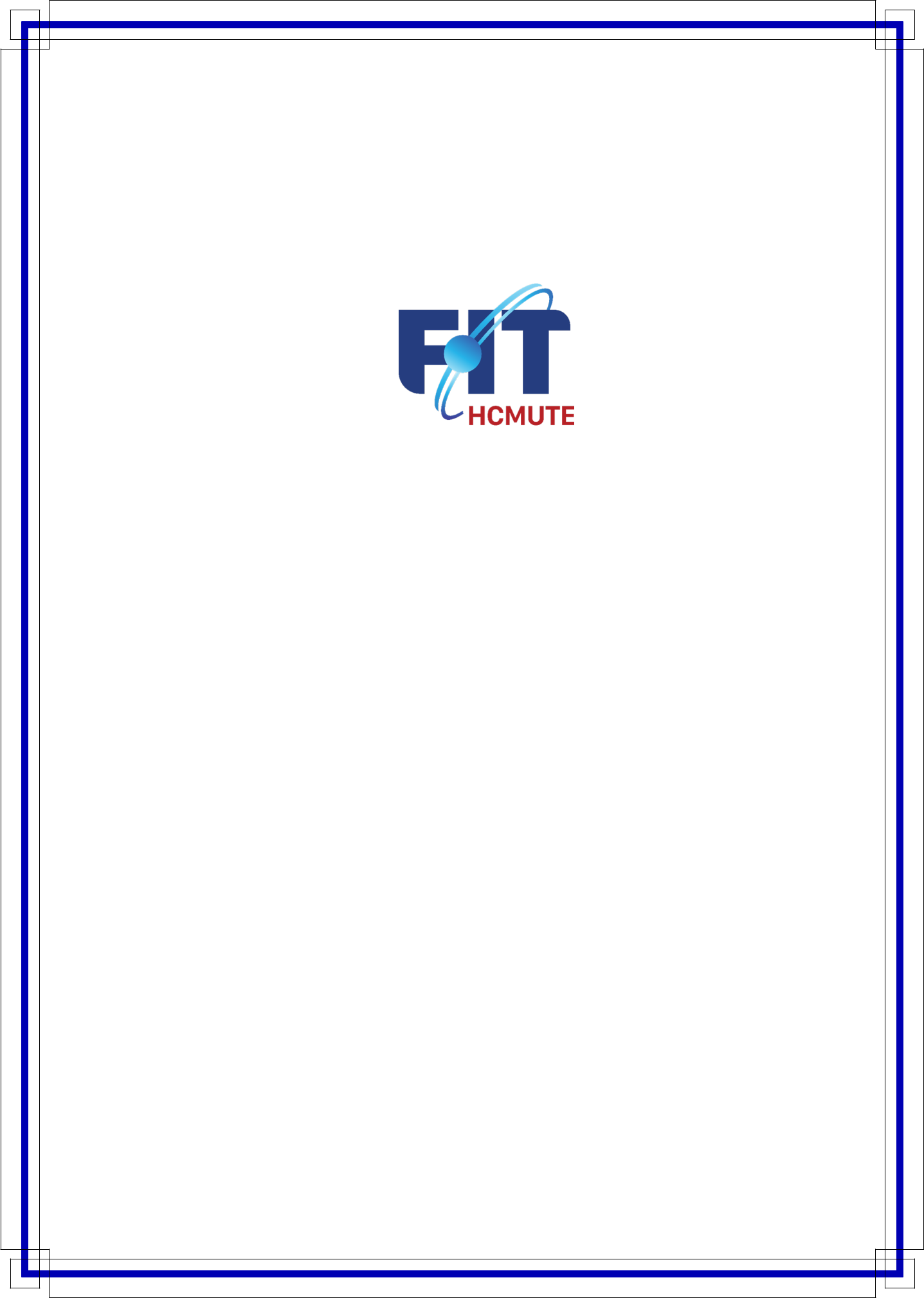
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**

------------------

**NGUYỄN MINH KHOA – 19133029**

Đề tài:

**TÌM HIỂU KNOWLEDGE DISTILLATION**

**VÀ ỨNG DỤNG**

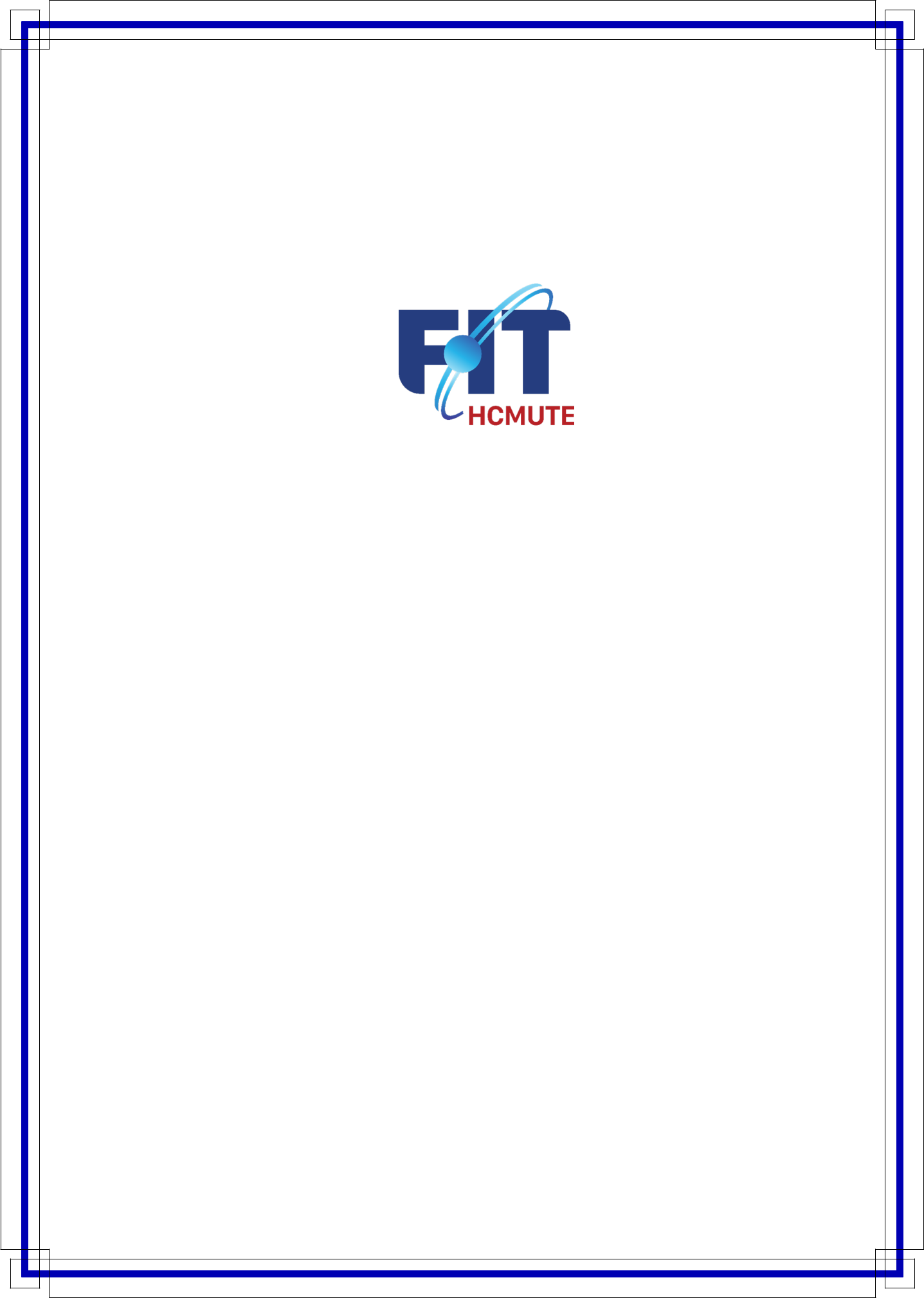
**KHOÁ LUẬN TỐT NGHIỆP**

**KỸ SƯ NGÀNH KỸ THUẬT DỮ LIỆU**

**GIÁO VIÊN HƯỚNG DẪN**

**ThS. QUÁCH ĐÌNH HOÀNG**

**KHOÁ 2019-2023**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**

------------------

**NGUYỄN MINH KHOA – 19133029**

Đề tài:

**TÌM HIỂU KNOWLEDGE DISTILLATION**

**VÀ ỨNG DỤNG**

**KHOÁ LUẬN TỐT NGHIỆP**

**KỸ SƯ NGÀNH KỸ THUẬT DỮ LIỆU**

**GIÁO VIÊN HƯỚNG DẪN**

**ThS. QUÁCH ĐÌNH HOÀNG**

**KHOÁ 2019-2023**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CNTT**  **\*\*\*\*\*\*\*\*** | **XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  **\*\*\*\*\*\*\*\*** |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: Nguyễn Minh Khoa MSSV: 19133029

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu Knowledge Distillation và ứng dụng

Họ và tên Giáo viên hướng dẫn:

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng thực hiện:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2. Ưu điểm:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

3. Khuyết điểm:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

4. Đề nghị cho bảo vệ hay không:

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2023

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CNTT**  **\*\*\*\*\*\*\*\*** | **XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  **\*\*\*\*\*\*\*\*** |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên Sinh viên 1: Nguyễn Minh Khoa MSSV: 19133029

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu Knowledge Distillation và ứng dụng

Họ và tên Giáo viên phản biện:

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng thực hiện:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2. Ưu điểm:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

3. Khuyết điểm:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

4. Đề nghị cho bảo vệ hay không:

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2023

Giáo viên phản biện

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên với sự chân thành và tình cảm sâu sắc nhất, tôi được bày tỏ lòng biết ơn đến tất cả các thầy cô trường đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo điều kiện, hỗ trợ, giúp đỡ tôi trong suốt quá trình học tập vừa qua.

Đồng thời, tôi xin gửi lời cảm ơn đến Ban Chủ nhiệm khoa Công nghệ Thông tin và các thầy cô khoa Công nghệ Thông tin - trường Đại học Sư phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo môi trường học tập và làm việc chuyên nghiệp, nhiệt tình giảng dạy chúng tôi thực hiện đề tài nói riêng và sinh viên trong khoa Công nghệ Thông tin nói chung trong quá trình học tập và làm việc tại trường.

Đặc biệt, tôi xin chân thành gửi lời cảm ơn tới thầy ThS. Quách Đình Hoàng - giáo viên hướng dẫn, người đã trực tiếp giúp đỡ, chỉ dẫn tận tình, hướng dẫn tôi hoàn thành tốt khoá luận tốt nghiệp trong thời gian qua.

Trong quá trình nghiên cứu đề tài, tôi có những cố gắng nhất định song do kinh nghiệm thực tiễn còn thiếu sót và kinh nghiệm chuyên môn còn non yếu, bài báo cáo vẫn có những thiếu sót và hạn chế nhất định. Kính mong nhận được những phản hồi, đóng góp ý kiến và chỉ bảo thêm của quý thầy cô để nhóm có thể đạt được những kiến thức hữu ích nhất, nâng cao ý thức để phục vụ cho kỹ năng sau này.

Tôi xin chân thành cảm ơn!

