

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: DỰ ĐOÁN KHÍ THẢI CO₂ BẰNG MÔ HÌNH ANN

Giáo viên hướng dẫn: TS. Trần Đăng Công

Sinh viên thực hiện:

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Lớp
1	1771020493	Doãn Hoài Nam	03/01/2005	CNTT 17-14
2	1771020724	Trần Quang Tùng	26/06/2005	CNTT 17-14
3	1771020337	Nguyễn Thị Quỳnh Hương	13/11/2005	CNTT 17-14

Hà Nội, năm 2025

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: DỰ ĐOÁN KHÍ THẢI CO₂ BẰNG MÔ HÌNH ANN

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	1771020493	Doãn Hoài Nam	03/01/2005		
2	1771020724	Trần Quang Tùng	26/06/2005		
3	1771020337	Nguyễn Thị Quỳnh Hương	13/11/2005		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong những thập kỷ gần đây, biến đổi khí hậu đã trở thành một trong những vấn đề cấp bách nhất đối với toàn cầu. Tác động của biến đổi khí hậu không chỉ giới hạn ở sự gia tăng nhiệt độ trung bình của Trái Đất mà còn kéo theo hàng loạt hệ lụy nghiêm trọng như nước biển dâng, thời tiết cực đoan và suy giảm đa dạng sinh học. Một trong những nguyên nhân chính dẫn đến tình trạng này là sự gia tăng lượng khí nhà kính, đặc biệt là khí CO₂, do các hoạt động của con người như công nghiệp, giao thông và sản xuất năng lượng.

Việc theo dõi và dự đoán chính xác lượng khí thải CO₂ đóng vai trò quan trọng trong việc xây dựng các chiến lược kiểm soát và giảm thiểu tác động tiêu cực đến môi trường. Tuy nhiên, do sự phức tạp của các yếu tố ảnh hưởng đến khí thải CO₂, các phương pháp truyền thống đôi khi không đạt được độ chính xác cao hoặc yêu cầu nhiều công sức trong việc thu thập và xử lý dữ liệu.

Nhờ vào sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI), đặc biệt là các mô hình học sâu (Deep Learning), việc dự đoán khí thải CO₂ đã trở nên khả thi và hiệu quả hơn. Trong số đó, mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một trong những công cụ mạnh mẽ giúp phân tích dữ liệu và dự báo xu hướng dựa trên các mẫu dữ liệu quá khứ. ANN có khả năng tự học từ dữ liệu, tìm ra các mối quan hệ phức tạp giữa các biến số đầu vào và cho ra kết quả có độ chính xác cao.

Tài liệu này sẽ trình bày toàn bộ quá trình xây dựng mô hình dự đoán khí thải CO₂ bằng ANN, bao gồm các bước thu thập dữ liệu, tiền xử lý, thiết kế mô hình, huấn luyện, đánh giá và tối ưu hóa kết quả. Ngoài ra, nghiên cứu cũng sẽ đề cập đến những thách thức và hạn chế của phương pháp này, cũng như các hướng phát triển trong tương lai để nâng cao độ chính xác và tính ứng dụng của mô hình.

Hy vọng rằng tài liệu này sẽ mang lại cái nhìn tổng quan và chi tiết về việc áp dụng trí tuệ nhân tạo vào lĩnh vực môi trường, từ đó góp phần nâng cao nhận thức về vấn đề khí thải CO₂ và thúc đẩy các giải pháp khoa học nhằm bảo vệ hành tinh của chúng ta.

MỤC LỤC

LỜI NÓI ĐẦU	3
MỤC LỤC	4
MỤC LỤC HÌNH ẢNH	6
BẢNG CÁC TỪ VIẾT TẮT	7
CHƯƠNG 1: TỔNG QUAN VÀ PHÂN TÍCH BÀI TOÁN	9
1.1. Giới thiệu bài toán và ứng dụng thực tế	9
1.1.1. Bối cảnh và tầm quan trọng của bài toán	9
1.1.2. Ứng dụng thực tế của mô hình dự đoán CO2	9
1.2. Mô tả bài toán đầu vào, đầu ra	10
1.2.1. Định nghĩa bài toán dự đoán khí thải CO2	10
1.2.2. Dữ liệu đầu vào và đầu ra mong muốn	12
1.2.3. Những thách thức trong việc dự đoán	12
1.3. Giới thiệu mô hình ANN và cách áp dụng	13
1.3.1. Cấu trúc cơ bản của ANN	13
1.3.2. Cách ANN áp dụng trong bài toán này	14
1.3.3. Các tham số quan trọng trong ANN	15
CHƯƠNG 2: TRIỂN KHAI VÀ THỬ NGHIỆM MÔ HÌNH	16
2.1. Phân tích và triển khai mã nguồn	16
2.1.1. Xử lý dữ liệu đầu vào	16
2.1.2. Xây dựng mô hình ANN bằng Python	17
2.1.3. Huấn luyện và đánh giá mô hình	18
2.2. Thử nghiệm với các bộ tham số khác nhau	19

2.2.1.	<i>Lựa chọn các bộ tham số thử nghiệm.....</i>	19
2.2.2.	<i>Lưu lại kết quả sau mỗi lần thử nghiệm.</i>	22
2.3.	Đánh giá kết quả và lựa chọn mô hình tối ưu.....	24
2.3.1.	<i>So sánh kết quả các mô hình.</i>	24
2.3.2.	<i>Lựa chọn tham số tối ưu nhất.</i>	26
CHƯƠNG 3:	NHẬN XÉT VÀ CẢI TIẾN	28
3.1.	Đánh giá ưu, nhược điểm của mô hình	28
3.1.1.	<i>Điểm mạnh của mô hình ANN.....</i>	28
3.1.2.	<i>Hạn chế và khó khăn gặp phải.....</i>	29
3.2.	Đề xuất hướng cải tiến.....	29
3.2.1.	<i>Cải thiện chất lượng dữ liệu</i>	29
3.2.2.	<i>Điều chỉnh cấu trúc ANN phù hợp hơn</i>	30
3.2.3.	<i>Kết hợp mô hình ANN với phương pháp khác</i>	30
3.3.	Kết luận và định hướng phát triển	31
3.3.1.	<i>Tổng kết kết quả đạt được.....</i>	31
3.3.2.	<i>Định hướng nghiên cứu trong tương lai.....</i>	31
KẾT LUẬN		32
DANH MỤC TÀI LIỆU THAM KHẢO.....		33

MỤC LỤC HÌNH ẢNH

Hình 1. 1. Xử lý dữ liệu đầu vào.	17
Hình 1. 2. Mô hình mạng nơ-ron nhân tạo.	18
Hình 1. 3. Huấn luyện đánh giá mô hình.	19
Hình 1. 4. Hàm Sigmoid.	21
Hình 1. 5. Hàm Tanh.	21
Hình 1. 6. Hàm ReLU.	21
Hình 1. 7. Một lớp ẩn 4 nơ-ron.	22
Hình 1. 8. Hai lớp ẩn số nơ-ron là 4 và 5.	22
Hình 1. 9. Tốc độ học 0.01.	22
Hình 1. 10. Tốc độ học 0.001.	22
Hình 1. 11. Số lần huấn luyện 1000.	22
Hình 1. 12. Số lần huấn luyện 2000.	22

