

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

TÌM HIỂU MÔ HÌNH ANN CHO VIỆC PHÂN LOẠI ẢNH

Giảng viên hướng dẫn:
ThS. Ngô Thanh Huy

Sinh viên thực hiện:
Họ tên: Phạm Khánh Đăng
MSSV: 110121009
Lớp: DA21TTB

Trà Vinh, tháng 12 năm 2024

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

TÌM HIỂU MÔ HÌNH ANN CHO VIỆC PHÂN LOẠI ẢNH

Giảng viên hướng dẫn:
ThS. Ngô Thanh Huy

Sinh viên thực hiện:
Họ tên: Phạm Khánh Đăng
MSSV: 110121009
Lớp: DA21TTB

Trà Vinh, tháng 12 năm 2024

[illegible]

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Ngô Thanh Huy đã dành thời gian, công sức trong suốt quá trình hướng dẫn em thực hiện đồ án. Những kiến thức quý báu và những lời chỉ bảo của thầy đã giúp em hoàn thành đồ án này một cách tốt nhất. Trong suốt quá trình làm việc, thầy luôn tận tình giải đáp các thắc mắc, chỉ dẫn chi tiết và đưa ra những góp ý sâu sắc để em có thể hoàn thiện đồ án một cách hoàn chỉnh.

Em xin gửi lời cảm ơn chân thành đến thầy!

Trân trọng,

Phạm Khánh Đăng

MỤC LỤC

MỤC LỤC	4
DANH MỤC HÌNH ẢNH	6
TÓM TẮT ĐỒ ÁN ĐỒ ÁN	7
1. Vấn đề nghiên cứu	7
2. Hướng tiếp cận	8
3. Cách giải quyết vấn đề	9
4. Kết quả đạt được	10
MỞ ĐẦU	11
1. Lý do chọn đề tài	11
2. Mục đích	11
3. Đối tượng	12
4. Phạm vi nghiên cứu	13
CHƯƠNG 1: TỔNG QUAN	15
1.1 Giới thiệu tổng quan	15
1.1.1 Mạng lưới nơron trong trí tuệ nhân tạo (ANN) là gì?	15
1.1.2 Cấu trúc của mạng nơron nhân tạo	15
1.1.3 Các loại mạng thần kinh nhân tạo ANN	16
1.1.4 Ứng dụng của mạng thần kinh nhân tạo ANN	16
1.2 Cách giải quyết những hạn chế của ANN	17
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	18
2.1 Cấu trúc mạng nơ-ron nhân tạo (ANN):	18
2.1.1 Lớp đầu vào (Input Layer)	18
2.1.2 Lớp ẩn (Hidden Layer)	19
2.1.3 Lớp đầu ra (Output Layer)	20
2.1.4 Trọng số (Weights) và Bias	20
2.1.5 Quá trình hoạt động của ANN	21
2.2 Quá trình phân loại ảnh bằng Mạng Nơ-ron Nhân tạo (ANN)	21
2.3 Ưu điểm và hạn chế của mô hình ANN trong phân loại ảnh	24
2.3.1 Ưu điểm	24
2.3.2 Hạn chế	24
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	26
3.1 Giới thiệu	26
3.1.1 Mục tiêu của nghiên cứu	26
3.1.2 Phạm vi nghiên cứu	26
3.2 Các bước nghiên cứu đã tiến hành	26
3.2.1 Tìm hiểu lý thuyết	26
3.2.2 Thu thập và phân tích dữ liệu	27
3.2.3 Lựa chọn công cụ và môi trường phát triển	27
3.3 Thiết kế hệ thống và bản mô tả chi tiết	27
3.3.1 Yêu cầu hệ thống	27
3.3.2 Phân tích và thiết kế hệ thống	28
3.3.3 Giải thích thiết kế mô hình ANN	29
3.4 Cách thức cài đặt chương trình	29
3.4.1. Cấu trúc thư mục	29
3.4.2 Chương trình chi tiết	30
3.5 Hiện thực hóa và kết quả	31
3.5.1 Trực quan hóa kết quả	31
3.5.2. Đánh giá mô hình	31
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	32
4.1 Cài đặt thư viện	32

4.2 Nhập các thư viện	32
4.3 Thực hiện tải dữ liệu	32
4.4 Xây dựng mô hình ANN.....	33
4.5 Biên dịch huấn luyện mô hình	33
4.6 Đánh giá mô hình.....	34
4.7 Dự đoán kết quả và hiển thị hình ảnh	35
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	36
5.1 Kết luận.....	36
5.2 Hướng phát triển	36
DANH MỤC TÀI LIỆU THAM KHẢO.....	37

DANH MỤC HÌNH ẢNH

Hình 1 Cấu trúc thư mục.....	30
Hình 2 Cài đặt thư viện.....	32
Hình 3 Nhập thư viện.....	32
Hình 4 Tải dữ liệu.....	33
Hình 5 Xây dựng mô hình ANN.....	33
Hình 6 Biên dịch mô hình.....	34
Hình 7 Đánh giá mô hình.....	34
Hình 8 Dự đoán kết quả.....	35
Hình 9 Hiện thị ảnh.....	35

TÓM TẮT ĐỒ ÁN ĐỒ ÁN

1. Vấn đề nghiên cứu

- Tiền xử lý dữ liệu ảnh
 - Kích thước ảnh: Ảnh đầu vào cần được chuẩn hóa về kích thước cố định để phù hợp với mô hình ANN.
 - Biến đổi dữ liệu (Data Augmentation): Áp dụng các kỹ thuật như xoay, lật, thay đổi độ sáng để tăng cường dữ liệu và giảm overfitting.
 - Chuẩn hóa pixel: Chuyển giá trị pixel (0-255) về một phạm vi nhất định (ví dụ: $[0, 1]$) để mô hình học tốt hơn.
 - Giảm chiều dữ liệu: Với ảnh có độ phân giải cao, cần xem xét giảm chiều để giảm độ phức tạp tính toán.
- Thiết kế kiến trúc ANN
 - Số lớp và số lượng neuron: cần xác định số lượng lớp (input, hidden, output) phù hợp với bài toán. Số neuron trong mỗi lớp hidden cần cân nhắc để tránh tình trạng underfitting hoặc overfitting.
 - Hàm kích hoạt (Activation Function): Lựa chọn hàm kích hoạt phù hợp như ReLU, Sigmoid, hoặc tanh tùy thuộc vào yêu cầu bài toán.
 - Regularization: Áp dụng Dropout hoặc L2 regularization để giảm overfitting.
 - Optimizer: Lựa chọn thuật toán tối ưu như SGD, Adam để cải thiện quá trình học.
- Hiệu suất mô hình
 - Tốc độ huấn luyện: ANN có thể chậm khi xử lý dữ liệu ảnh lớn. Xem xét việc sử dụng GPU hoặc tối ưu code để tăng tốc độ huấn luyện.
 - Độ chính xác: Đảm bảo mô hình đạt độ chính xác cao bằng cách tối ưu siêu tham số (learning rate, batch size).