**PHỤ LỤC**

[**1. MỞ ĐẦU** 1](#_Toc137801178)

[**1.1. Lý do chọn đề tài** 1](#_Toc137801179)

[**1.2. Mục tiêu của đề tài** 1](#_Toc137801180)

[**1.3. Đối tượng và phạm vi nghiên cứu** 1](#_Toc137801181)

[**1.3.1. Đối tượng nghiên cứu** 1](#_Toc137801182)

[**1.3.2.** **Phạm vi nghiên cứu** 1](#_Toc137801183)

[**1.4. Phương pháp nghiên cứu** 2](#_Toc137801184)

[**1.5. Kết quả dự kiến** 2](#_Toc137801185)

[**2. NỘI DUNG** 3](#_Toc137801186)

[**2.1. Tổng quan về Deep Learning** 3](#_Toc137801187)

[**2.1.1. Khái niệm Deep Learning** 3](#_Toc137801188)

[**2.1.2. Các mô hình trong Deep Learning** 3](#_Toc137801189)

[**2.1.3. Khái niệm của ANN** 5](#_Toc137801190)

[**2.1.4. Cấu tạo và quá trình xử lý thông tin của ANN** 6](#_Toc137801191)

[**2.1.5. Activation Function** 7](#_Toc137801192)

[**2.1.5.1. Hàm Sigmoid** 7](#_Toc137801193)

[**2.1.5.2. Hàm Tanh** 8](#_Toc137801194)

[**2.1.5.3. Hàm Relu** 9](#_Toc137801195)

[**2.1.5.4. Hàm Leakly ReLU** 10](#_Toc137801196)

[**2.1.5.5. Hàm Softmax** 11](#_Toc137801197)

[**2.2. Tổng quan về Knowledge Distillation** 12](#_Toc137801198)

[**2.2.1. Các dạng Knowledge Distillation** 13](#_Toc137801199)

[**2.2.1.1. Response-based knowledge** 14](#_Toc137801200)

[**2.2.1.2. Feature-based knowledge** 15](#_Toc137801201)

[**2.2.1.3. Relation-based knowledge** 16](#_Toc137801202)

[**2.2.2. Huấn luyện Knowledge Distillation** 16](#_Toc137801203)

[**2.2.2.1. Offline distillation** 17](#_Toc137801204)

[**2.2.2.2. Online distillation** 17](#_Toc137801205)

[**2.2.2.3. Self distillation** 18](#_Toc137801206)

[**2.2.3. Thuật toán Distillation** 18](#_Toc137801207)

[**2.2.3.1. Adversarial distillation** 18](#_Toc137801208)

[**2.2.3.2. Multi-Teacher distillation** 19](#_Toc137801209)

[**2.2.3.3. Cross-modal distillation** 20](#_Toc137801210)

[**2.3. Phương pháp Response-Based Knowledge** 22](#_Toc137801211)

[**2.3.1. Mô hình Teacher - Student** 22](#_Toc137801212)

[**2.3.2. Temperature scale** 22](#_Toc137801213)

[**2.3.3. Distillation loss** 23](#_Toc137801214)

[**2.4. Data augmentation** 26](#_Toc137801215)

[**2.4.1. Thách thức về vấn đề data collection** 26](#_Toc137801216)

[**2.4.2. Các phương pháp phổ biến trong Data Augmentation** 27](#_Toc137801217)

[**2.5. Ứng dụng** 29](#_Toc137801218)

[**2.5.1. Phân loại ảnh có áp dụng Knowledge Distillation** 29](#_Toc137801219)

[**2.5.1.1. Giới thiệu bài toán** 29](#_Toc137801220)

[**2.5.1.2. Vấn đề trong bài toán phân loại ảnh** 29](#_Toc137801221)

[**2.5.2. Giới thiệu tập dữ liệu** 29](#_Toc137801222)

[**2.5.3. Các thư viện cần thiết** 30](#_Toc137801223)

[**2.5.3.1. Môi trường** 31](#_Toc137801224)

[**2.5.3.2. Thư viện** 31](#_Toc137801225)

[**2.5.4. Thực nghiệm** 32](#_Toc137801226)

[**2.5.4.1. Chuẩn hoá dữ liệu** 32](#_Toc137801227)

[**2.5.4.2. Huấn luyện mô hình** 32](#_Toc137801228)

[**2.5.4.2.1. Mô hình teacher** 33](#_Toc137801229)

[**2.5.4.2.2. Mô hình student không sử dụng distillation loss** 35](#_Toc137801230)

[**2.5.4.2.3. Mô hình student sử dụng distillation loss** 38](#_Toc137801231)

[**2.5.3.3. Đánh giá mô hình** 38](#_Toc137801232)

[**3. Kết luận** 39](#_Toc137801233)

[**3.1. Kết quả đạt được** 39](#_Toc137801234)

[**3.2. Hướng phát triển** 39](#_Toc137801235)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Kiến trúc mạng ANN 6](#_Toc139796166)

[Hình 2. Quá trình xử lý của ANN 6](#_Toc139796167)

[Hình 3. Đồ thị hàm Sigmoid và đạo hàm của Sgimoid 8](#_Toc139796168)

[Hình 4. Đồ thị hàm Tanh và đạo hàm của Tanh 9](#_Toc139796169)

[Hình 5. Đồ thị hàm ReLU và đạo hàm của ReLU 10](#_Toc139796170)

[Hình 6. Đồ thị hàm Leaky ReLU 11](#_Toc139796171)

[Hình 7. Mô hình Teacher - Student cho Knowledge Distillation 12](#_Toc139796172)

[Hình 8. Các dạng Knowledge Distillation 13](#_Toc139796173)

[Hình 9. Response-based knowledge distillation 14](#_Toc139796174)

[Hình 10. Feature-based knowledge distillation 15](#_Toc139796175)

[Hình 11. Relation-based knowledge distillation 16](#_Toc139796176)

[Hình 12. Types of knowledge distillation training schemes 17](#_Toc139796177)

[Hình 13. Kiến trúc Adversarial Distillation 19](#_Toc139796178)

[Hình 14. Kiến trúc Multi-Teacher Distillation 20](#_Toc139796179)

[Hình 15. Kiến trúc Cross-Model Distillation 21](#_Toc139796180)

[Hình 16. Phân phối xác suất của softmax và softmax temperature 23](#_Toc139796181)

[Hình 17. Sơ đồ Response base distillation 25](#_Toc139796182)

[Hình 18. Các loại khớp dữ liệu có thể có trong mô hình 26](#_Toc139796183)

[Hình 19. Các phương pháp trong Data Augmentation 28](#_Toc139796184)

[Hình 20. Các vật thể trong tập CIFAR-10 30](#_Toc139796185)

[Hình 21. là ví dụ tensor khởi tạo ngẫu nhiên được tạo bằng torch.rand(5,3) 32](#_Toc139796186)

[Hình 22. Kiến trúc mạng mô hình Teacher 33](#_Toc139796187)

[Hình 23. Kiến trúc của Densenet trên tập CIFAR-10 33](#_Toc139796188)

[Hình 24. Conv1 34](#_Toc139796189)

[Hình 25. Cái nhìn sâu hơn về Dense-100-1. Dense Block và Transition Block 34](#_Toc139796190)

[Hình 26. Kiến trúc mạng mô hình Student 35](#_Toc139796191)

[Hình 27. Conv1 36](#_Toc139796192)

[Hình 28. Layer 1 36](#_Toc139796193)

[Hình 29. Layer 2, Block 1, Conv 37](#_Toc139796194)

[Hình 30. Layer 2 37](#_Toc139796195)

# **1. MỞ ĐẦU**

## **1.1. Lý do chọn đề tài**

Hiện nay với sự phát triển về phần cứng cũng như mạng Deep Neural Network càng trở nên phổ biến. Các vấn đề về Computer Vision, Natural Language Processing đều đã đạt được các thành công nhất định.

Nhưng bên cạnh đó thì mô hình Deep Neural Network thường rất sâu và có số lượng tham số cực kì lớn. Sẽ rất khó trong việc triển khai trên các thiết bị cấu hình yếu và giới hạn phần cứng.

Chính vì vậy mà Knowledge Distillation ra đời nhằm mục đích tạo ra mô hình nhỏ gọn được học từ mô hình phức tạp nhưng vẫn đảm bảo được độ chính xác cao, mô hình này sẽ rất phù hợp để triển khai trên các thiết bị cạnh, đặc biệt là thiết bị cấu hình yếu.

## **1.2. Mục tiêu của đề tài**

Trong đề tài này, tôi tập trung tìm hiểu về mạng Convolutional Neural Network, đồng thời áp dụng phương pháp Knowledge Distillation cho mô hình CNN. Từ đó sẽ cho ra được mô hình tuy nhỏ gọn nhưng vẫn đảm bảo độ chính xác cao.

## **1.3. Đối tượng và phạm vi nghiên cứu**

### **1.3.1. Đối tượng nghiên cứu**

Phân loại ảnh trong tập dữ liệu CIFAR-10 thông qua mô hình CNN và phương pháp Knowledge Distillation để nhận dạng và phân loại.

### **1.3.2. Phạm vi nghiên cứu**

Phạm vi: dữ liệu về bộ CIFAR-10, mô hình được tạo và huấn luyện trên Colab.

Phạm vi thời gian: trong khoảng 16 tuần từ 15/02/2023 đến 15/06/2023

Phạm vi nội dung: Tôi quan tâm chủ yếu tới việc xử lý dữ liệu đầu vào dạng ảnh, đồng thời sử dụng mô hình Convolutional Neural Network kết hợp với Knowledge Distillation để cho ra mô hình có kích thước nhỏ nhằm triển khai trên thiết bị cạnh.

## **1.4. Phương pháp nghiên cứu**

Phương pháp thu thập số liệu: Tìm kiếm và tổng hợp thông tin, kiến thức, lý thuyết từ các bài báo đã có sẵn.

Phương pháp toán học: Giải thích kiến trúc va công thức của phương pháp Knowledge Distillation.

Phương pháp thực nghiệm: Thử nghiệm, huấn luyện, đánh giá mô hình và phương pháp thông qua các metric cụ thể.

## **1.5. Kết quả dự kiến**

Kết quả phương pháp: mô hình Student khi sử dụng Distillation Loss sẽ cho độ chính xác tốt hơn so với mô hình Student khi không sử dụng Distillation, độ chính xác có thể đạt xấp xỉ mô hình Teacher.

# **2. NỘI DUNG**

## **2.1. Tổng quan về Deep Learning**

### **2.1.1. Khái niệm Deep Learning**

Deep Learning là một trong những phương thức chính của lĩnh vực trí tuệ nhân tạo (AI), phương pháp này dạy máy tính cách xử lý dữ liệu được lấy cảm hứng từ bộ não con người. Mô hình deep learning có thể nhận diện các hình mẫu phức tạp trong ảnh, văn bản, âm thanh và các dữ liệu khác để có thể cho ra thông tin chuyên sâu và dự đoán chính xác.

Mô hình deep learning cố gắng huấn luyện để giúp máy tính có thể tư duy như não bộ của con người. Công nghệ deep learning này thúc đẩy tạo ra nhiều ứng dụng AI được sử dụng phổ biến trong các sản phẩm hàng ngày, chẳng hạn như:

* Trợ lý kỹ thuật số
* Điều khiển tivi từ xa kích hoạt bằng giọng nói
* Phát hiện gian lận
* Nhận dạng khuôn mặt tự động
* Đây cũng là một thành phần quan trọng của các công nghệ mới nổi như xe tự lái, thực tế ảo, v.v.

Mô hình deep learning là tập hợp các tập tin máy tính đã được các nhà nghiên cứu huấn luyện sẳn để thực hiện các tác vụ bằng thuật toán hoặc hệ thống các bước đã được định sẵn. Hiện nay đã có số ít doanh nghiệp đã và đang sử dụng các mô hình deep learning để phân tích dữ liệu và đưa ra dự đoán trong nhiều lĩnh vực khác nhau.

