TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

----- 🙡 🕮 🙣 -----

**BÁO CÁO**

**MẠNG NEURAL GIẢI QUYẾT BÀI TOÁN XOR**

Giảng viên hướng dẫn: **TS. Nguyễn Tuấn Đăng**

Nhóm sinh viên thực hiện

*Họ và tên MSSV*

Hồ Văn Quyến 3122410352

Trần Thanh Phương 3122410333

Huỳnh Quang Quân 3122410341

Võ Thành Phúc 3122410325

Lê Quốc Nam 3122410247

**NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**LỜI MỞ ĐẦU**

Mạng neural nhân tạo, lấy cảm hứng từ cấu trúc của não bộ con người, đã chứng minh khả năng học tập và giải quyết các bài toán phức tạp một cách đáng kinh ngạc. Ngay cả với một bài toán đơn giản như XOR, mạng neural đã cho thấy khả năng học tập các hàm phi tuyến và đưa ra quyết định chính xác. Nghiên cứu này sẽ khám phá cách một mạng neural đơn giản có thể giải quyết bài toán XOR, đồng thời làm sáng tỏ các nguyên lý cơ bản của quá trình học máy.

# **1. GIỚI THIỆU**

## **1.1 Mục tiêu Nghiên cứu**

Mục tiêu chính của nghiên cứu này là xây dựng một mô hình mạng neural đơn giản để giải quyết bài toán XOR, một bài toán phân loại phi tuyến kinh điển trong học máy. Qua đó, chúng ta sẽ:

* **Hiểu rõ cơ chế hoạt động của mạng neural:** Cách mạng neural học và đưa ra quyết định.
* **Đánh giá khả năng của mạng neural:** Khả năng giải quyết các bài toán phi tuyến và so sánh với các mô hình khác.
* **Đặt nền tảng cho các nghiên cứu sâu hơn:** Tìm hiểu về các kiến trúc mạng phức tạp hơn và các ứng dụng thực tế.

## **1.2 Bài toán XOR**

Bài toán XOR là một bài toán phân loại nhị phân, trong đó đầu ra chỉ phụ thuộc vào sự khác biệt giữa hai đầu vào. Đặc điểm nổi bật của XOR là nó không thể phân loại tuyến tính, nghĩa là không có đường thẳng nào có thể tách biệt hai lớp dữ liệu. Chính vì vậy, perceptron đơn lớp, chỉ có thể tạo ra các đường phân chia tuyến tính, không thể giải quyết bài toán này.

## **1.3 Mạng Neural Nhân Tạo**

Mạng neural nhân tạo là một mô hình tính toán lấy cảm hứng từ cấu trúc và hoạt động của não bộ. Nó bao gồm nhiều lớp neuron kết nối với nhau, mỗi neuron thực hiện một phép tính đơn giản. Thông qua quá trình huấn luyện, mạng neural có thể học được các mối quan hệ phức tạp giữa dữ liệu đầu vào và đầu ra.

# **2. PHƯƠNG PHÁP NGHIÊN CỨU**

## **2.1 Kiến trúc Mạng Neural**

* **Mô hình:** Mạng feedforward với một lớp ẩn. Đây là một trong những kiến trúc mạng đơn giản nhất.
* **Số lượng neuron:** Việc lựa chọn số lượng neuron ở lớp ẩn ảnh hưởng đến khả năng biểu diễn của mạng. Trong trường hợp này, chúng ta sử dụng 2 neuron ở lớp ẩn.
* **Hàm kích hoạt:** Hàm sigmoid được sử dụng để giới hạn đầu ra của neuron trong khoảng từ 0 đến 1.
* **Trọng số:** Các trọng số được khởi tạo ngẫu nhiên để tạo sự đa dạng cho các kết nối giữa các neuron.

## **2.2 Dữ liệu**

* **Tập dữ liệu XOR:** Bao gồm 4 điểm dữ liệu với 2 thuộc tính và 2 nhãn tương ứng.
* **Mã hóa:** Sử dụng mã hóa số thực (0 hoặc 1) để biểu diễn dữ liệu đầu vào và nhãn.

