

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

**Xây dựng giải pháp chuyển đổi âm
thanh thành văn bản**

Giảng viên hướng dẫn: THS. TRẦN CHÂU THANH THIỆN
Sinh viên thực hiện: ĐẶNG THANH PHÚC
MSSV: 2100009466
Khoá: 2021
Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO

Tp HCM, tháng 12 năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

**Xây dựng giải pháp chuyển đổi âm
thanh thành văn bản**

Giảng viên hướng dẫn: THS. TRẦN CHÂU THANH THIỆN
Sinh viên thực hiện: ĐẶNG THANH PHÚC
MSSV: 2100009466
Khoá: 2021
Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO

Tp HCM, tháng 12 năm 2024

LỜI MỞ ĐẦU

Trong bối cảnh phát triển mạnh mẽ của công nghệ thông tin và trí tuệ nhân tạo, việc tối ưu hóa các công cụ giao tiếp giữa con người và máy móc đang ngày càng trở thành một nhu cầu thiết yếu. Một trong những lĩnh vực nổi bật trong sự phát triển này là chuyển đổi âm thanh thành văn bản, nơi các hệ thống máy tính có thể hiểu và xử lý ngôn ngữ nói một cách tự nhiên. Công nghệ này không chỉ mở ra những tiềm năng to lớn trong đời sống thường nhật mà còn mang lại hiệu quả cao trong nhiều lĩnh vực, từ giáo dục, kinh doanh, chăm sóc sức khỏe, đến các ứng dụng chuyên biệt như trợ lý ảo hay hỗ trợ người khuyết tật.

Chuyển đổi âm thanh thành văn bản không chỉ giúp tối ưu hóa quá trình nhập liệu, tiết kiệm thời gian, mà còn mở ra cơ hội để nâng cao trải nghiệm người dùng trong các hệ thống thông minh. Tuy nhiên, bài toán này không hề đơn giản, bởi nó đòi hỏi hệ thống phải xử lý được sự phức tạp của âm thanh trong các môi trường đa dạng, sự khác biệt về giọng nói, ngữ điệu, và từ vựng. Đặc biệt, với tiếng Việt – một ngôn ngữ đơn âm nhưng giàu thanh điệu, thách thức càng trở nên rõ rệt hơn khi tích hợp các hệ thống này vào thực tế.

Với những tiềm năng to lớn và tính ứng dụng cao, đề tài “Xây dựng giải pháp chuyển đổi âm thanh thành văn bản” được thực hiện với mục tiêu nghiên cứu và phát triển một giải pháp hiệu quả, có khả năng áp dụng trong các ngữ cảnh đa dạng, đặc biệt là trong môi trường sử dụng tiếng Việt. Đề tài tập trung vào nghiên cứu mô hình Conformer, là mô hình có một kiến trúc tiên tiến kết hợp ưu điểm của mạng nơ-ron tích chập và Transformer, giúp cải thiện khả năng nắm bắt ngữ cảnh ngắn hạn lẫn dài hạn trong tín hiệu âm thanh.

LỜI CẢM ƠN

Lời đầu tiên, cho phép em bày tỏ lòng biết ơn sâu sắc đến thày ThS. Trần Châu Thanh Thiện, người đã trực tiếp hướng dẫn, dìu dắt và tận tình giúp đỡ em trong suốt quá trình thực hiện khóa luận này. Không chỉ truyền đạt kiến thức chuyên môn, thày còn là nguồn động viên tinh thần to lớn, giúp em vững bước trên con đường học tập. Từ những hỗ trợ về tài liệu, kiến thức đến những lời khuyên quý báu, thày đã góp phần không nhỏ vào việc hoàn thiện nghiên cứu của em.

Em cũng xin gửi lời tri ân chân thành đến quý thày cô khoa Công nghệ Thông tin, trường Đại học Nguyễn Tất Thành. Sự tận tụy giảng dạy, truyền đạt kiến thức từ cơ bản đến chuyên sâu của quý thày cô trong suốt những năm học tập tại trường là hành trang vô giá cho em trên con đường nghiên cứu khoa học.

Mặc dù đã nỗ lực hết mình, song với kiến thức và kinh nghiệm còn hạn chế, khóa luận chắc chắn không tránh khỏi những thiếu sót. Kính mong quý thày cô lượng thứ và đóng góp ý kiến quý báu để em có thể hoàn thiện nghiên cứu này hơn nữa.

Sinh viên thực hiện

Đặng Thanh Phúc

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Hình thức (Bố cục, trình bày, lỗi, các mục, hình, bảng, công thức, phụ lục,)

2. Nội dung (mục tiêu, phương pháp, kết quả, sao chép, các chương, tài liệu,...)

.....

.....

.....

.....

- ### 3. Kết luận

TPHCM, Ngày tháng năm 2024

Giáo viên hướng dẫn

(Ký tên, ghi rõ họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

1. Hình thức (Bố cục, trình bày, lỗi, các mục, hình, bảng, công thức, phụ lục,)

2. Nội dung (mục tiêu, phương pháp, kết quả, sao chép, các chương, tài liệu,...)

3. Kết luận

.....

TPHCM, Ngày tháng năm 2024

Giáo viên phản biện

(Ký tên, ghi rõ họ tên)

MỤC LỤC

LỜI MỞ ĐẦU	i
LỜI CẢM ƠN	ii
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	iii
NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN	iv
MỤC LỤC.....	v
DANH MỤC HÌNH	vii
KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT	ix
CHƯƠNG 1 GIỚI THIỆU VỀ ĐỀ TÀI.....	1
1.1. Lý do chọn đề tài	1
1.2. Mục tiêu nghiên cứu	1
1.3. Đối tượng nghiên cứu	1
1.4. Phạm vi nghiên cứu	2
1.5. Bố cục đề tài	3
CHƯƠNG 2 CƠ SỞ LÝ LUẬN	4
2.1. Các nghiên cứu liên quan	4
2.2. Phương pháp nghiên cứu	5
2.3. Vấn đề cần giải quyết	7
2.4. Giải pháp.....	7
CHƯƠNG 3 MÔ HÌNH LÝ THUYẾT.....	9
3.1. Trí tuệ nhân tạo.....	9
3.1.1. Giới thiệu về Trí tuệ nhân tạo.....	9
3.1.2. Lịch sử phát triển của trí tuệ nhân tạo	10
3.1.3. Các phương pháp và thuật toán trong AI	12
3.1.4. Ứng dụng thực tiễn	14
3.1.5. Thách thức và cơ hội của AI	15
3.1.6. Tương lai của AI.....	16
3.2. Deep learning.....	16
3.2.1. Deep Learning là gì?	16
3.2.2. Cách thức Deep Learning hoạt động	17
3.2.3. Các ứng dụng của Deep Learning	21
3.2.4. Phân biệt Deep Learning và Machine Learning	27
3.3. Automatic Speech Recognition	31
3.3.1. Tổng quan	31
3.3.2. Lịch sử hình thành	32

3.3.3. Kiến trúc	34
3.3.4. Cách hoạt động	41
3.3.5. Các phương pháp chính trong ASR.....	42
3.4. Transformer	43
3.4.1. Sơ lược.....	43
3.4.2. <i>Lớp chú ý</i>	44
3.4.3. Encoder và Decoder	46
3.4.4. Encoder.....	47
3.4.5. Positional Encoding.....	47
3.4.6. Self-Attention	48
3.4.7. Decoder.....	50
3.4.8. Masked multi-head attention	50
3.4.9. Optimizer	51
3.4.10. Tokenizer.....	51
3.5. Conformer.....	52
3.5.1. Giới thiệu.....	52
3.5.2. Multi-headed Self Attention Module	53
3.5.3. Convolution Module.....	53
3.5.4. Feed Forward Module	54
3.5.5. Conformer Block	54
CHƯƠNG 4 MÔ HÌNH THỰC NGHIỆM	57
4.1. Chuẩn bị dữ liệu	57
4.1.1. Thu thập dữ liệu.....	57
4.1.2. Tiền xử lý dữ liệu	59
4.1.3. Chuẩn bị dữ liệu huấn luyện.....	60
4.2. Xây dựng mô hình.....	61
4.2.1. Lựa chọn mô hình.....	61
4.2.2. Huấn luyện mô hình	62
4.2.3. Đánh giá mô hình	64
4.3. Xây dựng giao diện và tích hợp mô hình	66
CHƯƠNG 5 KẾT LUẬN VÀ KIẾN NGHỊ	69
5.1. Kết luận	69
5.2. Kiến nghị và hướng phát triển.....	69
DANH MỤC TÀI LIỆU THAM KHẢO	70

DANH MỤC HÌNH

Hình 1. 1 Công nghệ chuyển đổi âm thanh thành văn bản	1
Hình 1. 2 Phạm vi nghiên cứu.....	3
Hình 2. 1 Bài báo nói về Conformer.....	4
Hình 2. 2 Cấu trúc mô hình tra Transformer.....	5
Hình 2. 3 Hiệu suất của mô hình Conformer so với các mô hình khác	6
Hình 3. 1 Trí tuệ nhân tạo AI là gì?	9
Hình 3. 2 Lịch sử phát triển của trí tuệ nhân tạo.....	10
Hình 3. 3 Dòng thời gian của trí tuệ nhân tạo	11
Hình 3. 4 Chat AI	12
Hình 3. 5 Vị trí của trí tuệ nhân tạo.....	13
Hình 3. 6 Ứng dụng của trí tuệ nhận tạo	15
Hình 3. 7 Deep learning là gì	16
Hình 3. 8 Cách thức Deep Learning hoạt động.....	17
Hình 3. 9 Kiến trúc mạng nơ-ron	18
Hình 3. 10 Feedforward	18
Hình 3. 11 Deep learning và con người	20
Hình 3. 12 Ứng dụng xe tự lái.....	21
Hình 3. 13 Ứng dụng trong Nhận dạng giọng nói	22
Hình 3. 14 Ứng dụng trong nhận dạng ảnh.....	23
Hình 3. 15 Ứng dụng trong tự động gọi ý.....	24
Hình 3. 16 Ứng dụng trong dịch máy	25
Hình 3. 17 Ứng dụng trong lĩnh vực y tế	26
Hình 3. 18 Mạng nơ-ron tích chập	28
Hình 3. 19 Học tăng cường sâu.....	30
Hình 3. 20 Automatic Speech Recognition.....	31
Hình 3. 21 Lịch sử hình thành và phát triển.....	32
Hình 3. 22 Sơ đồ kiến trúc của bộ nhận dạng giọng nói bộ mã hóa-giải mã.....	35
Hình 3. 23 Quá trình giải mã.....	37
Hình 3. 24 Kết hợp CTC and Encoder-Decoder	39

Hình 3. 25 Đầu ra của bộ mã hóa âm thanh.....	40
Hình 3. 26 Kiến trúc đầy đủ của Transformers.....	44
Hình 3. 27 Kiến trúc tự chú ý đa đầu (Multi-head self-Attention)	45
Hình 3. 28 Kiến trúc Scaled Dot-Product Attention	46
Hình 3. 29 Input Embedding	47
Hình 3. 30 Position in sentence.....	48
Hình 3. 31 vector Q, K, V	49
Hình 3. 32 Multi-head Attention.....	50
Hình 3. 33 Label Smoothing	51
Hình 3. 34 Multi-headed Self Attention Module	53
Hình 3. 35 Convolution Module	54
Hình 3. 36 Kiến trúc của mô đun Feed Forward.....	54
Hình 3. 37 Kiến trúc Conformer Block.....	55
Hình 4. 1 Phân phối độ dài tệp âm thanh	58
Hình 4. 2 Phân bố số lượng từ.....	58
Hình 4. 3 Thông tin về dữ liệu	59
Hình 4. 4 Chuẩn hóa âm thanh.....	60
Hình 4. 5 Cấu trúc thư mục của bộ dữ liệu	60
Hình 4. 6 Phân chia dữ liệu	61
Hình 4. 7 Cấu trúc huấn luyện mô hình	63
Hình 4. 8 Train 1/50 epoch	64
Hình 4. 9 Train 50/50 epoch	64
Hình 4. 10 Biểu đồ loss của 50 epoch.....	65
Hình 4. 11 Biểu đồ tỉ lệ lỗi từ và lỗi ký tự theo 50 epoch.....	65
Hình 4. 12 Giao diện chính	66
Hình 4. 13 Giao diện khi thu âm	67
Hình 4. 14 Giao diện khi chọn file từ máy tính	67
Hình 4. 15 Trả về kết quả sau khi xử lý	68

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
AI	Artificial Intelligence
ASR	Automatic Speech Recognition
CER	Character Error Rate
CNN	Convolutional Neural Networks
DNNs	Deep Neural Networks
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
RNNs	Recurrent Neural Networks
WER	Word Error Rate

CHƯƠNG 1 GIỚI THIỆU VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

1.2. Mục tiêu nghiên cứu

Nghiên cứu này nhằm mục tiêu xây dựng một giải pháp chuyển đổi âm thanh thành văn bản với độ chính xác cao, được tối ưu hóa cho ngôn ngữ tiếng Việt. Cụ thể, nghiên cứu tập trung vào:

Phát triển hệ thống nhận diện giọng nói (ASR) hiệu quả:

- Ứng dụng các công nghệ tiên tiến như trí tuệ nhân tạo (AI) và học sâu (Deep Learning) để xử lý tín hiệu âm thanh và chuyển đổi thành văn bản.
- Xây dựng mô hình âm học và ngôn ngữ đặc thù cho tiếng Việt, xử lý được các yếu tố như thanh điệu, ngữ âm phức tạp và giọng nói đa vùng miền.

Đảm bảo tính ứng dụng thực tiễn:

- Hệ thống phải hoạt động hiệu quả trong các môi trường đa dạng, bao gồm môi trường có tạp âm.
- Văn bản đầu ra cần rõ ràng, dễ đọc với dấu câu và ngữ pháp được xử lý tự động, nhằm đáp ứng nhu cầu thực tế trong các lĩnh vực như giáo dục, y tế, hành chính công và dịch vụ khách hàng.

Tăng cường hỗ trợ xã hội và phát triển cộng đồng:

- Đưa ra giải pháp hỗ trợ người khuyết tật hoặc những đối tượng gặp khó khăn trong giao tiếp, giúp họ tiếp cận thông tin dễ dàng hơn.
- Góp phần thúc đẩy việc ứng dụng công nghệ AI vào các hoạt động kinh tế - xã hội, tạo ra giá trị gia tăng và nâng cao hiệu quả lao động.

1.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này là công nghệ chuyển đổi âm thanh thành văn bản (Speech-to-Text), đặc biệt tập trung vào ngôn ngữ tiếng Việt. Cụ thể, nghiên cứu sẽ hướng đến việc phát triển một hệ thống Automatic Speech Recognition (ASR) có khả năng nhận diện và chuyển đổi giọng nói tiếng Việt thành văn bản một cách chính xác và hiệu quả.

Các yếu tố cụ thể trong đối tượng nghiên cứu bao gồm:

Dữ liệu âm thanh tiếng Việt:

- Các nguồn dữ liệu âm thanh sẽ được thu thập từ nhiều tình huống thực tế khác nhau, bao gồm các cuộc hội thoại, bài giảng, thuyết trình, và các cuộc gọi trong dịch vụ khách hàng.
- Dữ liệu này sẽ bao gồm nhiều giọng nói từ các vùng miền khác nhau và đa dạng về độ tuổi, giới tính, và ngữ cảnh, giúp mô phỏng các điều kiện môi trường và ngữ âm đa dạng của tiếng Việt.

Mô hình nhận diện giọng nói tiếng Việt:

- Nghiên cứu tập trung xây dựng và tối ưu hóa các mô hình học sâu (Deep Learning), đặc biệt là các kiến trúc hiện đại như **Conformer** và **Transformer**, nhằm nâng cao độ chính xác trong việc nhận diện và chuyển đổi tín hiệu âm thanh tiếng Việt thành văn bản.
- Các yếu tố đặc thù của tiếng Việt, bao gồm hệ thống âm vị phức tạp và sự khác biệt về thanh điệu, sẽ được xử lý kỹ lưỡng trong mô hình. Điều này đòi hỏi sự tích hợp hiệu quả giữa việc trích xuất đặc trưng âm thanh (acoustic features) và hiểu ngữ cảnh ngôn ngữ (language context).

Đánh giá và cải thiện độ chính xác hệ thống:

- Đối tượng nghiên cứu cũng bao gồm việc cải tiến và đánh giá độ chính xác của hệ thống Automatic Speech Recognition đối với tiếng Việt, đặc biệt là khả năng nhận diện giọng nói trong điều kiện có tiếng ồn hoặc âm thanh không rõ ràng.
- Các phương pháp đánh giá độ chính xác, tốc độ xử lý, và tính ổn định của hệ thống sẽ được triển khai để đảm bảo hiệu quả của hệ thống trong các ứng dụng thực tế.

1.4. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào phát triển hệ thống chuyển đổi âm thanh thành văn bản cho ngôn ngữ tiếng Việt

Nghiên cứu sẽ phát triển giải pháp chuyển đổi âm thanh thành văn bản cho tiếng Việt, một ngôn ngữ có hệ thống thanh điệu và ngữ âm phức tạp.



Hình 1. 2 Phạm vi nghiên cứu

Dữ liệu âm thanh sẽ được thu thập từ các cuộc hội thoại, thuyết trình, bài giảng và các cuộc gọi dịch vụ khách hàng, chủ yếu từ nhiều vùng miền khác nhau của Việt Nam để đảm bảo tính đa dạng.

Nghiên cứu sẽ thực hiện trong khoảng thời gian 2 tháng với nguồn lực hạn chế, chủ yếu phát triển mô hình thử nghiệm và đánh giá ban đầu.

1.5. Bố cục đề tài

Nội dung báo cáo gồm 5 chương:

Chương 1: Giới thiệu về đề tài.

Chương 2: Cơ sở lý luận.

Chương 3: Mô hình lý thuyết

Chương 4: Mô hình thực nghiệm.

Chương 5: Kết luận và kiến nghị.

CHƯƠNG 2 CƠ SỞ LÝ LUẬN

2.1. Các nghiên cứu liên quan

Công nghệ chuyên đổi âm thanh thành văn bản (Speech-to-Text) đã trải qua nhiều giai đoạn phát triển mạnh mẽ nhờ sự tiến bộ vượt bậc trong lĩnh vực học máy (Machine Learning) và học sâu (Deep Learning). Các mô hình nhận diện giọng nói (ASR - Automatic Speech Recognition) truyền thống sử dụng các thuật toán cơ bản như Hidden Markov Models (HMM) và Gaussian Mixture Models (GMM). Tuy nhiên, những phương pháp này dần bị thay thế bởi các mô hình học sâu có khả năng học các đặc trưng phức tạp hơn từ dữ liệu âm thanh.

Conformer: Convolution-augmented Transformer for Speech Recognition

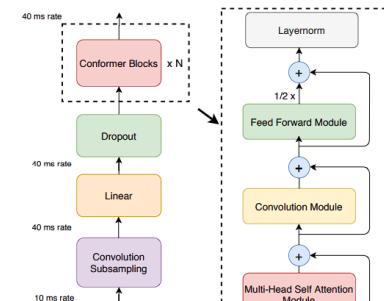
Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang

Google Inc.

{anmolgulati, jamesqin, chungchengc, nikip, ngyuzh, jiahuiyu, weihan, shibow, zhangzd, yonghui, rpang}@google.com

Abstract

Recently Transformer and Convolution neural network (CNN) based models have shown promising results in Automatic Speech Recognition (ASR), outperforming Recurrent neural networks (RNNs). Transformer models are good at capturing content-based global interactions, while CNNs exploit local features effectively. In this work, we achieve the best of both worlds by studying how to combine convolution neural networks and transformers to model both local and global dependencies of an audio sequence in a parameter-efficient way. To this regard, we propose the convolution-augmented transformer for speech recognition, named *Conformer*. *Conformer* significantly outperforms the previous Transformer and CNN



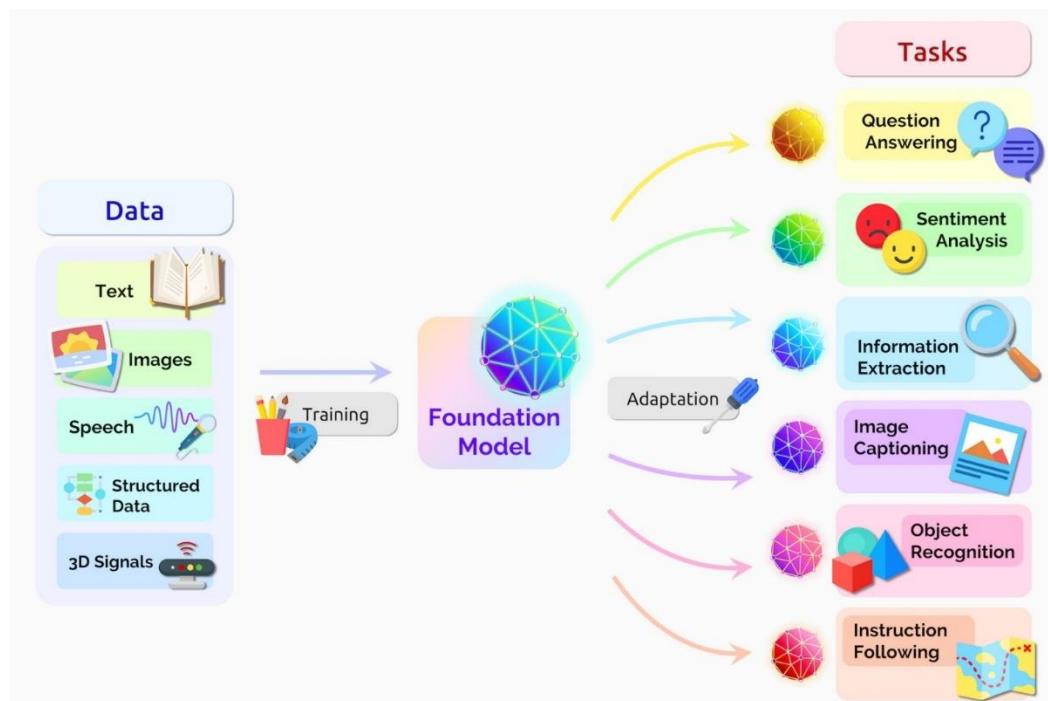
Hình 2. 1 Bài báo nói về Conformer

Một trong những bước đột phá trong lĩnh vực này là sự ra đời của mô hình Deep Neural Networks (DNNs), sau đó là các mô hình Long Short-Term Memory (LSTM) và Recurrent Neural Networks (RNNs). Các mô hình này đã mang lại sự cải thiện lớn về hiệu suất, đặc biệt là trong các tình huống nhận diện giọng nói liên tục và trong môi trường có tiếng ồn nền. Tuy nhiên, mặc dù các mô hình này rất mạnh mẽ trong việc xử lý chuỗi thời gian, nhưng vẫn có một số vấn đề chưa được giải quyết, đặc biệt là khả năng nhận diện chính xác trong các điều kiện phức tạp.