- Vấn đề overfitting/underfitting: Theo dõi và điều chỉnh mô hình để tránh overfitting (mô hình học quá kỹ dữ liệu huấn luyện) hoặc underfitting (mô hình học không đủ).
- Đánh giá hiệu suất mô hình
 - Các chỉ số đánh giá: Sử dụng các chỉ số như Accuracy, Precision, Recall, F1-Score để đánh giá chất lượng phân loại.
 - Ma trận nhầm lẫn (Confusion Matrix): Phân tích chi tiết các trường hợp mô hình dự đoán sai để hiểu rõ hơn về hiệu suất.
 - Cross-validation: Áp dụng k-fold cross-validation để đảm bảo kết quả huấn luyện ổn định.
- Ứng dụng và mở rộng
 - Ứng dụng thực tế: Nghiên cứu các ứng dụng thực tế của ANN trong phân loại ảnh (nhận diện chữ viết tay, phân loại đồ vật...).
 - Hướng phát triển: Tích hợp ANN với các mô hình khác (như CNN hoặc GAN) để tăng cường hiệu suất. Áp dụng ANN cho các bài toán khác như nhận diện đối tượng (object detection) hoặc phân đoạn ảnh (image segmentation).

2. Hướng tiếp cận

Kích thước và độ phức tạp của dữ liệu: Với dữ liệu lớn và phức tạp, các kiến trúc CNN sâu và rộng thường cho kết quả tốt hơn.

Yêu cầu về thời gian thực: Nếu cần thời gian thực, cần ưu tiên các mô hình nhỏ gọn và nhanh.

Tính chất của bài toán: Các bài toán khác nhau có thể yêu cầu các kiến trúc và kỹ thuật khác nhau.

Tài nguyên tính toán: Khả năng tính toán sẽ ảnh hưởng đến lựa chọn kiến trúc và kích thước mô hình.

3. Cách giải quyết vấn đề

❖ Xác định và phân tích bài toán

- Mục tiêu: Dự đoán nhãn phân loại cho từng ảnh từ tập dữ liệu.
- Đặc điểm dữ liệu:
 - Dữ liệu ảnh có nhiều kích thước và định dạng khác nhau.
 - Dữ liệu có thể có nhiều hoặc không cân bằng giữa các lớp.
- Đầu vào và đầu ra:
 - Đầu vào: Ảnh, được biểu diễn dưới dạng ma trận các pixel (ví dụ: ảnh grayscale hoặc RGB).
 - Đầu ra: Nhãn (label) tương ứng cho mỗi ảnh:
 - + Phân loại nhị phân (0 hoặc 1).
 - + Phân loại đa lớp (ví dụ: 10 chữ số từ 0-9).

❖ Chuẩn bị dữ liệu

- Thu thập và phân chia dữ liệu
 - Nguồn dữ liệu: Bộ dữ liệu phổ biến như MNIST (chữ viết tay), CIFAR-10 (phân loại đối tượng), hoặc tạo dữ liệu từ thực tế.
 - Phân chia dữ liệu:
 - + Tập huấn luyện (training set): 70-80% dữ liệu.
 - + Tập kiểm tra (test set): 10-20%.
 - + Tập xác thực (validation set): 10% (nếu cần).
- Tiền xử lý dữ liệu
 - Chuẩn hóa kích thước ảnh: Biến đổi tất cả ảnh về cùng một kích thước cố định (ví dụ: 28x28 hoặc 32x32 pixel).
 - Flatten dữ liệu: Chuyển ảnh 2D thành vector 1D để phù hợp với ANN.

- Chuẩn hóa giá trị pixel: Chia giá trị pixel (0-255) thành khoảng $[0, 1]$ hoặc chuẩn hóa về zero-mean.
- Biến đổi dữ liệu
 - Data Augmentation: Tăng cường dữ liệu bằng các kỹ thuật như xoay, lật, co giãn để giảm overfitting.
 - Encoding nhãn: Sử dụng One-hot encoding cho các nhãn phân loại đa lớp.

4. Kết quả đạt được

- Độ chính xác cao: Vượt trội so với các phương pháp truyền thống. ANN đạt được độ chính xác thậm chí còn vượt qua khả năng phân loại của con người.
- Khả năng tổng quát tốt: Ứng dụng cho nhiều loại hình ảnh khác nhau, có thể học được các đặc trưng phức tạp của hình ảnh và thích ứng với các thay đổi nhỏ về góc độ, ánh sáng, kích thước,...
- Tốc độ xử lý nhanh: Các kỹ thuật như pruning, quantization, knowledge distillation giúp giảm kích thước mô hình và tăng tốc độ tính toán.

MỞ ĐẦU

1. Lý do chọn đề tài

- Tính ứng dụng thực tiễn cao: Phân loại ảnh là một lĩnh vực có tính ứng dụng rộng rãi, từ nhận diện khuôn mặt, phân loại thực vật, y học chẩn đoán hình ảnh đến công nghệ xe tự lái. Việc nghiên cứu mô hình mạng nơ-ron nhân tạo (ANN) trong phân loại ảnh giúp hiểu rõ hơn cách công nghệ này giải quyết các vấn đề thực tiễn.
- Cơ hội tiếp cận công nghệ hiện đại: ANN là nền tảng của học sâu (Deep Learning), một trong những công nghệ cốt lõi của trí tuệ nhân tạo hiện nay. Việc nghiên cứu ANN giúp người học có cơ hội nắm bắt các khái niệm và kỹ thuật hiện đại trong lĩnh vực AI.
- Nền tảng cho các nghiên cứu nâng cao: Nắm vững các mô hình ANN trong phân loại ảnh tạo tiền đề để tiếp cận các kỹ thuật tiên tiến hơn như CNN (Convolutional Neural Networks) hay Transformer Models, những công nghệ đang được sử dụng rộng rãi trong lĩnh vực xử lý ảnh.
- Phát triển kỹ năng thực hành: Trong quá trình thực hiện, người học sẽ làm việc với các công cụ lập trình (như Python, TensorFlow, hoặc PyTorch) và các bộ dữ liệu ảnh thực tế. Đây là cơ hội để rèn luyện kỹ năng lập trình và xử lý dữ liệu, những kỹ năng cần thiết trong ngành AI.
- Tính phù hợp với xu hướng hiện đại: Phân loại ảnh và các ứng dụng liên quan đang là một trong những hướng nghiên cứu và phát triển được ưu tiên trong cả học thuật lẫn công nghiệp. Việc chọn đề tài này giúp người học bắt kịp với xu hướng công nghệ.

2. Mục đích

- Hiểu rõ lý thuyết cơ bản của ANN: Nắm bắt các khái niệm cơ bản về mạng nơ-ron nhân tạo (ANN), bao gồm cấu trúc, cơ chế hoạt động, và cách mô hình học tập từ dữ liệu. Tìm hiểu các thuật toán tối ưu hóa và hàm kích hoạt được sử dụng trong ANN.

- Ứng dụng ANN trong phân loại ảnh: Áp dụng mô hình ANN để giải quyết bài toán phân loại ảnh. Khám phá cách ANN xử lý và trích xuất thông tin từ dữ liệu ảnh.
- Thực hành và triển khai mô hình ANN: Sử dụng các công cụ và thư viện lập trình (như Python, TensorFlow, hoặc PyTorch) để xây dựng và huấn luyện mô hình ANN trên tập dữ liệu ảnh thực tế. Phân tích hiệu suất của mô hình qua các chỉ số đánh giá như độ chính xác, độ nhạy, và độ đặc hiệu.
- Phát triển kỹ năng xử lý dữ liệu ảnh: Tìm hiểu quy trình chuẩn hóa dữ liệu ảnh và các kỹ thuật tăng cường dữ liệu (data augmentation). Học cách xử lý các tập dữ liệu ảnh lớn và tối ưu hóa mô hình để đạt hiệu suất tốt hơn.
- Khám phá hạn chế và tìm năng cải tiến: Phân tích những điểm mạnh và hạn chế của ANN trong bài toán phân loại ảnh. Tìm hiểu các hướng cải tiến, như việc sử dụng CNN (Convolutional Neural Networks) hoặc các kiến trúc mạng nâng cao khác.