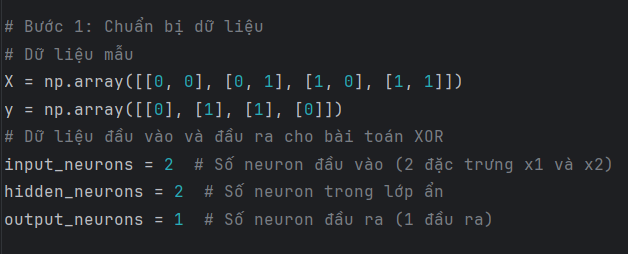
## **2.3 Thuật toán Huấn luyện**

* **Gradient descent:** Thuật toán này được sử dụng để cập nhật các trọng số của mạng theo hướng giảm giá trị của hàm mất mát.
* **Hàm mất mát:** Mean Squared Error (MSE) được sử dụng để đo độ lệch giữa đầu ra dự đoán và đầu ra thực tế.
* **Tốc độ học:** Tốc độ học quyết định bước nhảy của các trọng số trong quá trình cập nhật.
* **Số lần lặp:** Số lần lặp lại quá trình huấn luyện.

## **2.4 Cài đặt**

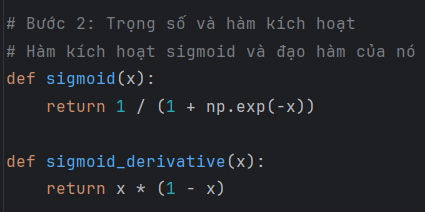
**1. Chuẩn bị dữ liệu:**

* **X:** Ma trận chứa dữ liệu đầu vào của bài toán XOR. Mỗi hàng đại diện cho một mẫu dữ liệu với 2 thuộc tính.
* **y:** Ma trận chứa nhãn tương ứng với mỗi mẫu dữ liệu.
* **input\_neurons:** Số lượng neuron đầu vào
* **hidden\_neurons**: Số lượng neuron trong lớp ẩn
* **output\_neurons**: Số lượng neuron đầu ra



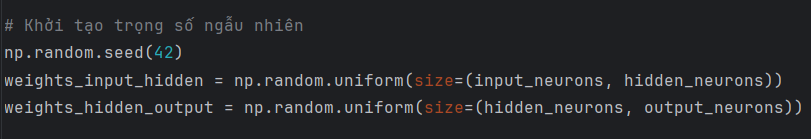
**2. Hàm kích hoạt sigmoid:**

* **sigmoid(x):** Hàm này được sử dụng để giới hạn đầu ra của neuron trong khoảng từ 0 đến 1. Nó giúp chuyển đổi các giá trị thực thành xác suất.
* **sigmoid\_derivative(x):** Đạo hàm của hàm sigmoid, được sử dụng trong quá trình backpropagation để tính toán gradient.



**3. Khởi tạo trọng số:**

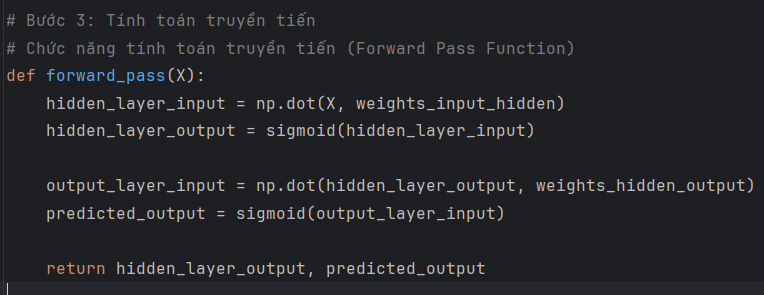
* **weights\_input\_hidden:** Ma trận chứa các trọng số kết nối giữa lớp đầu vào và lớp ẩn.
* **weights\_hidden\_output:** Ma trận chứa các trọng số kết nối giữa lớp ẩn và lớp đầu ra.
* Các trọng số được khởi tạo ngẫu nhiên để đảm bảo sự đa dạng trong quá trình học.