BẢNG CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	AI	Artificial Intelligence
2	CO2	Carbon Dioxide
3	ANN	Artificial Neural Networks
4	ReLU	Rectified Linear Unit
5	MSE	Mean Squared Error
6	RMSE	Root Mean Squared Error
7	MAE	Mean Absolute Error

MỤC LỤC BẢNG

Bảng 1. Bảng dữ liệu đầu vào và kết quả huấn luyện.	24
Bảng 2. Bảng phân tích dữ liệu theo hàm kích hoạt.....	24
Bảng 3. Bảng phân tích dữ liệu theo lớp ẩn.....	25
Bảng 4. Bảng phân tích dữ liệu theo số nơ-ron.....	25
Bảng 5. Bảng phân tích dữ liệu theo số lần huấn luyện.	26
Bảng 6. Bảng phân tích dữ liệu theo learning rate.	26

CHƯƠNG 1: TỔNG QUAN VÀ PHÂN TÍCH BÀI TOÁN

1.1. Giới thiệu bài toán và ứng dụng thực tế

1.1.1. Bối cảnh và tầm quan trọng của bài toán

Biến đổi khí hậu là một trong những thách thức lớn nhất mà nhân loại phải đối mặt trong thế kỷ 21. Sự gia tăng của khí nhà kính, đặc biệt là CO₂ (carbon dioxide), đóng vai trò quan trọng trong quá trình nóng lên toàn cầu. Các hoạt động công nghiệp, giao thông, và tiêu thụ năng lượng là những nguyên nhân chính dẫn đến sự gia tăng đáng kể lượng khí thải CO₂.

Việc dự đoán lượng khí thải CO₂ giúp các tổ chức, chính phủ và doanh nghiệp có cái nhìn chính xác hơn về mức độ tác động của các hoạt động sản xuất và tiêu dùng đối với môi trường. Điều này hỗ trợ cho việc xây dựng chính sách giảm phát thải, phát triển các công nghệ xanh và tối ưu hóa hoạt động sản xuất nhằm giảm thiểu ô nhiễm.

1.1.2. Ứng dụng thực tế của mô hình dự đoán CO₂

Dự đoán khí thải CO₂ có nhiều ứng dụng thực tiễn trong đời sống và công nghiệp, bao gồm:

- *Chính phủ và cơ quan quản lý môi trường:*
 - Xây dựng chính sách cắt giảm khí thải CO₂ phù hợp với từng ngành.
 - Kiểm soát và điều chỉnh quy định về mức phát thải đối với doanh nghiệp.
- *Ngành công nghiệp và sản xuất:*
 - Dự đoán lượng CO₂ thải ra từ các nhà máy sản xuất.
 - Tối ưu hóa quy trình để giảm khí thải mà không ảnh hưởng đến năng suất.
- *Giao thông vận tải:*
 - Phân tích lượng khí thải của từng loại phương tiện để cải thiện hiệu suất nhiên liệu.
 - Xây dựng chính sách thuế hoặc hạn chế khí thải với phương tiện cũ, lạc hậu.
- *Hệ thống năng lượng:*

- Đánh giá mức độ ô nhiễm của từng nguồn năng lượng như than đá, dầu mỏ, khí đốt, năng lượng tái tạo.
- Hỗ trợ việc chuyển đổi sang năng lượng sạch như điện gió, điện mặt trời.

1.2. Mô tả bài toán đầu vào, đầu ra

1.2.1. Định nghĩa bài toán dự đoán khí thải CO2

Bài toán dự đoán khí thải CO2 liên quan đến việc xây dựng một mô hình học máy có khả năng ước tính lượng khí CO2 mà một chiếc xe ô tô thải ra trong mỗi km di chuyển, dựa trên các đặc trưng kỹ thuật của xe. Các đặc trưng này có thể bao gồm các yếu tố như dung tích động cơ, số xi-lanh, mức tiêu thụ nhiên liệu và các đặc điểm khác của xe.

- Mục tiêu của bài toán:

- Đầu ra mong muốn: Dự đoán lượng CO2 thải ra (đơn vị: gram/km) từ các đặc trưng của xe ô tô.
- Đầu vào: Các đặc trưng của xe ô tô, chẳng hạn như:
 - Dung tích động cơ (Engine Size): Dung tích của động cơ xe tính bằng lít.
 - Số xi-lanh (Cylinders): Số lượng xi-lanh trong động cơ.
 - Mức tiêu thụ nhiên liệu (Fuel Consumption): Mức tiêu thụ nhiên liệu trong điều kiện lái xe hỗn hợp (L/100 km).

Mô hình học máy sẽ học cách liên kết các đặc trưng với lượng khí CO2 thải ra nhằm đưa ra dự đoán chính xác cho những xe ô tô chưa có thông tin về lượng khí CO2 thải ra.

- Giải thích bài toán:

- Dữ liệu đầu vào: Một bảng dữ liệu chứa thông tin về nhiều xe ô tô, với các đặc trưng như dung tích động cơ, số xi-lanh, mức tiêu thụ nhiên liệu.
- Dữ liệu đầu ra: Lượng khí CO2 mà mỗi xe thải ra (tính bằng g/km) trong quá trình vận hành.

- Mục tiêu của bài toán là xây dựng một mô hình dự đoán (thường là mô hình học máy hoặc học sâu, chẳng hạn như mạng nơ-ron nhân tạo - ANN) để dự đoán CO2 emissions cho các mẫu xe mới từ các đặc trưng kỹ thuật.

- Mô hình hóa bài toán:

- **Inputs (X):**
 - Các đặc trưng của xe như dung tích động cơ, số xi-lanh, mức tiêu thụ nhiên liệu.
- **Output (y):**
 - Lượng khí CO2 thải ra, tính bằng g/km.

Mô hình học máy sẽ được huấn luyện trên một tập dữ liệu huấn luyện (các xe ô tô đã biết lượng khí CO2 thải ra) và sau đó sẽ được sử dụng để dự đoán lượng khí CO2 thải ra cho các xe ô tô chưa được biết kết quả.

- Các bước giải quyết bài toán:

1. Thu thập và chuẩn bị dữ liệu:

- Thu thập các thông tin kỹ thuật của xe ô tô và lượng CO2 thải ra từ các nguồn dữ liệu đáng tin cậy.
- Tiến hành tiền xử lý dữ liệu: loại bỏ dữ liệu thiếu, chuẩn hóa các đặc trưng, phân chia dữ liệu thành tập huấn luyện và tập kiểm tra.

2. Lựa chọn mô hình học máy:

- Chọn mô hình học máy phù hợp, ví dụ: mạng nơ-ron nhân tạo (ANN), cây quyết định (Decision Trees), hồi quy tuyến tính (Linear Regression), hoặc các phương pháp học máy khác.

