|  |  |
| --- | --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **NGUYỄN ĐỨC HẬU**  **NGÔ VĂN THƯỜNG**  **KHÓA 15**  **HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **CHUYÊN NGÀNH: CÔNG NGHỆ DỮ LIỆU**  **ỨNG DỤNG KỸ THUẬT HỌC SÂU TRONG XÂY DỰNG DỊCH VỤ DỰ BÁO GIÁ PHÒNG CHO THUÊ**  **HÀ NỘI, NĂM** **2020** | |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **NGUYỄN ĐỨC HẬU**  **NGÔ VĂN THƯỜNG**  **KHÓA 15**  **HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH: CÔNG NGHỆ THÔNG TIN**.  **MÃ SỐ: 52480201**  **ỨNG DỤNG KỸ THUẬT HỌC SÂU TRONG XÂY DỰNG DỊCH VỤ DỰ BÁO GIÁ PHÒNG CHO THUÊ**  ***Cán bộ hướng dẫn: 2//, GV, TS. Cao Văn Lợi***  **HÀ NỘI, NĂM** **2020** | |
|  |  |

|  |  |
| --- | --- |
| BỘ QUỐC PHÒNG | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ** | **ĐỘC LẬP - TỰ DO - HẠNH PHÚC** |

**KHOA: CÔNG NGHỆ THÔNG TIN**

**Phê chuẩn**  Độ mật: …………

Ngày ….. tháng …. năm ……. Số: ………………

**CHỦ NHIỆM KHOA**

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: Nguyễn Đức Hậu Lớp: CNDL15 Khóa: 15

Ngô Văn Thường Lớp: CNDL15 Khóa :15

Ngành: Công nghệ thông tin Chuyên ngành: Công nghệ dữ liệu.

1. Tên đề tài: Ứng dụng kỹ thuật học sâu trong xây dựng dịch vụ dự báo giá phòng cho thuê.

2. Các số liệu ban đầu:

- Quyết định Giao đồ án tốt nghiệp đại học – Học viện KTQS

- Tài liệu tham khảo

3. Nội dung bản thuyết minh:

- Mở đầu

- Chương 1: Tổng quan về học sâu

- Chương 2: Tổng quan về học sâu cho dự báo giá phòng

- Chương 3: Học sâu cho định giá phòng tại Hà Nội

- Chương 4: Xây dựng chương trình định giá

- Kết luận

- Tài liệu tham khảo

4. Số lượng, nội dung các bản vẽ (ghi rõ loại, kích thước và cách thực hiện các bản vẽ) và các sản phẩm cụ thể (nếu có):

Được sử dụng máy tính và máy chiếu để trình chiếu.

5. Cán bộ hướng dẫn:

- Họ và tên: Cao Văn Lợi

- Cấp bậc: Trung tá

- Học hàm, học vị: GV, TS

- Đơn vị: Bộ môn An toàn thông tin - Khoa Công nghệ thông tin

- Hướng dẫn toàn bộ

Ngày giao: ......../......../2020 Ngày hoàn thành ........./........../2020

*Hà Nội, ngày ........ tháng ……..năm* 2020

**Chủ nhiệm bộ môn Cán bộ hướng dẫn**

(Ký, ghi rõ họ tên, học hàm, học vị)

**Học viên thực hiện**

Đã hoàn thành và nộp đồ án *........* tháng *……..*năm2020

(Ký và ghi rõ họ tên)

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc57676294)

[1. Lý do chọn đề tài 1](#_Toc57676295)

[3. Mục đích nghiên cứu 2](#_Toc57676296)

[4. Đối tượng và phạm vi nghiên cứu 3](#_Toc57676297)

[5. Nhiệm vụ nghiên cứu 3](#_Toc57676298)

[6. Phương pháp nghiên cứu 3](#_Toc57676299)

[CHƯƠNG 1 : CÁC KIẾN THỨC LIÊN QUAN 5](#_Toc57676300)

[1.1. Tổng quan về Khai phá dữ liệu 5](#_Toc57676301)

[1.1.1. Các bước khám phá trí thức (KDD) 5](#_Toc57676302)

[1.1.2. Các phương pháp trong Khai phá dữ liệu 7](#_Toc57676303)

[1.2. Giới thiệu về Neural Network 8](#_Toc57676304)

[1.2.1. Mô hình Neural Network 8](#_Toc57676305)

[1.2.2. Kiến thức nền tảng 9](#_Toc57676306)

[1.3. Giới thiệu về Học sâu 16](#_Toc57676307)

[1.4. Diễn giải 18](#_Toc57676308)

[1.4.1. Diễn giải Định lý xấp xỉ Phổ quát 18](#_Toc57676309)

[1.4.2. Diễn giải xác suất 18](#_Toc57676310)

[1.5. Các mô hình, kiến trúc học sâu 19](#_Toc57676311)

[1.5.1. Deep Neural Network (DNN) 19](#_Toc57676312)

[1.5.2. Deep Belief Network (DBN) 20](#_Toc57676313)

[1.5.3. Convolutional Neural Networks (CNN) 20](#_Toc57676314)

[1.5.4. Deep Predictive coding network (DPCN) 22](#_Toc57676315)

[1.5.5. Recurrent Neural Network (RNN) 22](#_Toc57676316)

[1.5.6. Feedforward Neural Network (FNN) 24](#_Toc57676317)

[1.6. Tổng kết chương 24](#_Toc57676318)

[CHƯƠNG 2 : MÔ HÌNH HỌC MÁY CHO ĐỊNH GIÁ PHÒNG 25](#_Toc57676319)

[2.1. Mô hình hồi quy tuyến tính 25](#_Toc57676320)

[2.2. Mô hình KNN 26](#_Toc57676321)

[CHƯƠNG 3 : MÔ HÌNH HỌC SÂU CHO ĐỊNH GIÁ PHÒNG 27](#_Toc57676322)

[3.1. Mô tả dữ liệu 27](#_Toc57676325)

[3.1.1. Nguồn gốc dữ liệu 27](#_Toc57676326)

[3.1.2. Mô tả dữ liệu thô 27](#_Toc57676327)

[3.1.3. Nhận xét đánh giá, xây dựng cơ sở dữ liệu 27](#_Toc57676328)

[3.2. Quy trình xử lý dữ liệu 27](#_Toc57676329)

[3.2.1. Thu thập dữ liệu 27](#_Toc57676330)

[3.2.2. Lựa chọn dữ liệu 27](#_Toc57676331)

[3.2.3. Xử lý dữ liệu 28](#_Toc57676332)

[3.2.4. Biến đổi dữ liệu 28](#_Toc57676333)

[3.3. Lựa chọn thuật toán, xây dựng mô hình học sâu 30](#_Toc57676334)

[3.4. Đánh giá chất lượng mô hình 31](#_Toc57676335)

[CHƯƠNG 4 KẾT QUẢ THỬ NGHIỆM, ĐÁNH GIÁ SO SÁNH 32](#_Toc57676336)

[KẾT LUẬN 32](#_Toc57676337)

[1. Kết luận 32](#_Toc57676338)

[2. Hướng phát triển 32](#_Toc57676339)

[TÀI LIỆU THAM KHẢO 33](#_Toc57676340)

# MỞ ĐẦU

## Lý do chọn đề tài

Không phải ai cũng biết định giá cho thuê phòng, nhiều chủ nhà do chưa biết

đánh giá định lượng giá cho phòng thuê nên thường đưa ra giá cho thuê không phù hợp so với giá cho thuê phòng của thị trường thực tế.Tương tự như vậy, với người thuê phòng cũng khó khăn trong việc định lượng giá phòng như vậy đã hợp lý chưa.