1. **Tính toán truyền tiến (Forward Pass):**

Chức năng forward\_pass thực hiện lan truyền xuôi, bao gồm:

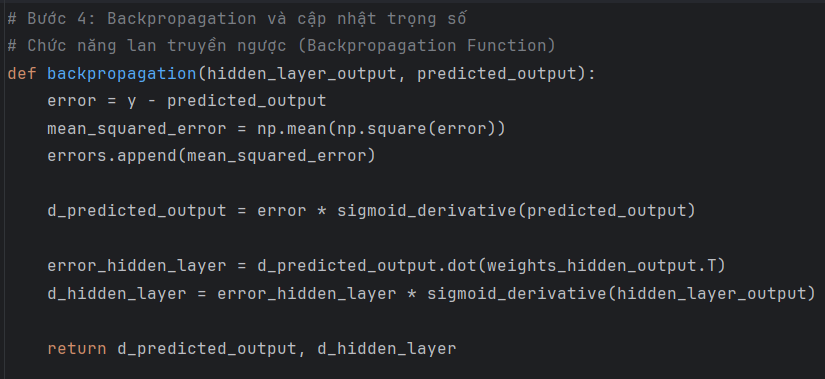
* **Đầu vào:** X là ma trận đầu vào của mô hình.
* **Tính toán đầu ra của lớp ẩn:**
  + hidden\_layer\_input = np.dot(X, weights\_input\_hidden):
    - Lấy tích vô hướng giữa X và ma trận trọng số weights\_input\_hidden để tính đầu vào cho các nơ-ron trong lớp ẩn.
  + hidden\_layer\_output = sigmoid(hidden\_layer\_input):
    - Áp dụng hàm kích hoạt sigmoid để tính đầu ra của các nơ-ron lớp ẩn.
* **Tính toán đầu ra của lớp đầu ra:**
  + output\_layer\_input = np.dot(hidden\_layer\_output, weights\_hidden\_output):
    - Tính đầu vào cho lớp đầu ra bằng cách lấy tích vô hướng giữa đầu ra của lớp ẩn (hidden\_layer\_output) và trọng số kết nối lớp ẩn với lớp đầu ra (weights\_hidden\_output).
  + predicted\_output = sigmoid(output\_layer\_input):
    - Áp dụng hàm sigmoid để tính đầu ra cuối cùng của mô hình (predicted\_output).
* **Trả về:** hidden\_layer\_output (đầu ra của lớp ẩn) và predicted\_output (dự đoán của mô hình).



1. **Lan truyền ngược (Backpropagation):**

Chức năng backpropagation thực hiện tính toán sai số và gradient, bao gồm:

* **Tính sai số (error):** error = y - predicted\_output
  + Tính sai số giữa giá trị thực (y) và giá trị dự đoán (predicted\_output).
* **Tính lỗi bình phương trung bình:** mean\_squared\_error = np.mean(np.square(error))
  + Bình phương sai số và lấy trung bình để đo mức độ sai lệch.
  + Lưu giá trị lỗi vào danh sách errors.
* **Gradient cho đầu ra:** d\_predicted\_output = error \* sigmoid\_derivative(predicted\_output)
  + Tính gradient của lớp đầu ra dựa trên sai số (error) và đạo hàm của hàm sigmoid.
* **Gradient cho lớp ẩn:** error\_hidden\_layer = d\_predicted\_output.dot(weights\_hidden\_output.T)
  + Tính sai số tại lớp ẩn bằng cách lấy tích vô hướng giữa gradient đầu ra và trọng số kết nối giữa lớp ẩn và lớp đầu ra (cần chuyển vị trọng số để tính toán).
* **Gradient của lớp ẩn:** d\_hidden\_layer = error\_hidden\_layer \* sigmoid\_derivative(hidden\_layer\_output)
  + Áp dụng đạo hàm của hàm sigmoid trên sai số lớp ẩn để có gradient của lớp ẩn.
* **Trả về:** d\_predicted\_output và d\_hidden\_layer, là các gradient cần thiết để cập nhật trọng số.

