I. Giới Thiệu   
II. Dataset  
III. Giới thiệu mô hình và phương pháp  
IV. Evaluation: giới thiệu thang đo đánh giá  
V. Quá trình huấn luyện  
VI. Đánh giá tổng quan  
 tổng hợp được bản kết quả thực nghiệm với mô hình tốt nhất ở mỗi phương pháp

VII. Tổng kết

# **Giới thiệu sơ lược về Neural Network**

### **Tổng quan**

Là một mô Neural hình tính toán bắt chước cách thức hoạt động của các tế bào thần kinh trong não người. Mạng nơ-ron nhân tạo (ANN) sử dụng các thuật toán learning có thể thực hiện các điều chỉnh một cách độc lập – hoặc học theo một nghĩa nào đó – khi chúng nhận được giá trị input mới.

Có rất nhiều loại mạng Neural Network (NN), được sử dụng trong thực tế, ngoài mạng ANN là mạng cơ bản nhất còn có một số mạng NN phức tạp hơn như: CNN, RNN, LSTM, GAN,

Mạng Neural Network là sự kết hợp của những tầng perceptron hay còn gọi là perceptron đa tầng. Và mỗi một mạng Neural Network thường bao gồm 3 kiểu tầng là:

* Tầng input layer (tầng vào): Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng.
* Tầng output layer (tầng ra): Là tầng bên phải cùng và nó thể hiện cho những đầu ra của mạng.
* Tầng hidden layer (tầng ẩn): Tầng này nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.

Thường sẽ nghe tới khái niệm CNN nhiều hơn, hoặc các khái niệm khác, nhưng mà mấy cái như CNN thì dùng để giải quyết các bài toán có dữ liệu phức tạp hơn: ảnh, âm thanh,

Bài toán của mình chỉ là dữ liệu dạng dataframe, 2 chiều, dạng bảng

Cấu trúc thông thường sẽ rất phức tạp, nhưng vẫn sẽ có đủ 3 tầng chính,

Input(CNN), hidden(multiple), output

A diagram of a network

Description automatically generated with low confidence

*Hình 37. Mô hình ANN đơn giản*

**Lưu ý:** Mỗi một Neural Network chỉ có duy nhất một tầng vào và 1 tầng ra nhưng lại có rất nhiều tầng ẩn.

Trong ANN, 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.

### **Các loại mạng trong Neural Network.**

#### Mạng lan truyền tiến (Feed Forward)

Mạng lan truyền tiến (FeedForward) có luồng thông tin 1 chiều, một đơn vị sẽ được sử dụng để gửi thông tin cho một đơn vị khác mà không nhận được bất kỳ thông tin nào. Chúng không có vòng phản hồi, thường dùng để nhận dạng một mẫu cụ thể vì chúng có đầu vào và đầu ra nhất định.

A picture containing circle, sketch, drawing

Description automatically generated

*Hình 38. Mạng lan truyền tiến*

#### Mạng lan truyền ngược (Back Forward)

Trong mạng Nơron nhân tạo này, chúng sẽ cho phép các vòng lặp phản hồi. Chúng ta thường sử dụng mô hình này trong các bộ nhớ có thể giải quyết nội dung.

A picture containing circle, sketch, drawing, diagram

Description automatically generated

*Hình 39. Mô hình Feedback ANN*

### **Cách hoạt động của Neural Network**

Mạng neural nhân tạo có khả năng sử dụng được như một loại cơ chế xấp xỉ hàm tùy ý mà học được từ việc dữ liệu quan sát. Tuy nhiên, việc sử dụng chúng khá khó và cần phải có sự hiểu biết tương đối về những lý thuyết cơ bản về mạng nơron này.

***Lựa chọn mô hình:*** Phụ thuộc vào cách trình bày dữ liệu và các ứng dụng của nó. Đây là mô hình khá phức tạp nên có thể dẫn đến nhiều thách thức cho quá trình học.

***Thuật toán học:*** Thường sẽ có rất nhiều thỏa thuận giữa các thuật toán học. Và hầu hết, chúng sẽ làm việc tốt với những tham số đúng nhằm huấn luyện trên dữ liệu mà không nhìn thấy yêu cầu một số lượng đáng kể các thử nghiệm.

**Mạnh mẽ**: Nếu như các mô hình, thuật toán học và hàm chi phí được lựa chọn một cách thích hợp thì Neural Network có thể cho ra kết quả vô cùng hợp lý.