Từ việc không đánh giá đúng giá trị phòng dẫn đến có chủ phòng thì cho thuê nhà giá thấp, có người thì đưa ra giá cho thuê quá cao so với thị trường.Trong một số trường hợp, chủ nhà cho thuê chỉ dựa trên một vài tiêu chí cá nhân để định giá phòng dẫn đến không đồng nhất trong thị trường thuê phòng.

Vì thế đề tài **“Nghiên cứu và ứng dụng học máy cho định giá phòng”** được ra đời nhằm nghiên cứu các giải pháp công nghệ sử dụng các kĩ thuật học sâu để có thể dự báo một cách tương đối nhất giá trị của phòng, căn nhà cho thuê. Điều này giúp không chỉ giúp cho việc định giá trở lên công bằng hơn giữa những người cho thuê, giữa các chủ đầu tư. Đồng thời cũng nó cũng là một kênh tham khảo với những người đi thuê phòng trước khi học tiến hành thuê.

1. **Cơ sở khoa học, tính thực tiễn đề tài**

Cơ sở khoa học:

* Cơ sở lý thuyết về các phương pháp học sâu.

Ý nghĩa thực tiễn đề tài:

Điều 473 tại Bộ Luật dân sự 2015 làm cơ sở định giá thuê nhà:

**“***1. Giá thêu do các bên thỏa thuận hoặc do bên thứ ba xác định theo yêu cầu của các bên thừ, trừ trường hợp luật có quy định khác [1].*

*2. Trường hợp không có thỏa thuận hoặc thỏa thuận không rõ ràng thì giá thuê được xác định theo giá thị trường tại đại điểm và thời điểm giao kết hợp đồng thuê***”**

Điều 129 Luật Nhà ở 2014 cũng quy định:

**“***1. Bên cho thuê và bên thuê nhà ở được thỏa thuận về thời hạn thuê, giá thuê và hình thức trả tiền thuê nhà theo định kỳ hoặc trả một lần; trường hợp Nhà nước có quy định về giá thuê nhà ở thì các bên phải thực hiện theo quy định đó [2].*

*2. Trường hợp chưa hết hạn hợp đồng thuê nhà ở mà bên cho thuê thực hiện cải tạo nhà ở và được bên thuê đồng ý thì bên cho thuê được quyền điều chỉnh giá thuê nhà ở. Giá thuê nhà ở mới do các bên thỏa thuận; trường hợp không thỏa thuận được thì bên cho thuê có quyền đơn phương chấm dứt hợp đồng thuê nhà ở và phải bồi thường cho bên thuê theo quy định của pháp luật.*

*3. Bên cho thuê và bên thuê nhà ở được Nhà nước bảo hộ các quyền và lợi ích hợp pháp trong quá trình thuê và cho thuê nhà ở*.**”**

Như vậy, để định giá một căn phòng cho thuê là thỏa thuận giữa hai bên gia chủ cho thuê và người thuê nhà, điều này làm cho giá thuê nhà rất khó có thể xác định. Việc này dẫn đến hệ lụy gia chủ cho thuê có thể cho thuê với mức giá cao và người thuê có thể phải thuê với một mức giá cao hơn giá thị trường.

Hiện nay, với sự phát triển của công nghệ đặc biệt là các công nghệ trí tuệ nhân tạo, học máy hoàn toàn có thể giải quyết được những khó khăn này bằng cách đưa ra dự báo giá nhà cho thuê một cách khách quan và chân thực nhất.

Với việc áp dụng công nghệ vào việc dự báo giá sẽ giúp cho người cho thuê có thể đưa ra một mức giá phù hợp nhất tránh tình trạng người cho thuê đưa ra mức giá qua cao dẫn đến không tìm được người thuê, còn người thuê sẽ thuê được phòng, căn nhà với mức giá phải chăng phù hợp với mình.

## Mục đích nghiên cứu

* Tìm hiểu một số phương pháp học sâu ứng dụng trong các bài toán định giá phòng.
* Xây dựng bộ dữ liệu cho định giá nhà cho thuê trên địa bàn một số quận/ huyện Hà Nội.

## Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của đề tài là nghiên cứu, thử nghiệm các phương pháp học sâu áp dụng cho định giá nhà sử dụng bộ dữ liệu nhà cho thuê tại Hà Nội.

## Nhiệm vụ nghiên cứu

* Xây dựng bộ dữ liệu cho định giá nhà cho thuê tại Hà Nội.
* Xây dựng mô hình định giá nhà cho thuê tại một số quận, huyện Hà Nội.

## Phương pháp nghiên cứu

Phương pháp nghiên cứu tài liệu:

* Đọc, tìm hiểu các bài báo về định giá nhà cho thuê trên thế giới.
* Đọc, tìm hiều cơ bản về học sâu, các phương pháp sử dụng cho định giá, các bước thực hiện đề tài.
* Tìm hiểu các công cụ lập trình hỗ trợ cho đề tài.

Phương pháp nghiên cứu thực hành: Lập trình thử nghiệm, đánh giá kết quả.

**Phân chia công việc**

|  |  |  |
| --- | --- | --- |
|  | Nguyễn Đức Hậu | Ngô Văn Thường |
| Mở đầu | Chung | Chung |
| Chương 1 | 1.3 Giới thiệu về Học sâu.  1.4 Diễn giải.  1.5 Các mô hình,kiến trúc học sâu. | 1.1 Tổng quan về khái phá dữ liệu.  1.2 Giới thiệu về Neural Network.  1.6 Tổng kết chương. |
| Chương 2 |  |  |
| Chương 3 | Quy trình xử lý dữ liệu   * Thu thập dữ liệu * Lựa chọn dữ liệu   (Tìm hiểu) | Quy trình xử lý dữ liệu   * Xử lý dữ liệu * Biến đổi dữ liệu   (Tìm hiểu) |

# : CÁC KIẾN THỨC LIÊN QUAN

## Tổng quan về Khai phá dữ liệu

Thuật ngữ Khai phá dữ liệu (KPDL) ra đời vào cuối những năm 80 thế kỷ trước. Có nhiều định nghĩa khác nhau về KPDL, nhưng diễn đạt một cách dễ hiểu thì KPDL là quá trình tìm kiếm những thông tin (tri thức) có ích, tiềm ẩn và mang tính dự đoán trong các khối cơ sở dữ liệu (CSDL) lớn.

Mục đích việc phát hiện tri thức từ dữ liệu KPDL là cốt lõi của **quá trình khám phá tri thức** gồm có các giải thuật KPDL chuyên dùng, dưới một số quy định về hiệu quả tính toán chấp nhận được. KPDL nhằm tìm ra những mẫu mới, những thông tin tiềm ẩn mang tính dự đoán chưa được biết đến, có khả năng mang lại lợi ích cho người sử dụng và KPDL là tìm ra các mẫu được quan tâm nhất tồn tại trong CSDL, nhưng chúng lại bị che giấu bởi một số lượng lớn dữ liệu.

Khai phá dữ liệu là một bước trong bảy bước của quá trình KDD (Knowleadge Discovery in Database).

Trong bài toán dự đoán giá phòng cho thuê cũng sẽ áp dụng các bước trong khái phá tri thức này.

### Các bước khám phá trí thức (KDD)



Hình 1.1. Quy trình khám phá tri thức

Bước 1- **Thu thập dữ liệu** (data gathering): Là bước lấy các dữ liệu lien quan đến vấn đề từ các nguồn khác nhau (Internet,…).

Bước 2- **Trích chọn dữ liệu** (data selection): Là bước trích chọn những tập dữ liệu cần được khai phá từ các tập dữ liệu lớn (databases,data warehouses).

Bước 3- **Tiền xử lý dữ liệu** (cleansing preprocessing): Là bước làm sạch dữ liệu (xử lý dữ liệu không đầy đủ, dữ liệu nhiễu, dữ liệu không nhất quán,...), rút gọn dữ liệu (sử dụng các phương pháp thu gọn dữ liệu, histograms, lấy mẫu,...), rời rạc hoá dữ liệu (dựa vào histograms, entropy, phân khoảng,…). Sau bước này, dữ liệu sẽ nhất quán, đầy đủ, được rút gọn và được rời rạc hoá.