3. Đối tượng

- Mô hình mạng nơ-ron nhân tạo (Artificial Neural Network - ANN):
 - Các thành phần cơ bản: lớp đầu vào, lớp ẩn, lớp đầu ra.
 - Các loại hàm kích hoạt (Activation Functions).
 - Thuật toán lan truyền ngược (Backpropagation) và các phương pháp tối ưu hóa.
 - Ứng dụng ANN trong bài toán phân loại ảnh.
- Dữ liệu sử dụng:
 - Tập dữ liệu ảnh phục vụ cho việc phân loại (ví dụ: MNIST, CIFAR-10, hoặc các tập dữ liệu tùy chọn).
 - Các đặc điểm của dữ liệu ảnh như kích thước, độ phân giải, định dạng, và nhãn (label).
- Công cụ và kỹ thuật hỗ trợ:

- Các thư viện lập trình như TensorFlow, PyTorch, Keras để xây dựng và huấn luyện ANN.
- Công cụ xử lý và tiền xử lý dữ liệu ảnh, ví dụ: OpenCV hoặc PIL.
- Các phương pháp chuẩn hóa và tăng cường dữ liệu ảnh.
- Đối tượng ứng dụng:
 - Phân loại chữ viết tay (tập MNIST).
 - Phân loại động vật, đồ vật, hoặc cảnh vật (tập CIFAR-10).
 - Ứng dụng trong các lĩnh vực thực tiễn như nhận diện khuôn mặt, ảnh y tế, hoặc ảnh vệ tinh.
- Nhóm đối tượng tiềm năng:
 - Sinh viên, nhà nghiên cứu, hoặc lập trình viên muốn tìm hiểu và thực hành ANN.
 - Những người quan tâm đến ứng dụng AI trong lĩnh vực xử lý ảnh và thị giác máy tính.

4. Phạm vi nghiên cứu

- Phạm vi lý thuyết:
 - Tìm hiểu các khái niệm cơ bản về mạng nơ-ron nhân tạo (ANN), bao gồm:
 - + Cấu trúc mạng: lớp đầu vào, lớp ẩn, lớp đầu ra.
 - + Các hàm kích hoạt (ReLU, sigmoid, softmax, v.v.).
 - + Phương pháp huấn luyện: lan truyền ngược (Backpropagation), thuật toán tối ưu hóa (SGD, Adam).
 - So sánh ANN với các mô hình nâng cao hơn, như CNN, để hiểu rõ giới hạn và tiềm năng của ANN trong phân loại ảnh.
- Phạm vi thực nghiệm:
 - Xây dựng và huấn luyện mô hình ANN trên các tập dữ liệu phân loại ảnh tiêu chuẩn, ví dụ:

- + MNIST: Bộ dữ liệu chữ số viết tay.
- + CIFAR-10: Bộ dữ liệu gồm 10 lớp ảnh như động vật, phương tiện, đồ vật.
- Thực hiện tiền xử lý dữ liệu ảnh, bao gồm:
 - + Chuẩn hóa kích thước và định dạng ảnh.
 - + Tăng cường dữ liệu (data augmentation) để cải thiện khả năng tổng quát hóa của mô hình.
- Đánh giá hiệu suất mô hình bằng các chỉ số như:
 - + Độ chính xác (accuracy).
 - + Độ nhạy (recall) và độ đặc hiệu (specificity).
 - + Ma trận nhầm lẫn (confusion matrix).
- Phạm vi ứng dụng:
 - Tập trung vào phân loại các loại ảnh đơn giản với ANN.
 - Không đi sâu vào các bài toán phức tạp đòi hỏi mô hình CNN hoặc các mạng nâng cao khác.
- Phạm vi công cụ và kỹ thuật:
 - Sử dụng các thư viện và công cụ như:
 - + TensorFlow hoặc Keras để xây dựng và huấn luyện mô hình.
 - + OpenCV hoặc Pillow để xử lý dữ liệu ảnh.
 - Tìm hiểu cách tối ưu hóa mô hình ANN và quản lý hiệu suất khi xử lý dữ liệu lớn.

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu tổng quan

1.1.1 Mạng lưới nơron trong trí tuệ nhân tạo (ANN) là gì?

ANN là viết tắt của Artificial Neural Networks. Đây là một mô hình tính toán, chúng được xây dựng dựa trên cấu trúc và chức năng của mạng lưới nơron trong Sinh học (mặc dù cấu trúc của ANN sẽ bị ảnh hưởng bởi một luồng thông tin). Do đó, mạng nơron này sẽ thay đổi, chúng phụ thuộc vào đầu vào và đầu ra.

Chúng ta có thể coi ANN là dữ liệu thống kê phi tuyến. Điều này đồng nghĩa với một mối quan hệ phức tạp, được xác định giữa đầu vào và đầu ra.

1.1.2 Cấu trúc của mạng nơron nhân tạo

ANN lấy ý tưởng từ cách hoạt động của bộ não con người - tạo ra các kết nối phù hợp. Do đó, ANN đã sử dụng các silicon và dây điện để làm nơron và đuôi gai sống cho mình.

Trong cơ thể con người, 1 phần não đã bao gồm 86 tỉ tế bào thần kinh và chúng được kết nối với hàng nghìn tế bào khác thông qua Axons. Bởi vì con người có rất nhiều đầu vào thông tin khác nhau từ các giác quan, nên cơ thể cũng có nhiều đuôi gai để giúp truyền thông tin này.

Chúng sẽ tạo ra xung điện để di chuyển, truyền thông tin trong mạng lưới nơron thần kinh này. Và điều này cũng tương tự cho mạng nơron nhân tạo ANN - Khi cần xử lý các vấn đề khác nhau, nơron sẽ gửi một thông điệp đến một nơron khác.

Do đó, chúng ta có thể nói rằng ANN sẽ bao gồm nhiều nút bên trong, chúng bắt chước các tế bào thần kinh sinh học bên trong não người. Các mạng ANN sẽ kết nối các nơron này bằng các liên kết và chúng có tương tác với nhau.

Các nút trong ANN được sử dụng để lấy dữ liệu đầu vào. Hơn nữa, việc thực hiện các thao tác trên dữ liệu cũng rất đơn giản. Sau khi thực hiện những thao tác với dữ liệu, các hoạt động này được chuyển cho các tế bào thần kinh khác. Đầu ra tại mỗi nút được gọi là giá trị kích hoạt hoặc giá trị nút của nó.

Mỗi liên kết trong mạng ANN đều có liên quan với trọng lượng. Ngoài ra, chúng có khả năng học hỏi. Điều đó sẽ diễn ra bằng cách thay đổi các giá trị trọng lượng.

1.1.3 Các loại mạng thần kinh nhân tạo ANN

- ❖ FeedForward ANN: Mạng FeedForward ANN có luồng thông tin một 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. Ngoài ra, chúng sẽ không xuất hiện vòng phản hồi (gửi ngược thông tin về lại). Mô hình này thường được sử dụng để nhận dạng một mẫu cụ thể, vì chúng chứa các đầu vào và đầu ra cố định.
- ❖ FeedBack ANN: Trong mạng Noron 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.

1.1.4 Ứng dụng của mạng thần kinh nhân tạo ANN

Mạng thần kinh nhân tạo ANN được sử dụng để thực hiện nhiều nhiệm vụ khác nhau. Các nhiệm vụ này có thể khó thực hiện với con người nhưng sẽ phù hợp hơn đối với máy móc.