### **2.1.2. Các mô hình trong Deep Learning**

Mô hình deep learning được sử dụng trong đa lĩnh vực như sản xuất, điện tử, ô tô, hàng không vũ trụ, nghiên cứu y học và nhiều lĩnh vực khác,...

* Xe tự hành: Sử dụng các mô hình deep learning để có thể tự động nhận dạng và các loại phát hiện biển báo giao thông cũng như người đi bộ.
* Hệ thống quốc phòng: Sử dụng mô hình deep learning để tự động gắn cờ các khu vực quan trọng trong ảnh vệ tinh.
* Phân tích hình ảnh y khoa: Sử dụng mô hình deep learning để có thể phát hiện ra các tế bào ung thư một cách sớm nhất.
* Các nhà máy: Sử dụng ứng dụng của deep learning để có thể phát hiện ra công nhân khi những người này nằm trong khoảng cách không an toàn với máy móc.

Ta có thể phân biệt các trường hợp sử dụng mô hình deep learning thành bốn nhóm chính: Thị giác máy tính, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên (NLP) và công cụ đề xuất.

Thị giác máy tính

Thị giác máy tính có thể trích xuất dữ liệu cũng như thông tin dạng ảnh và video. Máy tính có thể hiểu hình ảnh theo cách giống như não bộ của con người. Thị giác máy tính có những ứng dụng như:

* Kiểm duyệt để tự động loại bỏ nội dung không an toàn khỏi kho hình ảnh hoặc video.
* Nhận diện khuôn mặt.
* Phân loại hình ảnh để xác định các chi tiết cần thấy trong ảnh.

Nhận dạng giọng nói

Nhận dạng gióng nói có thể phân tích giọng nói của con người, bao gồm tone, ngôn ngữ, cao độ, và giọng vùng miền khác nhau. Có thể thực hiện các tác vụ như:

* Hỗ trợ nhân viên tổng đài và tự động phân loại cuộc gọi.
* Tạo phụ đề cho video và các bản ghi âm cuộc họp để dễ dàng tiếp cận đối với người dùng.

Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên có khả năng xử lý văn bản do con người tạo ra, bao gồm các chức năng như:

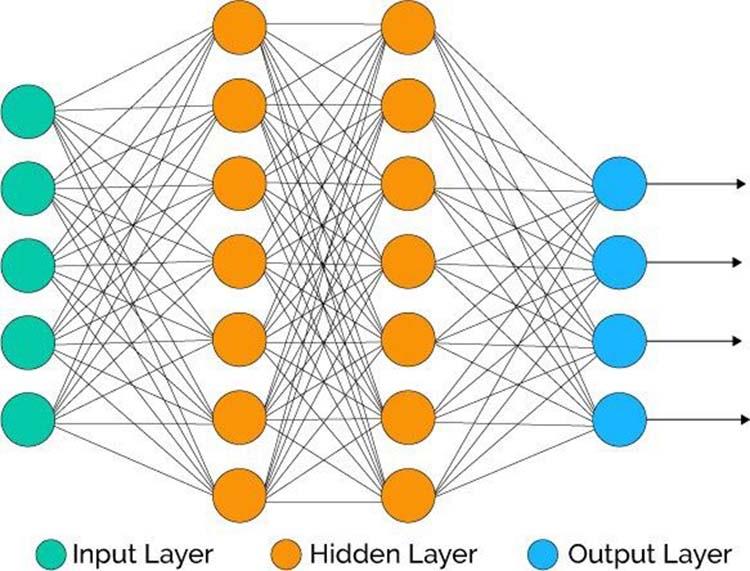
* Chatbot tự động.
* Tự động tóm tắt tin tức hoặc tài liệu.
* Lập chỉ mục các cụm từ quan trọng, cụ thể như những bình luận tiêu cực hoặc tích cực trên các nền tảng mạng xã hội.

Công cụ đề xuất

Phương pháp này có khả năng phân tích hành vi của người dùng nhằm có thể đưa ra nhưng đề xuất phì hợp, giúp người dùng có thể tiếp cận thêm các sản phẩm hoặc dịch vụ mới, tăng tính cá nhân hoá cho người dùng.

### **2.1.3. Khái niệm của ANN**

Mạng ANN là kỹ thuật dựa trên nghiên cứu về bộ não và hệ thần kinh của con người. Các mạng này mô phỏng lại các nơ-ron sinh học. Các tế bào thần kinh nơ-ron được kết nối với các bộ xử lý khác. Các nơ-ron được sắp xếp thành vecto với đầu ra đóng vai trò là đầu vào của lớp kế tiếp.



Hình . Kiến trúc mạng ANN

### **2.1.4. Cấu tạo và quá trình xử lý thông tin của ANN**

Mạng ANN gồm 3 thành phần chính là: Input layer, Hidden layer và Output layer.

Quá trình xử lý thông tin của một ANN:



Hình . Quá trình xử lý của ANN

* Input: Mỗi input tương ứng với 1 thuộc tính của dữ liệu
* Output: Kết quả của một ANN là giải pháp cho một vấn đề
* Connection Weights: Thể hiện mức độ quan trọng của dữ liệu đầu vào đối với quá trình xử lý thông tin.
* Summation Function: Tính tổng trọng số của tất cả các input được đưa vào mỗi nơ-ron.
* Transformation Function: Cho biết khả năng kích hoạt của nơ-ron đó.

### **2.1.5. Activation Function**

Activation function là hàm phi tuyến tại output của các nơ-ron.. Tính toán tổng các trọng số của đầu vào, thêm độ lệch và sau đó quyết địch có được kịch hoạt hay không. Với công thức tổng quát như sau:

#### **2.1.5.1. Hàm Sigmoid**

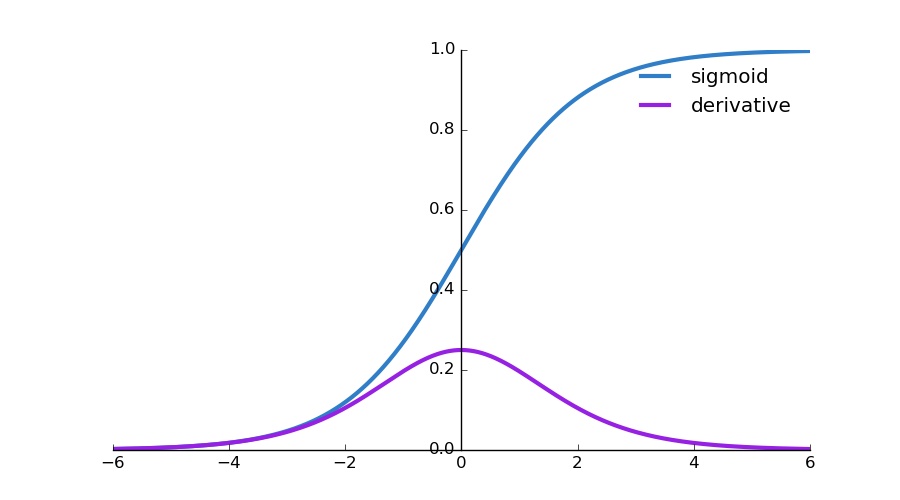
Phạm vi hàm Sigmoid trong khoảng từ 0 dến 1.

Hàm Sigmoid có công thức:

Đạo hàm Sigmoid:

Hàm Sigmoid nhận đầu vào là số thực (theo hình 1):

* Giá trị đầu vào âm sẽ thành 0
* Giá trị đầu vào dương sẽ thành 1



Hình . Đồ thị hàm Sigmoid và đạo hàm của Sgimoid

#### **2.1.5.2. Hàm Tanh**

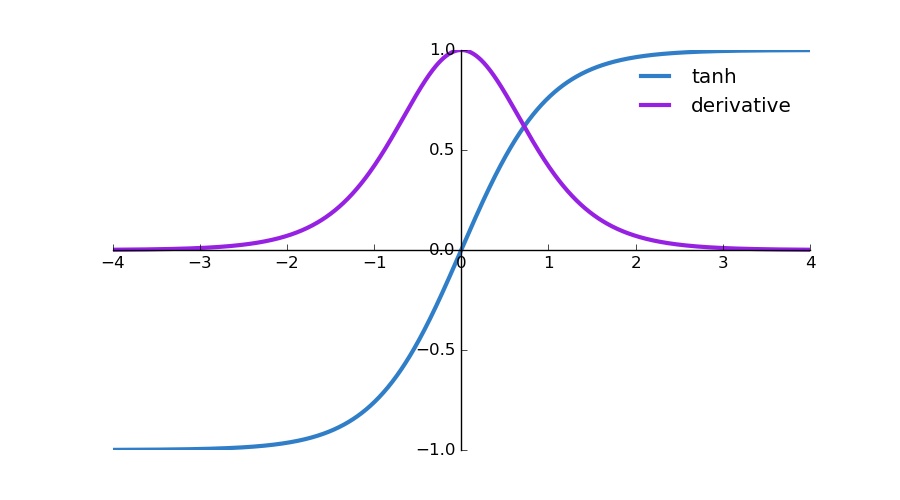
Phạm vi hàm Tanh trong khoảng từ -1 dến 1.

Hàm Tanh có công thức:

Được viết lại dựa trên hàm Sigmoid:

Đạo hàm Tanh:

Hàm Tanh có khoảng từ -1 đến 1 làm cho việc học của các lớp tiếp theo trở nên dễ dàng hơn.



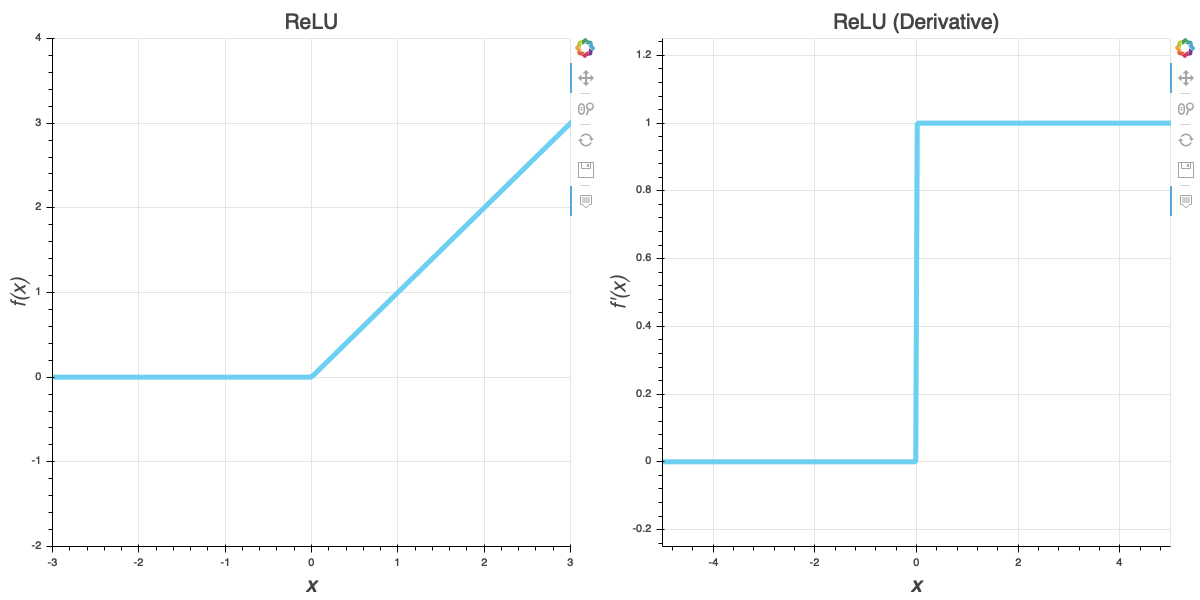
Hình . Đồ thị hàm Tanh và đạo hàm của Tanh

#### **2.1.5.3. Hàm Relu**

Hàm Relu được sử dụng rộng rãi nhờ tính đơn giản của nó.