Tiên phong trong việc ứng dụng các mô hình học sâu vào nhận diện giọng nói là nghiên cứu của **Gulati et al. (2020)** với bài báo "**Conformer: Convolution-augmented Transformer for Speech Recognition**". Nghiên cứu này giới thiệu **Conformer**, một mô hình kết hợp giữa Convolutional Neural Networks (CNN) và

Transformer, giúp giải quyết các vấn đề nhận diện giọng nói trong các môi trường phức tạp.

Conformer kết hợp ưu điểm của CNN trong việc xử lý các đặc trưng tần số ngắn hạn và Transformer trong việc xử lý các mối quan hệ dài hạn của tín hiệu âm thanh. Mô hình này đã đạt được những kết quả ấn tượng trong việc nhận diện giọng nói với độ chính xác cao và hiệu quả hơn so với các mô hình ASR truyền thống, đặc biệt là khi ứng dụng trong các tình huống có tạp âm hoặc giọng nói không rõ ràng. Đây là một cải tiến quan trọng, giúp nâng cao khả năng nhận diện trong các môi trường phức tạp và đa dạng.



Hình 2. 2 Cấu trúc mô hình tra Transformer

Bài báo của Gulati et al. chỉ ra rằng mô hình Conformer có thể cải thiện đáng kể hiệu suất nhận diện giọng nói, đặc biệt trong các tác vụ như nhận diện giọng nói liên tục và nhận diện giọng nói trong môi trường có nhiều tiếng ồn nồn, điều mà các mô hình truyền thống gặp khó khăn.

2.2. Phương pháp nghiên cứu

Mô hình Conformer sử dụng sự kết hợp giữa Convolutional Networks và Transformers để giải quyết các vấn đề trong nhận diện giọng nói. Cụ thể, Convolutional Neural Networks (CNN) giúp mô hình nhận diện các đặc trưng tần số ngắn hạn của âm thanh, trong khi Transformer xử lý các mối quan hệ dài hạn trong

chuỗi tín hiệu âm thanh. Phương pháp này giúp cải thiện hiệu suất hệ thống nhận diện giọng nói trong môi trường có tạp âm và tiếng ồn nền.

Convolutional Neural Networks (CNN):

CNN giúp nhận diện các đặc trưng ngắn hạn của tín hiệu âm thanh, như các mẫu tần số. Các lớp convolution trong Conformer cho phép mô hình tập trung vào việc phát hiện các đặc trưng quan trọng trong âm thanh đầu vào, chẳng hạn như sự thay đổi về tần số và cường độ của âm thanh theo thời gian. Điều này rất quan trọng trong việc xử lý giọng nói có sự biến đổi lớn về tần số, chẳng hạn như khi người nói thay đổi tốc độ hoặc khi có nhiều tạp âm.



Hình 2. 3 Hiệu suất của mô hình Conformer so với các mô hình khác

Transformer:

Trong khi CNN xử lý các đặc trưng ngắn hạn, Transformer giúp mô hình nhận diện các mối quan hệ dài hạn trong chuỗi âm thanh. Với cơ chế Self-Attention, Transformer cho phép mô hình học được sự phụ thuộc giữa các phần khác nhau trong chuỗi âm thanh, từ đó giúp cải thiện khả năng nhận diện giọng nói dài hạn và liên tục.

Mô hình Conformer:

Conformer kết hợp các ưu điểm của CNN và Transformer để xử lý cả đặc trưng ngắn hạn và dài hạn, giúp nhận diện giọng nói chính xác hơn, đặc biệt trong môi trường có tạp âm. Bài báo của Gulati et al. chỉ ra rằng việc bổ sung các lớp

Convolutional vào Transformer giúp tăng khả năng nhận diện trong các tình huống có tiếng ồn và cải thiện độ chính xác chung của hệ thống.

Lý thuyết về mô hình Transformer đã được chứng minh là rất hiệu quả trong các tác vụ ngôn ngữ tự nhiên (Natural Language Processing - NLP), và nghiên cứu này chứng minh rằng Conformer có thể mang lại những lợi ích tương tự trong lĩnh vực nhận diện giọng nói. Conformer cũng giải quyết được một số vấn đề mà các mô hình truyền thống như Deep Neural Networks (DNNs) hoặc Long Short-Term Memory (LSTM) không thể xử lý hiệu quả, đặc biệt khi dữ liệu âm thanh trở nên phức tạp.

2.3. Vấn đề cần giải quyết

Mặc dù mô hình Conformer đã đạt được nhiều thành công trong nhận diện giọng nói, nhưng các nghiên cứu hiện tại vẫn chưa hoàn toàn giải quyết được các vấn đề đặc thù của ngôn ngữ tiếng Việt. Tiếng Việt là một ngôn ngữ có hệ thống thanh điệu phức tạp và giọng điệu đa dạng, điều này tạo ra những thách thức lớn trong việc phát triển các hệ thống nhận diện giọng nói. Ngoài ra, việc áp dụng các mô hình như Conformer trong môi trường có tiếng ồn nền và giọng nói không rõ ràng vẫn là một vấn đề cần giải quyết.

Vấn đề chính cần giải quyết là làm thế nào để áp dụng mô hình Conformer hiệu quả cho tiếng Việt, một ngôn ngữ có nhiều đặc thù về ngữ âm và thanh điệu. Bài toán này đòi hỏi không chỉ cải tiến mô hình nhận diện giọng nói mà còn cần phải xây dựng cơ sở dữ liệu âm thanh đa dạng và phong phú từ nhiều vùng miền khác nhau của Việt Nam.

2.4. Giải pháp

Dựa trên các nghiên cứu trước đây và mô hình Conformer được giới thiệu trong bài báo "**Conformer: Convolution-augmented Transformer for Speech Recognition**", nghiên cứu này đề xuất phát triển một mô hình kết hợp Conformer với các kỹ thuật cải tiến khác để tối ưu hóa nhận diện giọng nói tiếng Việt. Các giải pháp đề xuất bao gồm:

Tối ưu hóa mô hình Conformer cho tiếng Việt:

Phát triển một phiên bản của mô hình **Conformer** phù hợp với các đặc thù của tiếng Việt, bao gồm thanh điệu và các biến thể trong giọng điệu. Mô hình này sẽ được huấn luyện trên cơ sở dữ liệu âm thanh phong phú của tiếng Việt.

Xử lý tạp âm và môi trường có tiếng ồn:

Áp dụng các phương pháp lọc tạp âm và tăng cường dữ liệu để cải thiện khả năng nhận diện giọng nói trong các điều kiện không lý tưởng, như trong các cuộc gọi dịch vụ khách hàng hay trong môi trường ngoài trời có nhiều tiếng ồn.

Xây dựng cơ sở dữ liệu âm thanh đa dạng:

Tạo ra một cơ sở dữ liệu âm thanh tiếng Việt lớn và đa dạng, bao gồm các giọng nói từ nhiều vùng miền khác nhau, để đảm bảo mô hình có thể nhận diện chính xác giọng nói trong các ngữ cảnh khác nhau.

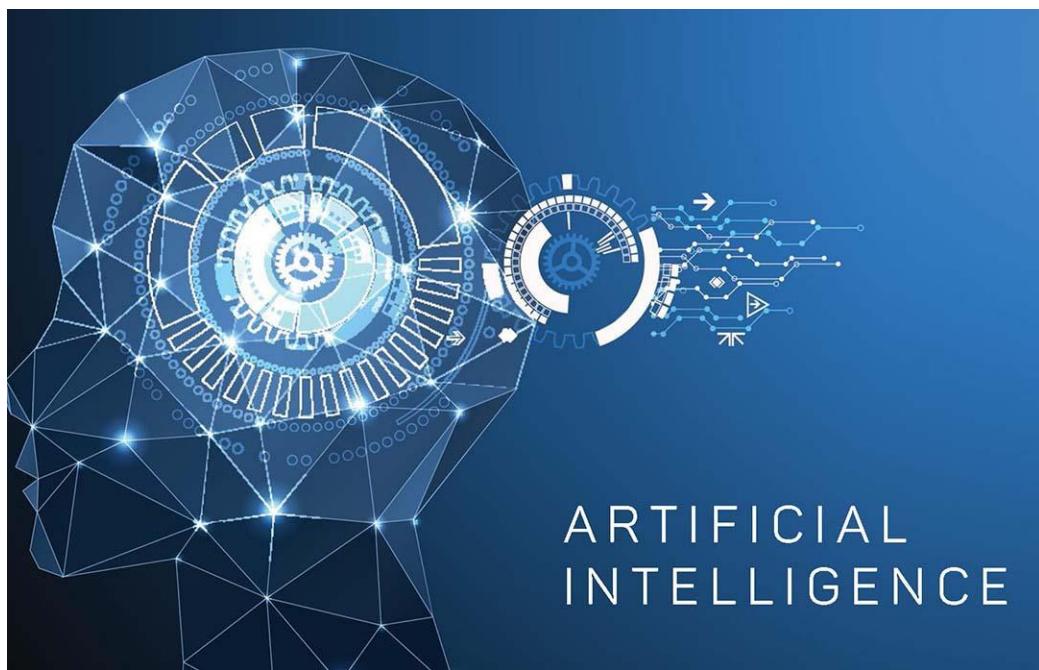
Qua những giải pháp này, nghiên cứu này kỳ vọng sẽ tạo ra một hệ thống nhận diện giọng nói tiếng Việt có độ chính xác cao, có thể ứng dụng trong nhiều lĩnh vực như giáo dục, y tế, và dịch vụ khách hàng.

CHƯƠNG 3 MÔ HÌNH LÝ THUYẾT

3.1. Trí tuệ nhân tạo

3.1.1. Giới thiệu về Trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là một lĩnh vực khoa học liên ngành kết hợp các khái niệm từ khoa học máy tính, toán học, và tâm lý học, với mục tiêu chính là xây dựng các hệ thống có khả năng mô phỏng hành vi thông minh của con người. AI không chỉ đơn giản là một công nghệ, mà còn là một cách tiếp cận nhằm giải quyết các vấn đề phức tạp trong nhiều lĩnh vực khác nhau.



Hình 3. 1 Trí tuệ nhân tạo AI là gì?

Cụ thể hơn, trí tuệ nhân tạo được chia thành các cấp độ:

- **AI hẹp (Weak AI):** Các hệ thống này tập trung vào một tác vụ duy nhất, chẳng hạn như chơi cờ vua, nhận diện hình ảnh, hoặc xử lý ngôn ngữ tự nhiên.
- **AI tổng quát (General AI):** Mục tiêu dài hạn của AI, nơi các hệ thống có thể thực hiện bất kỳ nhiệm vụ trí tuệ nào mà con người có thể làm.
- **AI siêu việt (Superintelligent AI):** Một cấp độ giả định nơi AI vượt qua trí thông minh của con người trong mọi lĩnh vực

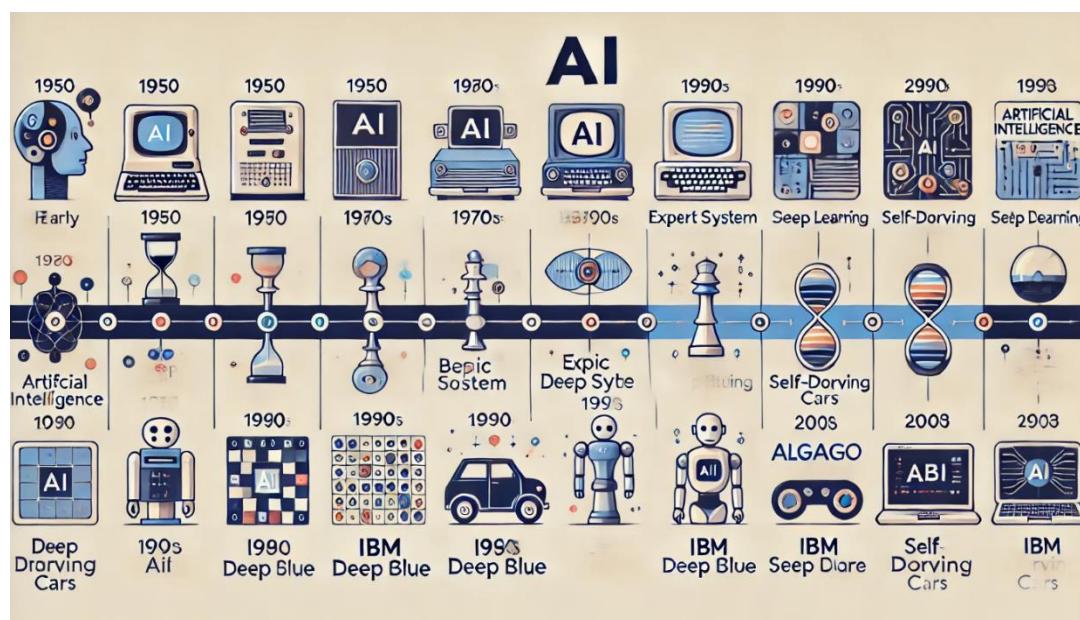
AI xây dựng các thuật toán để xử lý dữ liệu có khả năng tư duy, học hỏi và tương tác với môi trường như con người. Bên cạnh đó, nó mở rộng phạm vi của khoa

học dữ liệu và học máy thông qua việc tích hợp các kỹ thuật tiên tiến, giúp các hệ thống đưa ra quyết định thông minh và tự động hóa nhiều tác vụ phức tạp.

3.1.2. Lịch sử phát triển của trí tuệ nhân tạo

1940–1980: Những bước đi đầu tiên

Trí tuệ nhân tạo (AI) bắt đầu được định hình thông qua các nghiên cứu lý thuyết và thực nghiệm từ những năm 1940. Năm 1943, Warren McCulloch và Walter Pitts đã giới thiệu mô hình tế bào thần kinh nhân tạo, tạo nền tảng cho mạng nơ-ron, công nghệ cốt lõi của AI hiện đại.



Hình 3. 2 Lịch sử phát triển của trí tuệ nhân tạo

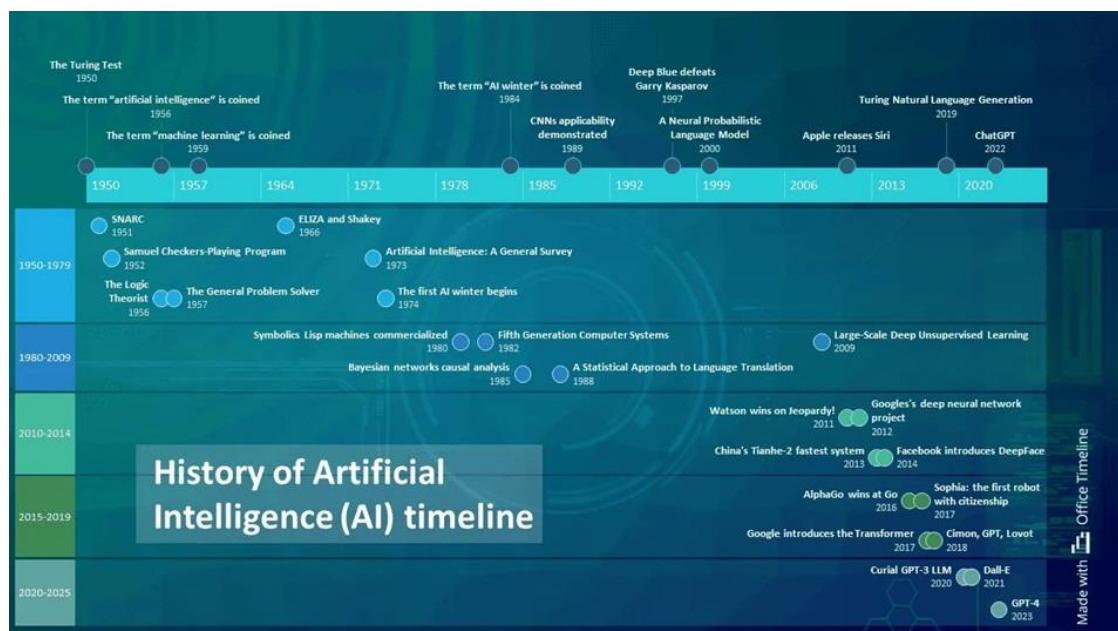
Đến năm 1950, Alan Turing công bố bài báo nổi tiếng “Máy tính và trí tuệ”, trong đó ông đưa ra khái niệm thử nghiệm Turing – một phương pháp đánh giá khả năng tư duy của máy móc. Những năm sau đó, các nhà khoa học khác tiếp tục đạt được nhiều thành tựu, bao gồm:

- Marvin Minsky và Dean Edmonds chế tạo cỗ máy mạng thần kinh đầu tiên có tên SNARC.
- Frank Rosenblatt phát triển mô hình **Perceptron**, một trong những mạng nơ-ron đầu tiên.
- Joseph Weizenbaum sáng tạo **ELIZA**, chatbot mô phỏng phương pháp trị liệu Rogerian vào những năm 1960.

Tuy nhiên, vào cuối thập niên 1960, Marvin Minsky đã chỉ ra những hạn chế của mạng nơ-ron, khiến nghiên cứu AI gặp khó khăn. Điều này, cùng với những rào cản về phần cứng, đã dẫn đến sự suy giảm quan tâm đến AI, khởi đầu cho "mùa đông AI" đầu tiên.

1980–2006: Sự hồi sinh và thách thức mới

Trong thập niên 1980, AI nhận được sự quan tâm trở lại, với nguồn tài trợ từ chính phủ tập trung vào các lĩnh vực như dịch thuật tự động và hỗ trợ quyết định. Các hệ thống chuyên gia như MYCIN, được thiết kế để mô phỏng cách con người ra quyết định trong y học, đã trở nên phổ biến.



Hình 3. 3 Dòng thời gian của trí tuệ nhân tạo

Sự hồi sinh của mạng nơ-ron cũng xuất hiện vào giai đoạn này. Các nhà nghiên cứu như David Rumelhart và John Hopfield đã phát triển các phương pháp học sâu, minh chứng rằng máy tính có thể học từ dữ liệu.

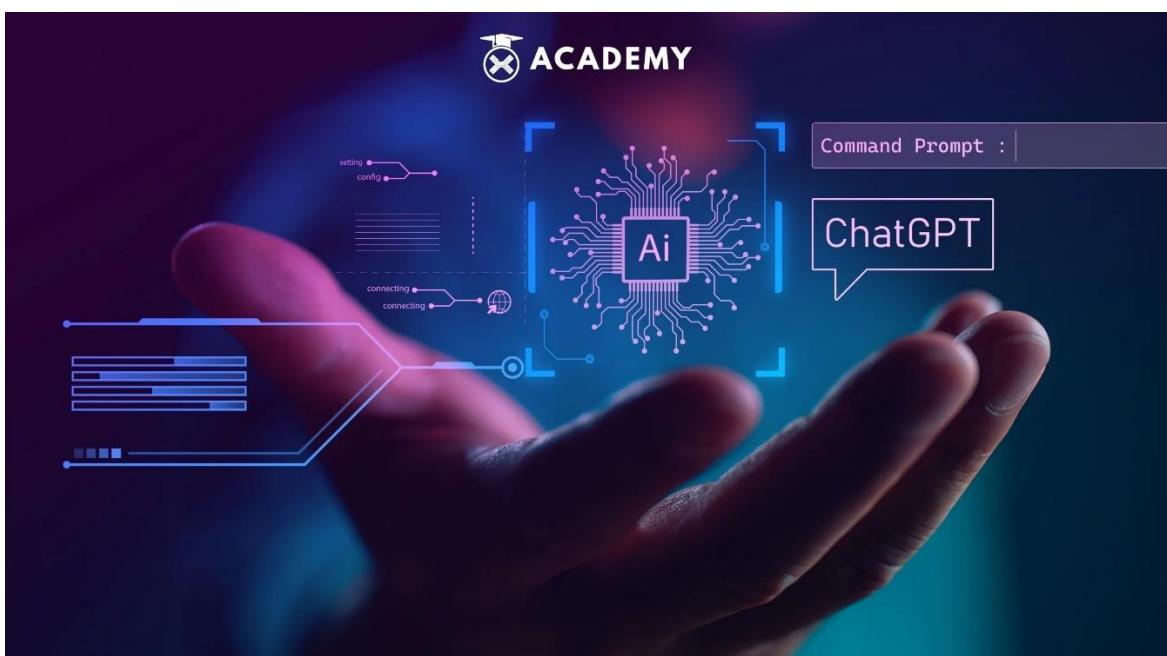
Tuy nhiên, đến cuối những năm 1980 và 1990, do các yếu tố kinh tế và sự bùng nổ của ngành công nghệ thông tin, mùa đông AI thứ hai xảy ra. Nghiên cứu AI trở nên phân tán, với các nhà khoa học tập trung vào các ứng dụng cụ thể.

Bước ngoặt lớn trong lĩnh vực này xảy ra vào năm 1997, khi Deep Blue, hệ thống chơi cờ của IBM, đánh bại nhà vô địch thế giới Garry Kasparov. Đồng thời, Geoffrey Hinton cùng các cộng sự đã làm sống lại mạng nơ-ron, mở đường cho những đột phá trong học sâu.

2007–nay: Kỷ nguyên bùng nổ của trí tuệ nhân tạo

Từ năm 2007, nhờ sự phát triển mạnh mẽ của điện toán đám mây, các công cụ AI trở nên dễ tiếp cận hơn. Điều này thúc đẩy đổi mới trong học máy và học sâu. Một số thành tựu nổi bật bao gồm:

- **AlexNet** (2012), mô hình mạng nơ-ron tích chập (CNN) của Alex Krizhevsky, Ilya Sutskever và Geoffrey Hinton, đã chiến thắng cuộc thi ImageNet, chứng minh sức mạnh của học sâu trong nhận dạng hình ảnh.
- **AlphaZero** (2017) của Google, một hệ thống AI có thể tự học chơi và xuất sắc trong các trò chơi như cờ vua, cờ shogi và cờ vây mà không cần dữ liệu từ con người.



