóa đánh giá kết quả dự đoán bài toán

**Nhận xét đánh giá mô hình:**

* Khi tăng xác suất đột biến, ta thấy mức dộ chính xác của mô hình bị giảm.
* Từ 2 ma trận nhiễu, ta thấy độ chính xác của 2 nhãn khá lệch. Nhãn 0 có độ chính xác cao hơn hẳn so với nhãn 1.
* Qua quá trình chạy thử với xắc suất lai ghép là [0,9; 0,95] , xắc suất đột biến là [0,05; 0,2] thì Accuracy luôn đạt trong khoảng [0,7845; 0,7917] và F1\_score luôn đạt trong khoảng [0,8702; 0,8801].
* Kết quả thử nghiệm tốt nhất đạt Accuracy=79,1667%, F1\_score=88,0055%.
* Từ các kết quả trên, ta thấy mô hình dự đoán với 2 nhãn 0,1 cho kết quả trung bình tốt nên độ tin cậy ở mức khá.

**Nhận xét chung:**

Sau khi chạy bộ dữ liệu dùng mạng noron và thuật toán di truyền trên python ta rút ra được ý nghĩa như sau:

* Đánh giá được mối tương quan giữa các thông tin của bệnh nhân với chi phí khám chữa bệnh của bệnh nhân. Từ đó, cho thấy khả năng chi trả của bệnh nhân.
* Thông qua các đánh giá trên mô hình, ta có thể đưa ra các đề xuất giúp các cơ sở y tế xây dựng nhiều kế hoạch để nâng cao chất lượng dịch vụ, đầu tư vào các dịch vụ chăm sức sức khỏe tiềm năng nhằm cung cấp dịch vụ tốt với chi phí thấp để tăng khả năng thu hồi kinh phí.

KẾT LUẬN

Hiện nay, phân tích chi phí bệnh viện (HCA) đã trở thành một chủ đề quan trọng và hàng đầu trong lĩnh vực phúc lợi xã hội và y tế. Dựa trên các kết quả dự báo về HCA, các đơn vị trong lĩnh vực phúc lợi xã hội và y tế có thể xây dựng nhiều kế hoạch để nâng cao chất lượng dịch vụ của bệnh viện, đầu tư vào các dịch vụ chăm sóc sức khỏe tiềm năng cũng như cung cấp dịch vụ chất lượng tốt với chi phí thấp hơn.

Đồ án tốt nghiệp với đề tài “Sử dụng mạng noron kết hợp với giải thuật di truyền trong phân tích chi phí Bệnh viện Đại học Y Hà Nội” em đã thu được một số kết quả cũng như nhận thấy một số hạn chế như sau:

Kết quả đạt được:

* Tìm hiểu tổng quan về mạng noron, mạng nơron nhân tạo truyền thẳng, thuật toán di truyền.
* Nghiên cứu về mô hình mạng nơron nhân tạo truyền thẳng và thuật toán di truyền.
* Nghiên cứu áp dụng mô hình mạng nơron nhân tạo truyền thẳng và thuật toán di truyền lập trình bằng ngôn ngữ Python
* Lập trình phân tích chi phí bệnh viện với mạng nơ-ron động và thuật toán di truyền.
* Đánh mô hình dựa trên bốn chỉ số Accuracy (độ chính xác), Precision, Recall, F1-score và ma trận nhiễu (Confusion matrix).

Một số hạn chế:

* Mô hình còn nhiều hạn chế nhưng tiêu biểu là còn làm trên tập dữ liệu nhỏ, kết quả huấn luyện mạng vẫn chưa thật sự đạt được độ chính xác cao, thời gian huấn luyện vẫn tương đối dài, chưa đưa được demo về chương trình hoàn chỉnh.
* Mặc dù đã rất cố gắng trong quá trình thực hiện để hoàn thành đồ án tốt nhất, song do năng lực và trình độ còn hạn chế nên khó tránh khỏi những thiếu sót. Kính mong các thầy cô và các bạn quan tâm và phản hồi để ứng dụng của em hoàn thiện tốt hơn.

Hướng phát triển:

Cải tiến thuật toán huấn luyện mạng nơron, kết hợp thuật toán di truyền với một số thuật toán khác như lan truyền ngược… trong việc tối ưu trọng số tăng cao độ chính xác và giảm thời gian học của mạng nơron truyền thẳng. Đối với ứng dụng mạng nơron nhân tạo truyền thẳng vào bài toán dự đoán độ tin cậy và tính hợp lệ của phân tích chi phí bệnh viện đây là mô hình có thể ứng dụng hiệu quả. Vì vậy, trong tương lai luận văn hướng phát triển là xây dựng phần mềm thành bộ công cụ hỗ trợ hoàn thiện hơn trong lĩnh vực nghiên cứu, đánh giá độ tin cậy và tính hợp lệ của phân tích chi phí bệnh viện.Tuy nhiên, em nhận thấy để kết quả ứng dụng phân tích chi phí bệnh viện đạt kết quả tốt ảnh hưởng rất lớn từ tập dữ liệu đầu vào. Vì thế, đồng hành với việc phát triển ứng dụng em cũng sẽ mở rộng thu thập, tìm kiếm thêm nhiều bộ số liệu phục vụ huấn luyện đầu vào.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | A. K. Jain, Mao, Jianchang and K. Mohiuddin, in *Artificial Neural Networks: A Tutorial*, IBM Alamaden Research Center, 1996, p. 31– 44. |
| [2] | C.-T. Lin and C. G. Lee, Neural fuzzy systems: a neurofuzzy synergism to intelligent systems, Prentice-Hall Inc, 1996. |
| [3] | "Python," [Online]. Available: https://www.duhoctrungquoc.vn/Python. |
| [4] | S. L. a. C. L. Giles, "Overfitting and Neural Networks: Conjugate Gradient and Backpropagation," *International Joint onference on Neural Networks,* p. 114–119., 2000. |
| [5] | L. H. Son, A. Ciaramella, D. T. T. Huyen, A. Staiano, T. M. Tuan and P. V. Hai, "Predictive reliability and validity of hospital cost analysis with dynamic neural network and genetic algorithm," *Springer,* 2018. |
| [6] | T. Vu, "Thinh Vu Blog," 2020. [Online]. Available: https://thinhvu.com/2021/07/29/huong-dan-su-dung-google-colab-tutorial-101/. |
| [7] | "caohoc," [Online]. Available: https://caohock24.files.wordpress.com/2012/11/bai-giang-khai-pha-du-lieu.doc. |
| [8] | T. X. Nam. [Online]. Available: https://txnam.net/course/nhap-mon-lap-trinh-khoa-hoc-du-lieu-k58. |