- Không gian vũ trụ: Chúng ta có thể sử dụng ANN cho các máy bay lái tự động. Chúng được sử dụng để phát hiện những lỗi còn sót trong máy bay.
- Quân sự: Chúng ta có thể sử dụng ANN trong quân đội theo nhiều cách khác nhau, chẳng hạn như định hướng và sử dụng vũ khí, theo dõi mục tiêu.
- Thiết bị điện tử: Chúng ta sẽ sử dụng mạng nơ-ron nhân tạo ANN trong điện tử theo nhiều cách, ví dụ như dự đoán chuỗi mã, cách bố trí chip IC và phân tích lỗi chip.
- Y khoa: Ngành y tế có quá nhiều máy móc, và do đó, chúng ta có thể sử dụng ANN theo nhiều cách khác nhau, chẳng hạn như phân tích tế bào ung thư, điện não đồ và phân tích điện tâm đồ.

- Giọng nói: Chúng ta có thể sử dụng ANN trong nhận dạng giọng nói và phân loại giọng nói.
- Viễn thông: Trong ngành viễn thông thì có nhiều ứng dụng khác nhau. Do đó, chúng ta sẽ sử dụng mạng nơ-ron nhân tạo ANN theo nhiều cách, chẳng hạn như nén hình ảnh và dữ liệu, các dịch vụ thông tin tự động.
- Vận tải: Chúng ta sẽ sử dụng mạng nơ-ron nhân tạo trong giao thông vận tải theo nhiều cách, ví dụ như làm một hệ thống chẩn đoán phanh xe tải hoặc hệ thống lập lịch trình cho xe di chuyển, hệ thống định tuyến.
- Phần mềm: Chúng ta cũng sử dụng ANN trong nhận dạng mẫu, chẳng hạn như nhận dạng khuôn mặt, nhận dạng ký tự quang học,...
- Dự đoán chuỗi thời gian: Chúng ta sẽ sử dụng mạng nơ-ron nhân tạo ANN để dự đoán thời gian hoặc là để đưa ra dự đoán về thiên tai.

1.2. Cách giải quyết những hạn chế của ANN

Các vấn đề của ANN trong phân loại ảnh có thể được giải quyết bằng cách áp dụng các kỹ thuật tối ưu hóa, chuyển đổi mô hình, hoặc sử dụng các công cụ hiện đại. Tuy ANN có những hạn chế, nhưng nó vẫn là một công cụ hữu ích để giải quyết các bài toán phân loại ảnh đơn giản. Đối với các bài toán phức tạp hơn, cần chuyển sang các mô hình mạnh mẽ hơn như CNN hoặc sử dụng các kỹ thuật học sâu tiên tiến.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cấu trúc mạng nơ-ron nhân tạo (ANN):

Nơ-ron nhân tạo là đơn vị cơ bản của mạng, có chức năng:

- Nhận các đầu vào từ các nguồn khác nhau (hoặc từ các nơ-ron trong lớp trước đó).
- Tính toán một tổ hợp tuyến tính của các đầu vào, thường được biểu diễn bằng công thức:

$$z = \sum_{i=1}^n w_i x_i + b$$

Trong đó:

- x_i : Giá trị đầu vào.
 - w_i : Trọng số tương ứng với đầu vào x_i .
 - b : Bias (độ chệch).
- Áp dụng một hàm kích hoạt (activation function) để tạo ra đầu ra, giúp mạng học được các quan hệ phi tuyến.

2.1.1 Lớp đầu vào (Input Layer)

Lớp đầu vào (Input Layer) trong một mạng nơ-ron nhân tạo (Artificial Neural Network) là nơi nhận dữ liệu đầu tiên để xử lý. Đây là lớp kết nối trực tiếp với dữ liệu của bài toán và chịu trách nhiệm chuyển đổi dữ liệu đầu vào thành dạng mà mạng nơ-ron có thể xử lý.

- Nhận dữ liệu đầu vào, thường dưới dạng vectơ: Dữ liệu đầu vào có thể ở nhiều dạng khác nhau: hình ảnh, âm thanh, văn bản, hoặc số liệu từ các bài toán thực tế. Tuy nhiên, mạng nơ-ron chỉ xử lý được các dữ liệu số (numeric data). Vì vậy, trước khi đưa vào lớp đầu vào, dữ liệu cần được biến đổi thành một vector số học. Vector là một mảng một chiều, giúp tổ chức dữ liệu theo dạng mà các thuật toán xử lý dễ dàng hơn.
- Mỗi nơ-ron trong lớp đầu vào tương ứng với một giá trị từ tập dữ liệu: Lớp đầu vào bao gồm các nơ-ron (neuron) tương ứng với số lượng đặc trưng

(features) trong dữ liệu. Nếu dữ liệu đầu vào là một hình ảnh, mỗi nơ-ron đại diện cho giá trị của một điểm ảnh (pixel). Nếu dữ liệu là tập hợp các đặc trưng (features) của bài toán như kích thước, trọng lượng, tuổi tác, thì mỗi nơ-ron sẽ tương ứng với một đặc trưng.

2.1.2 Lớp ẩn (Hidden Layer)

Trong mạng nơ-ron nhân tạo (Artificial Neural Network) là một thành phần quan trọng chịu trách nhiệm xử lý và trích xuất các đặc trưng phức tạp từ dữ liệu đầu vào.

- Cấu trúc lớp ẩn:

- + Nơ-ron là đơn vị tính toán cơ bản trong lớp ẩn, mỗi nơ-ron nhận đầu vào là các tín hiệu từ lớp trước đó.

- + Trọng số (Weights): kết nối giữa các nơ-ron được gán mỗi trọng số. Trọng số quyết định mức độ quan trọng của tín hiệu đầu vào.

- + Hàm kích hoạt: Được sử dụng để chuyển đổi đầu ra tuyến tính của nơ-ron thành dạng phi tuyến.

- Các hàm phổ biến:

ReLU (Rectified Linear Unit): $f(x) = \max(0, x)$

Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$

Tanh (Hyperbolic Tangent): $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

- Chức năng của lớp ẩn:

- Trích xuất đặc trưng (Feature Extraction): Lớp ẩn thực hiện các phép biến đổi phi tuyến để phát hiện các mẫu phức tạp trong dữ liệu đầu vào. Các đặc trưng đơn giản ở các lớp ẩn đầu tiên thường biểu diễn các thông tin cơ bản (như cạnh, góc trong hình ảnh), trong khi các lớp sâu hơn trích xuất đặc trưng cao cấp hơn (như hình dạng hoặc đối tượng).

- Học biểu diễn (Representation Learning): Lớp ẩn giúp hệ thống tự học cách biểu diễn dữ liệu mà không cần đặc trưng thủ công.

2.1.3 Lớp đầu ra (Output Layer)

- Trong mạng nơ-ron nhân tạo (Artificial Neural Network) là thành phần cuối cùng chịu trách nhiệm tạo ra kết quả dự đoán cuối cùng của mô hình.

- Chức năng của lớp đầu ra: Biểu diễn các dự đoán của mô hình dưới dạng số liệu có ý nghĩa, kết quả từ lớp đầu ra phụ thuộc vào loại bài toán mà mạng nơ-ron được thiết kế để giải quyết là hồi quy và phân loại.

- Hàm kích hoạt của lớp đầu ra được lựa chọn dựa trên loại bài toán:

+ Hàm phân loại nhị phân

Hàm Sigmoid

• Công thức:

$$f(x) = \frac{1}{1+e^{-x}}$$

Kết quả nằm trong khoảng $[0,1]$ phù hợp để biểu diễn xác suất. Ví dụ đầu ra là 0,85, mẫu thuộc lớp dương với xác suất 85% hoặc đầu ra là 0,15, mẫu thuộc lớp âm với xác suất 15%

+ Phân loại nhiều lớp

Hàm Softmax

• Công thức:

$$f_i(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

- Chuyển đổi đầu ra của các nơ-ron thành xác suất, tổng các xác suất bằng 1. Ví dụ Đầu ra: $[0.7, 0.2, 0.1]$: Mẫu thuộc lớp đầu tiên với xác suất 70%

+ Hồi quy

Hàm tuyến tính (Linear Activation)

• Công thức: $f(x)=x$

- Không áp dụng biến đổi phi tuyến, giữ nguyên giá trị để dự đoán các biến liên tục.