### **Ưu và nhược điểm của Neural Nerwork**

**Ưu điểm:**

**-** Phân tích dữ liệu một cách trực quan hiệu quả: Mạng nơ ron nhân tạo tương tự như mạng thần kinh của con người nên nó có thể xử lý một số các tác vụ phức tạp hơn so với các phương pháp khác. Mạng neural có thể phân tích các dữ liệu một cách hiệu qủa và tách chúng thành nhiều loại khác nhau.

**-** Xử lý hiệu quả các dữ liệu không có cấu trúc.

**-** Cấu trúc có tính thích ứng: bất kỳ mục đích nào mà ANN được áp dụng, nó sẽ thay đổi tiến trình cấu trúc của nó theo mục đích đó.

**Nhược điểm:**

**-** Yêu cầu phần cứng.

**-** Cấu trúc phức tạp và khó triển khai.

**-** Cần nhiều dữ liệu: Dữ liệu sẽ tỉ lệ thuận với độ chính xác.

### **Ứng dụng của Neural Network**

Mạng ANN hiện nay được ứng dụng rất nhiều trong rất nhiều ngành, nó có thể làm những việc mà con người gặp khó khăn một cách rất dễ dàng. Một số lĩnh vực sử dụng mạng nơron nhân tạo như: Hàng không vũ trụ, phần mềm, giao thông vận tải, viễn thông.

# **Lý thuyết toán học**

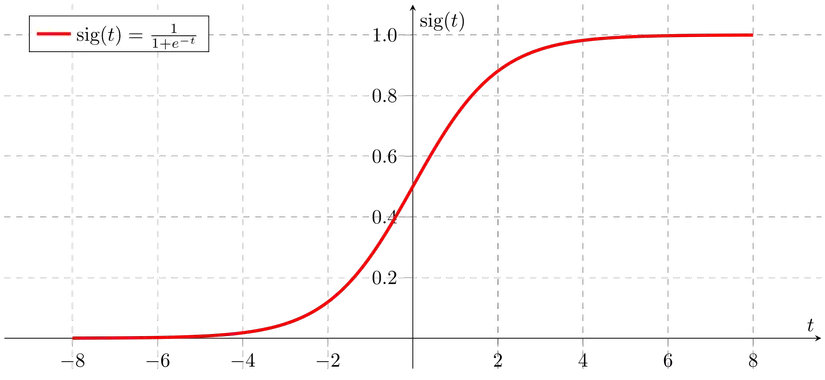
### **Sơ qua về Logistic Regression:**

Logistic Regression có activation function là hàm sigmoid:

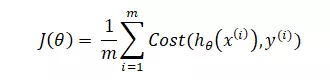


Hàm hypothesys:





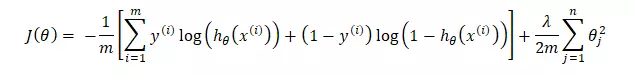
Cost function:



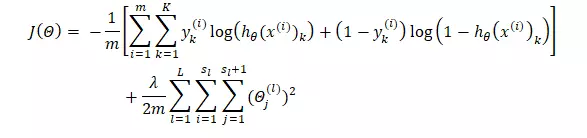
Với:

****

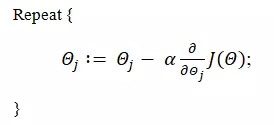
Vậy ta có cost function (kết hợp với Regurlarization):



Vậy với ANN với mỗi node thuộc layer khác input layer đều là một Logistic Regression ta sẽ có:

****

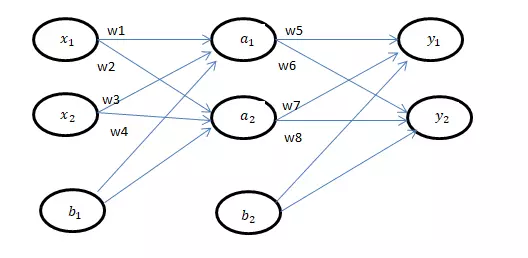
Công việc của chúng ta hiện tại là tìm ra được  sao cho  min.  
Để tìm cực tiểu của ta áp dụng thuật toán Gradient Descent.



Với α là learning rate.  
Để thực hiện được thì cần phải tính được  để tính được đạo hàm này là việc tương đối khó và ta cần thực hiện một thuật toán được gọi là **back propagation** để tính.

### **Forward Propagation**

Ta có mạng neural như sau :



**Chú thích**

 là các features của input.  
​ là các output.  
 là các bias.  
​ là các trọng số.  
Như cái tên của *forward propagation* , ta sẽ tiến hành tính toán ​,​ ​ từ trái qua phải.  
   