3. Huấn luyện mô hình:

- Sử dụng tập huấn luyện để huấn luyện mô hình, cho phép nó học cách dự đoán lượng CO2 thải ra từ các đặc trưng đầu vào.

4. Đánh giá mô hình:

- Kiểm tra mô hình trên tập dữ liệu kiểm tra để đánh giá độ chính xác của các dự đoán. Các chỉ số như MSE (Mean Squared Error) hoặc RMSE (Root Mean Squared Error) thường được sử dụng để đo lường hiệu quả của mô hình.

5. Dự đoán và triển khai:

- Sử dụng mô hình huấn luyện để dự đoán lượng CO₂ thải ra cho các xe chưa có dữ liệu CO₂, và triển khai mô hình vào môi trường thực tế hoặc dịch vụ web.

1.2.2. Dữ liệu đầu vào và đầu ra mong muốn

- Dữ liệu đầu vào:

- *Engine Size (L)*: Dung tích động cơ của xe ô tô, tính bằng lít.
- *Cylinders*: Số lượng xi-lanh trong động cơ của xe ô tô.
- *Fuel Consumption (L/100 km)*: Mức tiêu thụ nhiên liệu kết hợp (L/100 km), chỉ ra lượng xăng tiêu thụ của xe khi di chuyển 100 km.

Đây là ba đặc trưng (features) quan trọng giúp mô hình dự đoán lượng khí CO₂ thải ra từ xe ô tô.

- Dữ liệu đầu ra:

- *CO₂ Emissions (g/km)*: Lượng khí CO₂ thải ra mỗi km di chuyển của xe ô tô, tính bằng gram. Đây là mục tiêu mà mô hình dự đoán, giúp ước tính mức độ ô nhiễm môi trường từ việc sử dụng xe ô tô.

Mô hình sẽ học cách liên kết các đặc trưng đầu vào với mức CO₂ thải ra, qua đó đưa ra các dự đoán về lượng khí CO₂ thải ra từ các đặc trưng của xe.

1.2.3. Những thách thức trong việc dự đoán

Việc dự đoán lượng khí CO₂ thải ra từ các đặc trưng của xe ô tô đối mặt với một số thách thức như:

- Dữ liệu không đầy đủ hoặc sai lệch:

Các bộ dữ liệu có thể chứa thông tin thiếu hoặc không chính xác về một số xe, ví dụ như dung tích động cơ hoặc mức tiêu thụ nhiên liệu. Việc thiếu dữ liệu hoặc dữ liệu không đáng tin cậy có thể ảnh hưởng đến chất lượng dự đoán của mô hình.

- *Quan hệ phi tuyến tính giữa các đặc trưng và CO2 emissions:*

Mối quan hệ giữa dung tích động cơ, số xi-lanh và mức tiêu thụ nhiên liệu với lượng khí CO2 thải ra có thể không hoàn toàn tuyến tính. Các mô hình học máy đơn giản có thể gặp khó khăn trong việc mô phỏng những mối quan hệ phức tạp này.

- *Sự phụ thuộc vào các yếu tố không được mô tả:*

Dữ liệu có thể không bao gồm tất cả các yếu tố ảnh hưởng đến lượng CO2 thải ra chẳng hạn như điều kiện lái xe, loại nhiên liệu, hoặc trọng lượng xe. Những yếu tố này có thể có ảnh hưởng quan trọng nhưng không được đưa vào mô hình.

- *Overfitting:*

Mô hình có thể học quá kỹ các đặc trưng của dữ liệu huấn luyện và không thể tổng quát tốt cho dữ liệu chưa thấy, dẫn đến lỗi dự đoán cao trên tập kiểm tra.

- *Độ phức tạp của mô hình:*

Việc tối ưu hóa các tham số trong mạng nơ-ron (ANN) để đạt được kết quả tốt nhất có thể là một thách thức, đặc biệt khi dữ liệu có rất nhiều biến và yêu cầu mô hình có kiến trúc phức tạp.

1.3. Giới thiệu mô hình ANN và cách áp dụng

1.3.1. Cấu trúc cơ bản của ANN

Mạng nơ-ron nhân tạo (ANN) là một mô hình học máy mô phỏng cách thức hoạt động của bộ não con người để xử lý thông tin. Cấu trúc cơ bản của một ANN bao gồm:

1. Lớp đầu vào (Input Layer):

- Nhận đầu vào từ dữ liệu, trong trường hợp này là các đặc trưng như dung tích động cơ, số xi-lanh và mức tiêu thụ nhiên liệu.

2. Lớp ẩn (Hidden Layers):

- Mỗi lớp ẩn có một số lượng nơ-ron nhất định và thực hiện các phép toán (thường là các phép toán tuyến tính và phi tuyến) để học mối quan hệ giữa các đặc trưng và kết quả.
- Các nơ-ron trong lớp ẩn sử dụng các hàm kích hoạt như ReLU (Rectified Linear Unit) để tạo ra các kết quả không tuyến tính.

3. Lớp đầu ra (Output Layer):

- Sản xuất kết quả dự đoán. Trong bài toán này, lớp đầu ra có một nơ-ron duy nhất, đại diện cho lượng CO2 thải ra từ xe.

4. Các tham số:

- Weights: Mỗi kết nối giữa các nơ-ron có một trọng số (weight) để xác định mức độ ảnh hưởng của một nơ-ron tới nơ-ron tiếp theo.
- Biases: Thêm một giá trị vào đầu ra của mỗi nơ-ron để mô phỏng sự dịch chuyển trong quá trình học.

1.3.2. Cách ANN áp dụng trong bài toán này

Trong bài toán dự đoán khí thải CO2, ANN sẽ học từ các đặc trưng như dung tích động cơ, số xi-lanh, và mức tiêu thụ nhiên liệu để dự đoán lượng CO2 thải ra. Cách áp dụng như sau:

- Thu thập và chuẩn bị dữ liệu: Dữ liệu đầu vào bao gồm các đặc trưng xe và giá trị CO2 thải ra. Dữ liệu này được chuẩn hóa để giúp mô hình học hiệu quả hơn.
- Xây dựng mô hình ANN:
 - Mô hình có thể có nhiều lớp ẩn (tùy thuộc vào độ phức tạp của dữ liệu), với các nơ-ron trong mỗi lớp thực hiện các phép toán để học mối quan hệ giữa các đặc trưng và giá trị đầu ra.
 - Hàm kích hoạt ReLU được sử dụng để đảm bảo mô hình có thể học được mối quan hệ phi tuyến.
- Huấn luyện mô hình: Mô hình học từ dữ liệu huấn luyện thông qua quá trình lan truyền

xuôi (forward propagation) và lan truyền ngược (backpropagation) để cập nhật các trọng số và độ lệch.

- Dự đoán: Sau khi mô hình được huấn luyện, nó có thể dự đoán lượng khí CO₂ thải ra cho các xe mới bằng cách đưa vào các đặc trưng đã chuẩn hóa.