Hàm Relu có công thức:

Đầu ra hàm ReLU là tuyến tính nếu đầu vào dương, ngược = 0. Phạm vi hàm ReLU trong khoảng [0, max). Hàm ReLU có ưu điểm hơn các hàm khác nhờ khả năng tính toán nhanh hơn. Đạo hàm của ReLU không đổi khi x > 0. Ngoài ra hàm ReLU gặp phải khi gặp giá trị âm thì nó luôn trả về kết quả = 0. Khi đó thì đồ dốc của ReLU = 0



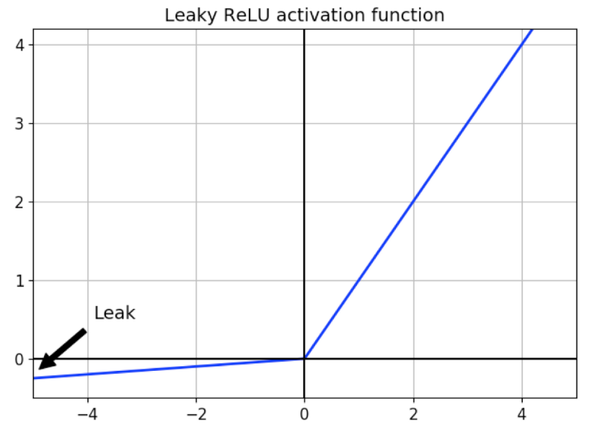
Hình . Đồ thị hàm ReLU và đạo hàm của ReLU

#### **2.1.5.4. Hàm Leakly ReLU**

Leaky ReLU là biến thể từ hàm ReLU

Hàm Leaky ReLU có công thức:

Hàm Leaky ReLU có siêu tham số là hằng số nhỏ xác định mức độ leak của hàm. Khi coi có giá trị không đổi là 1 khi và 0.01 khi . Điều này giúp cho Leaky ReLU tránh được vấn đề của ReLU.



Hình . Đồ thị hàm Leaky ReLU

#### **2.1.5.5. Hàm Softmax**

Hàm Softmax là hàm khá phố biển trong lĩnh vực học sâu. Được sử dụng để chuẩn hoá đầu ra của mạng thành phân phối xác suất trên các lớp đầu ra đã được dự đoán:

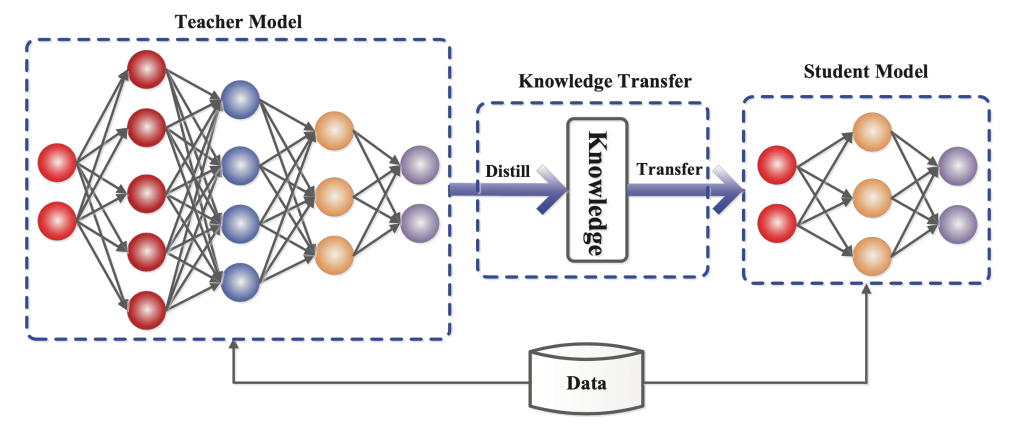
Hàm Softmax có công thức:

## **2.2. Tổng quan về Knowledge Distillation**

Knowledge distillation nhằm mục đích chuyển giao kiến thức từ mô hình lớn sang mô hình nhỏ hơn, có thể dùng triển khai thực tế trên các thiết bị cấu hình yếu.

Knowledge distillation được sử dụng phổ biến với các mô hình neural network có kiến trúc phức tạp. Do đó, với sự ra đời của deep learning trong thập kỷ qua và sự thành công của nó trong các lĩnh vực như speech recognition, image recognition, và computer vision. Kỹ thuật knowledge distillation đã trở nên nổ bật và được ứng dụng nhiều trong thực tế.

Vấn đề lớn trong việc triển khai các mô hình deep neural network trên các thiết bị cấu hình yếu đó là do sự giới hạn về khả năng lưu trữ và tính toán. Để giải quyết vấn đề này, phương pháp nén mô hình được đề xuất để chuyển giao knowledge từ một mô hình lớn sang nhỏ hơn mà không làm giảm hiệu suất. Quá trình học mô hình nhỏ từ mô hình lớn được gọi là knowledge distillation.

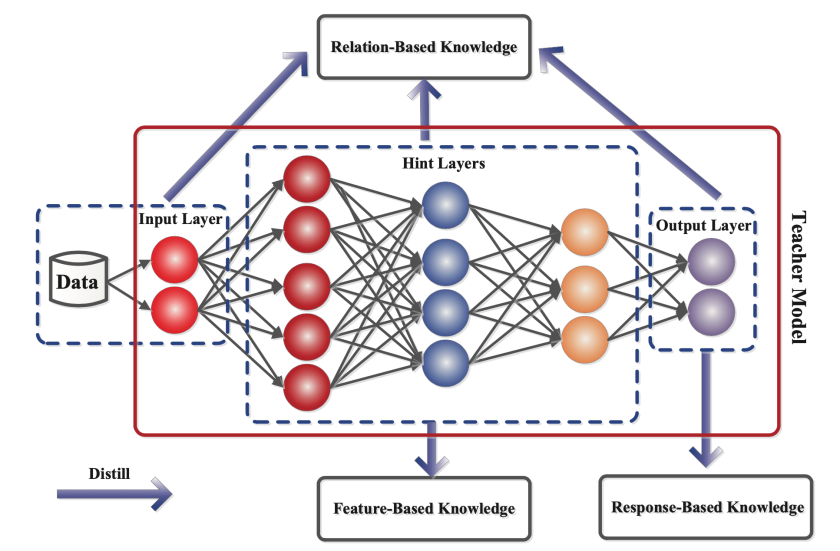


Hình . Mô hình Teacher - Student cho Knowledge Distillation

Như hình 7, trong quá trình knowledge distillation, mô hình Student học cách bắt chước mô hình Teacher và tận dụng kiến thức từ mô hình Teacher để đạt được độ chính xác tương đương hoặc cao hơn.

### **2.2.1. Các dạng Knowledge Distillation**

Trong mô hình neural network, kiến thức thường đề cập tới weight và bias. knowledge kistillation cơ bản sử dụng nguồn kiến thức từ giáo viên, trong khi những cái khác tập trung vào weight và activation function ở các lớp trung gian.



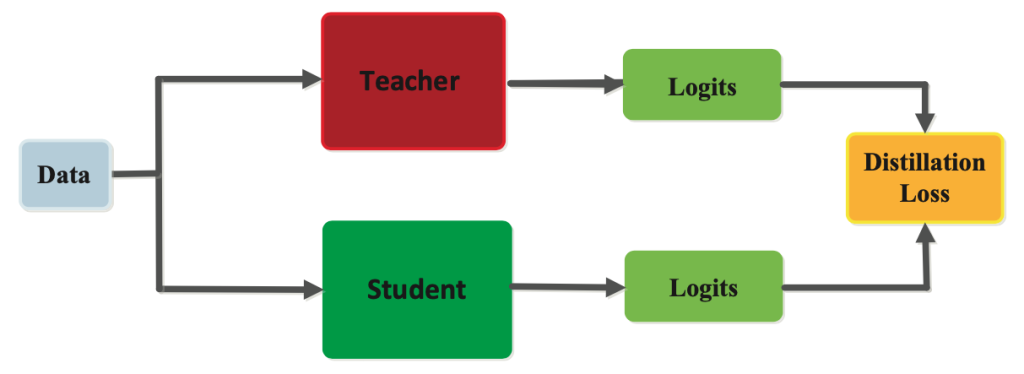
Hình . Các dạng Knowledge Distillation

Kiến thức được chia thành 3 loại: response-based knowledge, feature-based knowledge, và relation-based knowledge. Hình 8 mô tả 3 loại kiến thức từ mô hình Teacher.

Một hệ thống knowledge distillation gồm 3 thành phần chính: kiến thức, thuật toán distillation và kiến trúc Teacher - Student.

#### **2.2.1.1. Response-based knowledge**

Theo hình 8, response-based knowledge tập trung vào layer đầu ra của mô hình teacher. Giả thuyết là mô hình student sẽ bắt chước các dự đoán của mô hình teacher. Theo hình 9, điều này có thể đạt được bằng cách sử dụng loss function, còn được gọi là distillation loss. Đưa ra sự khác biệt logits của mô hình teacher và student tương ứng. Cực tiểu hoá loss function sẽ giúp mô hình student trở nê tốt hơn trong việc đưa ra các dự đoán giống với mô hình teacher.

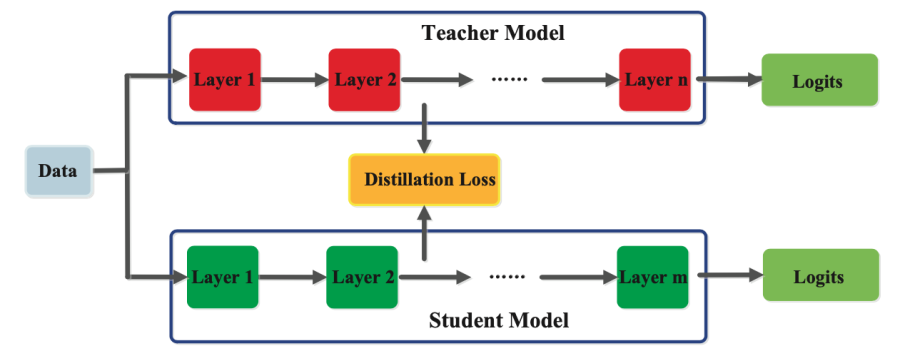


Hình . Response-based knowledge distillation

Đối với các tác vụ của Computer Vision như phân loại ảnh. Soft targets bao gồm response-based knowledge. Soft targets đại diện cho phân phối xác suất trên các lớp đầu ra, và thường được ước lượng bằng softmax function. Mỗi soft targets vào knowledge được điều chỉnh bằng cách sử dụng thêm 1 tham số gọi là temperature. Response-based knowledge chắt lọc kiến thức dựa trên soft targets thường được sử dụng trong bài toán supervised learning.