Bước 4- **Biến đổi dữ liệu** (data transformation): Là bước chuẩn hoá và làm mịn dữ liệu để đưa dữ liệu về dạng thuận lợi nhất nhằm phục vụ cho các kỹ thuật khai phá ở bước sau.

Bước 5- **Khai phá dữ liệu** (data mining): Đây là bước quan trọng và tốn nhiều thời gian nhất của quá trình **khám phá tri thức**, áp dụng các kỹ thuật khai phá (phần lớn là các kỹ thuật của machine learning) để khai phá, trích chọn được các mẫu (pattern) thông tin, các mối liên hệ đặc biệt trong dữ liệu.

Bước 6- **Đánh giá và biểu diễn tri thức** (knowledge representation & evaluation): Dùng các kỹ thuật hiển thị dữ liệu để trình bày các mẫu thông tin (tri thức) và mối liên hệ đặc biệt trong dữ liệu đã được khai phá ở bước trên biểu diễn theo dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật,… Đồng thời, bước này cũng đánh giá những tri thức khai phá được theo những tiêu chí nhất định.

Trong giai đoạn KPDL, có thể cần sự tương tác của người dùng để điều chỉnh và rút ra các tri thức cần thiết nhất. Các**tri thức** nhận được cũng có thể được lưu và sử dụng lại.

Trong đó đề tài sẽ nhấn mạnh vào 4 bước:***Thu thập dữ liệu ,*** ***Tiền xử lý dữ liệu , Biến đổi dữ liệu , Khai phá dữ liệu.***

### Các phương pháp trong Khai phá dữ liệu

#### Phương pháp luật kết hợp

Một trong những chủ đề phổ biến của KPDL là khai phá luật kết hợp. Mục đích của khai phá luật kết hợp là xác định mối quan hệ, sự kết hợp giữa các mục dữ liệu (item) trong một CSDL lớn.

#### Phương pháp cây quyết định

Mô tả tri thức dạng đơn giản nhằm phân các đối tượng dữ liệu thành một số lớp nhất định. Các nút của cây được gán nhãn là tên các mục dữ liệu, các cạnh được gán các giá trị có thể của các mục dữ liệu, các lá mô tả các lớp khác nhau. Các đối tượng được phân lớp theo các đường đi trên cây, qua các cạnh tương ứng với các giá trị của mục dữ liệu tới lá.

#### Phương pháp K-Mean

Có nhiều phương pháp được sử dụng trong phân cụm, phương pháp k-Mean được coi là các kỹ thuật cơ bản của phân cụm. Với phương pháp này sẽ chia tập có n đối tượng thành k cụm sao cho các đối tượng trong cùng một cụm thì giống nhau, các đối tượng khác cụm thì khác nhau.

#### Các phương pháp dựa trên mẫu

Phương pháp này sử dụng khai phá chuỗi theo thời gian (Sequential temporal patterns). Xét về mặt kỹ thuật thì tương tự như KPDL bằng luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Một luật mô tả mẫu tuần tự có dạng tiêu biểu X -> Y phản ánh sự xuất hiện của biến cố X sẽ dẫn đến việc xuất hiện kế tiếp biến cố Y. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán bởi chúng có tính dự báo cao.

## Giới thiệu về Neural Network

Mạng nơ-ron (Neural Network) hay có tên gọi đầy đủ là Artificial Neural network (ANN) là một chuỗi các thuật toán cố gắng nhận ra các mối quan hệ cơ bản trong một tập hợp dữ liệu thông qua một quá trình bắt chước cách bộ não con người hoạt động. Theo nghĩa này, mạng nơ-ron đề cập đến hệ thống nơ-ron, có bản chất sinh học hoặc nhân tạo. Mạng nơron có thể thích ứng với việc thay đổi đầu vào; vì vậy mạng tạo ra kết quả tốt nhất có thể mà không cần thiết kế lại các tiêu chí đầu ra. Khái niệm về mạng thần kinh, có nguồn gốc từ [trí tuệ nhân tạo](https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp) ([**artificial intelligence**](https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp)),và ngày càng nhanh chóng trở nên phổ biến trong sự phát triển của các hệ thống giao dịch.

### Mô hình Neural Network

Mạng nơ-ron ( Neural Network) hoạt động tương tự như mạng lưới thần kinh não bộ của con người. Một "nơ-ron" trong mạng nơ-ron là một hàm toán học thu thập và phân loại thông tin theo một kiến ​​trúc cụ thể. Mạng này khá tương đồng mạnh mẽ với các phương pháp thống kê như phân tích đường cong và phân tích hồi quy.

Một mạng nơ-ron chứa các lớp (layer) là tập hợp các nút được kết nối với nhau. Mỗi nút là một perceptron và tương tự như một [hồi quy tuyến tính bội](https://www.investopedia.com/terms/m/mlr.asp) ( [multiple linear regression](https://www.investopedia.com/terms/m/mlr.asp)). Perceptron cung cấp tín hiệu được tạo ra bởi một hồi quy nhiều tuyến tính thành một hàm kích hoạt có thể là phi tuyến.

Trong một perceptron nhiều lớp (Multi-layer Perceptron), các perceptron được sắp xếp thành các lớp liên kết với nhau. Lớp đầu vào thu thập các mẫu đầu vào. Lớp đầu ra có các phân loại hoặc tín hiệu đầu ra mà các mẫu đầu vào có thể ánh xạ. Ví dụ, các mẫu có thể bao gồm danh sách [các chỉ số kỹ thuật](https://www.investopedia.com/terms/t/technicalindicator.asp) về chứng khoán; đầu ra mong muốn có thể là “mua”, “giữ” hoặc “bán”.

Các lớp ẩn sẽ tinh chỉnh trọng số đầu vào cho đến khi biên độ lỗi của mạng nơ-ron là nhỏ nhất. Giả thuyết rằng các lớp ẩn ngoại suy các đặc điểm nổi bật trong dữ liệu đầu vào có khả năng dự đoán liên quan đến kết quả đầu ra. Điều này mô tả việc trích xuất tính năng, hoàn thành một tiện ích tương tự như các kỹ thuật thống kê, chẳng hạn như phân tích thành phần chính.



Hình 1.2. Kiến trúc cơ bản của Neural Network

* Input layer : Là tầng bao gồm các giá trị đầu vào của mạng.
* Hidden layer : Có thể gồm 1 hoặc nhiều tầng, các tầng này đóng vai trò như các logic, suy luận của mạng.
* Output layer : Thể hiện kết quả đầu ra của mạng.

### Kiến thức nền tảng

Trong phần này sẽ trình bày một số khái niệm liên quan đến Neural Network.Đây là nhưng kiến thức cơ bản nhất.

#### Layers

Layers (tầng ) là tập hợp các node cùng vị trí trong mạng nơ-ron. Một mạng nơ-ron cơ bản bao gồm 3 lớp: *Input layer , Hidden layer, Output layer* . Đôi khi một Multi-layer Perceptron (MLP) có thể có nhiều *Hidden layers* ở giữa. Các *Hidden layers* theo thứ tự từ input layer đến output layer được đánh số thứ thự là *Hidden layer 1, Hidden layer 2,…*

Trong mạng nơ-ron, trừ input layer thì tất cả các node thuộc các layer khác đều full-connected với các node thuộc layer trước nó. Mỗi node thuộc hidden layer nhận vào ma trận đầu vào từ layer trước và kết hợp với trọng số để ra được kết quả.