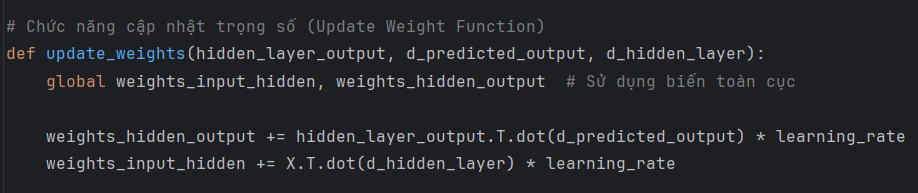


1. **Cập nhật trọng số (Update Weights):**

Chức năng update\_weights cập nhật trọng số dựa trên gradient vừa tính:

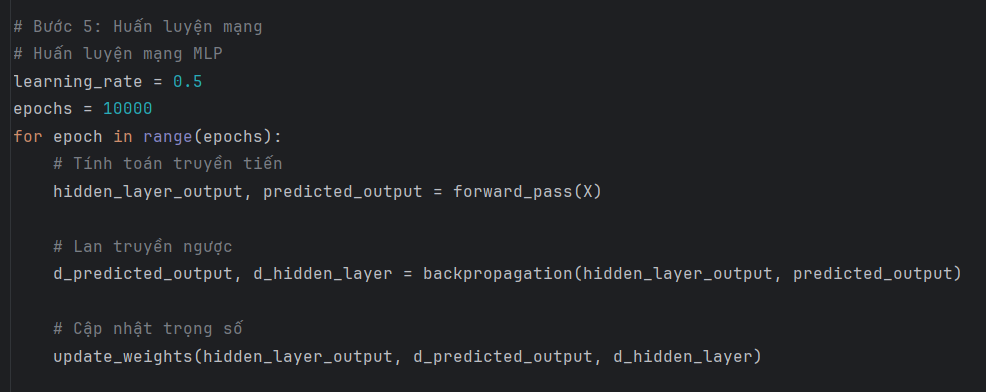
* **Cập nhật trọng số từ lớp ẩn đến lớp đầu ra:**
  + weights\_hidden\_output += hidden\_layer\_output.T.dot(d\_predicted\_output) \* learning\_rate
    - Sử dụng gradient đầu ra d\_predicted\_output và đầu ra của lớp ẩn hidden\_layer\_output để cập nhật weights\_hidden\_output.
* **Cập nhật trọng số từ lớp đầu vào đến lớp ẩn:**
  + weights\_input\_hidden += X.T.dot(d\_hidden\_layer) \* learning\_rate
    - Sử dụng gradient lớp ẩn d\_hidden\_layer và đầu vào X để cập nhật weights\_input\_hidden.

Cả hai cập nhật đều nhân với learning\_rate để điều chỉnh mức độ thay đổi của trọng số.