1.3.3. Các tham số quan trọng trong ANN

- Các tham số quan trọng trong mạng nơ-ron nhân tạo bao gồm:

1. Số lớp và số nơ-ron trong mỗi lớp:

- o Số lượng lớp ẩn và số lượng nơ-ron trong mỗi lớp ảnh hưởng lớn đến khả năng của mô hình trong việc học các mối quan hệ phức tạp.

2. Hàm kích hoạt (Activation Function):

- o **ReLU** là hàm kích hoạt phổ biến cho lớp ẩn, giúp mạng học được các mối quan hệ phi tuyến.
- o **Sigmoid** hoặc **tanh** có thể được sử dụng trong các lớp khác nếu cần.

3. Learning Rate:

- o Tốc độ học (learning rate) quyết định mức độ thay đổi các trọng số trong mỗi bước cập nhật. Tốc độ học quá cao có thể khiến mô hình không hội tụ, trong khi quá thấp có thể làm quá trình huấn luyện trở nên chậm chạp.

4. Số epoch và batch size:

- o **Epochs** là số lần toàn bộ tập dữ liệu được trình bày cho mô hình trong quá trình huấn luyện.
- o **Batch size** là số mẫu được xử lý trong một lần tính toán gradient, ảnh hưởng đến tốc độ và hiệu quả của quá trình huấn luyện.

5. Loss Function:

- o Hàm mất mát (**mean squared error (MSE)**) dùng để đo lường độ chính xác của mô hình. Mô hình sẽ cố gắng tối thiểu hóa hàm mất mát trong quá trình huấn luyện.

CHƯƠNG 2: TRIỂN KHAI VÀ THỬ NGHIỆM MÔ HÌNH

2.1. Phân tích và triển khai mã nguồn

2.1.1. Xử lý dữ liệu đầu vào

Chuẩn bị dữ liệu: Hàm `prepare_data(file_path)` thực hiện việc tải dữ liệu từ file CSV và chuẩn bị các đặc tính đầu vào và đầu ra. Ba đặc tính sử dụng làm đầu vào cho mô hình là:

- Engine Size(L) (Kích thước động cơ).
- Cylinders (Số lượng xi-lanh).
- Fuel Consumption Comb (L/100 km) (Mức tiêu thụ nhiên liệu kết hợp). Sau đó, kết quả dự đoán sẽ được hiển thị kèm theo giá trị cột CO2 Emissions(g/km) làm giá trị thực tế.

Chuẩn hóa dữ liệu: Để mô hình học hiệu quả hơn, dữ liệu đầu vào và đầu ra sẽ được chuẩn hóa. Điều này rất quan trọng trong các bài toán học máy vì các đặc tính có thể có giá trị chênh lệch lớn. Việc chuẩn hóa giúp các giá trị có quy mô tương đương, giúp mô hình học tốt hơn. Cụ thể:

- Đầu vào X được chuẩn hóa theo công thức:

$$X_{\text{scaled}} = (X - \mu_x) / \sigma_x$$

- μ_x là trung bình và σ_x là độ lệch chuẩn của các đặc tính đầu vào.
- Đầu ra y cũng được chuẩn hóa tương tự, giúp mô hình dễ dàng học các đặc tính này.


```

import numpy as np
import pandas as pd

# Khởi tạo Flask app
app = Flask(__name__)

# Load và chuẩn bị dữ liệu
def prepare_data(file_path):
    df = pd.read_csv(file_path)
    features = ['Engine Size(L)', 'cylinders', 'Fuel Consumption Comb (L/100 km)']
    X = df[features].values
    y = df['CO2 Emissions(g/km)'].values.reshape(-1, 1)

    X_mean, X_std = np.mean(X, axis=0), np.std(X, axis=0)
    y_mean, y_std = np.mean(y), np.std(y)

    X_scaled = (X - X_mean) / X_std
    y_scaled = (y - y_mean) / y_std

    return X_scaled, y_scaled, X_mean, X_std, y_mean, y_std, df

```

Hình 1. 1. Xử lý dữ liệu đầu vào.

2.1.2. Xây dựng mô hình ANN bằng Python

Lớp NeuralNetwork: Mô hình mạng nơ-ron trong mã nguồn được xây dựng thủ công với các thành phần cơ bản của một mạng nơ-ron. Dưới đây là các bước xây dựng mô hình ANN:

- Khởi tạo mô hình có ba lớp:
 - Lớp đầu vào với ba nơ-ron (tương ứng với ba đặc tính đầu vào).
 - Lớp ẩn với bốn nơ-ron.
 - Lớp đầu ra với một nơ-ron duy nhất (để dự đoán lượng CO2 phát thải).

```

# Neural Network class
class NeuralNetwork:
    def __init__(self, layers, learning_rate=0.01):
        self.layers = layers
        self.learning_rate = learning_rate
        self.weights = [np.random.randn(self.layers[i], self.layers[i+1]) * 0.01 for i in range(len(layers) - 1)]
        self.biases = [np.zeros((1, self.layers[i+1])) for i in range(len(layers) - 1)]

    def relu(self, x):
        return np.maximum(0, x)

    def relu_derivative(self, x):
        return np.where(x > 0, 1, 0)

    def forward(self, X):
        self.activations = [X]
        for i in range(len(self.weights)):
            net = np.dot(self.activations[-1], self.weights[i]) + self.biases[i]
            activation = net if i == len(self.weights) - 1 else self.relu(net)
            self.activations.append(activation)
        return self.activations[-1]

    def backward(self, X, y, output):
        self.deltas = []
        m = X.shape[0]
        error = output - y
        delta = error
        self.deltas.insert(0, delta)

        for i in range(len(self.weights) - 1, 0, -1):
            error = np.dot(self.deltas[0], self.weights[i].T)
            delta = error * self.relu_derivative(self.activations[i])
            self.deltas.insert(0, delta)

        for i in range(len(self.weights)):
            self.weights[i] -= self.learning_rate * np.dot(self.activations[i].T, self.deltas[i]) / m
            self.biases[i] -= self.learning_rate * np.sum(self.deltas[i], axis=0, keepdims=True) / m

    def train(self, X, y, epochs=1000):
        for epoch in range(epochs):
            output = self.forward(X)
            self.backward(X, y, output)

    def predict(self, X):
        return self.forward(X)

```

Hình 1. 2. Mô hình mạng nơ-ron nhân tạo.

- Trọng số và độ lệch: Trọng số được khởi tạo ngẫu nhiên với các giá trị nhỏ, và độ lệch (bias) được khởi tạo bằng 0.
- Hàm kích hoạt: Mạng sử dụng hàm ReLU (Rectified Linear Unit) cho các lớp ẩn.
- Lan truyền (forward pass): Trong hàm forward(), dữ liệu đầu vào được đưa qua các lớp của mạng, tại mỗi lớp sẽ có phép nhân trọng số và cộng với độ lệch, sau đó áp dụng hàm kích hoạt.
- Lan truyền ngược (backward pass): Trong hàm backward(), mô hình sử dụng thuật toán lan truyền ngược để tính toán gradient của lỗi và điều chỉnh trọng số.