#### **2.2.1.2. Feature-based knowledge**

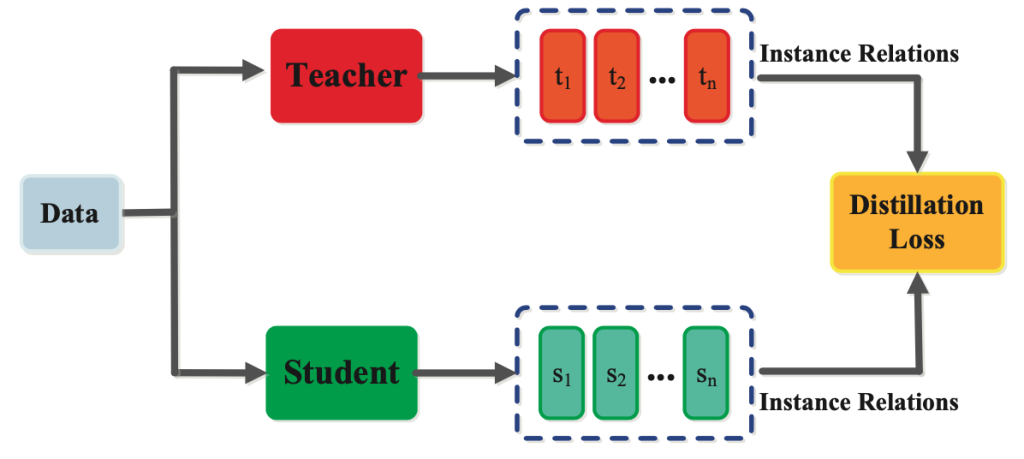
Mô hình teacher đã được huấn luyện sẽ nắm bắt kiến thức về dữ liệu trong các lớp trung gian của nó, điều này thích hợp với các mạng deep neural networks. Các lớp trung gian học cách phân biệt các feature cụ thể và kiến thức này có thể được sử dụng để huấn luyện mô hình student. Theo Hình 10, mục tiêu là huần luyện mô hình student học feature activation tương tự như mô hình teacher. Loss function đạt được điều này bằng cách cực tiểu hoá sự khác biệt giữa các feature activation của mô hình teacher và student.



Hình . Feature-based knowledge distillation

#### **2.2.1.3. Relation-based knowledge**

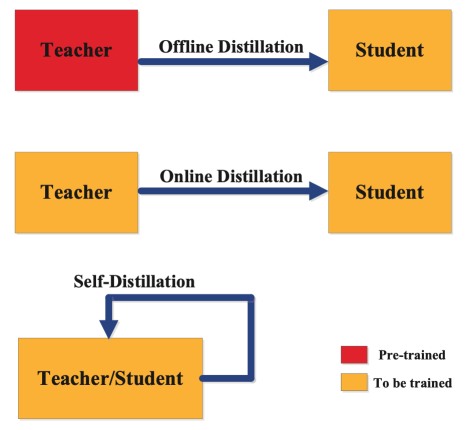
Ngoài kiến thức được biểu diễn trong các lớp đầu ra và các lớp trung gian của mô hình neural network, kiến thức nắm bắt mối quan hệ giữa các feautre maps cũng có thể được sử dụng để huấn luyện mô hình student. Dạng knowledge này được gọi là knowledge dựa trên mối quan hệ được mô tả trong Hình 11. Mối quan hệ này có thể được mô hình hóa dưới dạng mối tương quan giữa các feature maps, graphs, similarity matrix, feature embeddings hoặc phân phối xác suất dựa trên feature representations.



Hình . Relation-based knowledge distillation

### **2.2.2. Huấn luyện Knowledge Distillation**

Theo hình 12, có 3 phương pháp chính để huấn luyện mô hình teacher - student, đó là offline, online, và self distillation. Việc phân loại các phương pháp huấn luyện sẽ phụ thuộc vào việc liệu mô hình teacher có được modified cùng lúc với mô hình student hay không.



Hình . Types of knowledge distillation training schemes

#### **2.2.2.1. Offline distillation**

Offline distillation là phương pháp phổ biến nhất, trong đó mô hình teacher được sử dụng để huấn luyện mô hình student. Trong hình này, đầu tiên mô hình teacher được huấn luyện trước trên tập training, sau đó knowledge từ mô hình teacher sẽ được chắt lọc để huấn luyện mô hình student.

#### **2.2.2.2. Online distillation**

Đối với offline distillation, mô hình teacher được huấn luyện trước thường là deep neural network với kích thước vô cùng lớn. Đối với 1 số trường hợp, mô hình được huấn luyện trước có thể không phù hợp với offline distillation. Để giải quyết vấn đề này, ta có thể sử dụng online distillation cho cả mô hình teacher và student. Online distillation được vận hành bằng cách sử dụng tính toán song song, do đó nó là một phương pháp hiệu quả cao.

#### **2.2.2.3. Self distillation**

Theo hình 12, trong self-distillation, mô hình tương tự được sử dụng cho cả mô hình teacher và student.

Ví dụ, knowledge từ các lớp sâu hơn của deep neural network có thể được sử dụng để huấn luyện cho các lớp nông hơn.

Đây có thể được coi là 1 trường hợp đặc biệt của online distillation. Knowledge từ các epoch trước của mô hình teacher có thể được chuyển sang các epoch sau để huấn luyện mô hình student.

### **2.2.3. Thuật toán Distillation**

Các thuật toán khác nhau giúp cho quá trình transfer kiến thức từ mô hình teacher sang mô hình student được hiệu quả hơn trong các cài đặt phức tạp. Xem xét các thuật toán phổ biến này.

#### **2.2.3.1. Adversarial distillation**

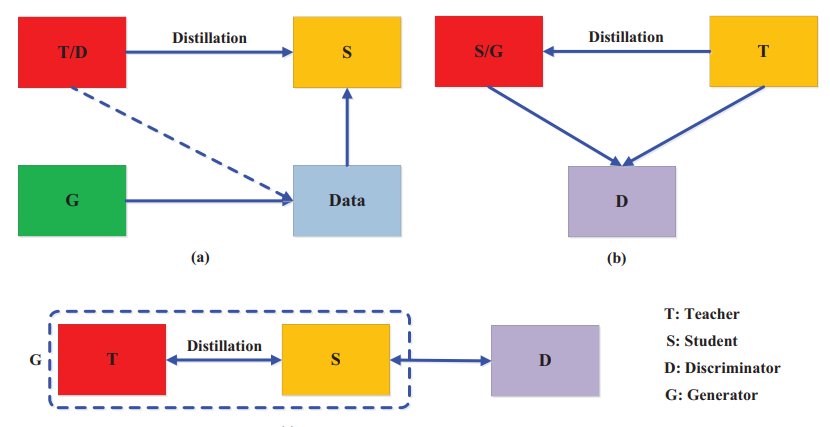
Adversarial distillation đã thành công trong mô hình tổng hợp dựa trên mạng GAN (Generative Adversarial Networks) đã được phát triển trước. Adversarial Learning cũng đã có ảnh hướng lớn trong Knowledge Distillation vì mạng GAN có thể được sử dụng để tăng cường tập train hiện có để có thể học tốt hơn hoặc cho phép mô hình teacher/student học data distribution tốt hơn.

Ý tưởng đằng sau Adversarial Distillation được mô tả trong hình 7 là chuyển giao kiến thức đã học được từ một mạng nơ-ron lớn, được gọi là mô hình Teacher sang một mạng nơ-ron nhỏ hơn, nông hơn được gọi là mô hình Student. Quá trình này giúp nén kiến thức từ mô hình lớn hơn thành một biểu diễn nhỏ gọn hơn mà không làm giảm hiệu suất.

Adversarial distillation liên quan đến việc đào tạo một mô hình nhỏ hơn, đơn giản hóa bằng cách sử dụng kiến thức và hướng dẫn từ một mô hình lớn hơn, phức tạp hơn. Nó đặc biệt được áp dụng trong kiến trúc mạng thần kinh.

Trong nhiều bối cảnh kiến trúc, Adversarial distillation có thể được sử dụng để đào tạo các mô hình nhỏ gọn và hiệu quả trong khi vẫn duy trì độ chính xác của chúng. Mô hình Teacher, điển hình là một cấu trúc phức tạp và hoạt động tốt, hoạt động như một nguồn kiến thức. Mặt khác, mô hình Student nhằm mục đích bắt chước hành vi và dự đoán của mô hình Teacher nhưng với ít tham số hơn.

Trong thực tế, adversarial distillation có thể được triển khai bằng cách sử dụng nhiều kỹ thuật, bao gồm adversarial training và adversarial regularization. Các kỹ thuật này liên quan đến việc sửa đổi loss function được sử dụng để huấn luyện mô hình student để tính đến sự hiện diện của các ví dụ adversarial.



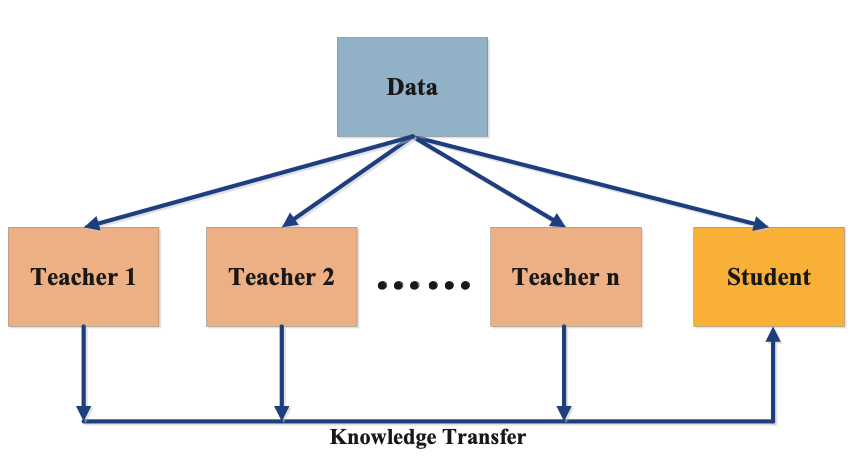
Hình . Kiến trúc Adversarial Distillation

#### **2.2.3.2. Multi-Teacher distillation**

Trong Multi-Teacher distillation, mô hình Student tiếp nhận kiến thức từ các mô hình Teacher khác nhau như hình 14. Sử dụng một tập hợp các mô hình Teacher có thể cung cấp cho mô hình Student các loại kiến thức riêng biệt, có thể sẽ đạt hiệu quả cáo hơn so với tiếp nhận kiến thức từ một mô hình teacher duy nhất.

Multi-teacher distillation được sử dụng để cải thiện hiệu suất của mô hình Student bằng cách tận dụng kiến thức từ nhiều mô hình Teacher. Nó đặc biệt hữu ích khi mô hình Teacher vượt trội ở các khía cạnh khác nhau hoặc có các lĩnh vực chuyên môn khác nhau và mục tiêu là kết hợp kiến thức của mô hình Teacher để tạo ra một mô hình Student mạnh mẽ và chính xác hơn.

Kiến thức từ nhiều mô hình teacher có thể được kết hợp lại bằng cách tính trung bình trên tất cả các mô hình teacher. Loại kiến thức được chuyển giao từ mô hình teacher được dựa vào logits và feature representation. Các mô hình teacher khác nhau có thể chuyển giao các loại kiến thức khác nhau.