Hình 1.3. MLP với hai hidden layers

Số lượng layer trong một MLP được tính bằng số hidden layers cộng với 1. Tức là khi đếm số layers của một MLP, ta không tính input layers. Số lượng layer trong một MLP thường được ký hiệu là  . Trong hình trên,  =3.

#### Weights và Biases

Có  ma trận trọng số cho một MLP có  layers. Các ma trận này được ký hiệu là   trong đó  thể hiện các kết nối từ layer thứ  tới layer thứ  (nếu ta coi input layer là layer thứ 0). Cụ thể hơn, phần tử  thể hiện kết nối từ node thứ    của layer thứ  tới node từ    của layer thứ . Các biases của layer thứ  được ký hiệu là . Các trọng số này được ký hiệu như trên *Hình 1.3*. Khi tối ưu một MLP cho một công việc nào đó, chúng ta cần đi tìm các weights và biases này.

Tập hợp các weights và biases lần lượt được ký hiệu làvà .

#### Units

Một Node hình tròn trong một layer được gọi là một unit. Unit ở các input layer, hidden layers, và output layer được lần lượt gọi là input unit, hidden unit, và output unit. Đầu vào của các hidden layer được ký hiệu bởi , đầu ra của mỗi unit thường được ký hiệu là  (thể hiện activation, tức giá trị của mỗi unit sau khi ta áp dụng activation function lên ). Đầu ra của unit thứ    trong layer thứ    được ký hiệu là . Giả sử thêm rằng số unit trong layer thứ  (không tính bias) là . Vector biểu diễn output của layer thứ    được ký hiệu là



Hình 1.4. Thông số đầu vào và ra giữa các layer

#### Activation functions

Trong mạng nơ-ron, một hàm kích hoạt chuẩn hóa đầu vào và tạo ra đầu ra, sau đó được chuyển tiếp vào lớp tiếp theo. Các chức năng kích hoạt thêm tính phi tuyến tính vào đầu ra cho phép mạng nơ-ron giải quyết các vấn đề phi tuyến tính. Nói cách khác, một mạng nơron không có chức năng kích hoạt về cơ bản chỉ là một mô hình hồi quy tuyến tính.

Mỗi output của một unit (trừ các input units) được tính dựa vào công thức:

Trong đó  là một (nonlinear) activation function. Ở dạng vector, biểu thức bên trên được viết là:

Khi activation function  được áp dụng cho một ma trận (hoặc vector), ta hiểu rằng nó được áp dụng cho từng thành phần của ma trận đó. Sau đó các thành phần này được sắp xếp lại đúng theo thứ tự để được một ma trận có kích thước bằng với ma trận input. Trong tiếng Anh, việc áp dụng lên từng phần tử như thế này được gọi là element-wise.

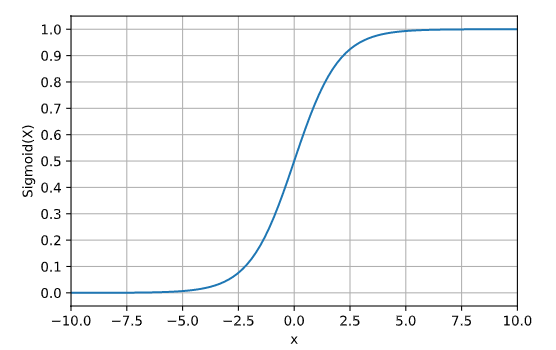
Một số hàm activation phổ biến **Linear**, **Sigmoid**, **Tanh**, and **ReLU**



Hình 1.5. Một số hàm activation

* *Hàm Sigmoid*

Công thức :



Hình 1.6. Đồ thị hàm sigmoid

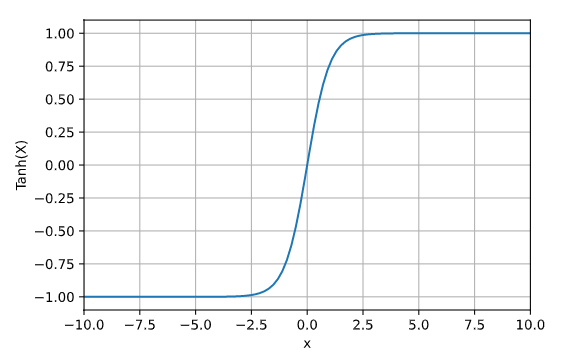
Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1) (xem đồ thị phía trên). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau:

* Hàm Sigmoid bão hào và triệt tiêu gradient**:** Một nhược điểm dễ nhận thấy là khi đầu vào có trị tuyệt đối lớn (rất âm hoặc rất dương), gradient của hàm số này sẽ rất gần với 0. Điều này đồng nghĩa với việc các hệ số tương ứng với unit đang xét sẽ gần như không được cập nhật (còn được gọi là *vanishing gradient*).
* **Hàm Sigmoid không có trung tâm là 0 gây khó khăn cho việc hội tụ.**

Để giải quyết yếu điểm này, chúng ta có thể giải quyết vấn đề này bằng cách chuẩn hoá dữ liệu về dạng có trung tâm là 0 (zero-centered) với các thuật toán batch/layer normalization.

* *Hàm Tanh*

Công thức :



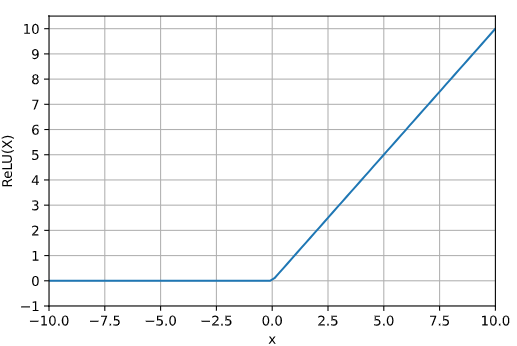
Hình 1.7. Đồ thị hàm Tang

Hàm tanh nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1; 1). Cũng như Sigmoid, hàm bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm lại đối xứng qua 0 nên khắc phục được một nhược điểm của Sigmoid.

Hàm  còn có thể được biểu diễn bằng hàm sigmoid như sau:

* *Hàm ReLU*

Công thức :



Hình 1.8. Đồ thị hàm ReLU

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0. Đựa vào công thức ta có thể dễ dàng hiểu được cách hoạt động của nó. Một số ưu điểm và nhược điểm của ReLU so với Sigmoid và Tanh:

Ưu điểm :

* Tốc độ hội tụ nhanh hơn hẳn. ReLU có tốc độ hội tụ nhanh gấp 6 lần Tanh [5]. Điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh.
* Tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm exp và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.

Nhược điểm :

* ReLU cũng có một nhược điểm: Với các node có giá trị nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đấy gọi là “Dying ReLU“. Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đấy cũng không được cập nhật với gradient descent.
* Khi learning rate lớn, các trọng số (weights) có thể thay đổi theo cách làm tất cả neuron dừng việc cập nhật.

Việc lựa chọn hàm activation cho mạng còn phụ thuộc vào nhiều yếu tố như : Độ sâu của mạng , output mong muốn , việc tìm hiều về các hàm activation sẽ phục vụ cho việc chọn hàm phù hợp khi ta xây dựng cho mô hình dự báo giá phòng được đề cập trong chương sau.

Một số chú ý về việc chọn activation function:

* Các hàm sigmoid và sự kết hợp của chúng thường phù hợp với những bài toán phân loại.
* Sigmoid và tanh đôi khi nên tránh sử dụng đồng thời vì có thể khiến gradient biến mất.
* ReLU là 1 activation function phổ biến và thường dùng nhất hiện nay.
* ReLU function chỉ có thể được sử dụng trong những hidden layer

Trên đây là kiến thức cơ bản, tổng hợp về mạng nơ-ron. Nó là tiền đề cho chúng ta nghiên cứu các mô hình học sâu ngay sau. Tóm tắt lại, mạng nơ-ron là nền tảng của học sâu, được sử dụng rộng rãi, với các ứng dụng cho hoạt động tài chính, lập kế hoạch doanh nghiệp, giao dịch, phân tích kinh doanh và bảo trì sản phẩm. Mạng nơ-ron cũng đã được áp dụng rộng rãi trong các ứng dụng kinh doanh như các giải pháp nghiên cứu tiếp thị và dự báo, phát hiện gian lận và [đánh giá rủi ro](https://www.investopedia.com/terms/r/risk-assessment.asp).