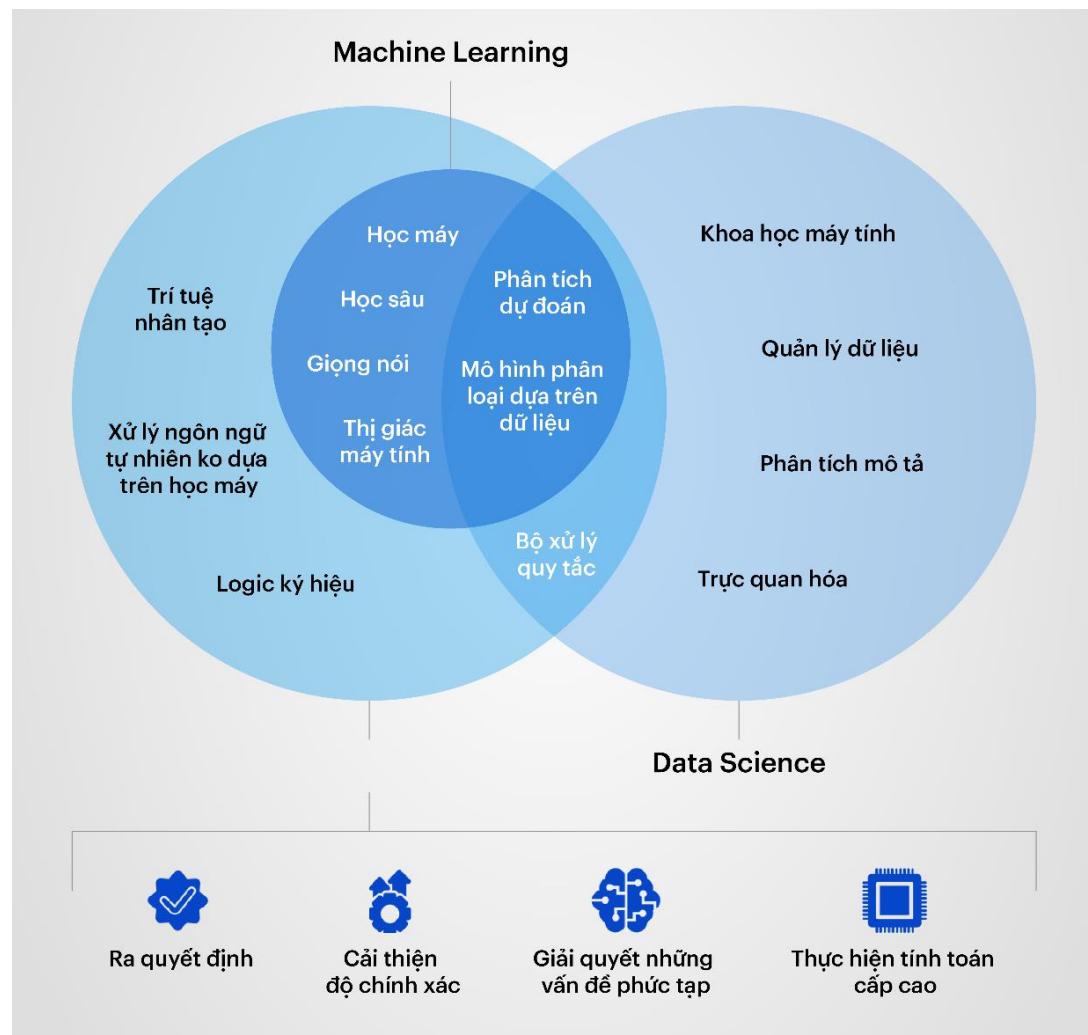
Hình 3. 4 Chat AI

Vào năm 2022, trí tuệ nhân tạo bước vào một làn sóng phát triển mới với sự ra đời của các chatbot như ChatGPT của OpenAI. Sử dụng xử lý ngôn ngữ tự nhiên (NLP), các chatbot này đã mở ra khả năng trò chuyện và thực hiện các nhiệm vụ giống như con người, đánh dấu một bước ngoặt quan trọng trong việc ứng dụng AI vào cuộc sống hàng ngày.

3.1.3. Các phương pháp và thuật toán trong AI

Học máy (**Machine Learning - ML**) là cốt lõi của AI, nơi các hệ thống được thiết kế để học hỏi từ dữ liệu. Điều này giúp AI không cần phải lập trình chi tiết từng

bước, mà thay vào đó học các mẫu hoặc mối quan hệ trong dữ liệu để đưa ra quyết định.



Hình 3. 5 Vị trí của trí tuệ nhân tạo

Học có giám sát (Supervised Learning): Dữ liệu đào tạo có sẵn nhãn, giúp mô hình học cách ánh xạ từ đầu vào đến đầu ra. Ví dụ: dự đoán giá nhà dựa trên dữ liệu giá trong quá khứ.

Học không giám sát (Unsupervised Learning): Mô hình tự tìm kiếm các cấu trúc ẩn trong dữ liệu chưa được gắn nhãn. Ví dụ: phân cụm khách hàng dựa trên hành vi mua sắm.

Học tăng cường (Reinforcement Learning): Tập trung vào việc học từ môi trường thông qua cơ chế "thưởng-phạt". Ví dụ: AI chơi game tự học cách chiến thắng qua thử nghiệm và sai lầm.

Học sâu (Deep Learning) là một nhánh tiên tiến của học máy, sử dụng mạng nơ-ron nhân tạo với nhiều tầng (deep neural networks) để xử lý các dữ liệu lớn và phức tạp.

- **Ưu điểm của học sâu:** Khả năng tự động trích xuất đặc trưng từ dữ liệu thô, giảm sự phụ thuộc vào chuyên gia xử lý dữ liệu.
- **Ứng dụng:** Nhận diện hình ảnh, xử lý giọng nói, dịch máy, và nhiều hơn nữa.

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là lĩnh vực giúp máy tính hiểu và tương tác với ngôn ngữ con người.

- Các bước trong NLP:
 - Tiền xử lý văn bản (loại bỏ dấu câu, chuẩn hóa từ ngữ).
 - Mô hình hóa ngôn ngữ (Language Modeling).
 - Dịch thuật, phân tích cảm xúc, tóm tắt văn bản.
- Các mô hình tiên tiến: GPT, BERT, Transformer, các hệ thống chatbot và trợ lý ảo.

Thị giác máy tính (Computer Vision) là một nhánh AI liên quan đến việc hiểu và phân tích dữ liệu hình ảnh hoặc video.

- Kỹ thuật cơ bản: Nhận diện đối tượng (Object Detection), phân loại hình ảnh (Image Classification), tái tạo hình ảnh (Image Reconstruction).
- Ứng dụng thực tiễn: Xe tự lái, phân tích hình ảnh y khoa, giám sát an ninh.

3.1.4. Ứng dụng thực tiễn

Trong công nghiệp và sản xuất

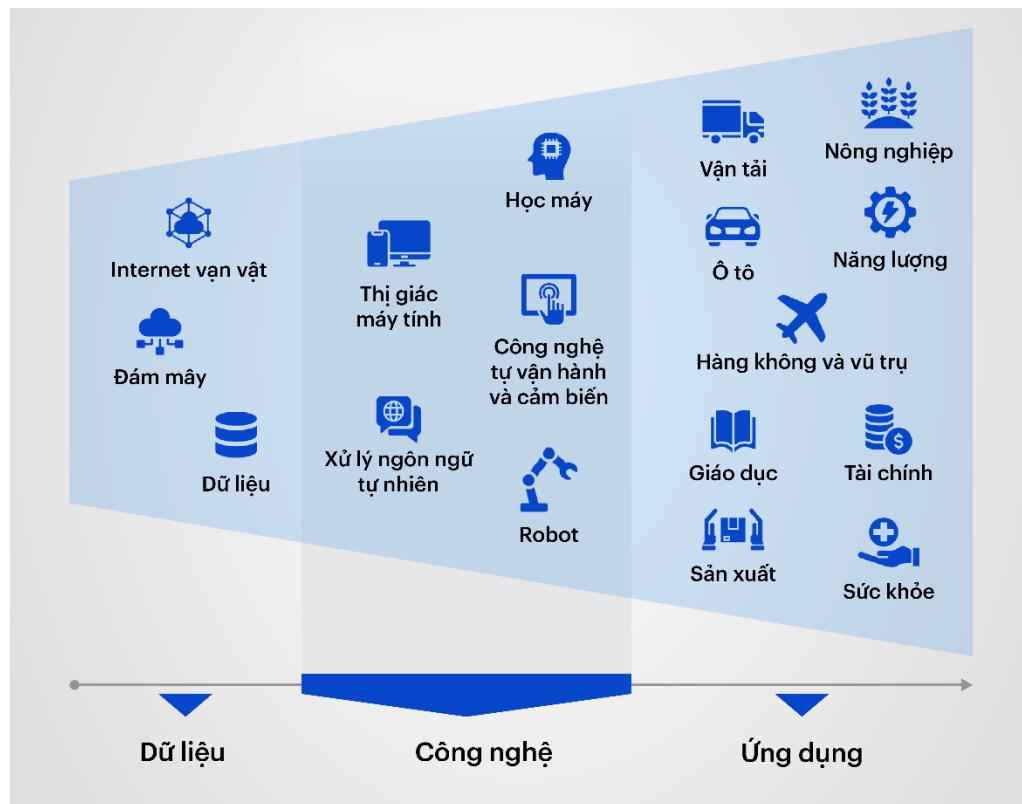
AI hỗ trợ tối ưu hóa quy trình sản xuất, bảo trì dự đoán và giảm chi phí vận hành. Các ứng dụng phổ biến:

- Sử dụng robot thông minh trong dây chuyền sản xuất.
- Phân tích dữ liệu để dự đoán hỏng hóc của máy móc.

Trong chăm sóc sức khỏe

AI hỗ trợ bác sĩ chẩn đoán bệnh nhanh chóng và chính xác hơn. Ví dụ:

- Phân tích hình ảnh X-quang, MRI để phát hiện ung thư sớm.
- Sử dụng chatbot y tế để hỗ trợ bệnh nhân từ xa.



Hình 3. 6 Ứng dụng của trí tuệ nhận tạo

Trong giáo dục

AI được sử dụng để cá nhân hóa quá trình học tập, giúp học sinh tiếp cận kiến thức phù hợp với năng lực của mình.

- Xây dựng hệ thống gia sư ảo.
- Phân tích kết quả học tập để đề xuất lộ trình học phù hợp.

3.1.5. Thách thức và cơ hội của AI

Thách thức

- **Đạo đức và pháp lý:** AI đặt ra câu hỏi về quyền riêng tư, phân biệt đối xử và trách nhiệm pháp lý.
- **Tính minh bạch:** Một số mô hình AI phức tạp (như học sâu) khó giải thích cách đưa ra quyết định.
- **Thiếu hụt dữ liệu:** Trong một số lĩnh vực, dữ liệu chất lượng cao vẫn chưa đủ.
- **Tài nguyên tính toán:** AI đòi hỏi các hệ thống máy tính mạnh mẽ, dẫn đến chi phí cao.

Cơ hội

- **Đổi mới công nghệ:** AI tạo ra các giải pháp mới trong nhiều lĩnh vực, từ y tế đến nông nghiệp.
- **Thị trường toàn cầu:** Quy mô thị trường AI được dự báo đạt hàng trăm tỷ USD vào năm 2030.
- **Tăng năng suất:** Tự động hóa các tác vụ lặp lại và cải thiện hiệu suất làm việc.

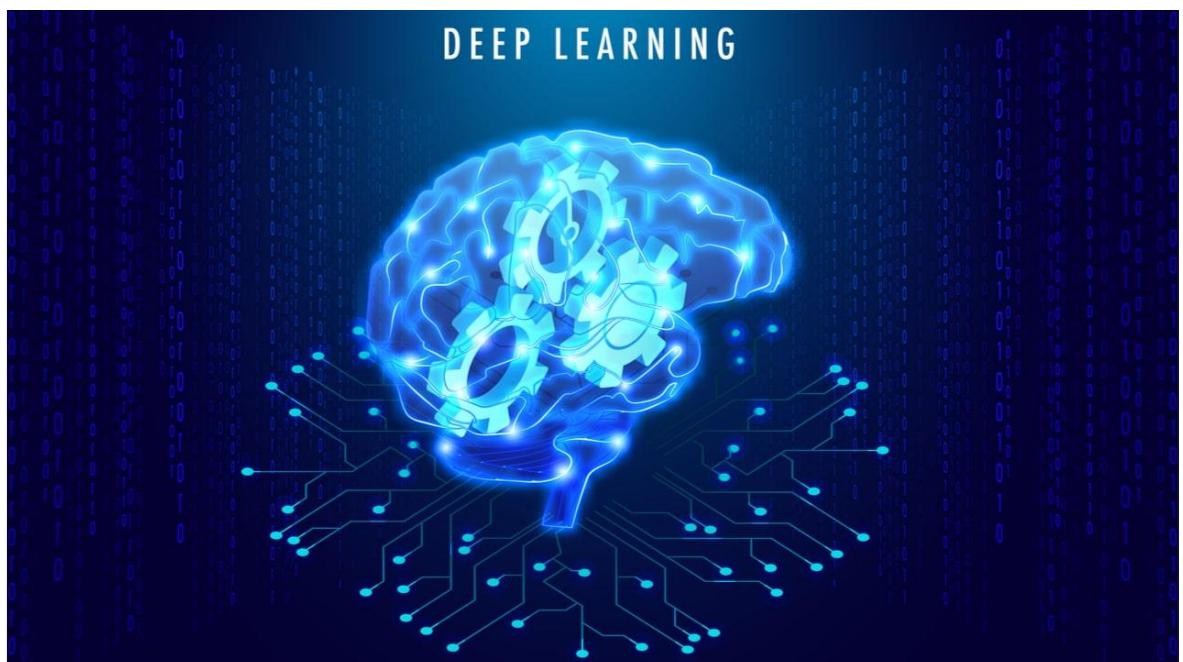
3.1.6. Tương lai của AI

Trong tương lai, AI được kỳ vọng sẽ vượt xa các ứng dụng hiện tại, từ phát triển các hệ thống tự động hóa hoàn toàn đến trí tuệ nhân tạo tổng quát. AI sẽ không chỉ làm thay đổi cách con người làm việc mà còn định hình lại cách chúng ta tương tác với thế giới.

3.2. Deep learning

3.2.1. Deep Learning là gì?

Deep Learning là gì? Deep Learning, hay học sâu, là một lĩnh vực con của Machine Learning (học máy), trong đó máy tính được lập trình để tự học và cải thiện hiệu suất thông qua các thuật toán. Deep Learning dựa trên các khái niệm phức tạp hơn so với Machine Learning truyền thống và chủ yếu sử dụng mạng nơ-ron nhân tạo để mô phỏng khả năng suy luận và tư duy của con người.

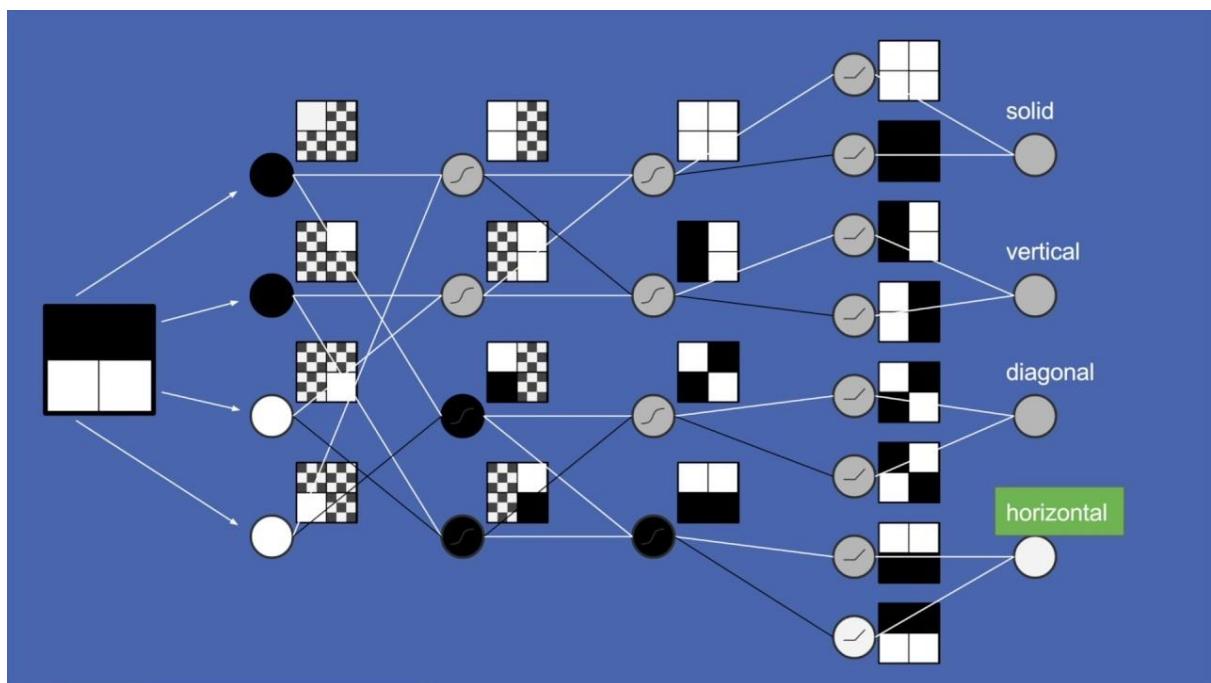


Hình 3. 7 Deep learning là gì

Mặc dù các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã xuất hiện từ những năm 1960, nhưng công nghệ này đã bị hạn chế bởi khả năng tính toán và khối lượng dữ liệu có sẵn trong thời điểm đó. Trong những năm gần đây, sự tiến bộ trong khai thác dữ liệu lớn (Big Data) đã cho phép chúng ta tận dụng tối đa khả năng của mạng nơ-ron nhân tạo.

Mạng nơ-ron nhân tạo là động lực chính sau sự phát triển của Deep Learning. Mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các phép tính với độ phức tạp rất cao. Deep Learning đang phát triển rất nhanh chóng và được coi là một trong những bước đột phá lớn nhất trong lĩnh vực Machine Learning

3.2.2. Cách thức Deep Learning hoạt động

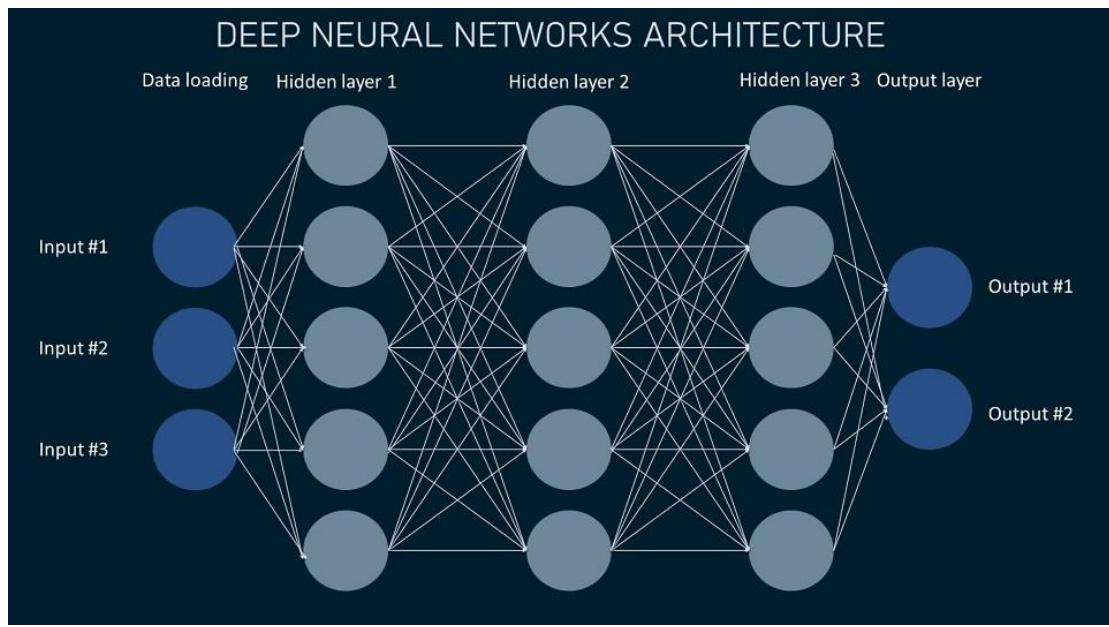


Hình 3. 8 Cách thức Deep Learning hoạt động

Sau khi tìm hiểu Deep Learning là gì qua định nghĩa trên, chúng ta cùng tìm hiểu cách thức hoạt động của Deep Learning là gì qua phần sau nhé!

Deep Learning là một phương pháp trong Machine Learning, trong đó mạng nơ-ron nhân tạo được sử dụng để mô phỏng khả năng tư duy của con người. Dưới đây là cách Deep Learning hoạt động:

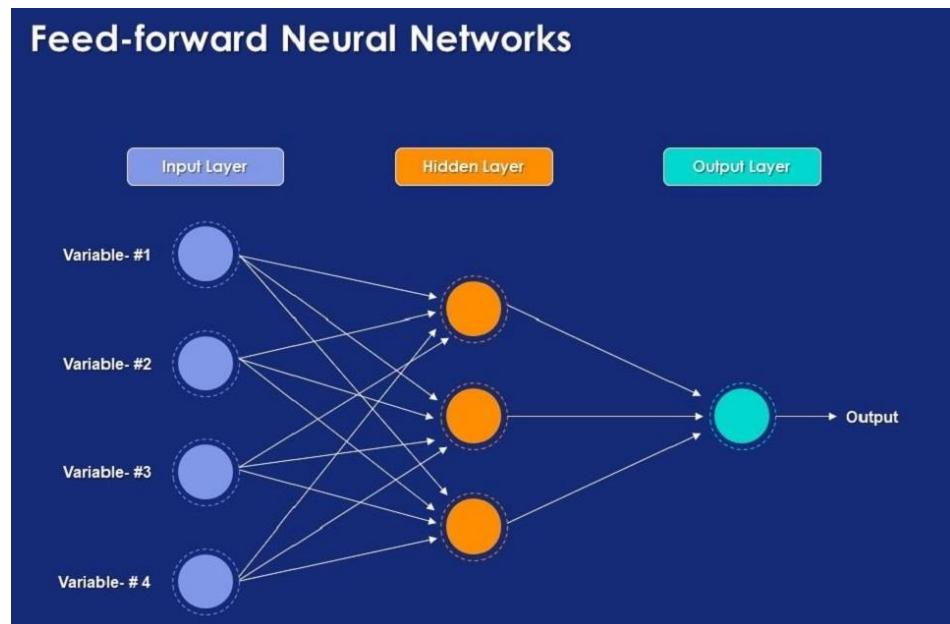
Kiến trúc mạng nơ-ron



Hình 3. 9 Kiến trúc mạng nơ-ron

- Mạng nơ-ron trong Deep Learning được xây dựng từ nhiều lớp, gọi là layer. Số lượng layer càng nhiều, mạng càng "sâu". Các layer được liên kết với nhau thông qua các kết nối nơ-ron và trọng số tương ứng.
- Mỗi nơ-ron trong mạng có một hàm kích hoạt, chịu trách nhiệm xử lý đầu vào và tạo ra đầu ra. Hàm kích hoạt giúp chuẩn hóa đầu ra của mỗi nơ-ron.

Feedforward (lai truyền thuận)



Hình 3. 10 Feedforward

- Dữ liệu được đưa vào mạng nơ-ron và truyền qua các layer theo chiều thuận từ layer đầu tiên đến layer cuối cùng, gọi là output layer.
- Trong quá trình này, các trọng số của mạng nơ-ron được áp dụng cho các kết nối nơ-ron và tính toán đầu ra của mỗi nơ-ron dựa trên đầu vào và hàm kích hoạt.