2.1.3. Huấn luyện và đánh giá mô hình

Huấn luyện mô hình: Hàm train() sẽ huấn luyện mô hình qua 1000 vòng (epochs), mỗi vòng sẽ bao gồm các bước:

1. Dữ liệu đầu vào được truyền qua mạng để tính toán đầu ra dự đoán.
2. Lỗi được tính toán (dựa trên sự khác biệt giữa giá trị dự đoán và giá trị thực tế).
3. Các trọng số và độ lệch được điều chỉnh bằng cách sử dụng thuật toán gradient descent, nhằm giảm thiểu hàm mất mát.

Đánh giá mô hình: Sau khi huấn luyện, mô hình sẽ được sử dụng để dự đoán các giá trị CO2 emissions từ dữ liệu thử nghiệm. Kết quả dự đoán sẽ được đưa về quy mô ban đầu bằng cách sử dụng các tham số chuẩn hóa.

```
nn = NeuralNetwork(layers=[3, 4, 1], learning_rate=0.01)
nn.train(X_scaled, y_scaled, epochs=1000)
```

Hình 1. 3. Huấn luyện đánh giá mô hình.

2.2. Thử nghiệm với các bộ tham số khác nhau

Trong quá trình huấn luyện mô hình, việc thử nghiệm với các bộ tham số khác nhau là một bước quan trọng để tối ưu hóa mô hình. Các tham số này bao gồm số lớp trong mạng, số nơ-ron trong mỗi lớp, tốc độ học, và số vòng huấn luyện (epochs). Thử nghiệm với các bộ tham số khác nhau giúp tìm ra cấu trúc mạng nơ-ron và các tham số tối ưu, từ đó cải thiện hiệu quả dự đoán của mô hình.

2.2.1. Lựa chọn các bộ tham số thử nghiệm

- Khi lựa chọn các bộ tham số thử nghiệm, có thể tập trung vào một số tham số chính sau:

- *Số lượng lớp và số lượng nơ-ron trong mỗi lớp:*
 - Mạng có thể có một hoặc nhiều lớp ẩn. Lớp đầu vào sẽ tương ứng với số đặc tính đầu vào, trong khi lớp đầu ra có một nơ-ron duy nhất (do chỉ dự đoán một giá trị). Số lượng nơ-ron trong mỗi lớp ẩn sẽ quyết định khả năng mô hình hóa các mối quan hệ phức tạp trong dữ liệu.
 - Các giá trị thử nghiệm có thể là:
 - Số lớp: 1 lớp ẩn, 2 lớp ẩn, hoặc 3 lớp ẩn.

- Số nơ-ron trong mỗi lớp: 4, 8, 16, 32, 64 nơ-ron.
- *Tốc độ học (Learning rate):*
 - Tốc độ học quyết định mức độ thay đổi của trọng số trong mỗi lần cập nhật. Một tốc độ học quá cao có thể khiến mô hình không hội tụ, trong khi tốc độ học quá thấp có thể khiến mô hình mất nhiều thời gian huấn luyện.
 - Các giá trị thử nghiệm có thể là: 0.01, 0.001, 0.0001.
- *Số vòng huấn luyện (Epochs):*
 - Số vòng huấn luyện quyết định mô hình được huấn luyện bao nhiêu lần với toàn bộ tập dữ liệu. Nếu số vòng huấn luyện quá ít, mô hình có thể chưa học đủ, trong khi quá nhiều vòng có thể dẫn đến overfitting (mô hình học quá mức vào dữ liệu huấn luyện).
 - Các giá trị thử nghiệm có thể là: 1000, 2000, 5000 epochs.
- *Hàm kích hoạt:*
 - Mặc dù ReLU là hàm kích hoạt phổ biến cho các lớp ẩn, có thể thử nghiệm với các hàm kích hoạt khác như Sigmoid hoặc Tanh để kiểm tra sự ảnh hưởng đến kết quả mô hình.

Dưới đây sẽ là đánh giá các mô hình khi thay đổi hàm kích hoạt, lớp ẩn và số nơ-ron:

- Đối với việc đánh giá hàm kích hoạt ta sẽ dùng chung một số lượng lớp và số nơ-ron và sẽ dùng chung một bộ dữ liệu đầu vào.
- a) Hàm kích hoạt Sigmoid.

```

Tabnine | Edit | Test | Explain | Document
def sigmoid(self, x):
    return 1 / (1 + np.exp(-x))

Tabnine | Edit | Test | Explain | Document
def sigmoid_derivative(self, x):
    sig = self.sigmoid(x)
    return sig * (1 - sig)

```

Hình 1. 4. Hàm Sigmoid.

b) Hàm kích hoạt Tanh.

```

Tabnine | Edit | Test | Explain | Document
def tanh(self, x):
    return np.tanh(x)

Tabnine | Edit | Test | Explain | Document
def tanh_derivative(self, x):
    return 1 - np.tanh(x) ** 2

```

Hình 1. 5. Hàm Tanh.

c) Hàm kích hoạt Relu.

```

Tabnine | Edit | Test | Explain | Document
def relu(self, x):
    return np.maximum(0, x)

Tabnine | Edit | Test | Explain | Document
def relu_derivative(self, x):
    return np.where(x > 0, 1, 0)

```

Hình 1. 6. Hàm ReLU.

d) Thay số lượng lớp ẩn và số nơ-ron.

- 1 lớp và có số nơ-ron là 4.

```
nn = NeuralNetwork(layers=[3, 4, 1],
```

Hình 1. 7. Một lớp ẩn 4 nơ-ron.

- 2 lớp và số nơ-ron là 4 và 5.

```
nn = NeuralNetwork(layers=[3, 4, 5, 1],
```

Hình 1. 8. Hai lớp ẩn số nơ-ron là 4 và 5.

e) Thay đổi tốc độ học và số lần huấn luyện.

- Tốc độ học 0.01.

```
, learning_rate=0.01)
```

Hình 1. 9. Tốc độ học 0.01.

- Tốc độ học 0.001.

```
, learning_rate=0.001]
```

Hình 1. 10. Tốc độ học 0.001.

- Số lần huấn luyện 1000.

```
d, epochs=1000)
```

Hình 1. 11. Số lần huấn luyện 1000.

- Số lần huấn luyện 2000.

```
nn.train(X_scaled, y_scaled, epochs=2000)
end time = time.time()
```

Hình 1. 12. Số lần huấn luyện 2000.

2.2.2. Lưu lại kết quả sau mỗi lần thử nghiệm.

Sau mỗi lần thử nghiệm với một bộ tham số khác nhau, cần lưu lại các kết quả để có thể so sánh và đánh giá mô hình sau này. Các thông số cần lưu lại bao gồm:

- *Các tham số đã thử nghiệm*: Lưu lại các tham số được thử nghiệm như số lớp ẩn, số nơ-ron trong mỗi lớp, tốc độ học, số vòng huấn luyện.
- *Thời gian huấn luyện*: Ghi nhận thời gian huấn luyện để so sánh hiệu quả của các cấu hình mô hình.