## Giới thiệu về Học sâu

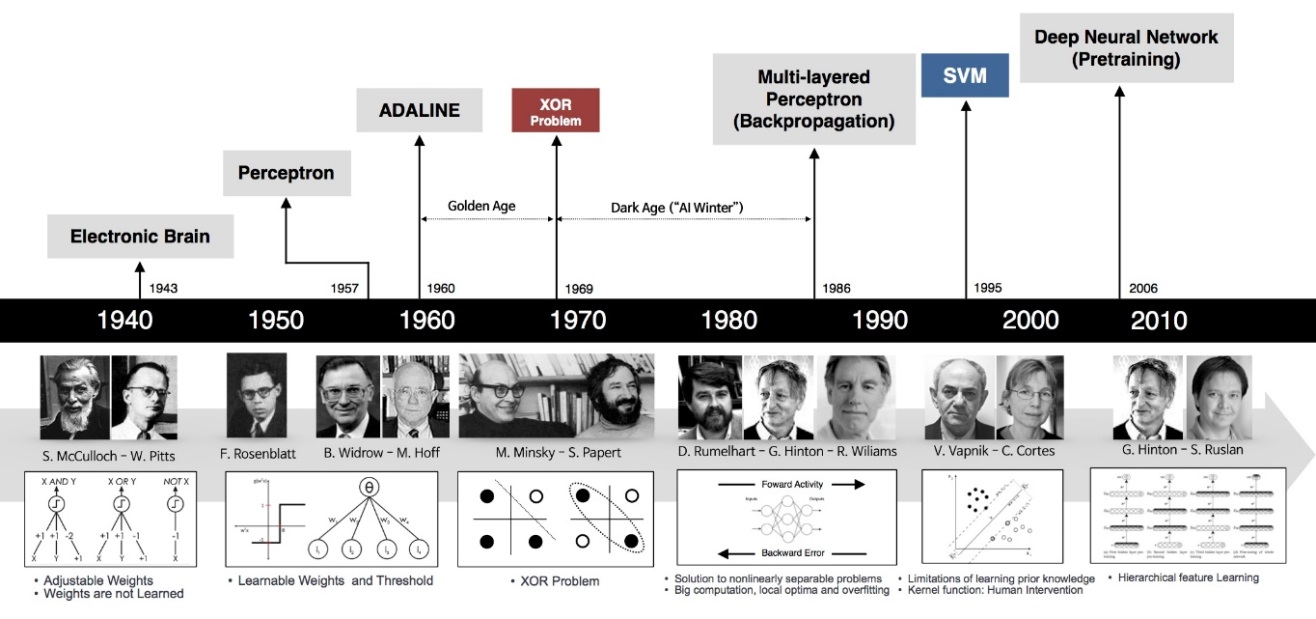
Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Học sâu ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): deep learning) là một chi của ngành [máy học](https://vi.wikipedia.org/wiki/M%C3%A1y_h%E1%BB%8Dc) dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Nhiều kiến trúc học sâu khác nhau như [mạng neuron sâu](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A1ng_n%C6%A1-ron_s%C3%A2u&action=edit&redlink=1), [mã mạng neuron tích chập sâu](https://vi.wikipedia.org/w/index.php?title=Convolutional_neuron_network&action=edit&redlink=1), [mạng niềm tin sâu](https://vi.wikipedia.org/w/index.php?title=Deep_belief_network&action=edit&redlink=1) và [mạng neuron tái phát](https://vi.wikipedia.org/w/index.php?title=Recurrent_neuron_network&action=edit&redlink=1) đã được áp dụng cho các lĩnh vực như [thị giác máy tính](https://vi.wikipedia.org/wiki/Th%E1%BB%8B_gi%C3%A1c_m%C3%A1y_t%C3%ADnh), [tự động nhận dạng giọng nói](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ti%E1%BA%BFng_n%C3%B3i), [xử lý ngôn ngữ tự nhiên](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%BD_ng%C3%B4n_ng%E1%BB%AF_t%E1%BB%B1_nhi%C3%AAn), nhận dạng âm thanh ngôn ngữ và [tin sinh học](https://vi.wikipedia.org/wiki/Tin_sinh_h%E1%BB%8Dc), chúng đã được chứng minh là tạo ra các kết quả rất tốt đối với nhiều nhiệm vụ khác nhau.

Các thuật toán học sâu tương phản với các thuật toán học nông bởi số biến đổi được tham số hóa một tín hiệu gặp phải khi nó lan truyền từ các lớp đầu vào đến lớp đầu ra, nơi một biến đổi được tham số hóa là một đơn vị xử lý có các thông số có thể huấn luyện được, chẳng hạn như trọng số và ngưỡng. Một chuỗi các biến đổi từ đầu vào đến đầu ra là một *đường gán kế thừa* (CAP- credit assignment path). CAP mô tả các kết nối quan hệ nhân quả tiềm năng giữa đầu vào và đầu ra và có thể thay đổi chiều dài. Đối với một mạng neuron nuôi tiến (feedforward), độ sâu của CAP, và do đó độ sâu của mạng đó, là số lượng các lớp ẩn cộng 1 (lớp đầu ra cũng là tham số hóa). Đối với [mạng neuron tái phát](https://vi.wikipedia.org/w/index.php?title=Recurrent_neuron_network&action=edit&redlink=1), trong đó một tín hiệu có thể truyền thông qua một lớp nhiều hơn một lần, CAPcó khả năng không bị giới hạn chiều dài. Không có sự thống nhất chung về ngưỡng của độ sâu chia học cạn với học sâu, nhưng hầu hết các nhà nghiên cứu trong lĩnh vực đồng ý rằng học sâu có nhiều lớp phi tuyến (CAP > 2) và Schmidhuber coi CAP > 10 để là học rất sâu

Lịch sử phát triển của Deep Learning:



Hình 1.9. Lịch sử phát triển của Deep Learning

## Diễn giải

Mạng neuron sâu thường được giải thích theo cách: định lý xấp xỉ tổng quát hoặc [Suy luận xác suất](https://vi.wikipedia.org/wiki/Suy_lu%E1%BA%ADn_Bayes).

### Diễn giải Định lý xấp xỉ Phổ quát

[Định lý xấp xỉ phổ quát](https://vi.wikipedia.org/w/index.php?title=Universal_approximation_theorem&action=edit&redlink=1) đề cập đến khả năng của [mạng neuron tiến tiếp](https://vi.wikipedia.org/w/index.php?title=Feedforward_neuron_networks&action=edit&redlink=1) (feedforward) với một lớp ẩn có kích thước hữu hạn đơn để xấp xỉ các [hàm liên tục](https://vi.wikipedia.org/wiki/H%C3%A0m_li%C3%AAn_t%E1%BB%A5c).