Huấn luyện (training)

- Quá trình huấn luyện mạng nơ-ron trong Deep Learning bao gồm việc tối ưu hóa các trọng số để đạt được hiệu suất tốt nhất.
- Đầu tiên, một hàm mất mát (loss function) được định nghĩa để đo lường sai lệch giữa kết quả dự đoán của mạng và giá trị thực tế.
- Sau đó, thuật toán lan truyền ngược (backpropagation) được sử dụng để lan truyền lỗi từ output layer về các layer trước đó. Quá trình này tính toán độ lỗi và điều chỉnh các trọng số của mạng nơ-ron dựa trên độ lỗi để cải thiện dự đoán.

Cập nhật trọng số

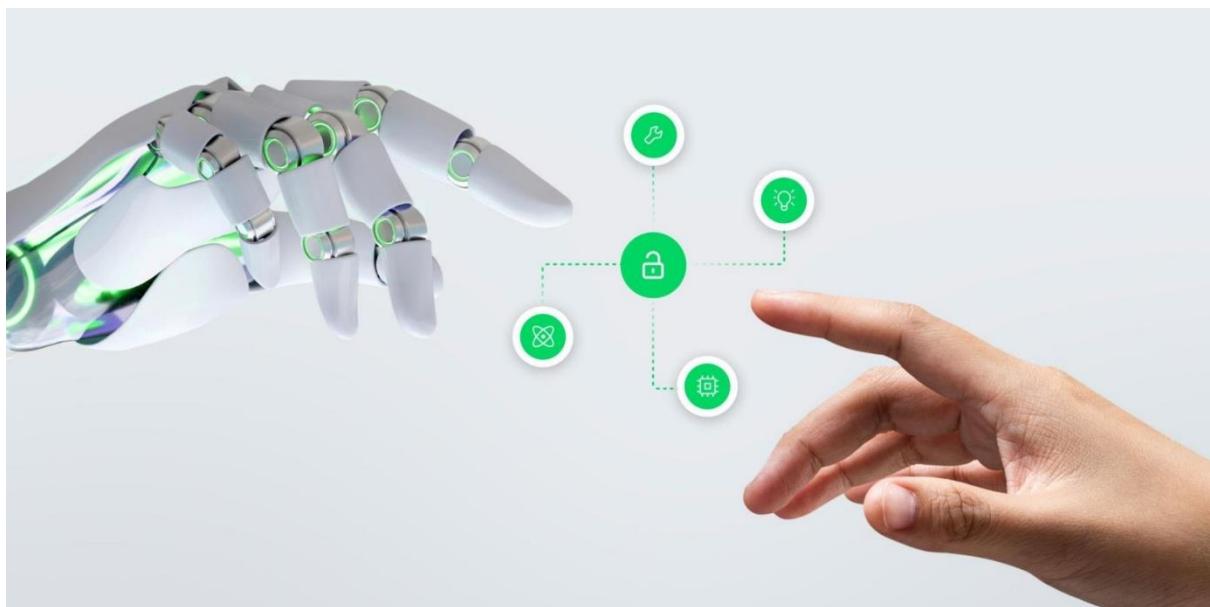
- Trong quá trình huấn luyện, các thuật toán tối ưu hóa như Gradient Descent được sử dụng để cập nhật các trọng số của mạng nơ-ron.
- Mục tiêu là điều chỉnh các trọng số sao cho giá trị hàm mất mát giảm xuống và đạt được dự đoán chính xác hơn trên tập huấn luyện.

Dự đoán

- Sau khi mạng nơ-ron đã được huấn luyện, nó có khả năng dự đoán đầu ra cho các dữ liệu mới.
- Đầu vào mới được đưa vào mạng nơ-ron và thông qua các trọng số đã được học để tạo ra dự đoán.

Deep Learning đòi hỏi phần cứng mạnh mẽ để xử lý lượng dữ liệu lớn và tính toán phức tạp. Thời gian huấn luyện mô hình Deep Learning có thể kéo dài từ vài giờ đến nhiều tháng, tùy thuộc vào kích thước của mạng và lượng dữ liệu.

Các ưu điểm và nhược điểm của Deep Learning



Hình 3. 11 Deep learning và con người

Dưới đây là các ưu và nhược điểm của Deep Learning:

Ưu điểm

- Độ linh hoạt của kiến trúc mạng nơ-ron cho phép dễ dàng thay đổi và tùy chỉnh để phù hợp với nhiều vấn đề khác nhau.
- Deep Learning có khả năng giải quyết các bài toán phức tạp với độ chính xác cao, đặc biệt là trong lĩnh vực nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, xử lý giọng nói, và nhiều lĩnh vực khác.
- Tính tự động hóa cao của Deep Learning cho phép mô hình tự điều chỉnh và tối ưu hóa chính nó dựa trên dữ liệu huấn luyện.
- Deep Learning có khả năng xử lý lượng dữ liệu lớn và thực hiện tính toán song song, giúp đạt hiệu năng cao.

Nhược điểm

- Deep Learning yêu cầu một khối lượng dữ liệu lớn để tận dụng được khả năng của nó. Việc thu thập và chuẩn bị dữ liệu có thể tốn kém và tốn thời gian. Để khai thác tối đa khả năng của Deep Learning, cần có một lượng dữ liệu lớn.

Tuy nhiên, việc thu thập và chuẩn bị dữ liệu có thể đòi hỏi thời gian và công sức đáng kể.

- Chi phí tính toán của Deep Learning cao, đặc biệt là khi xử lý các mô hình phức tạp và lượng dữ liệu lớn. Điều này đòi hỏi phần cứng mạnh mẽ và tài nguyên tính toán đáng kể.
- Hiện nay, vẫn chưa có nền tảng lý thuyết mạnh mẽ để lựa chọn công cụ tối ưu cho Deep Learning. Điều này có thể làm cho quá trình triển khai và tinh chỉnh mô hình trở nên phức tạp hơn.

Tóm lại, Deep Learning mang đến nhiều ưu điểm đáng kể trong việc giải quyết các bài toán phức tạp. Tuy nhiên, để khai thác tối đa khả năng của Deep Learning, cần phải xem xét các yếu tố như khối lượng dữ liệu, chi phí tính toán và công cụ tối ưu.

3.2.3. Các ứng dụng của Deep Learning

Dưới đây là một số ứng dụng phổ biến của Deep Learning mà chúng ta thường gặp trong cuộc sống hàng ngày:

Xe tự lái



Hình 3. 12 Ứng dụng xe tự lái

Một trong những ứng dụng đáng chú ý nhất của Deep Learning trong đời sống hàng ngày là công nghệ xe tự lái. Được sử dụng rộng rãi trong lĩnh vực này, Deep Learning giúp xe tự động nhận diện và hiểu môi trường xung quanh bằng cách sử dụng mạng nơ-ron phức tạp. Các mô hình Deep Learning có khả năng xử lý hình ảnh từ các camera và cảm biến để nhận biết các đối tượng, tính toán khoảng cách, phân loại làn đường, xác định tín hiệu giao thông và dự đoán hành vi của các phương tiện khác. Thông qua việc xử lý lượng lớn dữ liệu và áp dụng thuật toán học sâu, các hãng xe tiên phong như Tesla đã đạt được những tiến bộ đáng kể trong việc phát triển công nghệ xe tự lái.

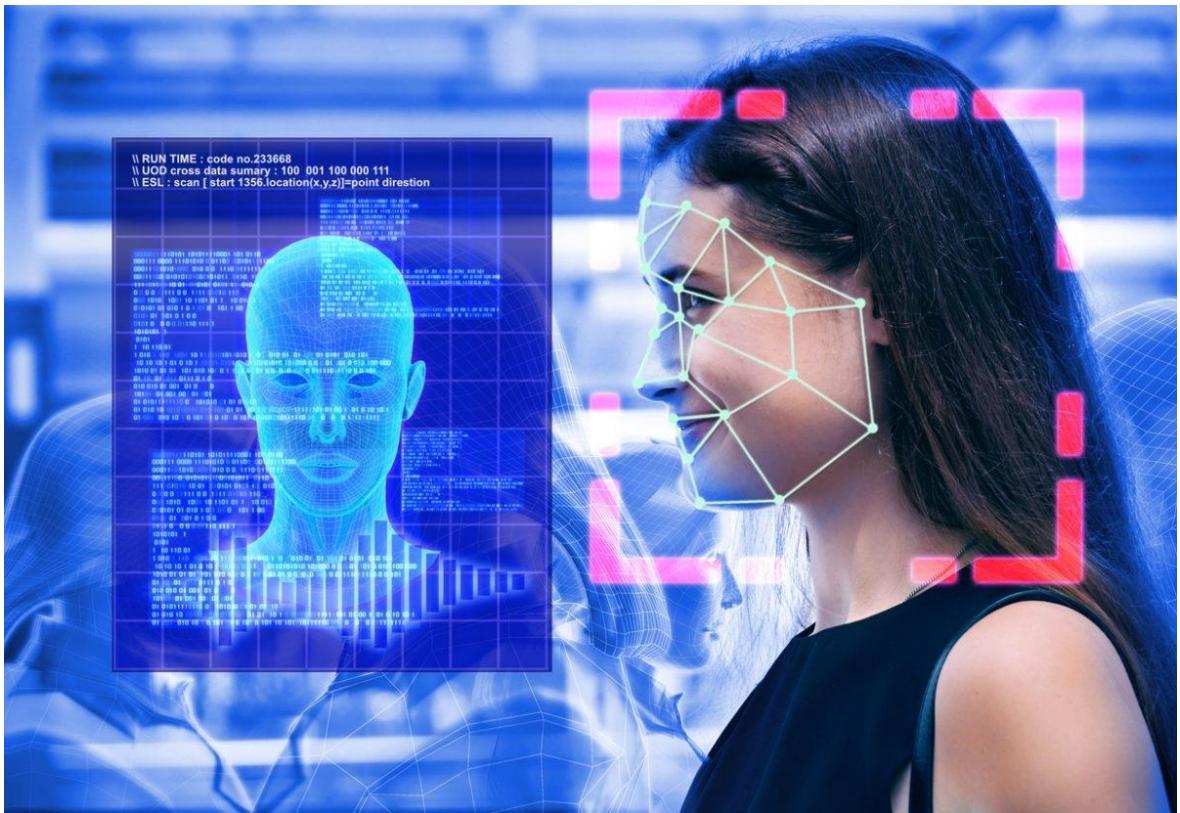
Nhận dạng giọng nói



Hình 3. 13 Ứng dụng trong Nhận dạng giọng nói

Deep Learning đã mang lại sự tiến bộ đáng kể trong việc nhận dạng và hiểu giọng nói trong cuộc sống hàng ngày. Các mô hình Deep Learning được áp dụng trong các trợ lý ảo như Siri, Google Assistant và Alexa, giúp chúng có khả năng hiểu và thực hiện các lệnh dựa trên giọng nói của người dùng. Bằng cách sử dụng mạng nơ-ron phức tạp, Deep Learning có khả năng phân tích âm thanh, xử lý ngôn ngữ tự nhiên và tìm hiểu ý nghĩa của câu nói. Điều này cho phép trợ lý ảo nhận diện và thực hiện các tác vụ như gọi điện, tìm kiếm thông tin, đặt lịch hẹn và điều khiển các thiết bị trong nhà chỉ bằng giọng nói. Sự phát triển của Deep Learning trong lĩnh vực này đã mang đến một trải nghiệm tương tác người - máy tiện lợi và tự nhiên hơn.

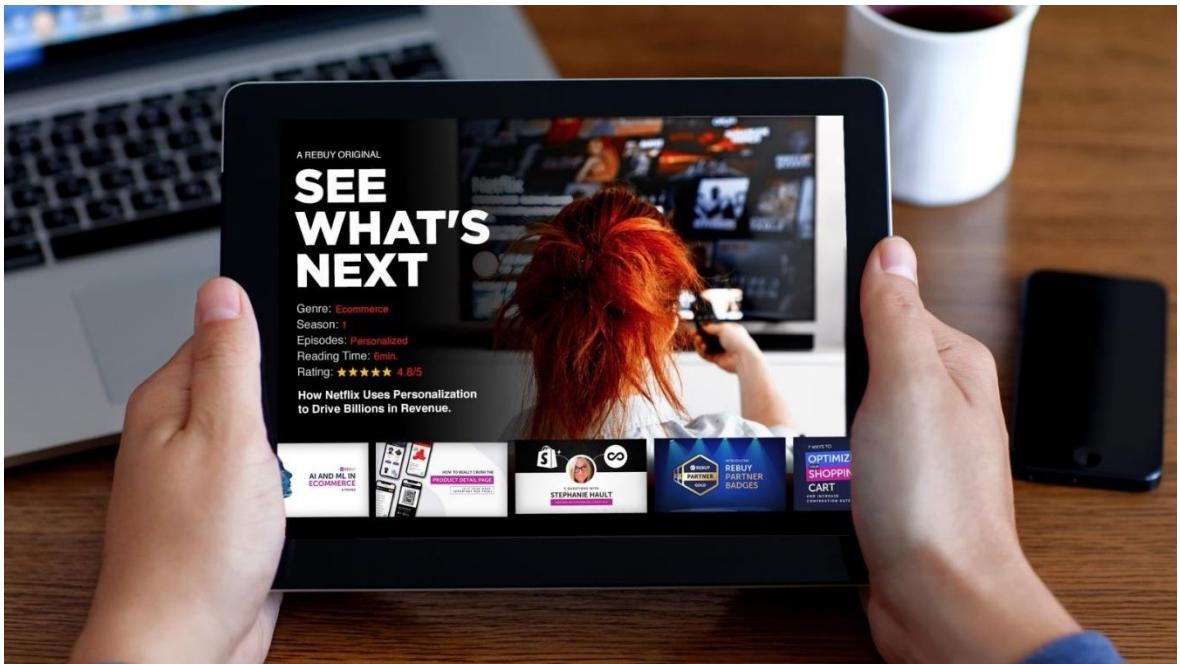
Nhận dạng ảnh



Hình 3. 14 Ứng dụng trong nhận dạng ảnh

Deep Learning đã có những đóng góp đáng kể trong việc nhận dạng và xử lý ảnh trong cuộc sống hàng ngày. Áp dụng Deep Learning vào lĩnh vực này, chúng ta đã có khả năng phân loại, nhận biết và hiểu nội dung của các hình ảnh. Các mô hình Deep Learning sử dụng mạng nơ-ron sâu để học và nhận diện các đặc trưng trong ảnh, từ đó đưa ra những dự đoán chính xác về nội dung của ảnh. Ví dụ, Deep Learning có thể được sử dụng để gắn nhãn tự động cho ảnh, nhận diện khuôn mặt, phân loại đối tượng, xử lý ảnh y tế, và nhiều ứng dụng khác. Công nghệ này đã mang lại tiện ích lớn cho việc tìm kiếm và quản lý ảnh số, cải thiện trải nghiệm người dùng trong các ứng dụng mạng xã hội, hỗ trợ trong nhiều lĩnh vực khoa học và y tế.

Tự động gợi ý và cá nhân hóa



Hình 3. 15 Ứng dụng trong tự động gợi ý

Deep Learning đã đóng góp quan trọng trong việc tạo ra các hệ thống gợi ý và cá nhân hóa trong cuộc sống hàng ngày. Các mô hình Deep Learning được sử dụng để phân tích và hiểu hành vi người dùng, từ đó đưa ra các đề xuất và gợi ý phù hợp với sở thích cá nhân. Ví dụ, các nền tảng như Netflix, Spotify và Amazon sử dụng Deep Learning để phân tích lịch sử xem, lắng nghe và mua sắm của người dùng. Dựa trên thông tin này, họ có thể đề xuất nội dung, bài hát, phim, sách và sản phẩm khác mà người dùng có thể quan tâm. Công nghệ Deep Learning đã cải thiện trải nghiệm người dùng, mang lại sự tiện ích và tạo ra cá nhân hóa trong các dịch vụ và sản phẩm mà chúng ta sử dụng hàng ngày.

Dịch máy



Hình 3. 16 Ứng dụng trong dịch máy

Deep Learning đã có những đóng góp quan trọng trong lĩnh vực dịch máy tự động, làm cho việc diễn giải và chuyển đổi giữa các ngôn ngữ trở nên hiệu quả hơn. Các mô hình Deep Learning sử dụng mạng nơ-ron sâu để học và hiểu cấu trúc ngôn ngữ, cùng với việc áp dụng các phương pháp xử lý ngôn ngữ tự nhiên. Kết quả là, chúng ta có khả năng dịch văn bản tự động với độ chính xác và dịch thuật gần với ngôn ngữ tự nhiên hơn. Deep Learning đã giúp cải thiện các công cụ dịch trực tuyến, ứng dụng di động và hệ thống dịch máy tự động, đồng thời tiếp cận và giao tiếp với người nói ngôn ngữ khác nhau trở nên dễ dàng hơn và thuận tiện hơn.

Phân tích cảm xúc

Lĩnh vực phân tích cảm xúc là một ứng dụng quan trọng của Deep Learning trong việc hiểu và đánh giá cảm xúc của con người thông qua văn bản, bình luận, đánh giá, và các dữ liệu ngôn ngữ khác. Công nghệ Deep Learning có thể được áp

dụng để xử lý và phân tích lượng lớn dữ liệu ngôn ngữ tự nhiên, từ đó nhận biết và phân loại cảm xúc như vui vẻ, buồn bã, hài lòng hay không hài lòng.

Các công ty và tổ chức có thể tận dụng Deep Learning để phân tích đánh giá, bình luận trên mạng xã hội, tweet, thông tin phản hồi từ khách hàng và nguồn dữ liệu khác. Qua đó, họ có thể hiểu được ý kiến, nhận định và cảm xúc của khách hàng đối với sản phẩm, dịch vụ, hoặc sự kiện cụ thể. Kết quả phân tích cảm xúc có thể cung cấp thông tin quý giá để hỗ trợ quyết định kinh doanh, phát triển chiến lược Marketing, cải thiện chất lượng sản phẩm và tương tác với khách hàng một cách tốt nhất.

Mạng xã hội

Deep Learning được sử dụng trong các nền tảng mạng xã hội như X, Instagram và Facebook để cải thiện trải nghiệm người dùng. Công nghệ này giúp phân tích dữ liệu lớn và hiểu sở thích của người dùng, đồng thời ngăn chặn nội dung bạo lực và không phù hợp. Deep Learning cũng được áp dụng để gợi ý nội dung, bạn bè và dịch vụ phù hợp, cùng nhận diện khuôn mặt trong ảnh.

Lĩnh vực y tế



Hình 3. 17 Ứng dụng trong lĩnh vực y tế

Deep Learning đã đóng góp đáng kể trong lĩnh vực y tế, đặc biệt là trong việc dự đoán tình trạng bệnh, chẩn đoán ung thư và phân tích kết quả hình ảnh y tế như MRI và X-ray.

Các mô hình Deep Learning được huấn luyện trên dữ liệu lớn từ hàng ngàn bệnh nhân, giúp xác định các mẫu và đặc trưng quan trọng trong dữ liệu y tế. Điều này cho phép họ tạo ra các mô hình dự đoán tình trạng bệnh, từ việc ước lượng nguy cơ mắc bệnh đến dự đoán kết quả điều trị.

Ngoài ra, Deep Learning cũng được sử dụng trong quá trình chẩn đoán ung thư. Các mô hình Deep Learning có khả năng phân tích và nhận dạng các đặc điểm bất thường trong hình ảnh y tế, giúp phát hiện sớm các dấu hiệu của ung thư và hỗ trợ các chuyên gia y tế trong quá trình chẩn đoán.

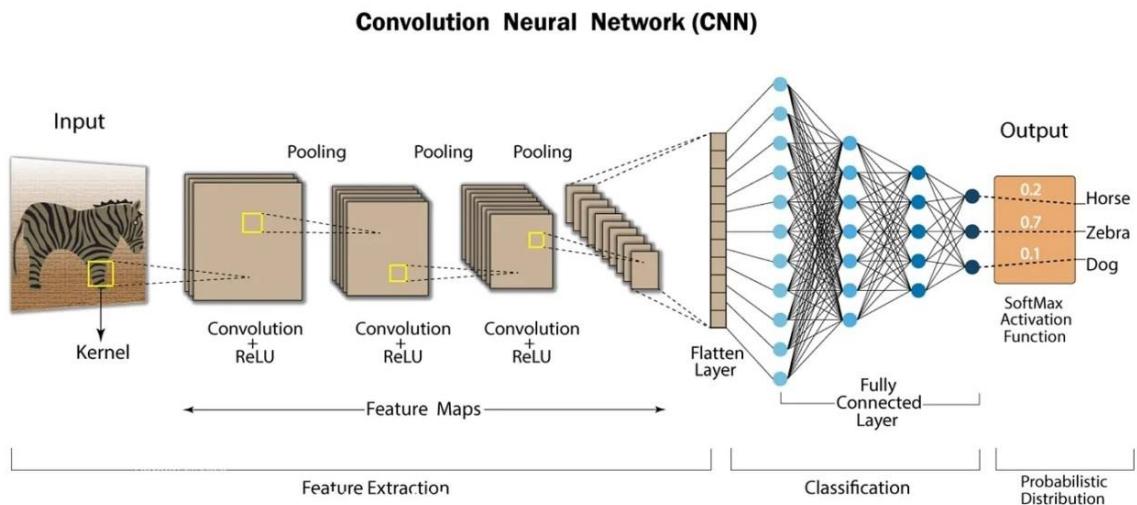
3.2.4. Phân biệt Deep Learning và Machine Learning

Lượng dữ liệu đầu vào	Có thể huấn luyện trên các tập dữ liệu nhỏ hơn	Yêu cầu lượng lớn dữ liệu
Cấu trúc mô hình	Sử dụng các mô hình học máy truyền thống như cây quyết định, hồi quy tuyến tính, SVM,...	Sử dụng các mạng nơ-ron nhân tạo (ANNs) phức tạp với nhiều lớp ẩn
Khả năng tự học	Cần nhiều sự can thiệp của con người hơn để sửa chữa và học hỏi	Tự học hỏi từ môi trường và những sai lầm trong quá khứ
Thời gian đào tạo	Đào tạo ngắn hơn và độ chính xác thấp hơn	Đào tạo lâu hơn và độ chính xác cao hơn
Môi tương quan	Tạo môi tương quan tuyến tính đơn giản	Tạo ra các môi tương quan phức tạp, phi tuyến tính
Thiết bị xử lý dữ liệu	Có thể huấn luyện trên CPU (bộ xử lý trung tâm)	Cần GPU (bộ xử lý đồ họa) chuyên dụng để huấn luyện
Ứng dụng	Thường được sử dụng cho các nhiệm vụ đơn giản như phân loại, phân nhóm, dự đoán	Thường được sử dụng cho các nhiệm vụ phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, xe tự lái, deepfake

3.2.5. Các phương pháp Deep Learning Mạng nơ-ron cổ điển

Mạng nơ-ron cỗ điển là một kiến trúc đơn giản của mạng nơ-ron, được xây dựng bằng cách kết nối perceptron đa lớp. Nó được sử dụng chủ yếu cho các bài toán phân loại nhị phân. Mạng nơ-ron cỗ điển sử dụng các hàm tuyến tính và phi tuyến, bao gồm sigmoid, tanh và ReLU. Nó phù hợp với dữ liệu dạng bảng, các bài toán phân loại và hồi quy với đầu vào là giá trị thực.