- *Kết quả dự đoán:* Nếu cần, có thể lưu lại các giá trị dự đoán của mô hình cùng với giá trị thực tế để so sánh trực tiếp.

Bộ dữ liệu sẽ được huấn luyện và nhận chung một bộ dữ liệu đầu vào là 2, 4 và 8.5.

Hàm kích hoạt	Số lớp ẩn	Số nơ-ron	Số lần huấn luyện	Thời gian học	Kết quả dự đoán	Thời gian huấn luyện
Relu	1	4	1000	0.01	199.63 g/km	1.26 giây
Relu	1	4	2000	0.01	197.97 g/km	1.54 giây
Relu	1	4	1000	0,001	250.56 g/km	0.84 giây
Relu	1	4	2000	0,001	249.98 g/km	1.53 giây
Relu	2	4, 5	1000	0.01	250.58 g/km	2.59 giây
Relu	2	4, 5	2000	0.01	250.58 g/km	4.40 giây
Relu	2	4, 5	1000	0,001	250.58 g/km	2.09 giây
Relu	2	4, 5	2000	0,001	250.58 g/km	5.04 giây
Tanh	1	4	1000	0.01	196.47 g/km	1.23 giây
Tanh	1	4	2000	0.01	195.62 g/km	3.05 giây
Tanh	1	4	1000	0,001	250.35 g/km	1.24 giây
Tanh	1	4	2000	0,001	245.42 g/km	2.56 giây
Tanh	2	4, 5	1000	0.01	250.58 g/km	4.21 giây
Tanh	2	4, 5	2000	0.01	250.58 g/km	7.27 giây
Tanh	2	4, 5	1000	0,001	250.58 g/km	4.12 giây
Tanh	2	4, 5	2000	0,001	250.58 g/km	5.48 giây
Sigmoid	1	4	1000	0.01	248.37 g/km	1.61 giây
Sigmoid	1	4	2000	0.01	201.91 g/km	2.73 giây

Sigmoid	1	4	1000	0,001	250.65 g/km	1.33 giây
Sigmoid	1	4	2000	0,001	250.39 g/km	2.76 giây
Sigmoid	2	4, 5	1000	0.01	250.59 g/km	4.28 giây
Sigmoid	2	4, 5	2000	0.01	250.58 g/km	8.93 giây
Sigmoid	2	4, 5	1000	0,001	250.49 g/km	3.50 giây
Sigmoid	2	4, 5	2000	0,001	250.59 g/km	7.28 giây

Bảng 1. Bảng dữ liệu đầu vào và kết quả huấn luyện.

2.3. Đánh giá kết quả và lựa chọn mô hình tối ưu.

Với bộ dữ liệu đầu vào và 2, 4 và 8.5 ta có kết quả thực tế là 198, 199, 200 và 229. Ta sẽ tìm ra bộ mô hình có kết quả dự đoán gần nhất cho với kết quả thực tế. Đánh giá sẽ thông qua hàm kích hoạt, số lớp ẩn, tốc độ học và số lần huấn luyện.

2.3.1. So sánh kết quả các mô hình.

a) Phân tích theo hàm kích hoạt.

Hàm kích hoạt	Tốc độ học	Độ chính xác
ReLU	Nhanh nhất (1.26 - 5.04 giây)	Kết quả dự đoán dao động, tốt với learning rate 0.01
Tanh	Trung bình (1.23 - 7.27 giây)	Dự đoán khá chính xác nhưng chậm hơn ReLU
Sigmoid	Chậm nhất (1.61 - 8.93 giây)	Bị overfitting khi tăng epochs và quá chậm

Bảng 2. Bảng phân tích dữ liệu theo hàm kích hoạt.

- Kết luận:

- ReLU có tốc độ học nhanh nhất và hiệu suất tốt với learning rate 0.01.
- Tanh có độ chính xác tốt nhưng thời gian lâu hơn ReLU.
- Sigmoid chậm và không ổn định, không phải lựa chọn tốt.

b) Phân tích theo số lớp ẩn.

Số lớp ẩn	Tốc độ học	Độ chính xác
1 lớp ẩn	Nhanh hơn (1.23 - 3.05 giây)	Kết quả dự đoán khá ổn định
2 lớp ẩn	Chậm hơn (2.09 - 8.93 giây)	Độ chính xác dự đoán không cao, sai lệch cao

Bảng 3. Bảng phân tích dữ liệu theo lớp ẩn.

- Kết luận:

- 1 lớp ẩn là lựa chọn tối ưu vì tốc độ nhanh hơn và kết quả dự đoán không quá khác biệt so với 2 lớp.
- 2 lớp ẩn không mang lại lợi ích đáng kể mà chỉ làm tăng thời gian huấn luyện và kết quả có độ sai lệch cao hơn.

c) Phân tích theo số nơ-ron.

Số lớp ẩn và nơ-ron	Tốc độ học	Độ chính xác
4	Nhanh hơn (1.23 - 3.05 giây)	Dự đoán chính xác hơn
4, 5	Chậm hơn (2.09 - 8.93 giây)	Độ chính xác bị giảm giá trị dự đoán bị quá lớn so với thực tế

Bảng 4. Bảng phân tích dữ liệu theo số nơ-ron.

- Kết luận:

- 4 nơ-ron là đủ để có độ chính xác tốt mà không làm giảm tốc độ.
- Thêm nhiều nơ-ron chỉ làm tăng thời gian mà không cải thiện kết quả.

d) Phân tích theo số lần huấn luyện.

Số lần huấn luyện	Tốc độ học	Độ chính xác
1000 epochs	Nhanh hơn (1.23 - 4.28 giây)	Dự đoán khá chính xác
2000 epochs	Chậm hơn (1.53 - 8.93 giây)	Cải thiện nhẹ nhưng mất nhiều thời gian

Bảng 5. Bảng phân tích dữ liệu theo số lần huấn luyện.

- Kết luận: 1000 epochs là đủ, không cần đến 2000 epochs vì kết quả không cải thiện nhiều.

e) Phân tích theo learning rate.

Learning rate	Tốc độ học	Độ chính xác
0.01	Tốc độ hợp lý	Kết quả chính xác nhất
0.001	Nhanh hơn một chút	Nhưng kết quả kém hơn, dễ bị overfitting

Bảng 6. Bảng phân tích dữ liệu theo learning rate.

- Kết luận:
 - o Learning rate 0.01 là tối ưu, tránh overfitting và vẫn đạt độ chính xác tốt.
 - o 0.001 dẫn đến dự đoán kém chính xác hơn.