Hình . Kiến trúc Multi-Teacher Distillation

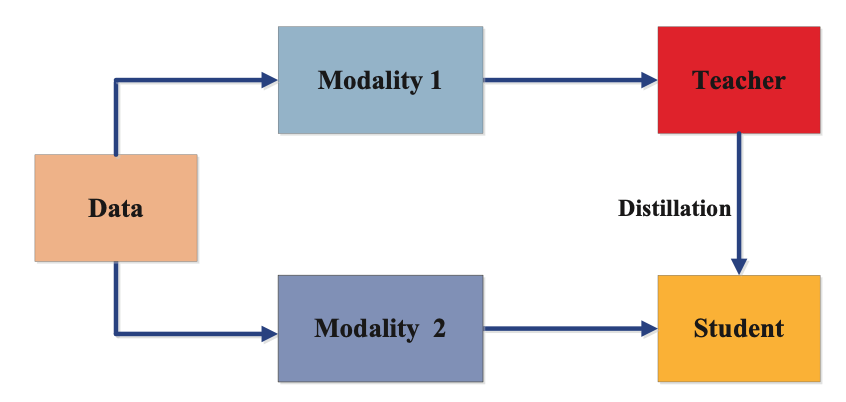
#### **2.2.3.3. Cross-modal distillation**

Hình 15 cho thấy Cross-modal distillation đề cập đến quá trình chuyển giao kiến thức từ một modality (ví dụ: hình ảnh) sang một modality khác (ví dụ: văn bản hoặc âm thanh) bằng cách sử dụng phương pháp Distillation. Nó liên quan đến việc huấn luyện một mô hình theo một modality cụ thể, được gọi là Teacher modality, sau đó sử dụng kiến thức của mô hình này được dùng để huấn luyện một mô hình khác theo một modality khác, được gọi là Student modality.

Đối với Cross-modal distillation, trọng tâm là chắt lọc kiến thức từ một kiến trúc quy mô lớn trong một modality sang một kiến trúc nhỏ hơn hoặc hiệu quả hơn trong một modality khác. Mục tiêu là tận dụng thông tin mà mô hình giáo viên đã học được để cải thiện hiệu suất của mô hình học sinh theo phương thức mục tiêu.

Cross-model distillation được sử dụng phổ biến trong các lĩnh vực computer vision.

Ví dụ: Kiến tức từ một giáo viên được đào tạo với dữ liệu đã được đánh nhãn có thể được sử dụng để chuyển giao sang mô hình student với có input chưa được đánh nhãn như dữ liệu về văn bản hoặc âm thanh. Trong trương hợp này, các features được học từ hình ảnh từ mô hình teacher sẽ được dùng để giám sát mô hình student. Cross-modal distillation rất hữu ích cho các ứng dụng như trả lời câu hỏi bằng hình ảnh, chú thích hình ảnh, ...



Hình . Kiến trúc Cross-Model Distillation

## **2.3. Phương pháp Response-Based Knowledge**

### **2.3.1. Mô hình Teacher - Student**

Mô hình Teacher: mô hình Teacher được huấn luyện trên tập dữ liệu đã được đánh nhãn. Bộ dữ liệu này cần có kích thước phù hợp để Teacher học được các trường hợp một cách tổng quát nhất. Sau khi mô hình Teacher đã đủ tốt, ta sẽ sử dụng mô hình Teacher để huấn luyện cho mô hình Student.

Mô hình Student: Cải thiện hiệu suất bằng cách sử dụng gợi ý từ mô hình Teacher. Nếu huấn luyện theo phương pháp thông thường thì mô hình Student sẽ áp dụng hàm loss function dạng cross entropy như sau:

Trong đó là số lượng classes.

Cross-entropy để đo lường sự tương quan giữa phân phối xác suất dự báo và ground-truth ( là một one-hot véc tơ có giá trị 1 tại index trùng với ground-truth). Cực tiểu hoá loss function để phân phối xác suất dự báo và ground-truth giống nhau nhất.

Trong loss function, dự báo từ teacher sẽ được sử dụng để thay thế cho ground-truth. Mục tiêu cực tiểu hoá loss function để phân phối xác suất dự báo của student gần với teacher nhất.

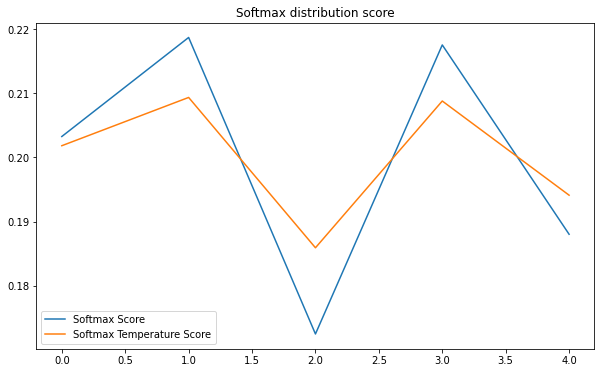
Distillation loss tại quan sát xi có dạng:

Trong đó là phân phối xác suất của dự báo từ teacher và là phân phối xác suất dự báo từ student tại quan sát thứ .

### **2.3.2. Temperature scale**

Đây là phương pháp calibration score giúp cho softmax score trở nên smoothing hơn. Giả sử ta có một phân phối xác suất dựa trên hàm softmax của vector input như sau:

Trong đó là số lượng classes output. Khi áp dụng temperature scale ta sẽ thu được xác suất mới như sau:



Hình . Phân phối xác suất của softmax và softmax temperature

Theo hình 16: Ta thấy đường màu cam của Softmax Temperature sẽ smoothing hơn so với đường màu xanh của Softmax Score. Thứ tự về độ lớn của xác suất không thay đổi ở cả hai distribution nên sau khi calibration không ảnh hưởng tới dự báo.

### **2.3.3. Distillation loss**

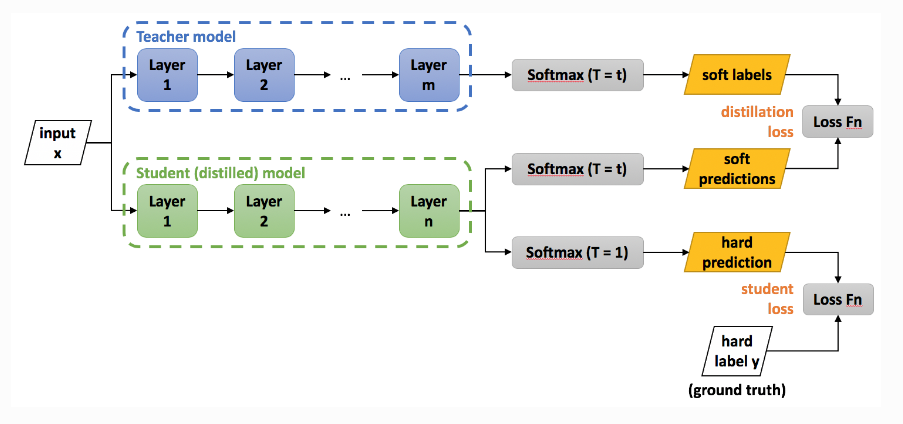
Khi áp dụng phương pháp temperature scale thì phân phối xác suất của teacher và student sẽ trở nên smoothing hơn. Ta ký hiệu lần lượt là và . Distillation loss sẽ là:

Trong đó ) là ký hiệu của hàm phân phối softmax sau khi làm mềm với đầu vào zi và hệ số temperature scale . Index và tương ứng với quan sát thứ và thứ của teacher và student.

Thông qua thực nghiệm của Geoffrey Hinton ta thấy việc học sẽ hiệu quả hơn nếu có sự kết hợp giữa học từ nhãn groud-truth và học từ dự báo của teacher. Do đó sẽ thêm student loss là một hàm cross-entropy thông thường vào distillation loss. Khi đó Distillation loss cuối cùng sẽ là:

Trong đó là vector one-hot của nhãn ground-truth đối với quan sát . Trường hợp sẽ tương ứng với softmax không áp dụng làm mềm trong student loss.

Thông thường, , sẽ được lựa chọn là một giá trị rất nhỏ so với b để teacher đóng vai trò lớn hơn trong quá trình huấn luyện.



Hình . Sơ đồ Response base distillation

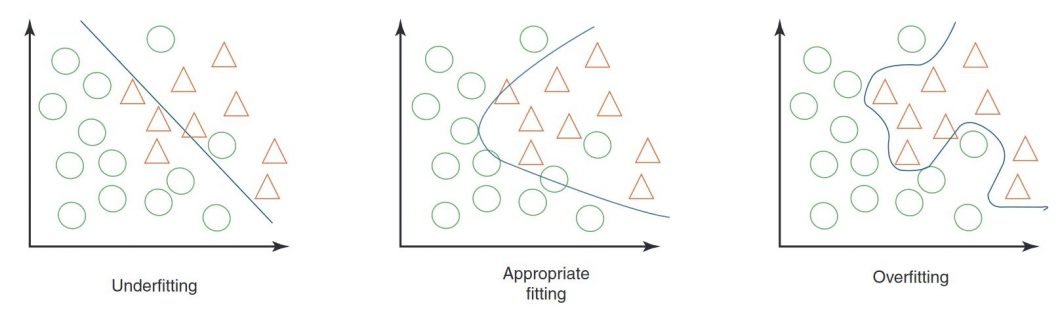
Theo hình 17: Mô hình teacher và student sẽ được kết hợp trong distillation loss. Phân phối xác suất của hai models này sẽ được làm mềm theo cùng hệ số temperature scale. Trong trường hợp chỉ huấn luyện student độc lập thì không làm mềm phân phối xác suất, đồng thời cross-entropy được tính dựa trên phân phối xác suất dự báo (hard prediction) và nhãn ground-truth.

## **2.4. Data augmentation**

Vấn đề dữ liệu có vai trò rất quan trọng trong lĩnh vực AI, đặc biệt là deep learning. Việc có ít dữ liệu sẽ làm cho model thiếu chính xác và kết quả không tốt như mong đợi. Chính vì vậy, phương pháp data augmentation ra đời với một tập hợp các kỹ thuật giúp tăng kích thước và nâng cao chất lượng của tập train, phương pháp này sẽ giúp train các mô hình deep learning tốt hơn.

Các thuật toán augmentation cho hình ảnh bao gồm geometric transformations, color space augmentation, kernel filtering, mixing images, random erasing, feature space augmentation, adversarial training, GAN (generative adversarial networks ).

Một trong những thử thách khó khăn nhất là khả năng tổng quát hoá của các mô hình deep learning. Mô hình có khả năng tổng quát hoá kém sẽ overfitting trên tập train. Để xây dựng được mô hình deep learning hiệu quả, data augmentation là một phương pháp rất mạnh để giảm overfitting. Các vấn đề về khớp dữ liệu được mô tả như trong hình 10.



Hình . Các loại khớp dữ liệu có thể có trong mô hình

### **2.4.1. Thách thức về vấn đề data collection**

* Việc thu thập thêm dữ liệu là cần thiết khi bộ dữ liệu về computer vision được public là không đủ. Tuy nhiên, điều đó sẽ khó thực hiện trong một vài tình huống, đặc biệt là sẽ khó khi xây dựng các ứng dụng về computer vision có mục đích. Một số vấn đề về thu thập dữ liệu như:
* Ứng dụng yêu cầu nhiều dữ liệu: Các ứng dụng thị giác máy tính trong thế giới thực liên quan đến các nhiệm vụ thị giác máy tính rất phức tạp đòi hỏi các mô hình, bộ dữ liệu và nhãn ngày càng phức tạp.
* Nguồn dữ liệu hạn chế: Khi các nhiệm vụ trở nên phức tạp hơn thì các yêu cầu thu thập dữ liệu trở nên khó khăn hơn. Một số tình huống có thể hiếm khi xảy ra trong thế giới thực, nhưng việc xử lý chính xác các tình huống này là rất quan trọng.
* Khó thu thập dữ liệu: Quá trình tạo dữ liệu đào tạo chất lượng cao rất khó khăn và tốn kém.
* Tăng chí phí: Chi phí sẽ tăng theo độ phức tạp của tác vụ và chuyển từ labeling frames sang labeling objects, keypoints và thậm chí cả pixel trong hình ảnh.
* Bảo mật dữ liệu: Quyền riêng tư trong thị giác máy tính đang trở thành vấn đề chính và làm phức tạp thêm việc thu thập dữ liệu. Điều này giới hạn mức độ dữ liệu trong thế giới thực có thể được thu thập và thúc đẩy nhu cầu đào tạo các mô hình học sâu trên các tập dữ liệu nhỏ hơn.