Mạng nơ-ron tích chập (CNN)



Hình 3. 18 Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (CNN) là một kiến trúc Neural Network được sử dụng chủ yếu cho xử lý hình ảnh và các bài toán phức tạp. Trong CNN, tích chập là một phép biến đổi tín hiệu đầu vào thông qua việc áp dụng bộ lọc để trích xuất các đặc trưng quan trọng.

Mô hình CNN bao gồm các layer đầu vào và đầu ra, cùng với các layer tích chập, lấy mẫu và kết nối hoàn toàn. Quá trình tích chập sử dụng các bộ lọc để tạo ra các tầng mới và trích xuất đặc trưng. Quá trình lấy mẫu giúp giảm kích thước đầu vào, trong khi quá trình kết nối hoàn toàn sử dụng các kết nối giữa các layer để tạo ra kết quả cuối cùng.

Các ứng dụng phổ biến của CNN bao gồm nhận diện, phân tích và phân khúc hình ảnh, phân tích video và xử lý ngôn ngữ tự nhiên.

Mạng nơ-ron hồi quy (RNN)

Mạng nơ-ron hồi quy (RNN) là một thuật toán được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên. RNN khác với các mô hình nơ-ron truyền thống bởi vì nó có

khả năng nhớ thông tin từ quá trình tính toán trước đó. Điều này cho phép RNN xử lý các chuỗi dữ liệu, với kết quả đầu ra phụ thuộc vào các phép tính trước đó.

Có hai dạng thiết kế chính của RNN:

- LSTM (Long Short-Term Memory): Được sử dụng để dự đoán dữ liệu theo thời gian, có khả năng lưu giữ hoặc loại bỏ thông tin, được điều chỉnh bởi các cổng như Input, Output và Forget.
- Gated RNN: Là một thiết kế phổ biến trong dự đoán chuỗi thời gian, với hai cổng là Update và Reset.

RNN có thể giải quyết các dạng bài toán nào?

- One to one: Một đầu vào kết nối với một đầu ra duy nhất, ví dụ như phân loại hình ảnh.
- One to many: Một đầu vào kết nối với nhiều đầu ra chuỗi, ví dụ như đặt chú thích cho ảnh.
- Many to one: Nhiều đầu vào nhưng chỉ có một đầu ra, ví dụ như phân loại cảm xúc.
- Many to many: Nhiều đầu vào và nhiều đầu ra, ví dụ như phân loại video.

Mạng sinh đối nghịch (GAN)

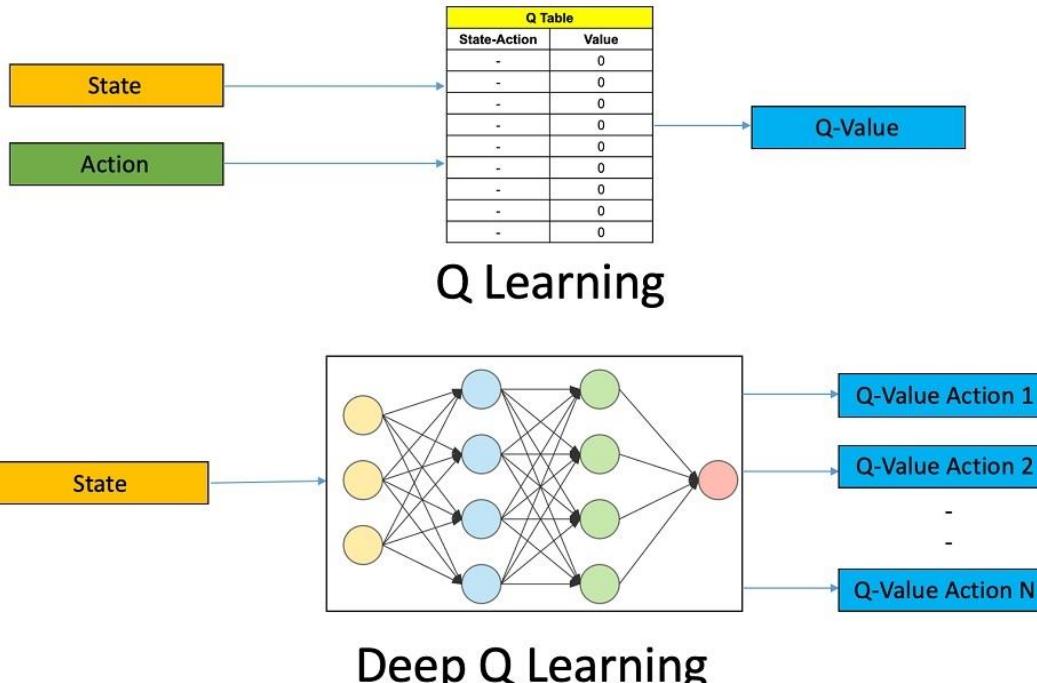
Mạng sinh đối nghịch (GAN) là một loại mô hình được sử dụng để tạo ra dữ liệu giả giống với dữ liệu thật. GAN bao gồm hai mạng chính là Generator (Người tạo) và Discriminator (Người phân biệt) hoạt động đối nghịch với nhau. Generator học cách tạo ra dữ liệu giả để đánh lừa Discriminator, trong khi Discriminator học cách phân biệt dữ liệu giả và dữ liệu thật. Cả hai mạng này được cải thiện qua quá trình huấn luyện.

GAN có nhiều ứng dụng phổ biến, ví dụ như tạo khuôn mặt người, thay đổi độ tuổi của khuôn mặt, tạo ảnh của các vật thể, tạo nhân vật hoạt hình và nhiều hơn nữa.

Boltzmann machine

Máy Boltzmann (Boltzmann machine) là một mô hình mạng không có hướng xác định, trong đó các node của mạng được kết nối thành một vòng tròn. Mô hình này thường được sử dụng để tạo ra các tham số cho mô hình. Các ứng dụng phổ biến của máy Boltzmann là giám sát hệ thống và xây dựng hệ thống khuyến nghị phân.

Deep Reinforcement Learning



Hình 3. 19 Học tăng cường sâu

Học tăng cường sâu (Deep Reinforcement Learning) là quá trình mà các tác tử tương tác với môi trường để thay đổi trạng thái của chính mình. Các tác tử quan sát và thực hiện hành động phù hợp để đạt được mục tiêu.

Mô hình học tăng cường sâu bao gồm input layer, output layer và nhiều hidden layer khác. Trạng thái của môi trường được đưa vào input layer. Mô hình được huấn luyện liên tục để dự đoán điểm đạt được sau mỗi hành động trong trạng thái cụ thể.

Học tăng cường sâu được áp dụng chủ yếu trong các trò chơi cờ vua, poker, xe tự lái, robot và nhiều ứng dụng khác.

Autoencoder

Autoencoder là một kỹ thuật Deep Learning phổ biến được sử dụng để học biểu diễn dữ liệu đầu vào mà không cần nhãn, tức là học không giám sát. Có một số loại autoencoder như Sparse (thưa), Denoising (lọc nhiễu), Contractive (hạn chế), và Stacked (xếp chồng).

Ứng dụng phổ biến của autoencoder bao gồm phát hiện đặc trưng, xây dựng hệ thống khuyến nghị, bổ sung đặc trưng cho tập dữ liệu và nhiều hơn nữa.

Backpropagation

Lan truyền ngược (backpropagation) là một kỹ thuật quan trọng trong mạng neuron. Nó giúp tính toán gradient từ layer cuối cùng đến layer đầu tiên của mạng. Qua

quá trình này, mang sẽ điều chỉnh các tham số dựa trên hàm mất mát. Lỗi tính toán được lan truyền ngược lại để điều chỉnh các tham số cho phù hợp.

Gradient Descent

Gradient Descent là một phương pháp quan trọng trong Deep Learning để tìm giá trị nhỏ nhất (hoặc lớn nhất) của một hàm số. Thay vì tìm nghiệm toàn cục, chúng ta thường tìm các điểm cực tiểu địa phương.

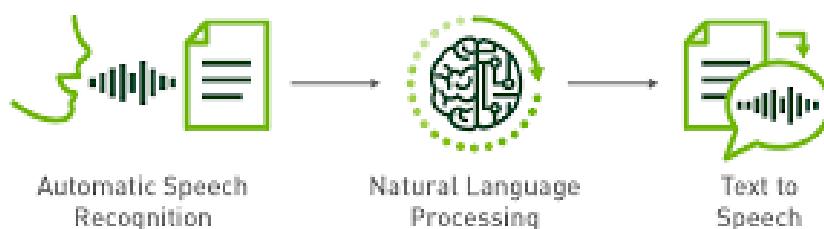
Phương pháp Gradient Descent dựa trên việc tính toán đạo hàm của hàm mất mát và di chuyển theo hướng ngược với đạo hàm để tiến gần đến điểm cực tiểu. Điểm khởi đầu được chọn và thuật toán lặp lại quá trình này cho đến khi đạt được kết quả mong muốn.

Trong Deep Learning, phương pháp Gradient Descent kết hợp với lan truyền ngược (backpropagation) giúp tối ưu hóa mạng nơ-ron nhanh chóng, đáng kể so với các phương pháp truyền thống.

3.3. Automatic Speech Recognition

3.3.1. Tổng quan

Automatic Speech Recognition (ASR), hay nhận dạng giọng nói tự động, là một công nghệ tiên tiến cho phép máy tính chuyển đổi tín hiệu âm thanh (speech) thành văn bản (text) một cách tự động và chính xác. ASR đóng vai trò quan trọng trong việc kết nối giữa con người và máy móc, giúp các hệ thống số hóa hiểu và xử lý thông tin từ giọng nói của người dùng.



Hình 3. 20 Automatic Speech Recognition

Công nghệ này hiện nay được tích hợp rộng rãi trong nhiều lĩnh vực, từ các trợ lý ảo như Siri, Alexa, Google Assistant, đến những ứng dụng đời sống như nhập liệu

bằng giọng nói, dịch thuật tự động, phụ đề trực tiếp, và hệ thống hỗ trợ khách hàng thông minh. ASR không chỉ làm thay đổi cách con người tương tác với máy tính mà còn mở ra các cơ hội trong các ngành công nghiệp như chăm sóc sức khỏe, giáo dục, giải trí, và thương mại.

Để đạt được khả năng chuyển đổi giọng nói thành văn bản, ASR phải dựa trên sự kết hợp của nhiều lĩnh vực khoa học và công nghệ khác nhau như:

- **Xử lý tín hiệu âm thanh:** Trích xuất các đặc trưng quan trọng từ tín hiệu giọng nói.
- **Xử lý ngôn ngữ tự nhiên (NLP):** Hiểu và diễn giải ngữ cảnh của ngôn ngữ đầu ra.
- **Âm học:** Nghiên cứu cách con người phát âm và cách âm thanh được truyền tải qua môi trường.

Với sự phát triển mạnh mẽ của công nghệ học sâu (Deep Learning) và các kiến trúc mạng thần kinh tiên tiến như Transformer, ASR ngày nay đã đạt được độ chính xác vượt trội, có thể xử lý các tình huống phức tạp như môi trường có tiếng ồn, giọng nói địa phương, hoặc tốc độ nói khác nhau. Điều này biến ASR trở thành một trong những công nghệ lõi trong kỷ nguyên số hóa và trí tuệ nhân tạo.

3.3.2. Lịch sử hình thành



Hình 3. 21 Lịch sử hình thành và phát triển

Quá trình phát triển của **Automatic Speech Recognition (ASR)** đã trải qua nhiều giai đoạn quan trọng, từ những nghiên cứu ban đầu vào giữa thế kỷ 20 đến các hệ thống hiện đại ngày nay. Lịch sử này gắn liền với sự tiến bộ của các lĩnh vực khoa học như xử lý tín hiệu, học máy, và trí tuệ nhân tạo (AI).

1950–1970: Những bước khởi đầu

ASR bắt đầu xuất hiện như một ý tưởng khoa học vào thập niên 1950. Trong thời kỳ này, các hệ thống nhận dạng giọng nói chỉ có khả năng nhận diện một số lượng hạn chế từ hoặc âm tiết đơn giản:

- **1952:** Bell Labs giới thiệu hệ thống **Audrey**, có thể nhận diện các chữ số từ 0 đến 9 thông qua giọng nói. Đây là một trong những hệ thống ASR đầu tiên, dựa trên việc phân tích các tín hiệu âm thanh bằng cách sử dụng các đặc trưng tần số.
- **1960s:** Các nhà nghiên cứu tại Hệ thống điện thoại quốc gia Nhật Bản (NTT) phát triển các hệ thống nhận diện từ vựng đơn giản dựa trên phân tích quang phổ và các mô hình toán học.

1970–1980: Giai đoạn ứng dụng các mô hình xác suất

Trong thập niên 1970, ASR bắt đầu áp dụng các kỹ thuật thống kê, đáng chú ý nhất là mô hình Markov ẩn (**Hidden Markov Model - HMM**). Đây là một bước ngoặt trong lĩnh vực nhận dạng giọng nói:

- **1971–1976:** Dự án DARPA Speech Understanding Research của Mỹ đánh dấu sự hợp tác lớn đầu tiên nhằm cải tiến ASR. Kết quả là hệ thống **Harpy**, được phát triển bởi Đại học Carnegie Mellon, có khả năng nhận diện được hơn 1.000 từ.
- **HMM** được sử dụng để mô hình hóa các chuỗi âm thanh, giúp hệ thống ASR dự đoán được các từ ngữ dựa trên xác suất. Điều này giúp cải thiện độ chính xác và khả năng nhận diện trong các hệ thống lớn hơn.

1980–1990: Sự bùng nổ nhờ máy tính

Nhờ vào sự phát triển mạnh mẽ của công nghệ máy tính trong thập niên 1980, ASR bắt đầu được mở rộng:

- **1980s:** Các hệ thống ASR thương mại đầu tiên xuất hiện, được thiết kế để nhận diện từ vựng hạn chế trong các ứng dụng công nghiệp, chẳng hạn như hệ thống điều khiển bằng giọng nói trong sản xuất.
- **1990s:** Sự kết hợp giữa mô hình Markov ẩn và mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN) mang lại những cải tiến đáng kể. Các hệ thống như **DragonDictate**, ra mắt vào năm 1990, là những phần mềm nhận diện giọng nói thương mại đầu tiên.

2000–2010: Kỷ nguyên của học máy

Vào đầu thế kỷ 21, ASR bước vào giai đoạn phát triển mạnh mẽ với sự hỗ trợ của các thuật toán học máy và cơ sở dữ liệu lớn:

- **Google** và **Microsoft** bắt đầu áp dụng ASR trong các ứng dụng tìm kiếm và hệ thống nhận diện giọng nói qua điện thoại.
- **2008:** Google giới thiệu công nghệ nhận diện giọng nói trong **Google Voice Search**, mở đầu cho việc tích hợp ASR vào các sản phẩm hàng ngày.

2010–nay: Thời đại của học sâu (Deep Learning)

Trong thập niên 2010, công nghệ ASR đạt được bước đột phá nhờ sự xuất hiện của các mạng thần kinh sâu (Deep Neural Networks - DNN) và các kiến trúc học sâu như **Transformer**:

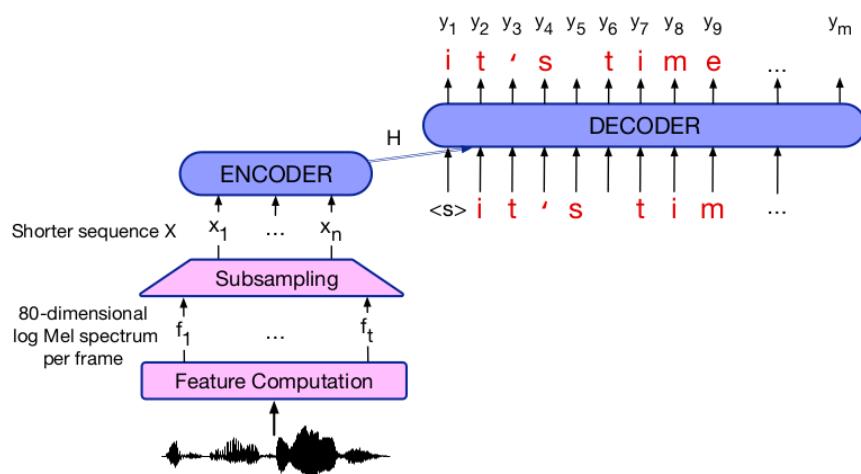
- **2012:** Hệ thống nhận diện giọng nói của Google sử dụng mạng nơ-ron sâu, cải thiện đáng kể độ chính xác và khả năng xử lý.
- **2016:** **Deep Speech** của Baidu sử dụng kiến trúc học sâu end-to-end, không cần phụ thuộc vào mô hình âm học hoặc ngữ pháp truyền thống.
- **2018:** **WaveNet** của Google DeepMind cài tiến khả năng tổng hợp giọng nói và nhận diện giọng nói, giúp các trợ lý ảo như Google Assistant trở nên tự nhiên hơn.
- **Hiện nay:** ASR được tích hợp sâu vào các thiết bị IoT, trợ lý ảo như Siri, Alexa, Google Assistant, và các ứng dụng dịch thuật tự động như Google Translate.

3.3.3. Kiến trúc

Kiến trúc cơ bản cho các tác vụ ASR là Encoder-Decoder (được áp dụng với RNNs và Transformers), khá giống với kiến trúc của task Machine Translation.

Thông thường, từ log mel spectral feature, ánh xạ tới các chữ cái, mặc dù cũng có thể ánh xạ tới morpheme như wordpiece hoặc BPE.

Hình bên dưới thể hiện kiến trúc encoder-decoder, khá tương tự như **attention-based encoder decoder** hay **AED**, hoặc **listen attend spell (LAS)**. Đầu vào của 1 chuỗi t vector đặc trưng $F=f_1, f_2, \dots, f_t$, $F=f_1, f_2, \dots, f_t$, 1 vector mỗi 10ms frame. Đầu ra có thể là chữ cái hoặc word-piece.



Hình 3. 22 Sơ đồ kiến trúc của bộ nhận dạng giọng nói bộ mã hóa-giải mã.

Bởi vì độ dài câu là khác nhau, nên kiến trúc encoder-decoder cho tiếng nói phải có 1 bước nén các thông tin từ khối encoder trước khi cho vào khối decoder. Mục tiêu của việc subsampling là sinh ra 1 chuỗi ngắn hơn $X=x_1, x_2, \dots, x_n$ để đầu vào cho khối decoder. Thuật toán đơn giản nhất gọi là low frame rate: với mỗi thời gian i , ta concatenate vector đặc trưng f_i với 2 vector trước f_{i-1}, f_{i-2} tạo thành 1 vector mới dài hơn 3 lần. Sau đó ta xóa đi f_{i-1}, f_{i-2} . Thay vì được 1 vector đặc trưng 40 chiều mỗi 10ms, ta thu được 1 vector 120 chiều mỗi 30ms, với độ dài chuỗi ngắn hơn $n=3t$.

Sau bước nén này, kiến trúc encoder-decoder cho giọng nói tương tự như kiến trúc cho Machine Translation (dịch máy), với sự kết hợp của mạng RNN hay Transformer.

$$p(y_1, \dots, y_n) = \prod_{i=1}^n p(y_i | y_1, \dots, y_{i-1}, X)$$

Ta có thể sinh ra mỗi chữ cái cho đầu ra nhờ vào thuật toán **greedy decoding**:

$$\hat{y}_i = \operatorname{argmax}_{\text{char} \in \text{Alphabet}} P(\text{char}|y_1 \dots y_{i-1}, X)$$

Hoặc là có thể sử dụng **beam search**, đây là thuật toán thường xuyên được sử dụng cho các bài toán **language model**. Beam search khởi đầu với 1 chuỗi rỗng. Tại mỗi bước, nó thực hiện tìm kiếm toàn bộ trên không gian của bước đó và lấy ra k kết quả có score cao nhất. Với λ là trọng số của language model, kết hợp với chuẩn hóa độ dài câu $|y|c|y|c$ và xác suất của language model $PLM(y)PLM(y)$, ta có phương trình tính score:

$$score(Y|X) = \frac{1}{|Y|_c} \log P(Y|X) + \lambda \log PLM(Y)$$

Learning

Encoder-decoder cho tiếng nói được huấn luyện với hàm loss **cross-entropy** được dùng cho các bài toán về language model. Tại timestep i của pha *decoding*, hàm loss chính là logarithm xác suất của kí tự đúng y_i :

$$L_{CE} = -\log p(y_i|y_1, \dots, y_{i-1}, X)$$

Hàm loss cho toàn bộ câu là tổng của các loss:

$$L_{CE} = -\sum_{i=1}^m \log p(y_i|y_1, \dots, y_{i-1}, X)$$

Chúng ta thường sử dụng **teacher forcing** sẽ giúp model hội tụ nhanh hơn.

CTC Inference

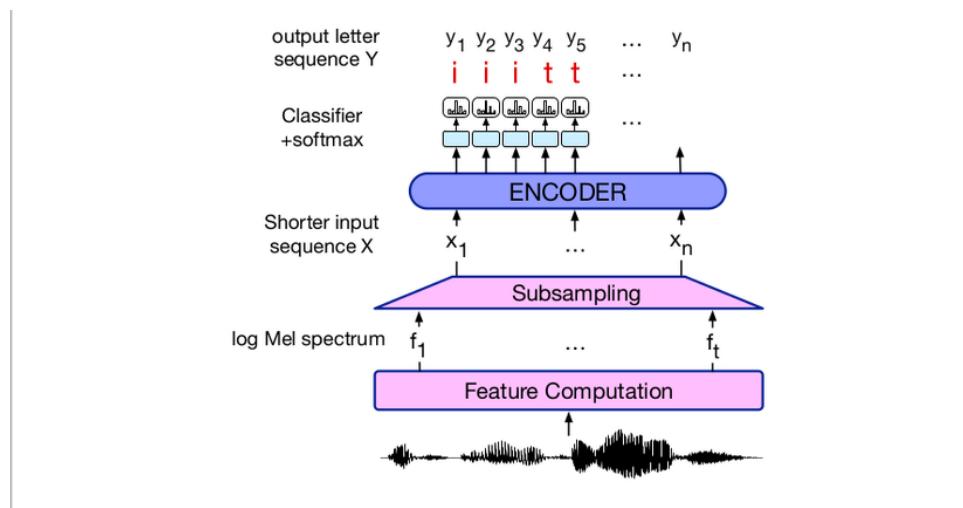
Trước khi tìm hiểu cách tính $PCTC(Y|X)PCTC(Y|X)$, đầu tiên cùng xem cách CTC gán xác suất cho mỗi alignment. CTC giả sử rằng với input X , CTC output a_t tại thời điểm t là độc lập với output ở các thời điểm a_i khác:

$$P_{CTC}(A|X) = \prod_{t=1}^T p(a_t|X)$$

Do đó để chọn ra alignment tốt nhất ta có thể chọn các kí tự có xác suất cao nhất tại mỗi timestep t :

$$\hat{a}_t = \arg \max_{c \in C} p_t(c|X)$$

Sau đó đưa alignment qua CTC cắt bớt các kí tự thừa để thu được đầu ra Y. Hình bên dưới, ta lấy encoder, sinh ra hidden state h_{t+1} tại mỗi timestep, và decode bằng cách lấy softmax trên tập các kí tự có trong từ điển.