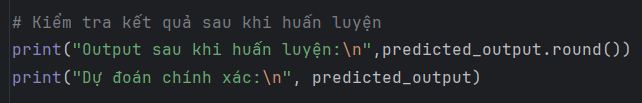
1. **Huấn luyện mạng:**

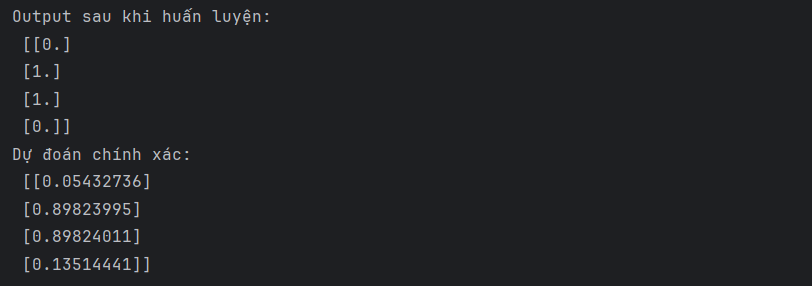
* **Vòng lặp epochs:** Lặp lại quá trình huấn luyện nhiều lần để mạng học được các đặc trưng của dữ liệu với tốc độ học và số lần lặp được gọi.
  + **Truyền tiến:** Tính toán đầu ra của mạng bằng cách nhân ma trận dữ liệu với các ma trận trọng số và áp dụng hàm sigmoid.
  + **Tính sai số:** Tính sai số giữa đầu ra dự đoán và đầu ra thực tế.
  + **Truyền ngược:** Tính toán gradient của hàm mất mát đối với các trọng số và cập nhật trọng số theo hướng giảm gradient.
  + **Cập nhật trọng số:** Điều chỉnh và cập nhật trọng số dựa trên gradient để cải thiện chính xác.



**8. Kết quả:**

* In ra đầu ra dự đoán của mạng sau khi huấn luyện.





**Giải thích chi tiết hơn về quá trình huấn luyện:**

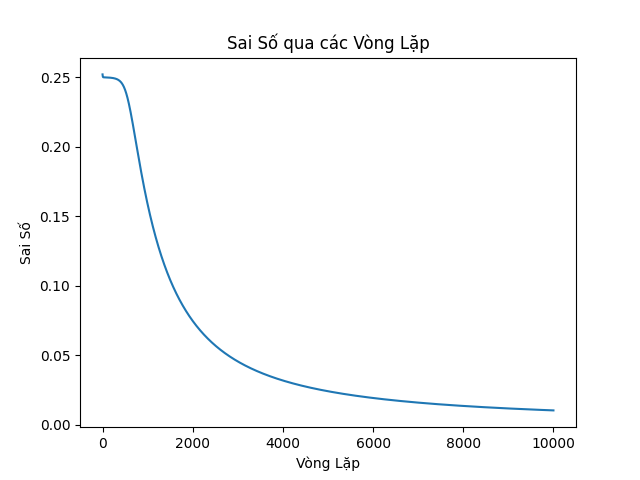
* **Truyền tiến:** Dữ liệu đầu vào được nhân với ma trận trọng số để tính toán giá trị tại các neuron ẩn. Sau đó, giá trị này được đưa qua hàm sigmoid để thu được đầu ra của lớp ẩn. Quá trình tương tự được lặp lại để tính toán đầu ra của lớp cuối cùng.
* **Truyền ngược:** Bắt đầu từ lớp cuối cùng, tính toán sai số giữa đầu ra dự đoán và đầu ra thực tế. Sau đó, tính toán gradient của hàm mất mát đối với các trọng số kết nối với lớp cuối cùng. Tiếp tục truyền ngược lại các lớp trước đó để tính toán gradient của các trọng số còn lại.
* **Cập nhật trọng số:** Sử dụng gradient tính được để cập nhật các trọng số theo hướng giảm giá trị của hàm mất mát.