### **2.4.2. Các phương pháp phổ biến trong Data Augmentation**

* Geometric transformations: Tăng cường dữ liệu dạng ảnh bằng cách lật ngang hoặc dọc, cắt ngẫu nhiên, xoay, dịch chuyển hình ảnh sang trái/phải/lên/xuống hoặc chèn nhiễu.
* Color distortion: Bao gồm thay đổi độ sáng, màu sắc hoặc độ bão hòa của hình ảnh. Việc thay đổi phân bố màu hoặc thao tác với biểu đồ kênh màu RGB được sử dụng để tăng khả năng chống lại các sai lệch ánh sáng của mô hình.
* Kernel filters: Sử dụng các kỹ thuật xử lý hình ảnh để làm sắc nét và làm mờ hình ảnh. Những phương pháp này nhằm mục đích tăng chi tiết về các đối tượng quan tâm hoặc để cải thiện khả năng chống nhòe do chuyển động.
* Mixing images: Áp dụng các kỹ thuật để trộn các hình ảnh khác nhau lại với nhau bằng cách lấy trung bình các giá trị pixel của chúng cho từng kênh RGB hoặc cắt xén và vá hình ảnh ngẫu nhiên. Phương pháp này cho thấy hiệu quả trong việc tăng hiệu suất mô hình.
* Information deletion: Phương pháp xóa ngẫu nhiên, cắt bỏ để che giấu các phần hình ảnh ngẫu nhiên, sử dụng các bản vá chứa đầy các giá trị pixel ngẫu nhiên một cách tối ưu. Xóa một mức thông tin được sử dụng để tăng khả năng chống tắc nghẽn trong nhận dạng hình ảnh.



Hình . Các phương pháp trong Data Augmentation

## **2.5. Ứng dụng**

### **2.5.1. Phân loại ảnh có áp dụng Knowledge Distillation**

#### **2.5.1.1. Giới thiệu bài toán**

Hiện nay với sự phát triển của công nghệ thì bài toán phân loại ảnh là một trong những nhiệm vụ phổ biến trong Computer Vision. Mục tiêu chính của bài toán này đó chính là phân loại một hình ảnh đầu vào thành một nhãn đầu ra.

#### **2.5.1.2. Vấn đề trong bài toán phân loại ảnh**

Vấn đề về dữ liệu

* Thu thập thêm dữ liệu: là cần thiết trong một số trường hợp đối với bài toán phân loại ảnh. Tuy nhiên, điều đó sẽ khó thực hiện trong một vài trường hợp.
* Dữ liệu bị hạn chế: Khi các yêu cầu trở nên phức tạp hơn thì các yêu cầu về thu thập dữ liệu trở nên khó khăn hơn. Một số yêu cầu có thể hiếm khi xảy ra, nhưng việc xử lý chính xác các yêu cầu này là rất quan trọng.
* Khó thu thập dữ liệu: Quá trình thu thập được dữ liệu có chất lượng cao sẽ rất khó khăn và tốn kém.
* Tăng chí phí: Chi phí sẽ tăng theo độ phức tạp của việc thu thâp dữ liệu.

### **2.5.2. Giới thiệu tập dữ liệu**

Tập dữ liệu CIFAR-10 (Canadian Institute for Advanced Research, 10 classes) là tập dữ liệu dạng ảnh, được sử dụng rộng rãi trong computer vision. Kể từ khi được phát hành, tập dữ liệu CIFAR-10 này thường được sử dụng trong các bài toán phân loại. Khi các kỹ thuật tiên tiến hơn được ra đời thì CIFAR-10 vẫn là bộ dữ liệu đáng tin cậy cho các nhà nghiên cứu cũng như người học.

Tập dữ liệu CIFAR-10 gồm 60000 ảnh, trong đó 45000 ảnh dùng cho tập train, 5000 ảnh cho tập val và 10000 ảnh dùng cho tập test. Các hình ảnh được đánh nhãn như: airplane, automobile (but not truck or pickup truck), bird, cat, deer, dog, frog, horse, ship, and truck (but not pickup truck). Hình ảnh tập dữ liệu CIFAR-10 ở dạng 32x32 (theo hình 14).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **airplane** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane10.png |
| **automobile** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile10.png |
| **bird** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird10.png |
| **cat** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat10.png |
| **deer** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer10.png |
| **dog** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog10.png |
| **frog** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog10.png |
| **horse** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse10.png |
| **ship** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship10.png |
| **truck** | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck10.png |

Hình . Các vật thể trong tập CIFAR-10

### **2.5.3. Các thư viện cần thiết**

Bài toán Knowledge Distillation được thực hiện trên Google Colab cùng với thư viện chính là pytorch, ngoài ra còn sử dụng thêm các thư viện khác như matplotlib, numpy và scikit – learn.

#### **2.5.3.1. Môi trường**

Google Colab là sản phẩm của Google Research cho thực thi chương trình bằng Python qua trình duyệt, được dùng để phân tích dữ liệu, thực hiện các mô hình machine learning, deep learning và giảng dạy. Google Colab không yêu cầu cài đặt hoặc cấu hình, tất cả đều chạy trên trình duyệt và sử dụng tài nguyên máy tính từ CPU, GPU và TPU tốc độ cao của Google.

Google Colab cung cấp nhiều loại GPU, nhưng người dùng không thể chọn loại GPU trong Colab, cấu hình GPU sẽ thay đổi theo thời gian. Vì là đây là dịch vụ miễn phí mà Google cung cấp nên Colab sẽ được ưu tiên trong việc sử dụng tài nguyên, bên cạnh đó sẽ có sự giới hạn về thời gian, thời gian sử dụng Colab có thể lên đến 12h.

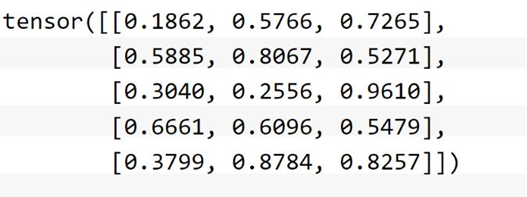
#### **2.5.3.2 . Thư viện**

Thư viện Pytorch

Pytorch là framework được phát triển bởi Facebook, đây là một framework khá được ưa chuộng đối với lĩnh vực Deep Learning. Ưu điểm của Pytorch là nhờ số lượng người dùng đông đảo, cộng đồng phát triển lớn mạnh. Cho đến năm 2019 thì framework này đã vươn lên vị trí thứ hai thế giới về số lượng người dùng trong lĩnh vực Deep Learning. Đây là package sử dụng thư viện CUDA và C++ giúp hỗ trợ tính toán trên GPU nhằm gia tăng tốc độ xử lý của mô hình. Package này hướng tới hai mục tiêu chính:

* Thay thế kiến trúc Numpy để tính toán trên GPU.
* Cung cấp tốc độ xử lý cao và linh hoạt

Dạng dữ liệu của pytorch là tensor. Tensor là dữ liệu nhiều chiều tương tự như ma trận trong numpy nhưng được thêm các tính chất để có thể hoạt động trên GPU nhằm gia tăng tốc độ tính toán.



Hình . là ví dụ tensor khởi tạo ngẫu nhiên được tạo bằng torch.rand(5,3)

Thư viện Scikit – learn

Scikit-learn một thư viện mã nguồn mở được cấp phép bởi BSD, khuyến khích sử dụng trong học thuật và thương mại. Ngoài ra đây còn là thư viện được sử dụng phổ biến nhất trong các bài toán machine learning. Cung cấp một tập công cụ để giải quyết các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

### **2.5.4. Thực nghiệm**

#### **2.5.4.1. Chuẩn hoá dữ liệu**

Trước khi vào xây dựng mô hình thì ảnh trong tập dữ liệu cần được xử lý với các bước sau:

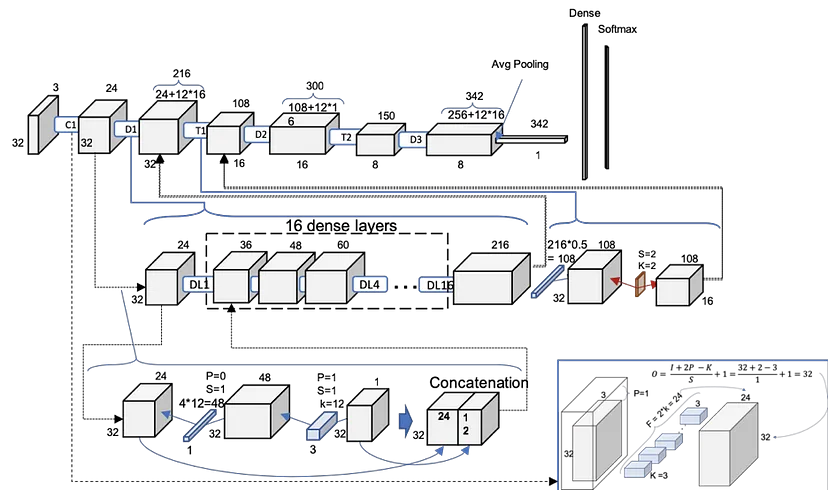
* Resize ảnh: Để đưa ảnh về dạng 32x32
* Transformation ảnh: Áp dụng các cách transformation khác nhau trên ảnh như xoay, lật, lọc màu, thay đổi độ sắc nét của ảnh, ...
* Normalization ảnh: Để giảm thiểu sự khác biệt về illumination.
* Add noise ảnh: Thêm ngẫu nhiên nhiễu vào ảnh để mô hình có thể khái quát tốt hơn.

#### **2.5.4.2. Huấn luyện mô hình**

Tập huấn luyện gồm 60000 ảnh được chia làm 3 phần, 45000 ảnh dùng cho tập train, 5000 ảnh dùng cho tập val và 10000 ảnh cho tập test.

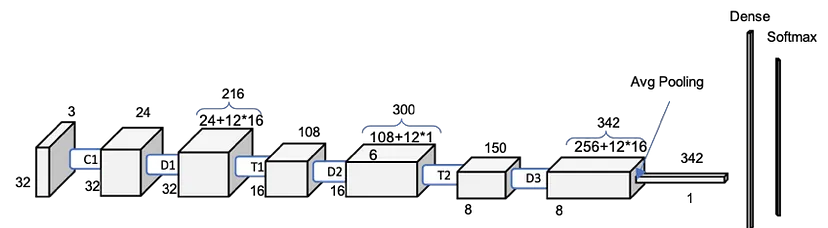
##### **2.5.4.2.1. Mô hình teacher**

Theo hình 22: Mô hình teacher là mô hình pretrained DenseNet-BC (k = 12, depth = 100). Mô hình sử dụng phương pháp optimizer = ’SGD’, learning rate = 0.1 và được train trên 182 epoch.