Hình 3. 23 Quá trình giải mã

Có 1 vấn đề phát sinh trong quá trình inference ở CTC: ta chọn alignment có khả năng cao nhất A, nhưng chưa chắc alignment xác suất cao nhất lại tương ứng với xác suất cao nhất của output Y sau khi collapse. Bởi vì có rất nhiều alignment cùng dẫn đến 1 chuỗi đầu ra giống nhau. Ví dụ, tưởng tượng alignment có khả năng cao nhất A với input X là chuỗi $[a b \epsilon]$ nhưng 2 alignment cao tiếp theo lại là $[b \epsilon \epsilon b]$ và $[\epsilon \epsilon b b]$. Khi đó, output $Y = [b b]$, bằng tổng xác suất 2 alignment kia có khả năng chính xác hơn $Y = [a b]$.

$$P_{CTC}(Y|X) = \sum_{A \in B^{-1}(Y)} P(A|X)$$

$$= \sum_{A \in B^{-1}(Y)} \prod_{t=1}^T p(a_t|h_t)$$

$$\hat{Y} = \arg \max_Y P_{CTC}(Y|X)$$

Tuy nhiên, việc tính tổng xác suất tất cả các alignment sẽ rất tốn kém do có quá nhiều alignment, do đó người ta đề xuất tính xác suất của tổng bằng cách sử dụng 1 biến thể của Viterbi beam search mà giữ trong beam các alignment xác suất cao nhất mà ánh xạ tới cùng 1 output.

CTC Training

Để huấn luyện 1 hệ thống ASR sử dụng CTC, chúng ta sử dụng hàm loss *negative log-likelihood*. Do đó, loss cho toàn bộ tập dữ liệu D là tổng của các negative log-likelihoods của correct output Y cho mỗi input X :

$$L_{CTC} = \sum_{(X,Y) \in D} -\log P_{CTC}(Y|X)$$

Để tính toán CTC loss cho mỗi cặp đầu vào (X, Y) , ta cần tính xác suất của Y cho trước X , hay chính là tính tổng tất cả các alignment mà collapse lại thành Y . Nói cách khác:

$$P_{CTC}(Y|X) = \sum_{A \in B^{-1}(Y)} \prod_{t=1}^T P(a_t|H_t)$$

Tính toán tất cả các alugnment có thể xảy ra là không khả thi vì có quá nhiều alignment. Tuy nhiên ta có thể tính tổng bằng cách sử dụng *dynamic programming* để merge các alignment.

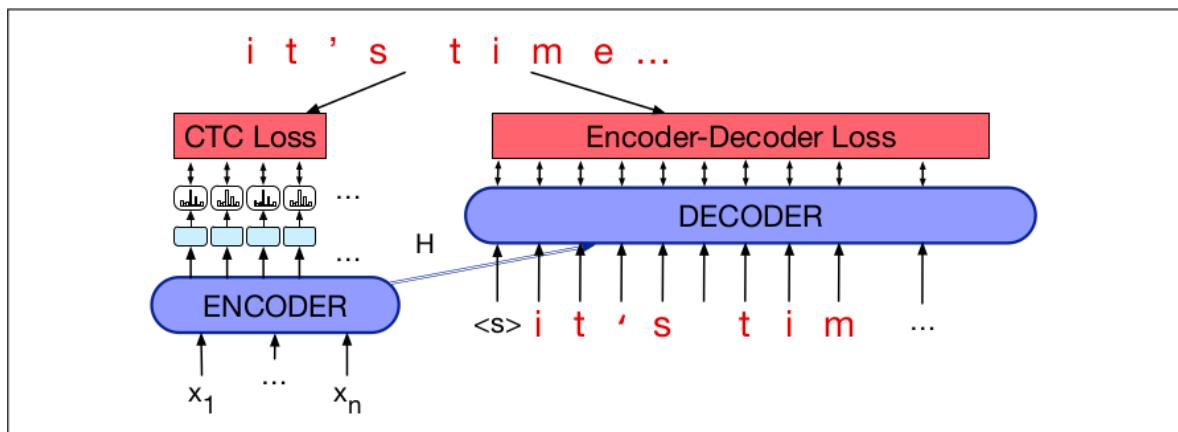
Kết hợp CTC and Encoder-Decoder

Chúng ta cũng có thể kết hợp 2 kiến trúc/loss function ta vừa mô tả ở trên, cross-entropy loss từ kiến trúc encoder-decoder, và CTC loss. Cho quá trình training, ta có thể đánh trọng số 2 loss với λ trên tập dev:

$$L = -\lambda \log P_{encdec}(Y|X) - (1 - \lambda) \log P_{ctc}(Y|X)$$

Cho quá trình inference, ta có thể kết hợp với 2 language model, cùng với *length penalty* và các trọng số đã được học:

$$\hat{Y} = \operatorname{argmax}_Y [\lambda \log P_{encdec}(Y|X) - (1 - \lambda) \log P_{CTC}(Y|X) + \gamma \log P_{LM}(Y)]$$



Hình 3. 24 Kết hợp CTC and Encoder-Decoder

Cải thiện CTC

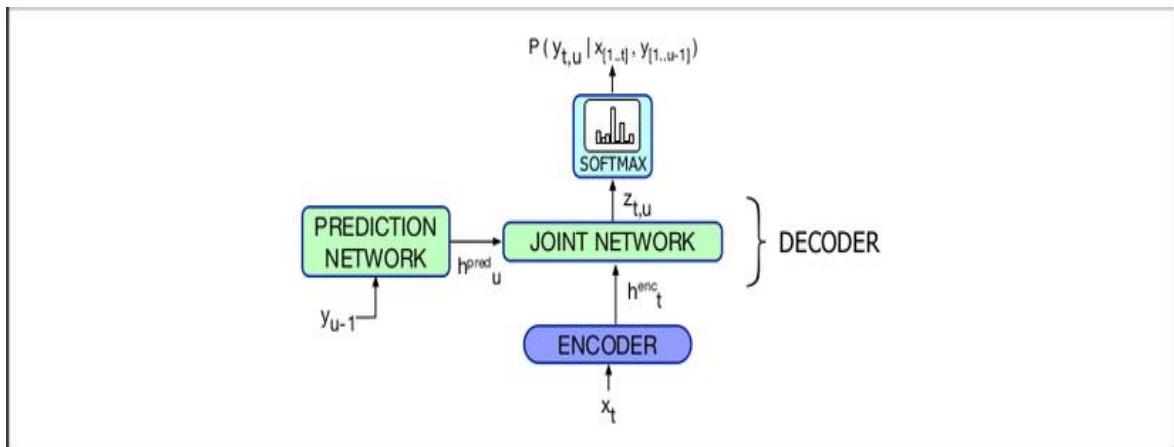
Bởi vì có 1 giả định độc lập trong CTC: cho rằng output tại thời điểm t là độc lập với output tại thời điểm $t-1$, nhận dạng trong CTC không đạt kết quả tốt như là nhận dạng attention-based encoder-decoder. Tuy nhiên, nhận dạng CTC có thể dùng cho bài toán **streaming**. Streaming là nhận dạng từ ngay lúc nói luôn hơn là chờ nói hết cả câu rồi mới nhận dạng. Các thuật toán sử dụng cơ chế attention cần phải tính toán chuỗi hidden state trên toàn bộ đầu vào rồi cung cấp phân phối attention cho decoder để nó tiến hành việc decode. Ngược lại, thuật toán CTC có thể input các chữ cái từ trái sang phải ngay lập tức.

Nếu chúng ta muốn làm việc với bài toán streaming, chúng ta cần 1 cách để cải thiện mô hình CTC recognition bằng cách loại bỏ giả định độc lập ở phía trên. Mô hình RNN-Transducer (RNN-T) là 1 model như vậy. RNN-T có 2 thành phần chính:

đầu tiên là CTC model, và 1 language model chia cắt gọi là **predictor** mà phụ thuộc vào đầu ra của các token ở phía trước. Tại mỗi timestep t , CTC encoder output 1 hidden state h_{ht} cho trước input $x_1 \dots x_t$. Language model predictor lấy đầu ra token ở trước làm input, và output hidden state h_{hu} . Sau đó h_{hu} được đưa qua 1 mạng khác có output được đưa qua lớp softmax để dự đoán kí tự tiếp theo.

$$P_{RNN-T}(Y|X) = \sum_{A \in B^{-1}(Y)} P(A|X)$$

$$= \sum_{A \in B^{-1}(Y)} \prod_{t=1}^T p(a_t | h_t, y_{< u_t})$$



Hình 3. 25 Đầu ra của bộ mã hóa âm thanh

ASR Evaluation: Word Error Rate

1 metric tiêu chuẩn dùng để đánh giá hệ thống ASR là **word error rate (WER)**. WER dựa vào có bao nhiêu từ được sinh ra từ mô hình khác biệt so với bản dịch gốc. Bước đầu tiên trong việc tính toán word error là tính toán **minimum edit distance** giữa các từ, sau đó trả về số lượng tối thiểu các từ **substitution**, **insertion**, và **deletion**. WER được định nghĩa như sau (do phương trình có bao gồm insertion nên có thể tỉ lệ WER sẽ lớn hơn 100%):

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

3.3.4. Cách hoạt động

Nhận dạng giọng nói tự động là một công nghệ khá tiên tiến, cực kỳ khó thiết kế và phát triển. Có hàng ngàn ngôn ngữ trên toàn thế giới với nhiều phương ngữ và trọng âm khác nhau. Nhận dạng giọng nói tự động sử dụng các khái niệm về xử lý ngôn ngữ tự nhiên và học máy để phát triển. Bằng cách kết hợp nhiều cơ chế học ngôn ngữ trong phần mềm, các nhà phát triển đảm bảo độ chính xác và hiệu quả của phần mềm nhận dạng giọng nói.

Một hệ thống ASR hoạt động qua nhiều giai đoạn, bao gồm:

Thu thập tín hiệu âm thanh (Speech Signal Acquisition)

- Thiết bị đầu vào: Âm thanh được thu qua microphone hoặc các thiết bị đầu vào âm thanh khác.
- Mẫu hóa tín hiệu: Tín hiệu âm thanh được số hóa bằng cách chuyển đổi từ dạng sóng âm thanh (analog) sang tín hiệu số (digital). Thông thường, quá trình này sử dụng tần số lấy mẫu phổ biến như 8kHz (điện thoại) hoặc 16kHz (độ phân giải cao).

Xử lý tín hiệu âm thanh (Pre-processing)

- Lọc nhiễu: Loại bỏ tạp âm và các yếu tố gây nhiễu trong tín hiệu âm thanh.
- Trích xuất đặc trưng (Feature Extraction): Các đặc trưng quan trọng của tín hiệu âm thanh được trích xuất để phục vụ cho quá trình nhận dạng. Các phương pháp phổ biến:
 - Mel Frequency Cepstral Coefficients (MFCC): Mô hình hóa phổ âm thanh theo cách tương tự như cách tai người nhận biết âm thanh.
 - Linear Predictive Coding (LPC): Dự đoán các mẫu sóng âm dựa trên thông tin tuyến tính.
 - Spectrogram: Biểu diễn năng lượng âm thanh theo thời gian và tần số.

Mô hình hóa âm vị (Acoustic Modeling)

- Mô hình âm học: Liên kết các đặc trưng âm thanh với các âm vị trong ngôn ngữ. Một âm vị là đơn vị nhỏ nhất của âm thanh trong một ngôn ngữ.
- Phương pháp sử dụng:

- Trước đây: Hidden Markov Models (HMM) được sử dụng để mô hình hóa mối quan hệ giữa âm thanh và âm vị.
- Hiện nay: Các phương pháp dựa trên mạng thần kinh sâu (Deep Neural Networks - DNN), như RNN hoặc Transformer, đang thay thế HMM nhờ độ chính xác cao hơn.

Mô hình hóa ngôn ngữ (Language Modeling)

- Vai trò: Dự đoán từ hoặc câu hợp lý dựa trên ngữ cảnh.
- Phương pháp:
 - Mô hình thống kê (Statistical Language Models - SLM): Dựa trên xác suất của các chuỗi từ, thường sử dụng N-grams.
 - Mô hình học sâu (Deep Language Models): Các mô hình hiện đại như BERT, GPT, hay Transformer giúp xử lý ngôn ngữ tự nhiên với ngữ cảnh phong phú hơn.

Phân tích và xuất kết quả (Decoding)

- **Bộ giải mã (Decoder):** Kết hợp các tín hiệu từ mô hình âm học, mô hình ngôn ngữ, và dữ liệu từ từ điển để chuyển đổi âm thanh thành văn bản.
- **Kết quả đầu ra:** Chuỗi văn bản cuối cùng được xử lý và hiển thị cho người dùng.

3.3.5. Các phương pháp chính trong ASR

Ban đầu, ASR chủ yếu dựa trên các mô hình truyền thống như Hidden Markov Model (HMM). Đây là công cụ cốt lõi trong việc xử lý chuỗi tín hiệu âm thanh, sử dụng các trạng thái ẩn để mô tả mối quan hệ giữa tín hiệu và các đơn vị âm vị (phoneme). HMM thường được kết hợp với Gaussian Mixture Models (GMM) để mô phỏng phân phối xác suất của tín hiệu âm thanh. Tuy nhiên, phương pháp này bị hạn chế trong việc mô hình hóa ngữ cảnh dài hạn và các đặc trưng phức tạp của âm thanh, dẫn đến nhu cầu cải tiến với các phương pháp học sâu hiện đại.

Các mạng học sâu như Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), và các biến thể của chúng như LSTM và GRU, đã cải thiện đáng kể hiệu suất ASR. DNN cho phép trích xuất và mô hình hóa các đặc trưng phức tạp của tín hiệu âm thanh, trong khi RNN và các biến thể của nó vượt trội trong việc xử

lý dữ liệu tuần tự, giúp ghi nhớ và sử dụng ngữ cảnh hiệu quả hơn. Tuy nhiên, RNN gặp khó khăn trong việc xử lý chuỗi dài do vấn đề về tính toán tuần tự, dẫn đến sự phát triển của các mô hình Transformer-based.

Transformer đã cách mạng hóa ASR hiện đại nhờ cơ chế self-attention, cho phép mô hình tập trung vào các phần quan trọng trong chuỗi dữ liệu mà không phụ thuộc vào thứ tự tuần tự. Một biến thể đặc biệt là Conformer (Convolutional Transformer), kết hợp khả năng xử lý ngữ cảnh toàn cục của Transformer với tính cục bộ của mạng tích chập (CNN). Điều này giúp Conformer trở thành kiến trúc hàng đầu trong việc tối ưu hóa hiệu suất và độ chính xác của ASR.

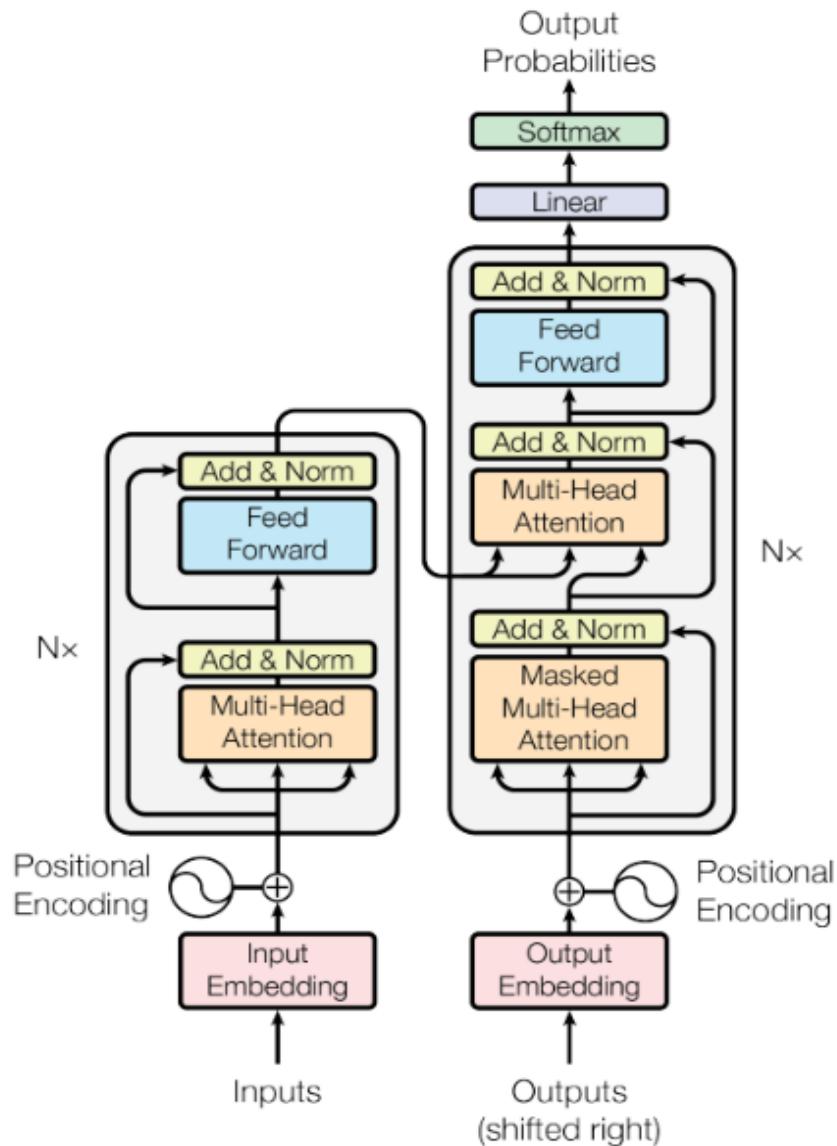
Ngoài ra, phương pháp End-to-End ASR đã đơn giản hóa quy trình xử lý truyền thống bằng cách sử dụng một mô hình duy nhất để chuyển đổi tín hiệu âm thanh thành văn bản. Các kiến trúc như Connectionist Temporal Classification (CTC), Sequence-to-Sequence (Seq2Seq) và các mô hình dựa trên Transformer được sử dụng rộng rãi trong hệ thống này. CTC giúp xử lý sự không đồng bộ giữa chuỗi âm thanh và chuỗi văn bản, trong khi Seq2Seq sử dụng cơ chế attention để tạo ra kết quả chính xác hơn.

3.4. Transformer

3.4.1. Sơ lược

Transformer là một mạng thần kinh được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Processing Language – NLP) và tín hiệu âm thanh (Audio Signal Processing - ASP).

Các ưu điểm của mạng có thể kể đến như tính song song, có thể tận dụng GPU trong quá trình huấn luyện. So với RNN hay LSTM, Transformer có thể làm việc với những phụ thuộc xa hơn (long-range dependencies), nghĩa là dữ liệu có chuỗi thời gian dài hơn mà lại ít mất thông tin hơn. Điều này mang lại nhiều lợi ích trong NLP và ASP vì trong tự nhiên, cả ngôn ngữ hay âm thanh đều rất lớn.



Hình 3. 26 Kiến trúc đầy đủ của Transformers

Tất cả mô hình Transformer như GPT, BERT, BART, T5,... được huấn luyện như những mô hình ngôn ngữ (language models). Với mô hình ngôn ngữ có nghĩa là mô hình được huấn luyện trên một số lượng lớn những văn bản thô trong quá trình học tập tự giám sát (self-supervised learning), đây là một loại học với mục tiêu là tính toán một cách tự động từ những đầu vào của mô hình. Cũng có nghĩa là con người không cần phải dán nhãn cho dữ liệu.

3.4.2. Lớp chú ý

Đặc trưng chủ yếu của Transformer chính là việc xây dựng lớp Attention. Lớp này sẽ nói với mô hình tập trung cụ thể vào những từ nhất định trong câu được đưa

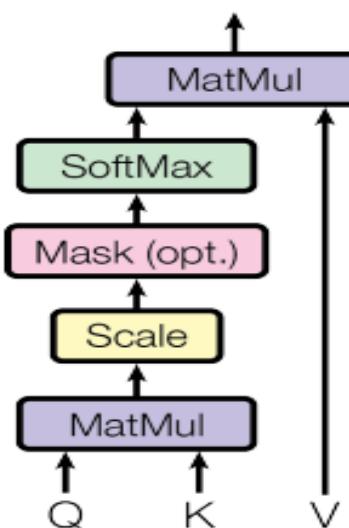
vào (và lờ đi nhiều hay ít những từ khác) khi xử lý với sự đại diện của từng từ. Hay nói cách khác chính là tính toán mức độ quan trọng (important) của một từ trong một câu (hoặc chuỗi) so với nhau.

Các thành phần chính bao gồm Query, Key, Value, Scaled Dot-Product Attention, Multi-Head Attention.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Cơ chế tự chú ý đa đầu (Multi-head Self-attention – MSA) đơn giản là Self-attention được thực hiện nhiều lần. Mỗi đầu (head) trong MSA là một bản sao của self-attention và có trọng số (weights) khác với các đầu còn lại.

Scaled Dot-Product Attention

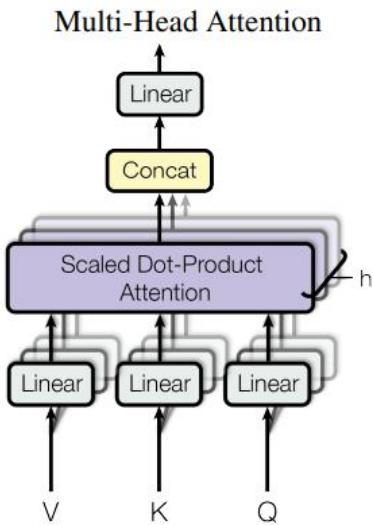


Hình 3. 27 Kiến trúc tự chú ý đa đầu (Multi-head self-Attention)

Mục đích của MSA chính là có thể học được các đại diện phức tạp và phi tuyến tính (non-linear) giữa các từ trong chuỗi hiệu quả hơn thông qua nhiều khía cạnh mà từ được nhìn nhận (sự tương quan và sự phụ thuộc) trong chuỗi.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where } \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

3.4.3. Encoder và Decoder



Hình 3. 28 Kiến trúc Scaled Dot-Product Attention

Tổng quan, Transformer chủ yếu gồm 2 khối kết hợp là Encoder và Decoder.

Tuy nhiên tùy vào bài toán mà mô hình Transformer chỉ có Encoder (mô hình Auto-Encoding) hoặc Decoder (mô hình Auto-Regressive) hoặc cả 2 (mô hình Sequence to Sequence hay Seq2Seq).

Encoder nhận vào một đầu vào và xây dựng nên một đại diện (representation) của đầu vào đó (features). Mô hình chỉ có Encoder phù hợp với các nhiệm vụ như phân loại từ (sequence classification), trả lời câu hỏi (question-answering), mô hình ngôn ngữ bị che (masked language modeling hay fill-mask) hay nhận diện thực thể được đặt tên (named entity recognition).