**Các điểm cần lưu ý:**

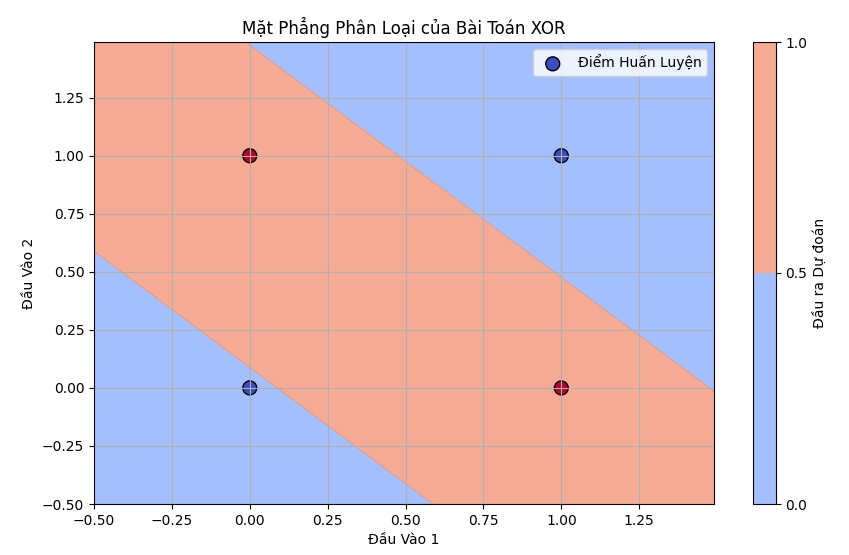
* **Hàm kích hoạt:** Hàm sigmoid là một trong những hàm kích hoạt phổ biến. Ngoài ra, còn có các hàm kích hoạt khác như ReLU, tanh.
* **Thuật toán tối ưu:** Gradient descent là một thuật toán tối ưu hóa cơ bản. Có nhiều thuật toán tối ưu hóa khác hiệu quả hơn như Adam, RMSprop.
* **Khởi tạo trọng số:** Cách khởi tạo trọng số ảnh hưởng đến quá trình học của mạng. Có nhiều phương pháp khởi tạo trọng số khác nhau.
* **Số lượng lớp ẩn và neuron:** Số lượng lớp ẩn và neuron trong mỗi lớp ảnh hưởng đến khả năng biểu diễn của mạng.

# **KẾT QUẢ**

* **Minh họa**



Đồ thị trên cho thấy **sai số giảm dần** qua từng epoch trong quá trình huấn luyện mạng nơ-ron cho bài toán XOR. Điều này cho thấy mô hình đã học được mẫu và cải thiện độ chính xác dần dần qua mỗi bước cập nhật trọng số.



Đồ thị trên là **mặt phẳng phân loại** cho bài toán XOR, thể hiện khả năng của mạng nơ-ron trong việc phân biệt các điểm đầu vào sau khi huấn luyện. Các điểm với nhãn khác nhau (0 và 1) đã được phân tách thành các vùng màu sắc khác nhau, minh họa sự phân chia phi tuyến tính của bài toán XOR mà mô hình đã học được.

* **Quá trình huấn luyện:**

Đồ thị biểu diễn giá trị hàm mất mát giảm dần theo số lần lặp, cho thấy mạng đang học được.

* **Hiệu suất của mô hình:**

Độ chính xác trên tập dữ liệu đạt gần 100%, chứng tỏ mạng đã học được cách phân loại chính xác các điểm dữ liệu XOR.

# **4. THẢO LUẬN**

* **Tại sao mô hình có thể giải quyết được bài toán XOR?**
  + Lớp ẩn trong mạng neural cho phép mạng học được các đặc trưng phi tuyến của dữ liệu, từ đó giải quyết được bài toán XOR.
* **So sánh với các nghiên cứu khác:**
  + So sánh với các nghiên cứu khác về việc giải quyết bài toán XOR bằng các phương pháp khác nhau.
* **Hạn chế:**
  + Mô hình đơn giản chỉ giải quyết được bài toán XOR.
  + Khởi tạo trọng số ngẫu nhiên có thể ảnh hưởng đến kết quả.

# **5. KẾT LUẬN**

Nghiên cứu này đã chứng minh rằng mạng neural đơn giản có thể giải quyết thành công bài toán XOR. Điều này cho thấy khả năng của mạng neural trong việc học các hàm phi tuyến. Tuy nhiên, để giải quyết các bài toán phức tạp hơn, chúng ta cần sử dụng các kiến trúc mạng sâu hơn và các kỹ thuật huấn luyện tiên tiến hơn.