Năm 1989, là bằng chứng đầu tiên được xuất bản bởi George Cybenko cho các hàm kích hoạt [h](https://vi.wikipedia.org/w/index.php?title=Sigmoid&action=edit&redlink=1)ình sigma và được mở rộng đối với các kiến trúc nuôi tiến nhiều lớp vào năm 1991 bởi Kurt Hornik

### Diễn giải xác suất

Diễn giải [xác suất](https://vi.wikipedia.org/wiki/X%C3%A1c_su%E1%BA%A5t) bắt nguồn từ lĩnh vực [máy học](https://vi.wikipedia.org/wiki/M%C3%A1y_h%E1%BB%8Dc). Nó có đặc điểm suy luận, cũng như các khái niệm [tối ưu hóa](https://vi.wikipedia.org/wiki/T%E1%BB%91i_%C6%B0u_h%C3%B3a_(to%C3%A1n_h%E1%BB%8Dc)) [huấn luyện](https://vi.wikipedia.org/wiki/%C4%90%C3%A0o_t%E1%BA%A1o) và và [kiểm tra](https://vi.wikipedia.org/w/index.php?title=Test_(assessment)&action=edit&redlink=1), liên quan đến việc phù hợp và [tổng quát hóa](https://vi.wikipedia.org/wiki/Generalization) tương ứng. Cụ thể hơn, diễn giải xác suất sẽ xem xét kích hoạt một cách phi tuyến như là một [hàm phân phối tích lũy](https://vi.wikipedia.org/wiki/H%C3%A0m_ph%C3%A2n_ph%E1%BB%91i_t%C3%ADch_l%C5%A9y). Xem [mạng tin sâu](https://vi.wikipedia.org/w/index.php?title=Deep_belief_network&action=edit&redlink=1). Diễn giải xác suất dẫn đến sự ra đời của [dropout](https://vi.wikipedia.org/w/index.php?title=Dropout_(neuron_networks)&action=edit&redlink=1) như [regularizer](https://vi.wikipedia.org/w/index.php?title=Regularization_(mathematics)&action=edit&redlink=1) trong mạng neuron.

Diễn giải xác suất đã được giới thiệu và phổ biến rộng rãi bởi những tiên phong như [Geoff Hinton](https://vi.wikipedia.org/w/index.php?title=Geoff_Hinton&action=edit&redlink=1), [Yoshua Bengio](https://vi.wikipedia.org/w/index.php?title=Yoshua_Bengio&action=edit&redlink=1), [Yann Le Cun](https://vi.wikipedia.org/w/index.php?title=Yann_Le_Cun&action=edit&redlink=1), [Juergen Schmidhuber](https://vi.wikipedia.org/w/index.php?title=Juergen_Schmidhuber&action=edit&redlink=1).

## Các mô hình, kiến trúc học sâu

Có một lượng rất lớn các biến thể của kiến trúc học sâu. Hầu hết chúng là nhánh sinh ra từ một số kiến trúc cha ban đầu. Không phải là luôn luôn có thể so sánh hiệu suất của nhiều kiến trúc cùng với nhau, vì chúng không phải là tất cả đánh giá trên cùng một tập dữ liệu. Học sâu học là một lĩnh vực phát triển nhanh, và các kiến trúc, biến thể, hoặc các thuật toán mới xuất hiện mỗi vài tuần.

### Deep Neural Network (DNN)

Mạng neuron sâu (DNN-Deep neural Network) là một [mạng neuron nhân tạo](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_n%C6%A1-ron_nh%C3%A2n_t%E1%BA%A1o) (ANN) với nhiều đơn vị lớp ẩn giữa lớp đầu vào và đầu ra. Tương tự như các ANN, các DNN nông có thể mô hình mối quan hệ phi tuyến phức tạp. Các kiến trúc DNN, ví dụ như để phát hiện và [phân tích](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%BD_ng%C3%B4n_ng%E1%BB%AF_t%E1%BB%B1_nhi%C3%AAn) đối tượng tạo ra các mô hình hỗn hợp trong đó đối tượng này được thể hiện như một thành phần được xếp lớp của các hình ảnh nguyên thủy. Các lớp phụ cho phép các thành phần của các đặc điểm từ các lớp thấp hơn, đem lại tiềm năng của mô hình hóa dữ liệu phức tạp với các đơn vị ít hơn so với một mạng lưới nông thực hiện tương tự như vậy.

Deep Neural Network được xây dựng với mục đích mô phỏng hoạt động não bộ phức tạp của con người và được áp dụng vào nhiều lĩnh vực khác nhau, mang lại thành công và những hiệu quả đáng kinh ngạc cho con người*.*



Hình 1.10. Mô hình Deep Neural Network

### Deep Belief Network (DBN)

Một mạng niềm tin sâu (DBN) là một mô hình xác suất [thể sinh](https://vi.wikipedia.org/w/index.php?title=Generative_model&action=edit&redlink=1), tạo thành bởi nhiều đơn vị ẩn nhiều lớp. Nó có thể được coi là một [hàm hợp](https://vi.wikipedia.org/wiki/H%C3%A0m_h%E1%BB%A3p) các mô-đun học đơn giản tạo thành mỗi lớp.

Một DBN có thể được sử dụng để huấn luyện trước khả sinh một DNN bằng cách sử dụng các trọng số DBN học như các trọng số DNN ban đầu. Các thuật toán truyền ngược hoặc suy xét khác sau đó có thể được áp dụng để điều chỉnh những trọng số này. Điều này đặc biệt hữu ích khi dữ liệu đào tạo giới hạn là có sẵn, vì các trọng số khởi tạo nghèo nàn có thể cản trở đáng kể hiệu suất của mô hình được học. Các trọng số đào tạo trước này là một vùng không gian trọng số là gần gũi hơn với trọng số tối ưu hơn là các trọng số ban đầu được chọn ngẫu nhiên. Điều này cho phép cả mô hình hóa được cải thiện và hội tụ tinh chỉnh pha nhanh hơn.



Hình 1.11. Mô hình kiến trúc DBN

### Convolutional Neural Networks (CNN)

Một CNN gồm có một hoặc nhiều hơn các lớp [tích chập](https://vi.wikipedia.org/wiki/T%C3%ADch_ch%E1%BA%ADp) với các lớp đầy đủ kết nối (đáp ứng phù hợp với những mạng neuron nhân tạo tiêu biểu) trên đỉnh. Nó cũng sử dụng trọng số gắn liền và các lớp thăm dò. Kiến trúc này cho phép các CNN tận dụng lợi thế của cấu trúc 2D của dữ liệu đầu vào. So với những kiến trúc sâu khác, mạng neuron tích chập đang bắt đầu thể hiện kết quả vượt trội trong các ứng dụng hình ảnh và giọng nói. Chúng cũng có thể được huấn luyện với tiêu chuẩn truyền ngược. CNN dễ dàng được đào tạo hơn các mạng nơ ron sâu nuôi tiến thông thường khác, và có ít thông số ước tính hơn, khiến cho chúng trở thành một kiến trúc rất hấp dẫn để sử dụng.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



Hình 1.12. Ví dụ mô hình CNN

### Deep Predictive coding network (DPCN)

Có những lợi thế của một mô hình mà có thể chủ động cập nhật bản thân từ ngữ cảnh trong dữ liệu. Mạng lập trình (DPCN) là một chương trình lập trình [tiên đoán](https://vi.wikipedia.org/w/index.php?title=Predictive_modelling&action=edit&redlink=1), trong đó thông tin từ trên xuống được sử dụng để điều chỉnh theo kinh nghiệm của những cái trước đó cần thiết cho một thủ tục [suy luận](https://vi.wikipedia.org/wiki/Suy_lu%E1%BA%ADn) từ dưới lên bằng các phương tiện của một [mô hình thể sinh](https://vi.wikipedia.org/w/index.php?title=M%C3%B4_h%C3%ACnh_th%E1%BB%83_sinh&action=edit&redlink=1) kết nối cục bộ sâu. Điều này hoạt động bằng cách chiết tách các đặc điểm rời rạc các quan sát biến đổi theo thời gian bằng cách sử dụng một mô hình động học tuyến tính. Sau đó, một chiến lược thăm dò được sử dụng để học các đại diện đặc điểm bất biến. Các đơn vị này tập hợp lại để tạo thành một kiến trúc sâu và được huấn luyện bởi [học không giám sát](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_kh%C3%B4ng_c%C3%B3_gi%C3%A1m_s%C3%A1t) layer-wise [tham lam](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_tham_lam). Các lớp tạo thành một loại [xích Markov](https://vi.wikipedia.org/wiki/X%C3%ADch_Markov) mà các trạng thái tại bất kỳ lớp nào cũng chỉ phụ thuộc vào các lớp trước và các lớp sau (kế thừa).