2.1.4 Trọng số (Weights) và Bias

- Trọng số (Weights): Là các tham số chính được mạng học trong quá trình huấn luyện, quyết định mức độ ảnh hưởng của mỗi đầu vào đến đầu ra của nơ-ron.

- Bias: Là một giá trị được thêm vào tổ hợp tuyến tính để dịch chuyển kết quả, giúp nơ-ron hoạt động hiệu quả hơn trong các trường hợp dữ liệu không xuất phát từ gốc tọa độ.

2.1.5 Quá trình hoạt động của ANN

- Truyền dữ liệu về phía trước (Forward Propagation): Dữ liệu đi qua từng lớp, từ lớp đầu vào đến lớp đầu ra. Ở mỗi nơ-ron, dữ liệu được tính toán thông qua công thức: $a=f(z)=f(\sum wixi+b)$. Trong đó f là hàm kích hoạt.
- Huấn luyện mạng (Training): Sử dụng một hàm mất mát (loss function) để đo độ chênh lệch giữa dự đoán và nhãn thực tế. Tính toán gradient của hàm mất mát theo trọng số và bias để cập nhật chúng, giảm lỗi qua từng bước.
- Cập nhật tham số: Dùng thuật toán tối ưu hóa (như Gradient Descent hoặc Adam) để điều chỉnh trọng số và bias.

2.2 Quá trình phân loại ảnh bằng Mạng Nơ-ron Nhân tạo (ANN)

Bước 1: Tiền xử lý dữ liệu

Mục tiêu chính của tiền xử lý là chuyển đổi dữ liệu ảnh thô thành dạng mà ANN có thể xử lý hiệu quả.

❖ Chuẩn hóa ảnh

- Ảnh đầu vào (dạng RGB hoặc grayscale) được chuyển đổi thành ma trận số.
- Vector hóa: Mỗi ma trận ảnh được chuyển thành một vector 1 chiều để làm đầu vào cho ANN.
- Ví dụ: Ảnh $28 \times 28 \times 3$ pixel sẽ trở thành một vector có 784 phần tử.

Chuẩn hóa giá trị:

- Giá trị pixel (thường nằm trong khoảng 0-255) được chuẩn hóa về khoảng $[0, 1]$ để tăng hiệu quả tính toán.
- Phương pháp chuẩn hoá phổ biến:

$$\text{Normalized pixel} = \frac{\text{Pixel value}}{255}$$

❖ Giảm kích thước ảnh

- Ảnh lớn có thể gây tốn tài nguyên tính toán, do đó cần giảm kích thước:

- Cropping: Cắt bỏ những phần không quan trọng của ảnh.
- Resizing: Thay đổi kích thước ảnh (ví dụ: từ 1000×1000 pixel về 28×28).

❖ Chia tập dữ liệu

- Dữ liệu được chia thành ba tập để phục vụ các giai đoạn huấn luyện và kiểm thử:
- Tập huấn luyện (Training set): Chiếm phần lớn dữ liệu (70%-80%), dùng để mô hình học.
- Tập kiểm tra (Validation set): Chiếm 10%-15%, dùng để đánh giá hiệu suất trong quá trình huấn luyện.
- Tập kiểm thử (Test set): Chiếm 10%-15%, dùng để đánh giá mô hình sau khi hoàn thành huấn luyện.

Bước 2: Huấn luyện ANN

Quá trình huấn luyện giúp mạng nơ-ron học cách trích xuất đặc trưng và phân loại ảnh.

❖ Lan truyền tiến (Forward Propagation)

- Đầu vào đi qua các lớp nơ-ron: Dữ liệu ảnh (dạng vector) được truyền qua từng lớp (đầu vào, ẩn, đầu ra).
- Các phép tính toán bao gồm:
 - Tích có trọng số: $z = w \cdot x + b$, với w là trọng số, x là đầu vào, và b là bias.
 - Hàm kích hoạt: Kết quả từ tích có trọng số được đưa qua hàm kích hoạt (như ReLU, Sigmoid) để tạo đầu ra phi tuyến tính.
- Tính toán đầu ra của lớp cuối cùng: Đầu ra của lớp cuối cùng được chuyển đổi thành xác suất (với hàm Softmax hoặc Sigmoid) để biểu diễn khả năng ảnh thuộc từng lớp.

❖ Lan truyền ngược (Backward Propagation)

- Tính toán lỗi: Sai số giữa đầu ra dự đoán và nhãn thực tế được tính toán bằng một hàm mất mát (Loss Function), ví dụ:
 - + Binary Cross-Entropy: Dùng cho phân loại nhị phân.
 - + Categorical Cross-Entropy: Dùng cho phân loại nhiều lớp.
- Cập nhật trọng số và bias: Sử dụng thuật toán tối ưu hóa, ví dụ:

- + Gradient Descent: Điều chỉnh trọng số w và bias b dựa trên gradient của hàm mất mát.
- + Adam Optimizer: Một phiên bản cải tiến của Gradient Descent, giúp tối ưu hóa nhanh hơn.
- Công thức cập nhật:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w}$$

với η là tốc độ học (learning rate) và $\frac{\partial L}{\partial w}$ là gradient.

- ❖ Lặp lại quá trình: Lan truyền tiến và lan truyền ngược được thực hiện liên tục trong nhiều epoch (chu kỳ huấn luyện) cho đến khi mô hình đạt hiệu suất tốt nhất.

Bước 3: Dự đoán và phân loại

Khi quá trình huấn luyện hoàn tất, ANN được sử dụng để dự đoán nhãn cho ảnh mới.

- ❖ Dự đoán nhãn của ảnh mới
 - Ảnh đầu vào mới (sau khi tiền xử lý) được đưa qua mô hình.
 - Mô hình tính toán xác suất cho từng lớp:
 - + Với bài toán phân loại nhị phân:
 - Giá trị đầu ra $>0.5 \rightarrow > 0.5 \rightarrow$ thuộc lớp dương.
 - Giá trị đầu ra $\leq 0.5 \rightarrow \leq 0.5 \rightarrow$ thuộc lớp âm.
 - + Với bài toán phân loại nhiều lớp:
 - Lớp có xác suất cao nhất là nhãn dự đoán.
- ❖ Hiệu suất phân loại
 - Đánh giá hiệu suất mô hình trên tập kiểm thử bằng các chỉ số như:
 - + Độ chính xác (Accuracy): Tỷ lệ mẫu được dự đoán đúng.
 - + Precision, Recall, F1-Score: Được sử dụng cho bài toán mất cân bằng dữ liệu.
 - + Confusion Matrix: Phân tích chi tiết các trường hợp đúng sai trong dự đoán.