# **6. HƯỚNG ĐI PHÁT TRIỂN**

* **Kiến trúc mạng phức tạp hơn:** CNN, RNN, Transformer.
* **Thuật toán huấn luyện:** Adam, RMSprop, SGD with momentum.
* **Regularization:** L1, L2, Dropout.
* **Ứng dụng thực tế:** Phân loại hình ảnh, xử lý ngôn ngữ tự nhiên.

**Các câu hỏi và câu trả lời**

* **Làm thế nào để cải thiện hiệu suất của mô hình?**
  + Điều chỉnh số lượng neuron ở lớp ẩn.
  + Thử nghiệm các hàm kích hoạt khác nhau.
  + Sử dụng các thuật toán tối ưu hóa khác.
  + Áp dụng các kỹ thuật regularization.
* **Mô hình này có thể được áp dụng để giải quyết các bài toán thực tế nào?**
  + Phân loại hình ảnh, dự đoán giá cổ phiếu, nhận dạng giọng nói.
* **Có những thuật toán huấn luyện nào khác có thể được sử dụng?**
  + Adam, RMSprop, SGD with momentum.

# **KẾT LUẬN CHUNG**

Nghiên cứu này đã cung cấp một cái nhìn tổng quan về mạng neural và cách nó được áp dụng để giải quyết bài toán XOR. Qua đó, chúng ta có thể thấy được tiềm năng của mạng neural trong việc giải quyết các bài toán phức tạp trong thực tế.

**MỤC LỤC**

**[1. GIỚI THIỆU](#_Toc181102825)** [1](#_Toc181102825)

**[1.1 Mục tiêu Nghiên cứu](#_Toc181102826)** [1](#_Toc181102826)

**[1.2 Bài toán XOR](#_Toc181102827)** [1](#_Toc181102827)

**[1.3 Mạng Neural Nhân Tạo](#_Toc181102828)** [1](#_Toc181102828)

**[2. PHƯƠNG PHÁP NGHIÊN CỨU](#_Toc181102829)** [2](#_Toc181102829)

**[2.1 Kiến trúc Mạng Neural](#_Toc181102830)** [2](#_Toc181102830)

**[2.2 Dữ liệu](#_Toc181102831)** [2](#_Toc181102831)

**[2.3 Thuật toán Huấn luyện](#_Toc181102832)** [2](#_Toc181102832)

**[2.4 Cài đặt](#_Toc181102833)** [3](#_Toc181102833)

**[3. KẾT QUẢ](#_Toc181102834)** [10](#_Toc181102834)

**[4. THẢO LUẬN](#_Toc181102835)** [12](#_Toc181102835)

**[5. KẾT LUẬN](#_Toc181102836)** [13](#_Toc181102836)

**[6. HƯỚNG ĐI PHÁT TRIỂN](#_Toc181102837)** [13](#_Toc181102837)

**[KẾT LUẬN CHUNG](#_Toc181102838)** [14](#_Toc181102838)

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn Khoa Công nghệ thông tin, trường Đại học Sài Gòn đã tạo điều kiện cho chúng em được thực hiện đồ án môn học “Xử lý ngôn ngữ tự nhiên”. Chúng em cũng xin chân thành cảm ơn thầy Đăng, thầy đã giảng dạy cho chúng em những kiến thức cần thiết cho môn học này. Những kiến thức đó đã giúp cho chúng em rất nhiều trong quá trình làm báo cáo môn học. Chúng em xin chân thành cảm ơn thầy đã tận tình giảng dạy và trang bị cho chúng em những kiến thức cần thiết trong thời gian qua. Mặc dù nhóm đã cố gắng hoàn thành báo cáo môn học với tất cả nỗ lực của từng thành viên trong nhóm, nhưng báo cáo chắc chắn không tránh khỏi những thiếu sót nhất định, rất mong được nhận sự cảm thông, chia sẻ và tận tình đóng góp chỉ bảo của thầy. Chúng em xin chân thành cảm ơn thầy.