Mạng lập tình dự đoán sâu (DPCN) dự đoán đại diện của lớp, bằng cách sử dụng một cách tiếp cận từ trên xuống bằng cách sử dụng thông tin ở lớp trên và các phụ thuộc thời gian từ các trạng thái trước đó.

### Recurrent Neural Network (RNN)

RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào chứ nhỉ? RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó (ta cùng bàn cụ thể vấn đề này sau) mà thôi. Về cơ bản một mạng RNN có dạng như sau:



Hình 1.13. Hình dạng cơ bản của một mạng RNN

Huấn luyện mạng RNN cũng tương tự như các mạng nơ-ron truyền thống, tuy nhiên giải thuật lan truyền ngược (backpropagation) phải thay đổi một chút. Đạo hàm tại mỗi đầu ra phụ thuộc không chỉ vào các tính toán tại bước đó, mà còn phụ thuộc vào các bước trước đó nữa, vì các tham số trong mạng RNN được sử dụng chung cho tất cả các bước trong mạng. Ví dụ, để tính đạo hàm tại t = 4 ta phải lan truyền ngược cả 3 bước phía trước rồi cộng tổng đạo hàm của chúng lại với nhau. Việc tính đạo hàm kiểu này được gọi là lan truyền ngược liên hồi ([BPTT](https://en.wikipedia.org/wiki/Backpropagation_through_time) - Backpropagation Through Time). Nếu giờ bạn chưa thể hiểu được BPTT thế nào thì cũng đừng lo sợ vì trong bài sau ta sẽ xem xét cụ thể nó là gì sau. Còn giờ, chỉ cần nhớ rằng với các bước phụ thuộc càng xa thì việc học sẽ [càng khó khăn hơn](https://arxiv.org/pdf/1211.5063v2.pdf) vì sẽ xuất hiện vấn đề hao hụt/bùng nổ (vanishing/exploding) của đạo hàm. Có một vài phương pháp được đề xuất để giải quyết vấn đề này và các kiểu mạng RNN hiện nay đã được thiết kế để triệt tiêu bớt chúng như LSTM chẳng hạn.

### Feedforward Neural Network (FNN)

## Tổng kết chương

Như vậy trong chương này chúng ta có được cái nhìn tổng thể về neural network và học sâu, một số mô hình học sâu. Từ đó có thể phát triển hướng nghiên cứu tìm ra phương pháp để xây dựng bài toán dự báo giá phòng cho thuê.

# : MÔ HÌNH HỌC MÁY CHO ĐỊNH GIÁ PHÒNG

Hiện nay, đã có nhiều đề tài đề cập đến các phương pháp giải quyết vấn đề cho bài toán định giá hồi quy. Trong đó nổi bật phải kể đế một vài thuật toán cho bài toán hồi quy như : Linear regression, KNN, SVR, Decision Tree. Trong phần tiếp theo sẽ nghiên cứu, đánh giá các thuật toán trên từ đó ta có thể rút ra những ưu nhược điểm của nó so với phương pháp học sâu đã đề cập ở trên:

## Mô hình hồi quy tuyến tính

"Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục.

Một ví dụ liên quan đến

Một căn nhà rộng , có phòng ngủ và cách trung tâm thành phố  km có giá là bao nhiêu. Giả sử chúng ta đã có số liệu thống kê từ 1000 căn nhà trong thành phố đó, liệu rằng khi có một căn nhà mới với các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm, chúng ta có thể dự đoán được giá của căn nhà đó không? Nếu có thì hàm dự đoán sẽ có dạng như thế nào. Ở đây là một vector hàng chứa thông tin *input*, ***y*** là một số vô hướng (scalar) biểu diễn *output* (tức giá của căn nhà trong ví dụ này).

Ta có thể suy luận đơn giản nhất , chúng ta thấy :

* Diện tích nhà càng lớn thì giá nhà càng cao;
* Số lượng phòng ngủ càng lớn thì giá nhà càng cao;
* Càng xa trung tâm thì giá nhà càng giảm.

Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa giá phòng và 3 đại lượng đầu vào là:

trong đó :

* là các hằng số,
* được gọi là bias.
* Mối quan hệ  bên trên là một mối quan hệ tuyến tính (linear).

Ưu điểm :

Nhược điểm:

## Mô hình KNN

# : MÔ HÌNH HỌC SÂU CHO ĐỊNH GIÁ PHÒNG



## Mô tả dữ liệu

### Nguồn gốc dữ liệu

### Mô tả dữ liệu thô

### Nhận xét đánh giá, xây dựng cơ sở dữ liệu

## Quy trình xử lý dữ liệu

Áp dụng theo các bước của quả trình khám phá tri thức (KDD) ta có các chi tiết các bước thực hiện xây dựng mô hình định giá phòng như sau:

### Thu thập dữ liệu

Bước này nhằm mục đích nghiên cứu, đưa ra các hướng giải pháp trong việc thu thập toàn bộ dữ liệu về thông tin phòng cho thuê từ các nguồn thông tin khác nhau trên Internet. Tiêu chí đặt ra trong bước này là không nhằm thu thập tối đa thông tin cần thiết tránh bỏ sót thông tin, sẽ ảnh hưởng đến quá quá trình xứ lý dữ liệu và khai phá.

### Lựa chọn dữ liệu

Tại bước này chúng ta tìm kiếm thu thập lựa chọn những bộ dữ liệu về phòng với đầy đủ các thông tin mô tả đặc điểm của phòng, những đặc điểm mô tả này phải là những đặc điểm có ảnh hưởng tới định giá giá phòng. Những dữ liệu này phải là những bộ dữ liệu uy tín, chính xác, thể hiện được giá cả thị trường nhà đất. Nó được phân loại theo khu vực, diện tích ,kiểu phòng trọ , kiểu vệ sinh, có view như thế nào, … . Điều quan trọng nhất trong bước này là đánh giá, lựa chọn ra các thuộc tính có ảnh hưởng đến giá phòng và có khả năng mô hình hóa nó cho việc định giá .

### Xử lý dữ liệu

Sau khi lựa chọn được bộ dữ liệu và khám phá nó để thấy được những đặc điểm của bộ dữ liệu chúng ta cần thực hiện các phương pháp để làm sạch dữ liệu, lọc bỏ dữ liệu nhiễu và impute missing value.

Lọc dữ liệu nhiễu (outlier) dữ liệu có những giá trị bất thường, (do giá trị đo của các trường hợp rất đặc biệt, do lỗi của công cụ đo lường,…). Lỗi này là lỗi khó phát hiện nhất, gây ra các sai sót nghiêm trọng làm sai lệch, giảm hiệu quả thực hiện các thuật toán khai phá.

Xử lý dữ liệu khuyết thiếu (impute missing value): là công việc chúng ta phải tìm một giá trị phù hợp để điền vào vị trí bị khuyết thiếu trong bộ dữ liệu. Có một số phương pháp để xử lý dữ liệu khuyết thiếu như:

* Một giá trị hằng có ý nghĩa trong miền xác định của dữ liệu, ví dụ như 0.
* Một giá trị của một đặc trưng từ một mẫu dữ liệu ngẫu nhiên trong tập dữ liệu.
* Các giá trị thống kê cơ bản như giá trị trung bình, giá trị trung vị hay giá trị mốt (mode) của cột.
* Một giá trị được ước lượng từ một mô hình dự đoán khác.

Mỗi phương pháp sẽ phù hợp với từng bộ dữ liệu và từng thuộc trường cụ thể.