   
   
  
Tương tự :  
  
=​+​​+​  
​=​

​

*Forward propagation* là một công đoạn tính toán giá trị tại từng node để phục vụ việc tính toán trong *Back propagation*.

### **Back Propagation**

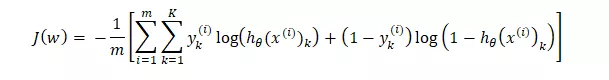
Như đã nói ở trên, mục tiêu của *back propagation* là tính

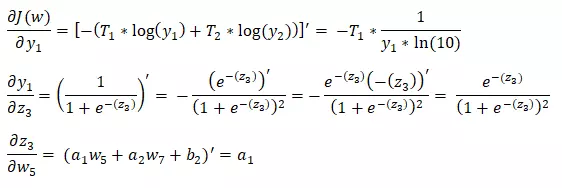
Giả sử ta đang cần , Áp dụng chain rule, ta có:

=

Trong đó:

Để dễ hiểu, chúng ta có thể giả sử không có regularization và m = 1 với tập dữ liệu kết quả trong training set ứng với là





Tất cả các kết quả của ​​ đều có thể tính được thông qua kết quả thu được từ  
*forward propagation*. Vậy ta có thể tính được 

Tương tự như vậy ta có thể lần lượt tính được giá với j = 1, 2, …, 8 trong trường hợp này.  
Như vậy nhờ vào *back propagation* ta đã có thể tính được từ đó giúp thuật toán Gradient Descent có thể hoạt động và ta có thể tìm ra tập Θ sao cho Cost function là nhỏ nhất.

# **Xây dựng mô hình**

Giới thiệu thư viện và tham số: Xem thêm về tài liệu tại [đây](https://keras.io/about/).

Sử dụng Keras, một API miễn phí với thư viện mở chạy trên nền tảng máy học Tensorflow.  
Trong Keras, mô hình **Sequential** cho phép người dùng để xây dựng các mạng neural network có kiến trúc đơn giản, liên tiếp các layer với nhau theo thứ tự.

Sử dụng các phương thức thao tác cơ bản như add, compile, fit, predict.

Các siêu tham số được sử dụng trong mô hình: **(CHƯA HOÀN THÀNH, CÒN BỔ SUNG THÊM RÕ HƠN VỀ CÁC SIÊU THAM SỐ, BỔ SUNG THÊM KHẢO SÁT GRIDSEARCH)**

Trong lớp Dense thuộc phương thức add: s=6, kernel\_initializer='glorot\_uniform', activation='relu'

Trong phương thức compile: optimizer = 'adam', loss = 'binary\_crossentropy', metrics = ['accuracy']

Trong phương thức fit: batch\_size=10,epochs = 100

Trong phương thức đánh giá: average='macro'

Kết quả thực nghiệm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Số hidden layer** | **Số node mỗi hidden layer** | **Hàm kích hoạt của hidden layer** | **ACC** | **Precision** | **Recall** | **F1\_score** |
| **1** | (6) | Relu | 85.06 | 83.75 | 82.99 | 83.33 |
| **2** | **(6,6)** | **Relu** | **86.70** | **84.99** | **85.51** | **85.19** |
| **3** | (6,6) | Sigmoid | 84.30 | 83.14 | 82.03 | 82.45 |
| **4** | (6,10) | Relu | 84.08 | 83.47 | 81.68 | 82.28 |
| **5** | (6,10) | Sigmoid | 84.30 | 83.14 | 82.03 | 82.45 |
| **6** | (6,15) | Relu | 85.17 | 84.31 | 82.78 | 83.41 |
| **7** | (6,6,10) | Relu | 83.43 | 81.75 | 81.37 | 81.43 |
| **8** | (6,10,10) | Relu | 85.50 | 83.74 | 83.95 | 83.82 |
| **9** | (6,10,10,10) | Relu | 86.37 | 84.94 | 84.84 | 84.82 |

**Nhận xét :**

- Ta thấy với bộ dữ liệu heart.csv, mô hình có độ chính xác khi có 2 tầng ẩn với số node lần lượt là 6, 6.

- Ta thấy mô hình có xư hướng tăng độ chính xác khi tăng số node hoặc số tầng ẩn.

**Phân tích kết quả thực nghiệm:**

<https://viblo.asia/p/tong-quan-ve-artificial-neural-network-1VgZvwYrlAw>

<https://viettelidc.com.vn/tin-tuc/cam-nang-ai-artificial-neural-network-la-gi-cau-truc-cach-hoat-dong-va-ung-dung-cua-mo-hinh-nay>

<https://keras.io/api/>