Decoder sử dụng đại diện của Encoder đọc theo những đầu vào khác để tạo ra chuỗi mục tiêu. Đầu vào của Mô hình Decoder được tối ưu cho việc tạo sinh. Mô hình chỉ có Decoder phù hợp cho các nhiệm vụ như tạo sinh văn bản.

Bên trong mô hình Seq2Seq, mô hình chỉ có Encoder được tối ưu cho hiểu biết từ đầu vào. Đầu ra của Encoder sẽ được chuyển trực tiếp cho Decoder. Đồng thời Decoder cũng có một đầu vào khác (nếu không có thì có thể cho một giá trị mà tại đó chỉ ra sự bắt đầu giải mã một chuỗi), đầu vào này sẽ kết hợp với đầu vào được đưa vào bởi Encoder và sau khi kết thúc giải mã thì đầu ra của Decoder sẽ thu được một từ.

Trong suốt quá trình huấn luyện mô hình Seq2Seq, đầu vào của Encoder là những câu ở một ngôn ngữ cụ thể và attention trong Encoder có thể sử dụng tất cả từ trong câu đó. Còn đầu vào của Decoder là những câu tương tự nhưng ở một ngôn ngữ khác. Decoder làm việc tuần tự và chỉ tập trung vào những từ trong câu mà đã được dịch. Có một điều thú vị rằng trọng số của Encoder và Decoder tách biệt nhau và không có sự chia sẻ trọng số nào được diễn ra.

3.4.4. Encoder

Input Embedding: Máy tính không hiểu câu chữ mà chỉ đọc được số, vector, ma trận vì vậy ta phải biểu diễn câu chữ dưới dạng vector, gọi là input embedding. Điều này đảm bảo các từ gần nghĩa có vector gần giống nhau. Hiện đã có khá nhiều pretrained word embeddings như GloVe, Fasttext, gensim Word2Vec,...

Input Embedding



Hình 3. 29 Input Embedding

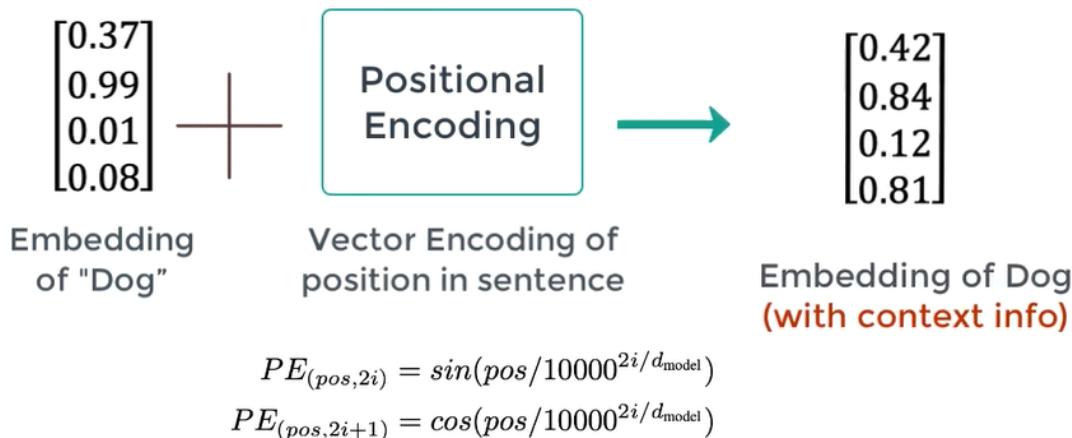
3.4.5. Positional Encoding

Word embeddings phần nào cho giúp ta biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Đó là lý do Transformers có thêm một phần Positional Encoding để inject thêm thông tin về vị trí của một từ

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Trong đó pos là vị trí của từ trong câu, PE là giá trị phần tử thứ i trong embedding có độ dài d_model. Sau đó ta cộng PE vector và Embedding vector.



Hình 3. 30 Position in sentence

3.4.6. Self-Attention

Self-Attention là cơ chế giúp Transformers "hiểu" được sự liên quan giữa các từ trong một câu. Ví dụ như từ "kicked" trong câu "I kicked the ball" (tôi đã đá quả bóng) liên quan như thế nào đến các từ khác? Rõ ràng nó liên quan mật thiết đến từ "I" (chủ ngữ), "kicked" là chính nó lên sẽ luôn "liên quan mạnh" và "ball" (vị ngữ). Ngoài ra từ "the" là giới từ nên sự liên kết với từ "kicked" gần như không có.

Quay trở lại với kiến trúc tổng thể ở trên, ta có thể thấy đầu vào của các module Multi-head Attention (bản chất là Self-Attention) có 3 mũi tên, đó chính là 3 vectors Querys (Q), Keys (K) và Values (V). Từ 3 vectors này, ta sẽ tính vector attention Z cho một từ theo công thức sau:

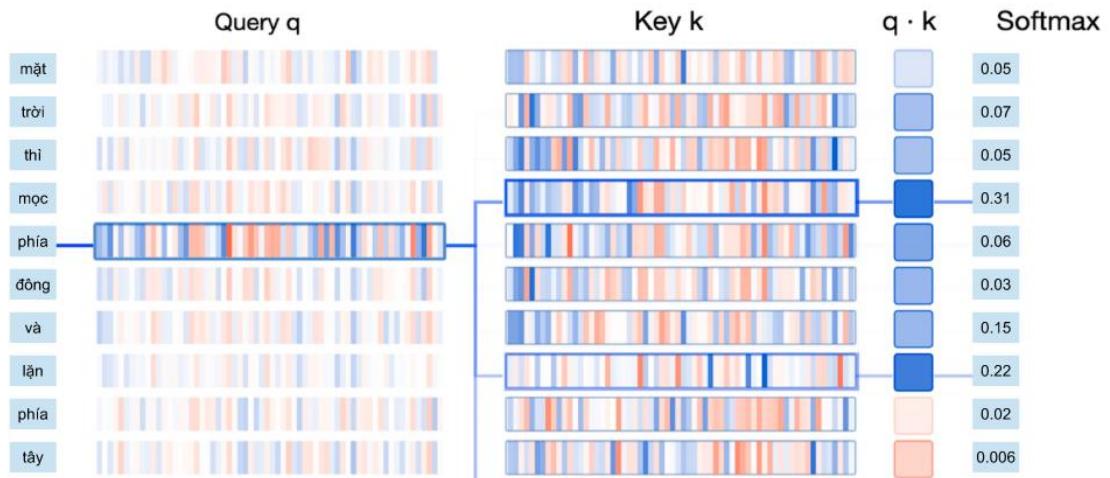
$$Z = \text{softmax} \left(\frac{Q \cdot K^T}{\sqrt{\text{Dimension of vector } Q, K \text{ or } V}} \right) \cdot V$$

Trong đó:

- Q là query vector: vector dùng để chứa thông tin của từ được tìm kiếm, so sánh. Giống như là câu query của google search.
- K là key vector: vector dùng để biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên. Ví dụ, như các trang web mà google sẽ so sánh với từ khóa mà bạn tìm kiếm.

- V là value vector: vector biểu diễn nội dung, ý nghĩa của các từ. Các bạn có thể tượng tượng, nó như là nội dung trang web được hiển thị cho người dùng sau khi tìm kiếm.

Để tính tương quan, chúng ta đơn giản chỉ cần tính tích vô hướng dựa các vector query và key. Sau đó dùng hàm softmax để chuẩn hóa chỉ số tương quan trong đoạn 0-1, và cuối cùng, tính trung bình cộng có trọng số giữa các vector values sử dụng chỉ số tương quan mới tính được.



Hình 3. 31 vector Q, K, V

Cụ thể hơn, quá trình tính toán attention vector có thể được tóm tắt làm 3 bước như sau:

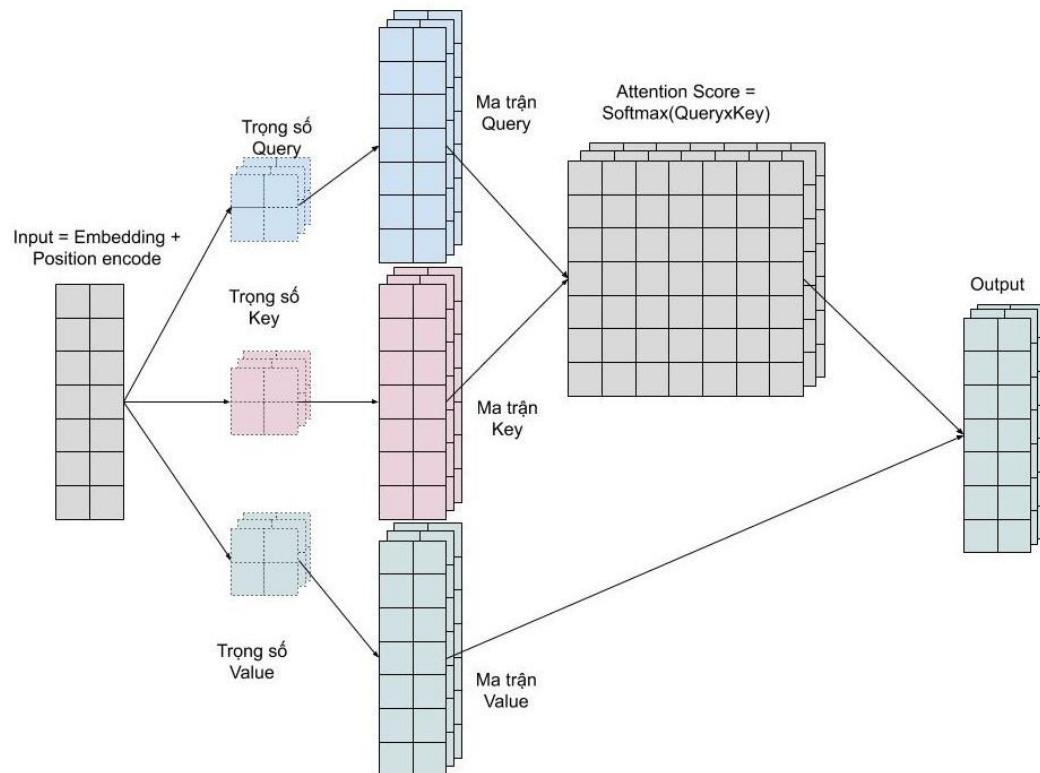
Bước 1: Tính ma trận query, key, value bằng cách khởi tạo 3 ma trận trọng số query, key, vector. Sau đó nhân input với các ma trận trọng số này để tạo thành 3 ma trận tương ứng.

Bước 2: Tính attention weights. Nhân 2 ma trận key, query vừa được tính ở trên với nhau để với ý nghĩa là so sánh giữa câu query và key để học mối tương quan. Sau đó thì chuẩn hóa về đoạn [0-1] bằng hàm softmax. 1 có nghĩa là câu query giống với key, 0 có nghĩa là không giống.

Bước 3: Tính output. Nhân attention weights với ma trận value. Điều này có nghĩa là chúng ta biểu diễn một từ bằng trung bình có trọng số (attention weights) của ma trận value.

Chúng ta muốn mô hình có thể học nhiều kiểu mối quan hệ giữa các từ với nhau. Với mỗi self-attention, chúng ta học được một kiểu pattern, do đó để có thể mở rộng khả năng này, chúng ta đơn giản là thêm nhiều self-attention. Tức là chúng ta cần

nhiều ma trận query, key, value mà thôi. Giờ đây ma trận trọng số key, query, value sẽ có thêm 1 chiều depth nữa.



Hình 3. 32 Multi-head Attention

Trong kiến trúc của mô hình transformer, residuals connection và normalization layer được sử dụng mọi nơi, giống như tinh thần của nó. 2 kỹ thuật giúp cho mô hình huấn luyện nhanh hơn và trách mắng thông tin trong quá trình huấn luyện mô hình, ví dụ như là thông tin của vị trí các từ được mã hóa.

3.4.7. Decoder

Decoder thực hiện chức năng giải mã vector của câu nguồn thành câu đích, do đó decoder sẽ nhận thông tin từ encoder là 2 vector key và value. Kiến trúc của decoder rất giống với encoder, ngoại trừ có thêm một multi head attention nằm ở giữa dùng để học mối liên quan giữ từ đang được dịch với các từ được ở câu nguồn.

3.4.8. Masked multi-head attention

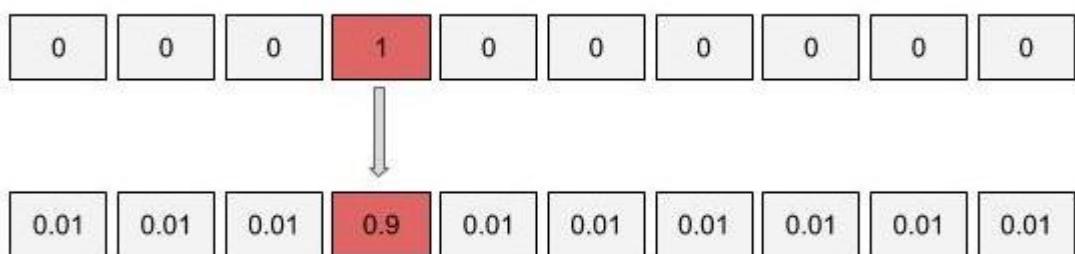
Masked Multi Head Attention tất nhiên là multi head attention mà chúng ta đã nói đến ở trên, có chức năng dùng để encode các từ câu câu đích trong quá trình dịch, tuy nhiên, lúc cài đặt chúng ta cần lưu ý rằng phải che đi các từ ở tương lai chưa được mô hình dịch đến, để làm việc này thì đơn giản là chúng ta chỉ cần nhân với một vector chứa các giá trị 0,1.

Trong decoder còn có một multi head attention khác có chức năng chú ý các từ ở mô hình encoder, layer này nhận vector key và value từ mô hình encoder, và output từ layer phía dưới. Đơn giản bởi vì chúng ta muốn so sánh sự tương quan giữa từ đang được dịch với các từ nguồn.

3.4.9. Optimizer

Để huấn luyện mô hình transformer, các bạn vẫn sử dụng Adam, tuy nhiên, learning rate cần phải được điều chỉnh trong suốt quá trình học. Cơ bản thì learning rate sẽ tăng dần trong các lần cập nhật đầu tiên, các bước này được gọi là warm up step, lúc này mô hình sẽ chạy. Sau đó learning rate lại giảm dần, để mô hình hội tụ.

Với mô hình nhiều triệu tham số của transformer, thì việt overfit là chuyện dễ dàng xảy ra. Để hạn chế hiện tượng overfit, chúng ta có thể sử dụng kỹ thuật label smoothing. Về cơ bản thì ý tưởng của kỹ thuật này khá đơn giản, chúng ta sẽ phạt mô hình khi nó quá tự tin vào việc dự đoán của mình. Thay vì mã hóa nhãn là một one-hot vector, các bạn sẽ thay đổi nhãn này một chút bằng cách phân bổ một tí xác suất vào các trường hợp còn lại.



Hình 3. 33 Label Smoothing

Giờ thì chúng ta sẽ an tâm khi có thể train với epoch lớn mà không lo rằng mô hình sẽ overfit nặng nề.

3.4.10. Tokenizer

Là một phần không thể thiếu khi huấn luyện Transformer. Nó thực hiện nhiệm vụ xử lý câu (chuỗi) thành những đầu vào dạng số và ngược lại. Tokenizer giúp cho việc huấn luyện mô hình diễn ra thuận lợi hơn. Mục tiêu của nó là tìm ra đại diện có ý nghĩa nhất cho mô hình và đồng thời là đại diện có kích thước nhỏ nhất có thể.

Có 3 loại tokenizer thường gặp trong Transformer là word-based tokenization (chia theo từ), character-based tokenization (chia theo ký tự) và subword-based tokenization (kết hợp 2 loại trên).

3.5. Conformer

3.5.1. Giới thiệu

Conformer (Convolution-augmented Transformer) là một mô hình tiên tiến được thiết kế để xử lý các tác vụ nhận dạng giọng nói tự động (ASR) và các ứng dụng âm thanh liên quan. Mô hình này được giới thiệu lần đầu tiên bởi Google vào năm 2020 trong bài báo "Conformer: Convolution-augmented Transformer for Speech Recognition". Conformer được phát triển để khắc phục những hạn chế của các mô hình truyền thống trong ASR, đặc biệt là những mô hình sử dụng các mạng nơ-ron hồi tiếp (RNN) hoặc Transformer đơn thuần.

Cốt lõi của Conformer là sự kết hợp giữa hai phương pháp mạnh mẽ trong lĩnh vực học sâu: self-attention từ Transformer và convolutional layers từ mạng nơ-ron tích chập (CNN). Trong khi Transformer được biết đến với khả năng nắm bắt các tương tác toàn cục trong dữ liệu, self-attention giúp mô hình học được các mối quan hệ xa, dài hạn trong chuỗi tín hiệu âm thanh, thì CNN lại nổi bật ở khả năng khai thác thông tin cục bộ. Convolution giúp mô hình bắt được các đặc trưng địa phương như âm điệu, ngữ điệu, hoặc các đặc điểm tần số ngăn chặn của tín hiệu âm thanh.

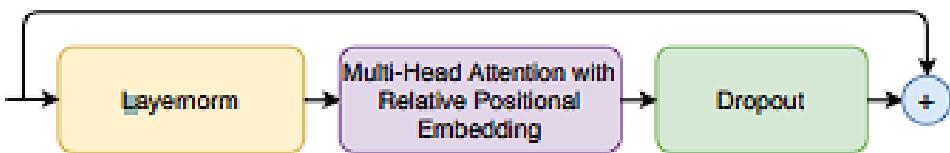
Sự kết hợp này giúp Conformer giải quyết được vấn đề mà các mô hình chỉ sử dụng Transformer gặp phải: chúng có thể nắm bắt ngữ cảnh toàn cục nhưng lại gặp khó khăn khi trích xuất các đặc trưng tinh tế ở cấp độ cục bộ. Bằng cách tích hợp cả convolution và self-attention, Conformer có thể học được cả các đặc trưng cục bộ và toàn cục trong một chuỗi tín hiệu âm thanh, mang lại khả năng nhận diện giọng nói chính xác hơn.

Cấu trúc của Conformer bao gồm các khối Conformer, mỗi khối này bao gồm một mô-đun Feed-forward, một mô-đun Multi-headed Self-Attention, một mô-đun Convolution, và một mô-đun Feed-forward thứ hai. Các mô-đun này được xếp chồng lên nhau theo một cách thức đặc biệt, với các lớp Feed-forward được đặt ở giữa các lớp Self-Attention và Convolution, tạo thành một cấu trúc "macaron-like". Cách thức này giúp tối ưu hóa việc kết hợp thông tin cục bộ và toàn cục trong một chuỗi tín hiệu âm thanh.

Kết quả thực nghiệm cho thấy, Conformer đạt được hiệu quả vượt trội trong các tác vụ nhận dạng giọng nói so với các mô hình trước đây, chẳng hạn như Transformer và RNN-based models, với độ chính xác (WER) cao hơn, đặc biệt trong các tập dữ liệu như LibriSpeech. Các nghiên cứu cũng chỉ ra rằng mô hình Conformer không chỉ tối ưu hóa hiệu suất nhận dạng giọng nói mà còn đạt được sự cân bằng tốt giữa chi phí tính toán và hiệu suất mô hình, mang lại một lựa chọn lý tưởng cho các hệ thống ASR hiện đại.

3.5.2. Multi-headed Self Attention Module

Chúng tôi sử dụng Multi-headed Self Attention Module (MHSA) kết hợp với một kỹ thuật quan trọng từ Transformer, đó là phương pháp mã hóa vị trí sóng hài tương đối. Mã hóa vị trí tương đối giúp mô-đun self-attention tổng quát hóa tốt hơn trên các đầu vào có độ dài khác nhau, làm cho bộ mã hóa trở nên mạnh mẽ hơn trước sự biến đổi về độ dài câu. Chúng tôi áp dụng các đơn vị residual theo kiểu prenorm cùng với dropout để hỗ trợ việc huấn luyện và điều chỉnh các mô hình sâu.



Hình 3. 34 Multi-headed Self Attention Module

3.5.3. Convolution Module

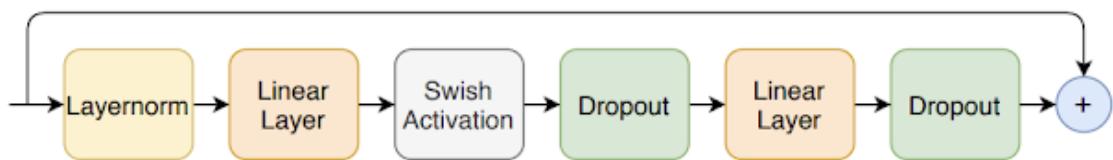
Convolution Module là thành phần quan trọng nhất trong việc cải thiện độ chính xác của Conformer, nhờ khả năng khai thác thông tin tương quan cục bộ trong dữ liệu âm thanh. Bên cạnh đó, cách thức sắp xếp Convolution Module và Multi-Head Self-Attention trong khối Conformer cũng đóng một vai trò quan trọng. Mô-đun Convolution được xây dựng dựa trên ý tưởng cơ chế cồng và bao gồm một số lớp cơ bản: lớp GLU (Gated Linear Unit) đầu tiên, sau là một lớp 1D Depthwise Convolution, tiếp đó là một lớp BatchNorm giúp cải thiện khả năng học sâu của mô hình, và lớp kích hoạt Swish ở cuối cùng. Xung quanh phần lõi này, hai lớp Pointwise Convolution được sử dụng để hoàn thiện mô-đun. Ngoài ra, mô-đun còn sử dụng LayerNorm ở đầu vào và Dropout ở cuối để giảm thiểu hiện tượng overfitting.



Hình 3. 35 Convolution Module

Việc bổ sung mô-đun Convolution để hỗ trợ cho MHSA trong Transformer mang lại hiệu quả cao, tuy nhiên, vị trí và cách thức sắp xếp của mô-đun này trong mạng cũng rất quan trọng.