### Biến đổi dữ liệu

Biến đổi dữ liệu là việc chuyển toàn bộ tập giá trị của một thuộc tính sang một tập các giá trị thay thế sao ho mỗi giá trị cũ tương ứng với một giá trị mới. Trong bước này chúng ta lựa chọn và thực hiện các công việc tùy thuộc vào mục đích và đặc điểm của bộ dữ liêu để đưa dữ liệu về dạng chuẩn:

* Điều chỉnh lại tỉ lệ dữ liệu trên các đặc tính (Rescale)
* Chuẩn hóa dữ liệu (Standardization)
* Bình thường hóa dữ liệu (normalization)
* Số hóa dữ liệu (Digitalization)

Điều chỉnh tỷ lệ (rescale) Dữ liệu gồm nhiều đặc tính (cột), và mỗi đặc tính thì lại có các đơn vị và độ lớn nhỏ khác nhau. Điều này tác động tới tính hiệu quả của nhiều thuật toán, ví dụ thời gian thực hiện, quá trình hội tụ, hay thậm chí ảnh hưởng cả tới độ chính xác của thuật toán. Chính vì vậy, người ta thường tiến hành điều chỉnh dữ liệu để các đặc tính cùng có chung một tỉ lệ (data scaling) . Và thường để các đặc tính có giá trị trong khoảng [0, 1]. Kết quả sẽ giúp cho nhiều thuật toán quan trọng trong Máy học sử dụng kĩ thuật Gradient Descent hội tụ nhanh. Việc điều chỉnh tỉ lệ thường dùng công thức sau đây (giả sử chúng ta đang làm trên một cột dữ liệu số cụ thể, gọi là F):

Trong đó: là gì trị ban đầu trong cột F

là giá trị sau khi được điều chỉnh

là gì trị nhỏ nhất trong cột F

là giá trị lớn nhất trong cột F

Phương pháp trên được gọi là MinMaxScaler sẽ đưa bộ dữ liệu của chúng ta về khoảng giá trị [0,1].

Chuẩn hóa dữ liệu (Standardize Data): Nhiều thuật toán trong Máy học giả định rằng dữ liệu đầu vào có phân phối Gauss. Chính vì vậy, khi chuẩn hóa dữ liệu về dạng chuẩn phân phối Gauss với giá trị trung bình bằng 0 và phương sai bằng 1. Nhờ việc chuẩn hóa, các thuật toán như linear regression, logistic regression được cải thiện. Công thức được sử dụng trong phương pháp này là :

Trong đó: giá trị trung bình

Độ lệch chuẩn :

Bình thường hóa dữ liệu (Normalize Data): Bình thường hóa dữ liệu là sự điều chỉnh tỉ lệ dữ liệu sao cho mỗi thể hiện (trên hàng) đều cho độ dài là 1. Kĩ thuật này rất cần thiết cho dữ liệu thưa (gồm nhiều số 0) trên mỗi cột đặc tính. Điều này đặc biệt ảnh hưởng tới các thuật toán lấy trọng số của các giá trị nhập vào như neuron networks, hay các thuật toán dùng độ đo khoảng cách (như k-Nearest Neighbors)

Số hóa dữ liệu (Digitalization): Rất nhiều thuật toán học máy hiện nay chỉ chấp nhận dữ liệu dạng số. Những những liệu dạng category chúng ta đều phải chuyển sang dạng số. Ở đây chúng ta có 3 phương pháp đễ mã hóa dữ liệu, chuyển dữ liệu category về dạng sô. Ví dụ một đặc tính có N giá trị khác nhau

* Mã hoá một trong N (One of N encoding): sử dụng một tập gồm N các cột nguyên để chuẩn hoá, low (0, 0, 1), mid (0, 1, 0), high (1,0,0)
* Mã hoá nhị phân (Binary encoding): chuẩn hoá thành M cột nhị phân, với M l= [log2N]. low (0,0), mid (1,0), high (1,1)
* Mã hoá số nguyên (Numeric encoding): chuẩn hoá thành một cột có giá trị nguyên, mỗi giá trị của cột tương ứng với thứ tự của giá trị gốc có trong tập giá trị ban đầu: low (1), mid (2), high (3)

Bên cạnh đó, trong bước này chúng ta còn thực hiện rút gọn dữ liệu, lựa chọn các đặc trưng phù hợp để đưa vào mô hình hay còn gọi là feature selection

## Lựa chọn thuật toán, xây dựng mô hình học sâu

Định giá phòng là một bài toán hồi quy. Chính vì thế các thuật toán được lựa chọn là các thuật toán học máy phục vụ cho bài toán phổ biển và thông dụng nhất hiện nay. Nghĩ đến bài toán hồi quy thì thuật toán đầu tiên được nghĩ đến đó chính là Linear Regression.Nhưng trong đề tài chúng ta sẽ không đề cập đến nó ,mà đề cập đến các mô hình Deeplearning.Về cơ bản với một bài toán định giá thì việc lựa chọn thuật toán này là không thực sự cần thiết, nhưng như đã nói ở trên ,nhóm vẫn muốn tìm ra một hướng giải quyết từ đó có cái nhìn tổng quan và những đánh giáchính xác về các phương pháp tiếp cận bài toán.

## Đánh giá chất lượng mô hình

Ta đánh giá chất lượng mô hình dựa vào các độ đo là sai số tuyệt đối trung bình MAE (mean absolute error) và phần trăm sai số tuyệt đối trung bình MAPE (mean absolute percentage error). Đây là hai độ đo được tính toán trên kết quả ở cả tập training và tập test ở tất cả các mô hình được xây dựng để lựa chọn đánh giá mô hình.

Một mô hình tốt là một mô hình có kết quả các độ đo MAE và MAPE càng nhỏ càng tốt

Ngoài ra có một khái niện nữa để đánh giá chất lượng mô hình là overfitting. Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu training. Việc quá khớp này có thể dẫn đến việc dự đoán nhầm nhiễu, và chất lượng mô hình không còn tốt trên dữ liệu test nữa. Dữ liệu test được giả sử là không được biết trước, và không được sử dụng để xây dựng các mô hình Machine Learning. Về cơ bản, overfitting xảy ra khi mô hình quá phức tạp để mô phỏng training data. Điều này đặc biệt xảy ra khi lượng dữ liệu training quá nhỏ trong khi độ phức tạp của mô hình quá cao và không bị hiện tượng overfitting tức sự chênh lệch giữa các độ đô trên tập training và tập testing càng nhỏ càng tốt. Một mô hình tốt sẽ không xảy ra hiện tượng overfitting hoặc sự chênh lệch này được giảm xuống thấp nhất có thể.

Từ viện đánh giá chất lượng mô hình chúng ta sẽ quyết định cải tiến mô hình bằng cách thay đổi, bổ sung phương pháp xử lý dữ liệu hay chuẩn hóa dữ liệu hay tiến hành tunning tham số của thuật toán để tăng độ chính xác và giảm hiện tượng overfitting của mô hình.

# KẾT QUẢ THỬ NGHIỆM, ĐÁNH GIÁ SO SÁNH

# KẾT LUẬN

## Kết luận

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "Điêu 473 tại Bô Luật dân sự 2015". |
| [2] | "Điều 129 Luật Nhà ở 2014". |
| [3] | Byeonghwa Park, Jae Kwon Bae, "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data," *Expert Systems with Applications,* vol. 42, pp. 2928-2934, 2015. |
| [4] | Tiêu chuẩn thẩm định giá Việt Nam số 02 Giá trị thị trường làm cơ sở cho thẩm định giá (Ký hiệu: TĐGVN 02), “Ban hành kèm theo Thông tư số 158/2014/TT-BTC ngày 27 tháng 10 năm 2014 của Bộ trưởng Bộ Tài chính”. |