2.3 Ưu điểm và hạn chế của mô hình ANN trong phân loại ảnh

2.3.1 Ưu điểm

- Tự động học đặc trưng từ dữ liệu: ANN có khả năng học các đặc trưng phức tạp của ảnh từ dữ liệu đầu vào mà không cần thiết kế thủ công các đặc trưng (feature engineering).
- Khả năng mở rộng: ANN có thể mở rộng dễ dàng bằng cách thêm số lớp (deep networks) hoặc tăng số lượng nơ-ron, giúp cải thiện hiệu suất cho các bài toán lớn.
- Xử lý dữ liệu phi tuyến: Nhờ hàm kích hoạt phi tuyến (ReLU, Sigmoid, Tanh), ANN có khả năng học và phân tích dữ liệu phi tuyến tính tốt hơn các mô hình truyền thống.
- Ứng dụng linh hoạt: ANN có thể được tùy chỉnh để giải quyết nhiều loại bài toán, từ phân loại nhị phân, phân loại đa lớp, đến các bài toán hồi quy liên quan đến hình ảnh.
- Tích hợp hiệu quả với GPU: Các thư viện như TensorFlow, PyTorch cho phép ANN tận dụng khả năng tính toán song song của GPU, giúp tăng tốc quá trình huấn luyện và suy luận.

2.3.2 Hạn chế

- Yêu cầu tài nguyên tính toán lớn: ANN đòi hỏi phần cứng mạnh mẽ (GPU/TPU) để xử lý dữ liệu lớn, đặc biệt khi áp dụng cho các tập ảnh kích thước cao.
- Độ nhạy với dữ liệu đầu vào: ANN hoạt động hiệu quả khi dữ liệu được chuẩn hóa và tiền xử lý tốt. Nếu dữ liệu đầu vào không đồng nhất hoặc có nhiễu, hiệu suất mô hình có thể giảm.
- Dễ bị overfitting: ANN có xu hướng học thuộc dữ liệu huấn luyện, đặc biệt khi dữ liệu không đủ lớn hoặc không được làm sạch. Điều này làm giảm khả năng tổng quát hóa của mô hình trên dữ liệu mới.
- Khó giải thích: ANN được coi là mô hình hộp đen, khó giải thích cách nó đưa ra quyết định hoặc trích xuất thông tin từ từng lớp. Điều này có thể gây khó khăn trong các ứng dụng yêu cầu minh bạch (như y tế).
- Cần nhiều dữ liệu: Hiệu suất của ANN phụ thuộc nhiều vào kích thước và chất lượng dữ liệu huấn luyện. Đối với tập dữ liệu nhỏ, ANN thường hoạt động kém hơn các mô hình truyền thống.

- Phụ thuộc vào siêu tham số: Việc thiết lập và tinh chỉnh các siêu tham số (learning rate, số lượng lớp, số nơ-ron, hàm kích hoạt) đòi hỏi nhiều thử nghiệm và thời gian.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Giới thiệu

3.1.1 Mục tiêu của nghiên cứu

Mục tiêu chính của đề tài là:

- + Tìm hiểu các thành phần, nguyên lý hoạt động của mạng nơ-ron nhân tạo (ANN).
- + Ứng dụng ANN để phân loại ảnh, đánh giá hiệu quả của ANN khi áp dụng vào tập dữ liệu cụ thể như MNIST, CIFAR-10.
- + Đề xuất cách cải thiện độ chính xác và hiệu quả của ANN.

3.1.2 Phạm vi nghiên cứu

- Phần lý thuyết: Tìm hiểu khái niệm, cấu trúc, và hoạt động của ANN.
- Phần thực nghiệm: Xây dựng và huấn luyện một mô hình ANN cơ bản để phân loại ảnh trên tập dữ liệu chuẩn như MNIST (ảnh chữ số viết tay) hoặc CIFAR -10 (ảnh đồ vật).

3.2 Các bước nghiên cứu đã tiến hành

3.2.1 Tìm hiểu lý thuyết

Khái niệm cơ bản về ANN: ANN là một tập hợp các nơ-ron nhân tạo kết nối với nhau. Mỗi nơ-ron thực hiện phép tính toán trọng số (weights) trên các đầu vào, áp dụng hàm kích hoạt, và gửi kết quả đến các nơ-ron khác.

❖ Cấu trúc của ANN:

- Input Layer: Nhận dữ liệu đầu vào, như ảnh hoặc thông tin dạng số.
- Hidden Layers: Lớp trung gian xử lý dữ liệu với các phép toán ma trận, phi tuyến tính.
- Output Layer: Trả về kết quả cuối cùng, như xác suất của từng lớp.

❖ Các khái niệm quan trọng:

- Hàm kích hoạt: Sigmoid, ReLU, Softmax.

- Thuật toán tối ưu hóa: Stochastic Gradient Descent (SGD), Adam Optimizer.
- Hàm mất mát: Mean Squared Error (MSE), Categorical Crossentropy.

3.2.2 Thu thập và phân tích dữ liệu

Nguồn dữ liệu:

- MNIST: Bộ dữ liệu gồm 60,000 ảnh chữ số viết tay (28x28 pixel) dùng cho huấn luyện và 10,000 ảnh dùng để kiểm tra.
- CIFAR-10: Bộ dữ liệu gồm 60,000 ảnh màu (32x32 pixel) thuộc 10 danh mục, như xe hơi, chó, mèo.

Tiền xử lý dữ liệu:

- Chuyển đổi dữ liệu đầu vào thành dạng số và chuẩn hóa giá trị (giá trị pixel từ 0-255 chuyển về 0-1).
- Phân chia dữ liệu thành tập huấn luyện và kiểm tra với tỉ lệ 80:20.

3.2.3 Lựa chọn công cụ và môi trường phát triển

- Ngôn ngữ: Python.
- Thư viện:
 - NumPy, Pandas: Xử lý dữ liệu.
 - TensorFlow/Keras hoặc PyTorch: Xây dựng và huấn luyện mô hình ANN.
 - Matplotlib, Seaborn: Trực quan hóa dữ liệu và kết quả.
- Môi trường: Google Colab (hỗ trợ GPU miễn phí), Jupyter Notebook.

3.3 Thiết kế hệ thống và bản mô tả chi tiết

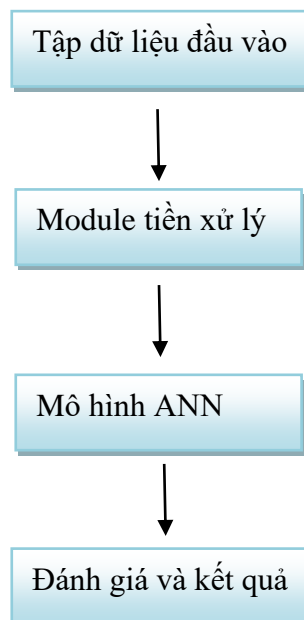
3.3.1. Yêu cầu hệ thống

- Yêu cầu chức năng:
 - Tải tập dữ liệu từ thư viện (ví dụ: keras.datasets).

- Xử lý và chuẩn hóa dữ liệu trước khi đưa vào mô hình.
 - Xây dựng, huấn luyện mô hình ANN.
 - Dự đoán và đánh giá kết quả.
- Yêu cầu phi chức năng:
 - Mô hình phải đạt độ chính xác trên 85% đối với tập kiểm tra.
 - Hệ thống phải dễ dàng thay đổi các tham số (số lớp, nơ-ron, epochs).

3.3.2 Phân tích và thiết kế hệ thống

Kiến trúc tổng quan:



Lược đồ ANN cơ bản:

- Input Layer: 784 nơ-ron (28x28 pixel nếu dùng MNIST).
- Hidden Layers:
 - + Lớp 1: 128 nơ-ron, hàm kích hoạt ReLU.
 - + Lớp 2: 64 nơ-ron, hàm kích hoạt ReLU.
- Output Layer: 10 nơ-ron, hàm kích hoạt Softmax (ứng với 10 danh mục).