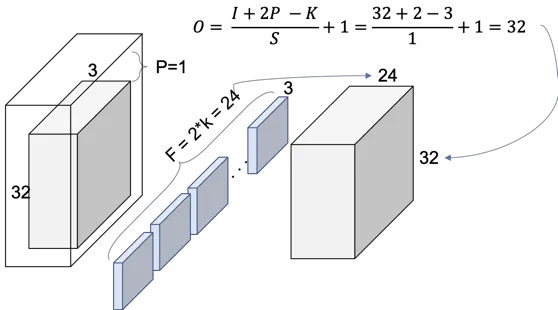
Hình . Kiến trúc mạng mô hình Teacher

Kiến trúc mô hình DenseNet-BC



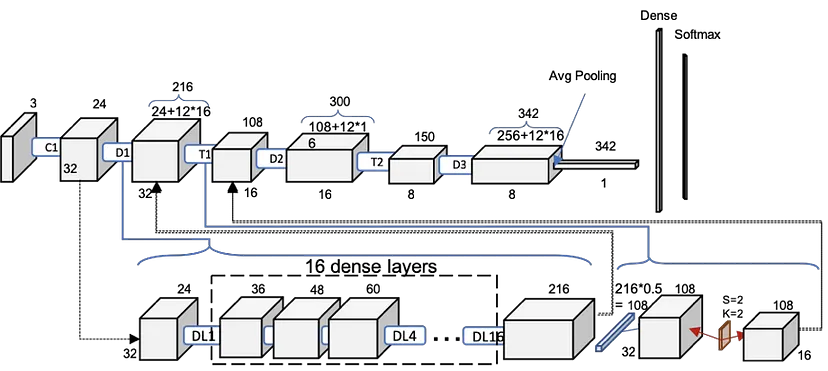
Hình . Kiến trúc của Densenet trên tập CIFAR-10

Theo hình 23, Kiến trúc của DenseNet-BC gồm 1 lớp convolution, theo sau là 2 cặp dense-block với transition-block, tiếp theo là dense-block thứ 3 và theo sau đó là global average pooling để giảm về 1x1x342. Vector này sẽ cung cấp cho lớp dense.



Hình . Conv1

Theo hình 24, Sử dụng lớp 3x3 convolution với batch normalization, với padding và stride = 1.



Hình . Cái nhìn sâu hơn về Dense-100-1. Dense Block và Transition Block

Các Dense Block được cấu tạo bởi 16 lớp Dense. Số Feature Maps tăng theo tốc độ 24, 36, 48, ở lớp Dense Block 1 cuối cùng là 216

Transition Block nằm giữa 2 dense-block để giảm feature map xuống 1 nửa với theta = 0.5. Giảm size feature map xuống 1 nửa với stride = 2, kernel size = 2 và padding = 1.

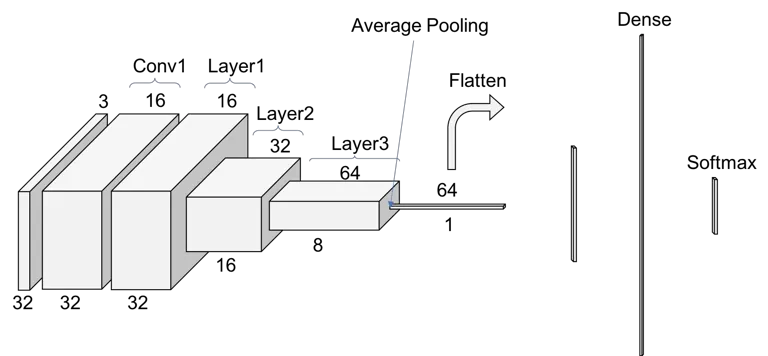
Mô hình teacher với số lượng tham số khoảng: 0.769 triệu tham số

Mô hình DenseNet-BC (k = 12, depth = 100) là pretrained model.

Kết quả của mô hình teacher đánh giá trên tập val với Accuracy = 0.967

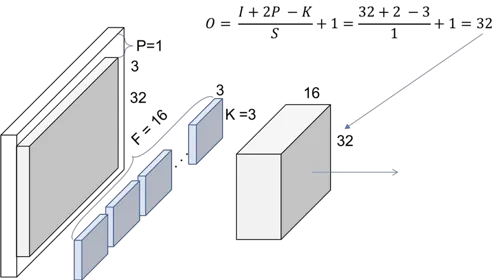
##### **2.5.4.2.2. Mô hình student không sử dụng distillation loss**

Theo hình 26: Mô hình student là mô hình Resnet-20. Mô hình sử dụng phương pháp optimizer = SGD, learning\_rate = 0.1 và được train trên 10 epoch.



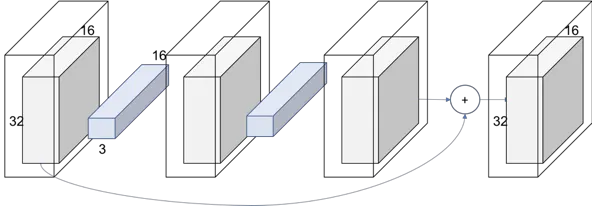
Hình . Kiến trúc mạng mô hình Student

Kiến trúc mô hình Resnet-20



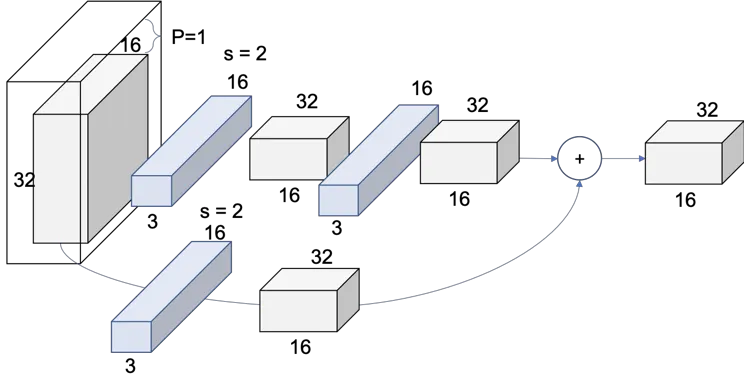
Hình . Conv1

Đưa qua lớp conv 3x3. Với Stride = 1, Padding = 1



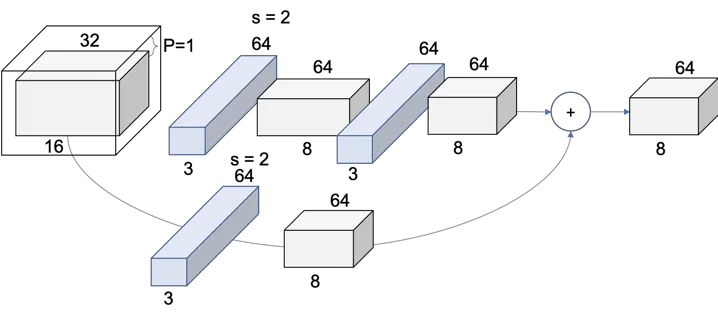
Hình . Layer 1

* Sử dụng 6 lớp Conv 3x3
* Feature Maps có kích thước 32, 16, 8 với 2 lớp Conv cho mỗi Feature maps. Ngoài ra số Filter là 16, 32, 64
* Down Sampling đạt được khi tăng Stride lên 2



Hình . Layer 2, Block 1, Conv

Layer 2 và 3 tương tự Layer 1, ngoại trừ Stride = 2, Padding = 1



Hình . Layer 2

Mô hình student với số lượng tham số khoảng: 0.27 triệu tham số

Mô hình student (không sử dụng distillation loss) được train bằng google colab với thời gian khoảng 0.98 giờ.

Kết quả của mô hình student (không sử dụng distillation loss) đánh giá trên tập val với accuracy = 0.91

##### **2.5.4.2.3. Mô hình student sử dụng distillation loss**

Mô hình student khi sử dụng distillation sẽ sử dụng phương pháp optimizer = ’SGD’ với temperature = 4 và được train trên 182 epoch.

Mô hình student (sử dụng distillation loss) được train bằng google colab với thời gian khoảng 2.4 giờ.

Kết quả của mô hình student (khi sử dụng distillation loss) trên tâp val với accuracy = 0.925.

#### **2.5.3.3. Đánh giá mô hình**

Đối với bài toán này thì độ đo được sử dụng là accuracy.

Bảng so sánh mô hình student (khi không sử dụng distillation loss) với mô hình teacher và mô hình student (khi sử dụng distillation loss)

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Số lượng tham số | Thời gian huấn luyện | Val Accuracy |
| Teacher | 0.769 triệu tham số | Pretrained model | 0.967 |
| Student (không sử dụng distillation loss) | 0.27 triệu tham số | 1.5 giờ | 0.913 |
| Student (sử dụng distillation loss) | 0.27 triệu tham số | 2.5 giờ | 0.925 |

Sau khi áp dụng distillation loss lên mô hình student thì accuracy đạt được trên tập test xấp xỉ 0.93. Đây là một kết quả khá tốt, accuracy tốt hơn mô hình student khi không áp dụng distillation loss và xấp xỉ với mô hình teacher dù tham số của mô hình student nhỏ hơn rất nhiều so với mô hình teacher. Từ đó có thể thấy knowledge distillation rất hiểu quả trong việc nén mô hình nhưng vẫn đảm bảo được độ chính xác cao.

# **3. Kết luận**

## **3.1. Kết quả đạt được**

Sau quá trình tìm hiểu Knowledge Distillation và ứng dụng trên bộ CIFAR-10. Kết quả đạt được của mô hinh Student khi áp dụng Distillation Loss sẽ cho độ chinh xác cao hơn so với khi không sử dụng, và cao xấp xỉ mô hình Teacher. Từ đó có thể thấy phuong pháp Knowledge Distillation hoạt động khá hiệu quả trong việc thu gọn mô hình nhưng vẫn đảm bảo độ chính xác cao.

Về mặt lý thuyết:

* Hiểu rõ về Deep Learning, Neural Network, bài toán phân loại ảnh
* Hiểu rõ về các tính chất, đặc trưng của Knowledge Distillation.

Về mặt ứng dụng:

* Áp dụng được Knowledge Distillation vào bài toán phân loại ảnh giúp tăng độ chính xác của mô hình có kích thước nhỏ.

## **3.2. Hướng phát triển**

Dựa trên cơ sở và kiến thức có sẵn, trong tương lai tôi sẽ áp dụng Knowledge Distillation vào các bài toán phức tạp hơn, đặc biệt là mảng Computer Vision

**TÀI LIỆU THAM KHẢO**

[1] Geoffrey Hinton el al., “Distilling the Knowledge in a Neural Network”, arXiv preprint arXiv:1503.02531, 2015.

[2] Lin Wang and Kuk-Jin Yoon, “Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[3] Jianping Gou et al., “Knowledge Distillation: A Survey, International Journal of Computer Vision”, 2021.

[4] Geoffrey Hinton, Oriol Vinyals and Jeff Dean. “Distilling the Knowledge in a Neural Network” (Deep Learning and Representation Learning Workshop, 2014

[5] Pengguang Chen, Shu Liu, Hengshuang Zhao, Jiaya Jia. “Distilling Knowledge via Knowledge Review”, 2021

[6] H. Li, “Exploring knowledge distillation of Deep neural nets for efficient hardware solutions,” CS230 Report, 2018

[7] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. Fitnets: “Hints for thin deep nets”. arXiv preprint arXiv:1412.6550, 2014

[8] DenseNet on CIFAR10 [Online]. Available: https://towardsdatascience.com/densenet-on-cifar10-d5651294a1a8

[9] ResNets for CIFAR-10 [Online]. Available: https://towardsdatascience.com/resnets-for-cifar-10-e63e900524e0

[10] Image Data Augmentation for Computer Vision (2023 Guide) [Online]. Available: https://viso.ai/computer-vision/image-data-augmentation-for-computer-vision/

[11] Knowledge distillation recipes [Online]. Available: https://keras.io/examples/keras\_recipes/better\_knowledge\_distillation/