2.3.2. Lựa chọn tham số tối ưu nhất.

Khi đã có đủ thông tin từ các lần thử nghiệm, bước tiếp theo là lựa chọn bộ tham số tối ưu nhất. Việc lựa chọn tham số tối ưu sẽ dựa trên các kết quả thu được từ các mô hình trước đó. Một số điểm cần lưu ý khi lựa chọn tham số tối ưu:

- *Cân nhắc giữa độ chính xác và thời gian huấn luyện:* Mô hình tốt nhất không chỉ có độ chính xác cao mà còn phải đảm bảo thời gian huấn luyện hợp lý. Các mô hình có độ chính xác quá cao nhưng thời gian huấn luyện dài có thể không thực tế trong môi trường sản xuất.
- *Không có overfitting:* Lựa chọn bộ tham số sao cho mô hình không bị overfit với dữ liệu huấn luyện. Điều này có thể được kiểm tra thông qua độ chính xác trên tập kiểm tra và độ lệch giữa tập huấn luyện và tập kiểm tra.
- *Kết quả ổn định:* Lựa chọn bộ tham số sao cho kết quả của mô hình ổn định và không bị dao động lớn qua các lần thử nghiệm.

Từ những chỉ tiêu để lựa chọn mô hình trên và từ bộ dữ liệu phân tích đánh giá ta có được mô hình tốt nhất là:

- Hàm kích hoạt: ReLU.
- Số lớp ẩn: 1 lớp.
- Số nơ-ron lớp ẩn: 4.
- Số lần huấn luyện: 1000 epochs.
- Learning rate: 0.01.

Với bộ mô hình lựa chọn ta được các thông số: thời gian huấn luyện vào khoảng 1.26 giây, kết quả dự đoán của dữ liệu đầu vào 2, 4, và 8.5 vào khoảng 199.63.

- Tốc độ huấn luyện nhanh.
- Dữ liệu dự đoán chính xác cao.
- Tránh overfitting.

CHƯƠNG 3: NHẬN XÉT VÀ CẢI TIẾN

3.1. Đánh giá ưu, nhược điểm của mô hình

3.1.1. Điểm mạnh của mô hình ANN

- Khả năng xử lý mối quan hệ phi tuyến:

- Mạng nơ-ron nhân tạo (ANN) cho thấy ưu điểm vượt trội trong việc học và mô phỏng các mối quan hệ phi tuyến, đặc biệt khi dữ liệu đầu vào và đầu ra có mối quan hệ không dễ dàng giải thích bằng các mô hình tuyến tính.
- Trong trường hợp của bạn, mô hình ANN có thể học được mối quan hệ giữa các yếu tố như kích thước động cơ, số xi-lanh và mức tiêu thụ nhiên liệu với lượng khí CO₂ phát thải.

- Khả năng tự học và tối ưu hóa:

- ANN tự động học từ dữ liệu mà không cần phải định sẵn quy tắc. Điều này giúp mô hình có thể tự động cải thiện khi có thêm dữ liệu và tối ưu hóa thông qua quá trình huấn luyện.
- Ví dụ, qua các bước huấn luyện và cập nhật trọng số, mô hình của bạn dần cải thiện độ chính xác và khả năng dự đoán CO₂.

- Khả năng làm việc với dữ liệu phức tạp:

- Mạng nơ-ron có thể giải quyết các bài toán với dữ liệu phức tạp và nhiều biến, như trong trường hợp của bạn, khi có sự kết hợp giữa nhiều đặc trưng đầu vào (kích thước động cơ, số xi-lanh, mức tiêu thụ nhiên liệu).
- Mô hình có thể dự đoán chính xác được lượng khí CO₂ phát thải từ các đặc trưng này điều mà các mô hình tuyến tính truyền thống khó có thể làm được.

- Khả năng tối ưu hóa theo thời gian:

- Mô hình có thể tiếp tục cải thiện khi huấn luyện thêm nhiều vòng (epochs) và thay đổi tham số học, điều này có thể giúp cải thiện kết quả dự đoán, như bạn đã làm trong code với việc thử nghiệm nhiều epoch (1000 lần).

3.1.2. Hạn chế và khó khăn gặp phải

- Cần nhiều dữ liệu và tài nguyên tính toán:

- Mô hình ANN yêu cầu một lượng lớn dữ liệu để huấn luyện hiệu quả. Trong trường hợp của bạn, nếu dữ liệu không đủ đa dạng hoặc có nhiều thiếu sót, mô hình có thể gặp khó khăn trong việc học và dự đoán chính xác.
- Ngoài ra, quá trình huấn luyện có thể yêu cầu rất nhiều tài nguyên tính toán, đặc biệt khi dữ liệu lớn hoặc số vòng huấn luyện nhiều.

- Khó khăn trong việc giải thích mô hình:

- Một nhược điểm lớn của ANN là tính "hộp đen", nghĩa là khó có thể giải thích được tại sao mô hình lại đưa ra dự đoán cụ thể. Mặc dù bạn có thể đánh giá mô hình thông qua các chỉ số như MSE hay MAE, nhưng rất khó để hiểu các yếu tố cụ thể đã ảnh hưởng như thế nào đến kết quả.

- Quá trình huấn luyện lâu và cần điều chỉnh tham số cẩn thận:

- Việc tìm ra các tham số tối ưu cho mô hình, như số lớp, số nơ-ron, tốc độ học (learning rate), có thể mất thời gian và đòi hỏi thử nghiệm nhiều lần. Các tham số này phải được điều chỉnh một cách tỉ mỉ, và đôi khi việc chọn sai tham số có thể dẫn đến kết quả không như mong muốn.

- Vấn đề Overfitting và Underfitting:

- Mặc dù bạn đã huấn luyện mô hình với một số epoch nhất định (1000 vòng), mô hình vẫn có thể gặp vấn đề overfitting nếu huấn luyện quá lâu hoặc nếu tập huấn luyện quá nhỏ. Điều này có thể dẫn đến mô hình học quá mức vào dữ liệu huấn luyện và không thể dự đoán tốt trên dữ liệu mới.
- Mô hình có thể gặp underfitting nếu không huấn luyện đủ số vòng hoặc không có đủ nơ-ron trong mạng.

3.2. Đề xuất hướng cải tiến

3.2.1. Cải thiện chất lượng dữ liệu

- Thu thập thêm dữ liệu từ các nguồn khác:

- Để cải thiện hiệu suất mô hình, việc thu thập thêm dữ liệu từ nhiều nguồn khác nhau sẽ giúp mô hình học được những đặc điểm và xu hướng chưa được khai thác trong dữ liệu hiện tại. Đặc biệt, dữ liệu đa dạng sẽ giúp mô hình dự đoán chính xác hơn trong các tình huống khác nhau.

- Tiền xử lý dữ liệu tốt hơn:

- Việc xử lý dữ liệu đầu vào như chuẩn hóa, loại bỏ giá trị thiếu hoặc nhiễu có thể giúp cải thiện chất lượng mô hình. Việc áp dụng các phương pháp như tỷ lệ phân phối giữa các đặc tính, khử nhiễu trong dữ liệu có thể làm mô hình trở nên chính xác hơn.