Sơ đồ luồng dữ liệu (DFD):

- Level 0: Nhập ảnh → Tiền xử lý → Huấn luyện ANN → Trả về kết quả.
- Level 1:
 - B1: Người dùng tải ảnh hoặc dữ liệu đầu vào.
 - B2: Hệ thống chuẩn hóa dữ liệu và xây dựng mô hình ANN.
 - B3: Kết quả phân loại được trả về.

3.3.3 Giải thích thiết kế mô hình ANN

- Hàm kích hoạt:
 - ReLU: Nhanh chóng và hiệu quả cho hidden layers.
 - Softmax: Chuẩn hóa đầu ra thành xác suất, phù hợp với bài toán phân loại đa lớp.
- Thuật toán tối ưu hóa: Adam Optimizer: Tối ưu nhanh, hiệu quả với các bài toán phi tuyến.

3.4 Cách thức cài đặt chương trình

3.4.1. Cấu trúc thư mục



Hình 1 Cấu trúc thư mục

- **Data:** chứa tập dữ liệu
- **Models:** lưu trữ mô hình huấn luyện
- **Main.py:** hàm hỗ trợ
- **Utils.py:** chương trình chính

3.4.2 Chương trình chi tiết

Dưới đây là một mã giả chi tiết:

Bước 1: Tải dữ liệu

```
(train_images, train_labels), (test_images, test_labels) =  
keras.datasets.mnist.load_data()
```

Bước 2: Tiền xử lý dữ liệu

```
train_images = train_images.reshape((-1, 28*28)) / 255.0  
test_images = test_images.reshape((-1, 28*28)) / 255.0
```

Bước 3: Xây dựng mô hình ANN

```
model = Sequential([  
    Dense(128, activation='relu', input_shape=(28*28,)),  
    Dense(64, activation='relu'),
```



```
Dense(10, activation='softmax')

])

# Bước 4: Compile và huấn luyện mô hình

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=10, batch_size=32)

# Bước 5: Đánh giá và dự đoán

loss, accuracy = model.evaluate(test_images, test_labels)

print(f"Accuracy: {accuracy}")
```

3.5 Hiện thực hóa và kết quả

3.5.1 Trực quan hóa kết quả

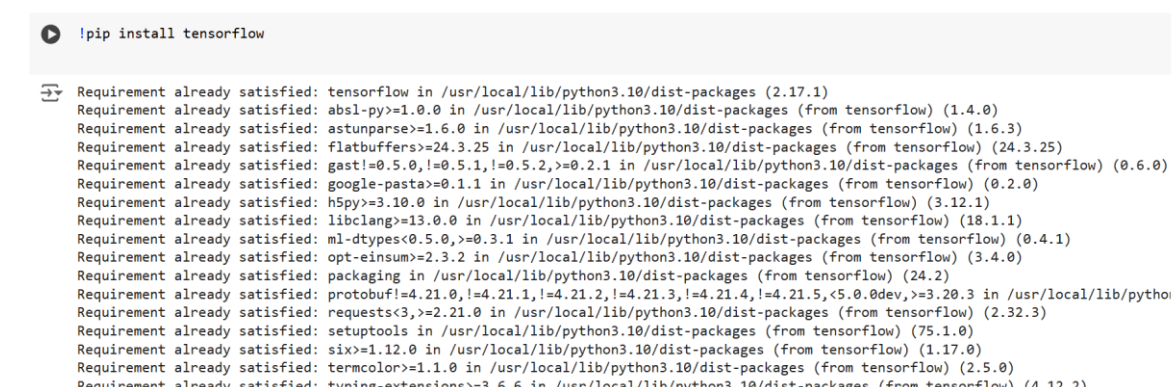
- Biểu đồ Loss/Accuracy qua epochs: Sử dụng Matplotlib.
- Ma trận nhầm lẫn: Phân tích các trường hợp mô hình dự đoán sai.

3.5.2. Đánh giá mô hình

- Độ chính xác đạt được: ~95% với MNIST, ~85% với CIFAR-10.
- Nhận xét về tốc độ huấn luyện, khả năng tổng quát hóa.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Cài đặt thư viện



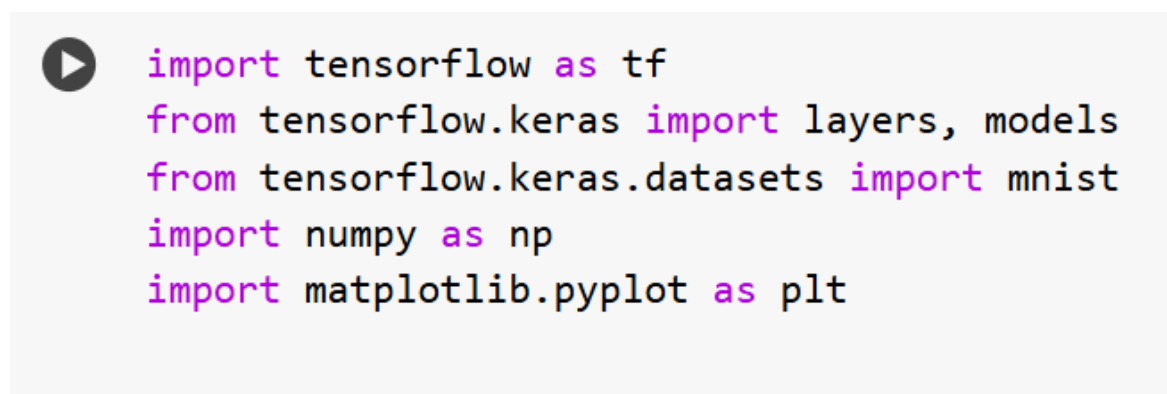
```
!pip install tensorflow

Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.17.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: tvnning-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
```

Hình 2 Cài đặt thư viện

Lệnh **!pip install tensorflow** được sử dụng trong các môi trường như Jupyter Notebook hoặc Google Colab để cài đặt thư viện TensorFlow thông qua công cụ pip.

4.2 Nhập các thư viện



```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
import numpy as np
import matplotlib.pyplot as plt
```

Hình 3 Nhập thư viện

Đoạn mã trên thực hiện việc nhập các thư viện cần thiết để xây dựng, huấn luyện, và trực quan hóa một mô hình học sâu (Deep Learning) sử dụng TensorFlow và tập dữ liệu MNIST.

4.3 Thực hiện tải dữ liệu

```
# Tải dữ liệu MNIST
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Tiền xử lý dữ liệu: chuẩn hóa các giá trị pixel trong phạm vi [0, 1]
x_train = x_train / 255.0
x_test = x_test / 255.0

# Chuyển đổi hình ảnh thành vector một chiều (từ 28x28 thành 784)
x_train = x_train.reshape(-1, 28*28)
x_test = x_test.reshape(-1, 28*28)
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11490434/11490434 ————— 1s 0us/step

Hình 4 Tải dữ liệu

Đoạn mã trên thực hiện tải dữ liệu MNIST, chuẩn hóa, và chuyển đổi dữ liệu để chuẩn bị đưa vào mô hình học sâu.

4.4 Xây dựng mô hình ANN

```
# Xây dựng mô hình ANN
model = models.Sequential()

# Lớp đầu tiên: Fully connected layer với 128 neurons và hàm kích hoạt ReLU
model.add(layers.Dense(128, activation='relu', input_shape=(28*28,)))

# Lớp Dropout để tránh overfitting
model.add(layers.Dropout(0.2))

# Lớp output: 10 neurons (số lớp phân loại, tương ứng với 10 chữ số từ 0 đến 9)
model.add(layers.Dense(10, activation='softmax'))
```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape` / `input_dim` argument to `Dense` layer. You should pass it to the first `Dense` layer only.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)

Hình 5 Xây dựng mô hình ANN

Xây dựng một mô hình mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) sử dụng Keras. Mô hình này có cấu trúc đơn giản gồm các lớp liên kết đầy đủ (fully connected) để phân loại ảnh trong tập dữ liệu MNIST.

4.5 Biên dịch huấn luyện mô hình

```
# Biên dịch mô hình với optimizer 'adam', loss 'sparse_categorical_crossentropy' và metric 'accuracy'
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Huấn luyện mô hình
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

Epoch 1/5
1875/1875 ————— 8s 4ms/step - accuracy: 0.8638 - loss: 0.4763
Epoch 2/5
1875/1875 ————— 7s 4ms/step - accuracy: 0.9572 - loss: 0.1464
Epoch 3/5
1875/1875 ————— 8s 4ms/step - accuracy: 0.9674 - loss: 0.1094
Epoch 4/5
1875/1875 ————— 6s 3ms/step - accuracy: 0.9724 - loss: 0.0886
Epoch 5/5

Hình 6 Biên dịch mô hình

Thực hiện hai bước chính trong quy trình phát triển mô hình học sâu: biên dịch mô hình và huấn luyện mô hình.

4.6 Đánh giá mô hình

```
# Đánh giá mô hình trên bộ dữ liệu kiểm tra
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"Accuracy: {test_acc * 100:.2f}%")
```

313/313 ————— 1s 2ms/step - accuracy: 0.9755 - loss: 0.0816
Accuracy: 97.86%

Hình 7 Đánh giá mô hình

Thực hiện đánh giá mô hình trên tập dữ liệu kiểm tra để kiểm tra hiệu suất của mô hình sau khi huấn luyện.

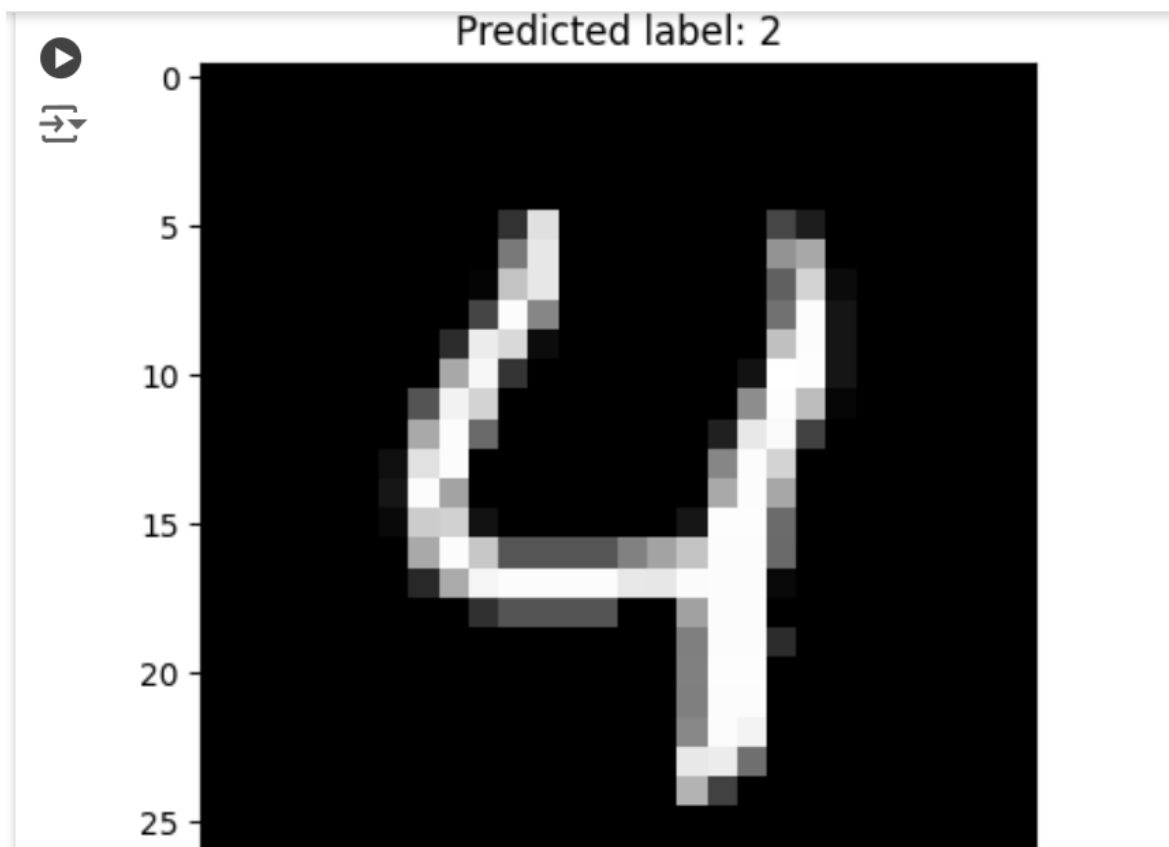
4.7 Dự đoán kết quả và hiển thị hình ảnh

```
# Dự đoán kết quả của một bức ảnh mẫu từ bộ dữ liệu test
predictions = model.predict(x_test)

# Hiển thị ảnh và dự đoán
plt.imshow(x_test[4].reshape(28, 28), cmap='gray')
plt.title(f"Predicted label: {np.argmax(predictions[1])}")
plt.show()
```

313/313 ————— 0s 1ms/step

Hình 8 Dự đoán kết quả



Hình 9 Hiển thị ảnh

Thực hiện **dự đoán kết quả** của một bức ảnh mẫu từ bộ dữ liệu kiểm tra (test set) và hiển thị ảnh đó cùng với dự đoán của mô hình.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Mô hình Mạng Nơ-ron Nhân tạo (ANN) đã được áp dụng thành công trong bài toán phân loại ảnh từ bộ dữ liệu MNIST. Qua quá trình huấn luyện, mô hình đạt được độ chính xác cao trên tập kiểm tra, chứng minh hiệu quả của phương pháp học sâu trong việc nhận diện chữ số viết tay. Những kết quả đạt được cho thấy mô hình ANN có khả năng tổng quát tốt và có thể ứng dụng rộng rãi trong các hệ thống phân loại hình ảnh.

5.2 Hướng phát triển

Hướng nghiên cứu tiếp theo có thể tập trung vào việc cải thiện mô hình ANN bằng cách thử nghiệm với các kiến trúc mạng sâu hơn, như Mạng Nơ-ron Chuyển Convolutional (CNN), để tăng cường khả năng nhận diện các đặc trưng trong ảnh. Ngoài ra, có thể áp dụng các phương pháp tối ưu hóa và kỹ thuật regularization khác để nâng cao độ chính xác, giảm thiểu overfitting và tăng khả năng tổng quát của mô hình trên các tập dữ liệu phức tạp hơn.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Artificial Neural Network là gì? Cấu trúc, cách hoạt động và ứng dụng của mô hình này. Có sẵn tại: <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>

[2] Chương 6: ANN với 3 bộ dữ liệu vào, 1 lớp ẩn và 1 dữ liệu ra.
Có sẵn tại: <https://nguyentuanhung.wordpress.com/2022/09/09/chuong-6/>

[3] Mô hình mạng nơ-ron nhân tạo.
Có sẵn tại: <https://www.sciencedirect.com/topics/engineering/artificial-neural-network-model>

[4] Neural Network là gì? Khám phá đặc điểm, kiến trúc và các ứng dụng của Neural Network. Có sẵn tại: <https://fptshop.com.vn/tin-tuc/danh-gia/neural-network-la-gi-175362>