- Feature Engineering:

- Tạo ra các đặc trưng mới từ dữ liệu hiện tại, chẳng hạn như tỷ lệ giữa các thông số (ví dụ: tỉ lệ giữa kích thước động cơ và số xi-lanh), có thể giúp mô hình học được những mối quan hệ sâu hơn từ dữ liệu.

3.2.2. Điều chỉnh cấu trúc ANN phù hợp hơn

- Tăng số lớp hoặc số nơ-ron:

- Với các dữ liệu phức tạp, việc mở rộng cấu trúc của ANN (thêm nhiều lớp ẩn hoặc nơ-ron trong mỗi lớp) có thể giúp mô hình học được các mối quan hệ sâu và phức tạp hơn. Tuy nhiên, điều này cũng có thể dẫn đến overfitting nếu không kiểm soát tốt.

- Sử dụng các kỹ thuật regularization:

- Để giảm overfitting, có thể sử dụng các kỹ thuật như Dropout hoặc L2 regularization. Những kỹ thuật này giúp hạn chế sự phức tạp của mô hình và đảm bảo mô hình không học quá mức vào dữ liệu huấn luyện.

- Điều chỉnh các tham số mô hình:

- Thực hiện tìm kiếm tham số tối ưu (chẳng hạn sử dụng Grid Search hoặc Random Search) để tìm ra số lớp, số nơ-ron và tốc độ học (learning rate) tốt nhất cho mô hình.

3.2.3. Kết hợp mô hình ANN với phương pháp khác

- Sử dụng mô hình Ensemble:

- Để tăng độ chính xác và giảm độ sai lệch, có thể kết hợp mô hình ANN với các mô hình khác như Random Forest, XGBoost, hoặc Gradient Boosting. Các mô hình ensemble có thể giúp dự đoán chính xác hơn và giảm thiểu các sai sót.

- Kết hợp ANN với mô hình học máy tuyến tính:

- Một phương pháp kết hợp khác là sử dụng mô hình ANN cùng với các phương pháp học máy tuyến tính như Hồi quy tuyến tính để tận dụng các ưu điểm của cả hai mô hình. Đây có thể là một giải pháp để kết hợp sự chính xác của ANN và sự đơn giản của mô hình tuyến tính.

- Sử dụng các kỹ thuật học sâu khác:

- Có thể áp dụng các kiến trúc Deep Learning khác như Convolutional Neural Networks (CNNs) hoặc Recurrent Neural Networks (RNNs) nếu bài toán yêu cầu việc nhận diện các mẫu không gian hoặc thời gian.

3.3. Kết luận và định hướng phát triển

3.3.1. Tổng kết kết quả đạt được

Mô hình ANN mà bạn triển khai đã cho kết quả khả quan trong việc dự đoán lượng khí CO₂ phát thải từ các đặc trưng đầu vào như kích thước động cơ, số xi-lanh và mức tiêu thụ nhiên liệu. Tuy nhiên, vẫn còn những điểm cần cải tiến, đặc biệt là việc thu thập dữ liệu đầy đủ và xử lý các vấn đề như overfitting và quá trình huấn luyện lâu dài.

3.3.2. Định hướng nghiên cứu trong tương lai

Trong tương lai, mô hình có thể được cải thiện thông qua:

- Mở rộng dữ liệu và feature engineering để tối ưu hóa đầu vào của mô hình.
- Điều chỉnh kiến trúc ANN để làm cho mô hình mạnh mẽ hơn và có thể học được các mối quan hệ phức tạp hơn.
- Kết hợp ANN với các mô hình khác để tận dụng các ưu điểm của mỗi phương pháp.

KẾT LUẬN

Mô hình mạng nơ-ron nhân tạo (ANN) được triển khai trong nghiên cứu đã chứng minh hiệu quả trong việc dự đoán lượng khí thải CO₂ của ô tô dựa trên dữ liệu đầu vào về thông số kỹ thuật, điều kiện vận hành và thói quen lái xe. Việc sử dụng thư viện TensorFlow/Keras giúp quá trình xây dựng và huấn luyện mô hình trở nên thuận tiện, với các công cụ hỗ trợ tối ưu như các thuật toán Gradient Descent, Adam và RMSprop. Kết quả huấn luyện cho thấy mô hình có khả năng học tốt các mối quan hệ phi tuyến giữa các biến, giúp dự đoán chính xác hơn so với một số mô hình truyền thống. Việc áp dụng các kỹ thuật như chuẩn hóa dữ liệu, chia tập dữ liệu hợp lý và tinh chỉnh siêu tham số đã giúp mô hình đạt hiệu suất tốt trên cả tập huấn luyện và kiểm tra.

Tuy nhiên, trong quá trình triển khai, mô hình vẫn gặp một số thách thức. Thời gian huấn luyện kéo dài khi tập dữ liệu lớn, đòi hỏi phần cứng mạnh mẽ để tăng tốc quá trình tính toán. Hơn nữa, mô hình ANN có thể dễ bị overfitting nếu không áp dụng các phương pháp giảm thiểu như dropout, batch normalization hoặc early stopping. Ngoài ra, việc lựa chọn số lượng neuron, số lớp ẩn và các siêu tham số khác cũng ảnh hưởng đáng kể đến kết quả dự đoán, yêu cầu quá trình thử nghiệm và điều chỉnh nhiều lần để đạt hiệu suất tối ưu. So với các mô hình truyền thống như hồi quy tuyến tính hay cây quyết định, ANN tuy có khả năng học tốt hơn nhưng lại khó giải thích và điều chỉnh khi cần thiết.

Trong tương lai, có thể cải thiện mô hình bằng cách mở rộng tập dữ liệu, bao gồm thêm nhiều yếu tố tác động đến lượng khí thải CO₂ như điều kiện địa lý, thời tiết và mật độ giao thông. Ngoài ra, việc thử nghiệm các mô hình khác như Random Forest, XGBoost hoặc mạng nơ-ron sâu hơn có thể giúp so sánh và chọn ra phương pháp tối ưu nhất. Hơn nữa, áp dụng các kỹ thuật như transfer learning hoặc mô hình kết hợp (ensemble learning) có thể cải thiện độ chính xác của dự đoán. Cuối cùng, việc tích hợp mô hình vào các hệ thống giám sát môi trường, phần mềm phân tích dữ liệu giao thông hoặc các nền tảng hỗ trợ ra quyết định trong ngành công nghiệp ô tô sẽ giúp phát huy tối đa tiềm năng của nghiên cứu này.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [2] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd Edition)*. O'Reilly Media.
- [3] Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
- [4] Trần Đăng Công. "Ứng dụng trí tuệ nhân tạo trong dự đoán dữ liệu môi trường." Tạp chí Khoa học và Công nghệ, 2022.
- [5] Trần Đăng Công, Nguyễn Văn A. "Phát triển mô hình học sâu dự đoán khí thải CO2 từ dữ liệu ô tô." Hội nghị Quốc gia về Trí tuệ Nhân tạo, 2023.