3.5.4. Feed Forward Module

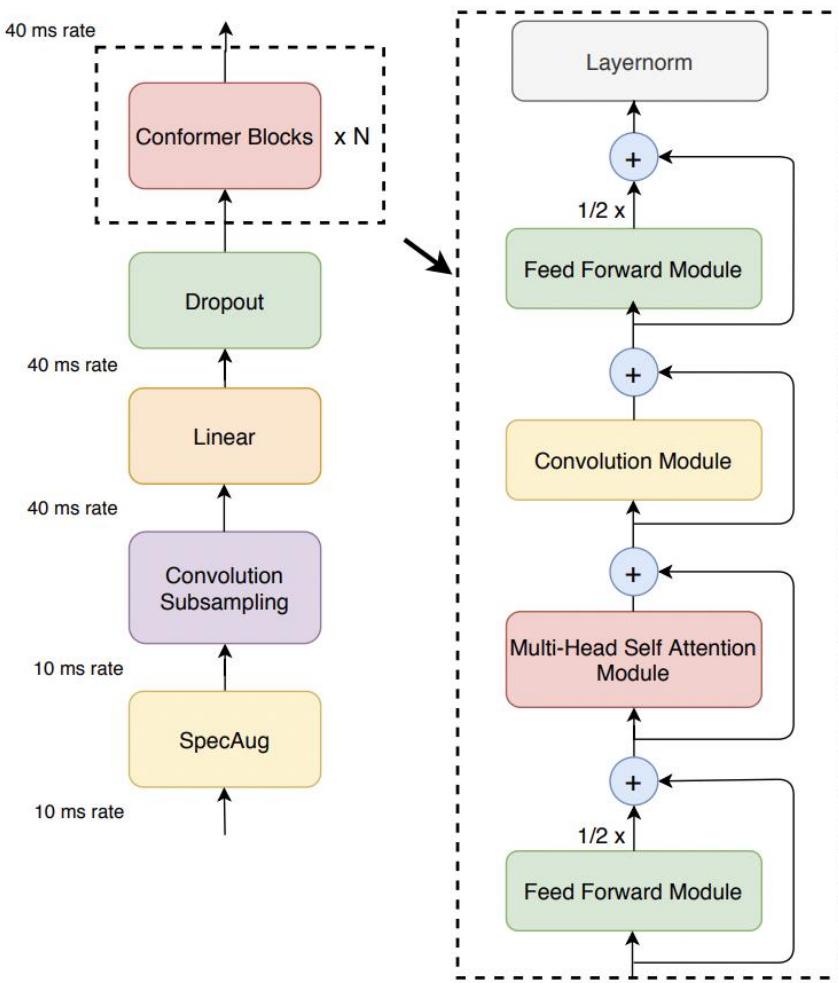


Hình 3. 36 Kiến trúc của mô đun Feed Forward

Mô đun Feed Forward trong Conformer được phát triển lên từ mô đun feed forward trong Transformer ASR [4]. Trong đó kiến trúc chi tiết của mô đun này được thể hiện trong hình 4, trong đó một LayerNorm được đặt trước LinearLayer đầu tiên, sau Linear layer thì Swish Activation và Dropout được sử dụng để giúp “regularizing” model. Ngoài ra một residual connection giữa đầu vào và đầu ra của layer này cũng được sử dụng để giúp model học được sâu hơn.

3.5.5. Conformer Block

Trong bộ mã hóa (Encoder) của mô hình Transducer, các khối Transformer truyền thống được thay thế bằng các khối Conformer. Mỗi khối Conformer được cấu tạo từ bốn mô-đun chính: một mô-đun Feed-forward, một mô-đun Multi-headed Self-Attention (MHSA), một mô-đun Convolution, và một mô-đun Feed-forward thứ hai. Kiến trúc tổng thể của khối Conformer .



Hình 3. 37 Kiến trúc Conformer Block

Phát triển từ ý tưởng của Macron-Net (theo phong cách Macron style), thay vì chỉ sử dụng một mô-đun Feed-forward (FFN), Conformer Block tích hợp hai mô-đun FFN "nửa bước" (half-step FFN), được đặt trước và sau các mô-đun MHSA và Convolution. Cấu trúc toán học của một khối Conformer được biểu diễn như sau:

$$\begin{aligned}
 \tilde{x}_i &= x_i + \frac{1}{2} \text{FFN}(x_i) \\
 x'_i &= \tilde{x}_i + \text{MHSA}(\tilde{x}_i) \\
 x''_i &= x'_i + \text{Conv}(x'_i) \\
 y_i &= \text{Layernorm}(x''_i + \frac{1}{2} \text{FFN}(x''_i))
 \end{aligned}$$

Hệ số 1/2 trước mỗi FFN biểu thị trọng số du nửa bước (half-step residual) theo phong cách Macron style. Theo bài báo gốc về Conformer, việc áp dụng Macron style giúp cải thiện đáng kể tỷ lệ lỗi từ (WER), so với việc chỉ sử dụng một FFN duy nhất. Tuy nhiên, kết quả thử nghiệm trong phần nghiên cứu Ablation Studies (Bảng 3 và Bảng 5) lại không hoàn toàn khẳng định điều này. Trong phần lớn các bài kiểm tra, việc sử dụng Macron style không mang lại sự cải thiện rõ rệt, mà chỉ giúp giảm 0.2% WER trong trường hợp test_other.

Dù phương pháp Macron style mang lại một số lợi ích nhỏ về độ chính xác, lợi ích đó chưa đủ thuyết phục để bù đắp chi phí tính toán bổ sung mà nó gây ra. Do đó, vai trò thực sự của Macron style trong việc cải thiện hiệu quả nhận dạng vẫn còn là một câu hỏi cần nghiên cứu thêm.

CHƯƠNG 4 MÔ HÌNH THỰC NGHIỆM

4.1. Chuẩn bị dữ liệu

4.1.1. Thu thập dữ liệu

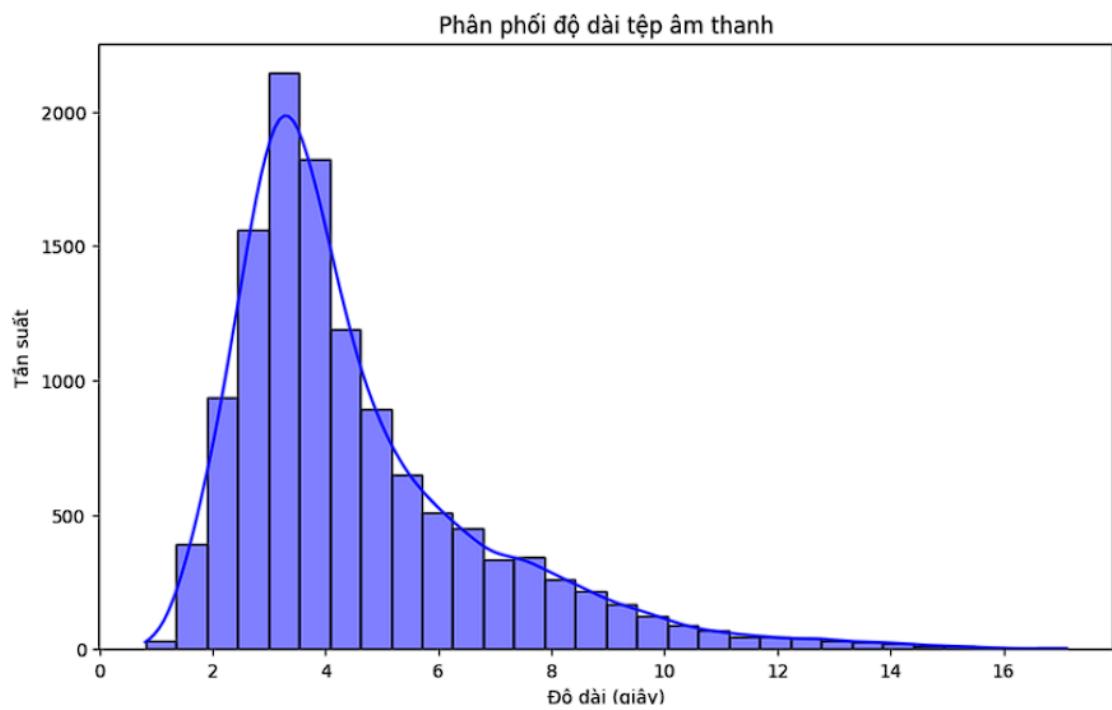
Nguồn: Tập dữ liệu này được tạo ra trong khuôn khổ dự án VIVOS, được phát triển bởi AILAB, thuộc Đại học Khoa học Tự nhiên, Đại học Quốc gia TP. Hồ Chí Minh.

Mục đích: Nhằm hỗ trợ các nghiên cứu về nhận diện giọng nói và xử lý ngôn ngữ tự nhiên tiếng Việt.

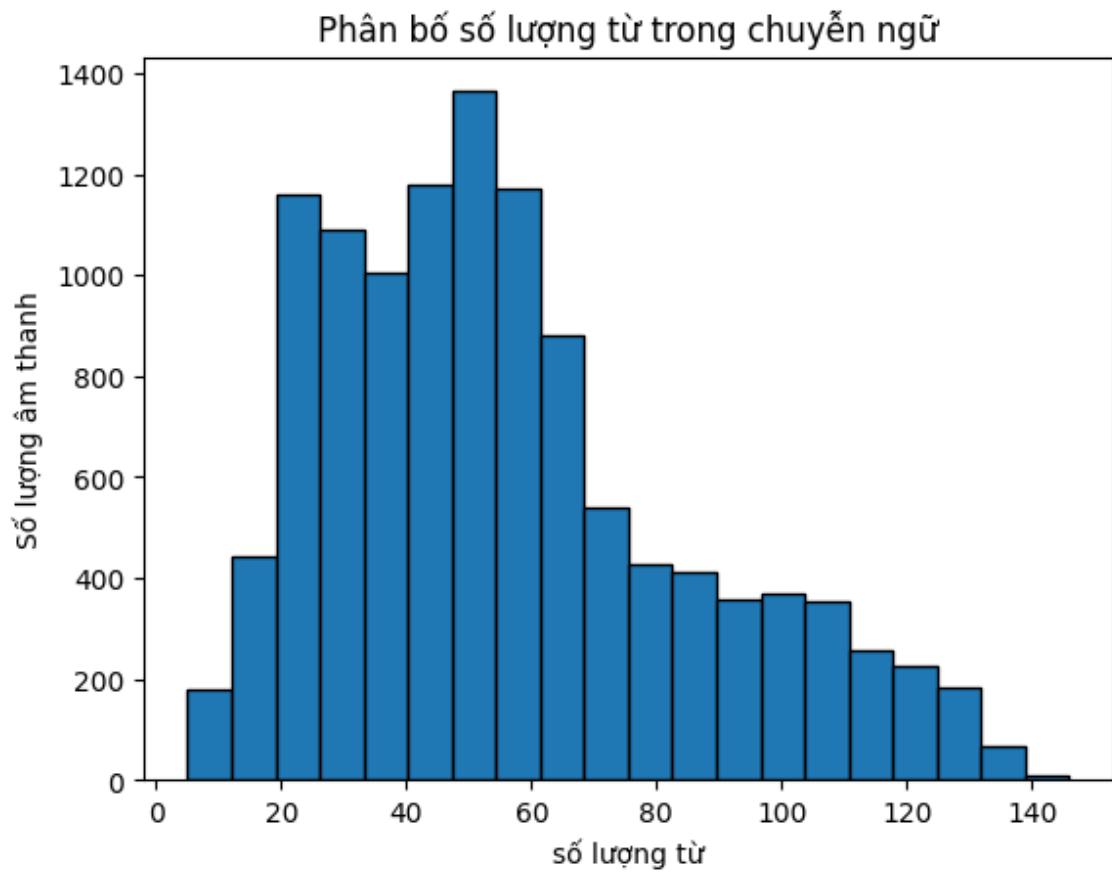
Dữ liệu: [VIVOS Vietnamese | Kaggle](#)

Kích thước:

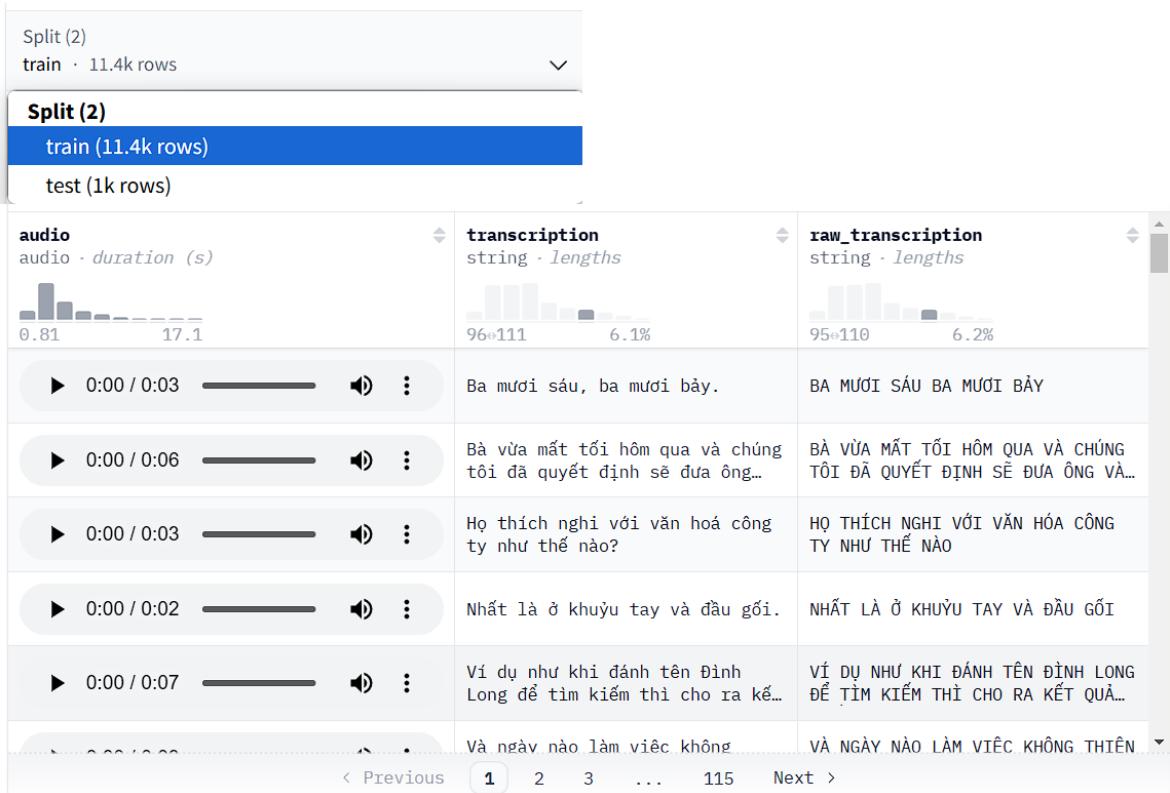
- Số lượng bản ghi âm: 12420
- Tổng thời lượng: 15.67 giờ
- Độ dài trung bình: 8 giây
- Độ dài tối đa: 29.4 giây
- Độ dài tối thiểu: 1.42 giây
- Nội dung: Đa dạng
- Số lượng người nói: 65
- Vùng miền: Đa dạng
- File âm thanh: .wav



Hình 4. 1 Phân phối độ dài tệp âm thanh



Hình 4. 2 Phân bố số lượng từ



Hình 4. 3 Thông tin về dữ liệu

4.1.2. Tiền xử lý dữ liệu

Chuẩn hóa tần số lấy mẫu (Resampling):

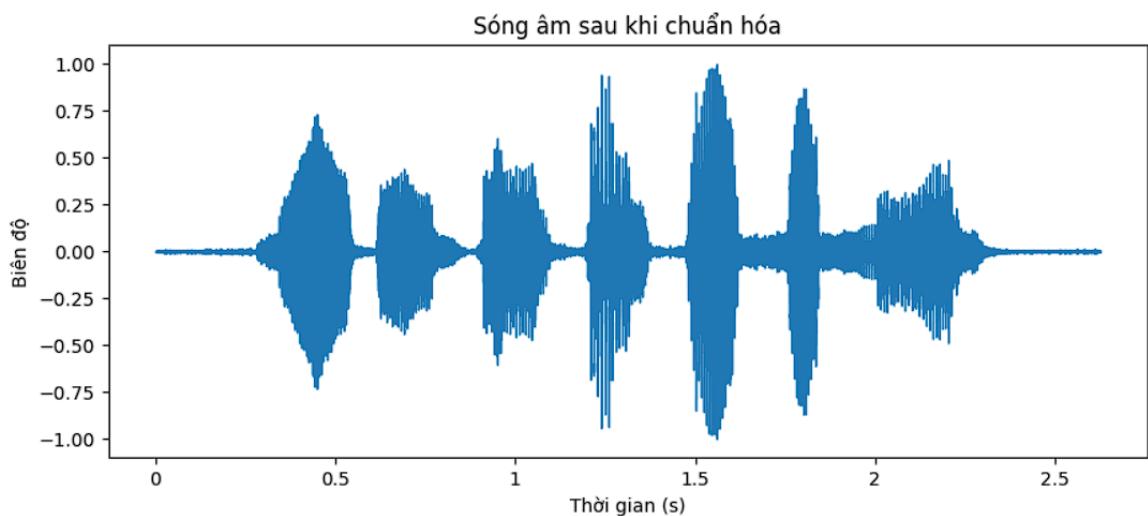
- Đảm bảo tất cả các tệp âm thanh có cùng tần số lấy mẫu.
- Chuyển đổi tần số lấy mẫu về 16kHz phổ biến cho các hệ thống ASR

Chuyển đổi âm thanh sang dạng đơn âm (Mono Conversion):

- Mục tiêu: Nếu âm thanh có nhiều kênh (stereo), chuyển đổi về một kênh duy nhất (mono).
- Sử dụng công cụ xử lý âm thanh để trộn các kênh thành mono.

Chuẩn hóa âm lượng:

- Điều chỉnh âm lượng của tất cả các file âm thanh về cùng một mức. Điều này giúp tránh trường hợp một số file quá to hoặc quá nhỏ so với các file khác, ảnh hưởng đến quá trình huấn luyện
- Giảm nhiễu và cải thiện chất lượng



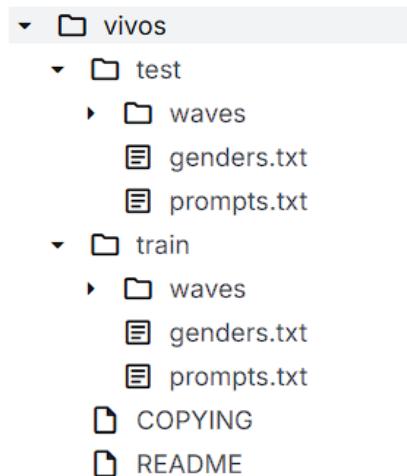
Hình 4. 4 Chuẩn hóa âm thanh

4.1.3. Chuẩn bị dữ liệu huấn luyện

Phân chia dữ liệu:

- Tập huấn luyện (train): ~11.400 mẫu.
- Tập kiểm tra (test): ~1.000 mẫu.

Cấu trúc thư mục của bộ dữ liệu



Summary

▼	12.4k files	
♫	.wav	12.4k
☰	.txt	4
📁		2

Hình 4. 5 Cấu trúc thư mục của bộ dữ liệu

```

[18] 1 !python preprocessing/preprocess_vivos.py
      2 ..... --dataset_dir=/content/drive/MyDrive/ColabNotebooks/KHOALUAN/vivos
      3 ..... --subset=train
      4 ..... --save_filepath=data/data.txt

→ Loading data from: /content/drive/MyDrive/ColabNotebooks/KHOALUAN/vivos
100% 11420/11420 [00:00<00:00, 152196.43it/s]
100% 11420/11420 [00:02<00:00, 288.57it/s]
saved to: data/data.txt

▶ 1 !python preprocessing/preprocess_vivos.py
      2 ..... --dataset_dir=/content/drive/MyDrive/ColabNotebooks/KHOALUAN/vivos
      3 ..... --subset=test
      4 ..... --save_filepath=data/test.txt

→ Loading data from: /content/drive/MyDrive/ColabNotebooks/KHOALUAN/vivos
100% 1000/1000 [00:00<00:00, 161001.62it/s]
100% 1000/1000 [00:02<00:00, 257.53it/s]
saved to: data/test.txt

```

Hình 4. 6 Phân chia dữ liệu

4.2. Xây dựng mô hình

4.2.1. Lựa chọn mô hình

Thông qua bài báo “Conformer: Convolution-augmented Transformer for Speech Recognition” nên tôi quyết định lựa chọn mô hình Conformer cho bài toán nhận dạng giọng nói, với khả năng vượt trội của nó trong việc nắm bắt cả thông tin cục bộ và phụ thuộc dài hạn trong tín hiệu âm thanh. Kiến trúc này kết hợp hiệu quả CNN và Transformer, đã được chứng minh đạt hiệu suất cao trên nhiều bộ dữ liệu.

Ưu điểm của Conformer

- Kết hợp CNN và Transformer: Conformer kết hợp khả năng học ngữ cảnh toàn cục của Transformer và đặc trưng cục bộ của CNN. Điều này rất phù hợp với dữ liệu âm thanh, nơi cả thông tin ngữ cảnh toàn cục và cấu trúc cục bộ đều cần được chú trọng.
- Hiệu quả với dữ liệu âm thanh: Module Convolution trong Conformer đặc biệt được thiết kế để học các đặc trưng phổ biến trong tín hiệu âm thanh, chẳng hạn như biến thiên tần số ngắn hạn và cấu trúc âm thanh theo thời gian.
- Hiệu năng vượt trội: Theo kết quả từ bài báo gốc, Conformer vượt qua Transformer trong hầu hết các bài toán ASR. Điều này cho thấy Conformer không chỉ mạnh mẽ hơn mà còn tối ưu hơn về mặt tài nguyên.

So sánh với các mô hình khác:

Mô hình	Ưu điểm	Nhược điểm
LSTM	Tốt cho chuỗi tuần tự ngắn, dễ triển khai.	Khó xử lý ngữ cảnh dài, hiệu năng thấp.
Transformer	Xử lý tốt ngữ cảnh toàn cục, song song hóa hiệu quả.	Hiệu năng kém trên dữ liệu âm thanh dài.
CNN	Tốt trong học thông tin cục bộ, tính toán nhanh.	Không xử lý được ngữ cảnh toàn cục.
Conformer	Kết hợp ưu điểm của Transformer và CNN.	Đòi hỏi tài nguyên tính toán lớn hơn các mô hình khác.

4.2.2. Huấn luyện mô hình

Thành phần trong cấu trúc train:

- **output_dir** (Directory to store training results): Thư mục mà các kết quả, mô hình, và log sẽ được lưu trữ trong quá trình huấn luyện.
- **num_epoch** (Number of epochs): Số lần toàn bộ dữ liệu được đưa qua mô hình để huấn luyện.
- **warmup_steps** (Warmup steps for learning rate scheduling): Số bước trong giai đoạn warmup, nơi tốc độ học tăng dần từ 0 đến giá trị tối đa.
- **lr** (Learning rate): Tốc độ học cơ bản được sử dụng trong quá trình tối ưu hóa.
- **weight_decay** (Weight decay for regularization): Hệ số giảm trọng số, giúp giảm overfitting.
- **acc_steps** (Gradient accumulation steps): Số bước để tích lũy gradient trước khi thực hiện cập nhật trọng số.
- **pretrained_path** (Path to pretrained checkpoint): Đường dẫn tới một checkpoint của mô hình đã được huấn luyện trước, dùng để tiếp tục huấn luyện từ trạng thái hiện tại.
- **wandb_config** (Configuration for monitoring tools like Weights & Biases): Cấu hình cho công cụ theo dõi hiệu suất mô hình.
- **tag** (Model tag): Nhãn hoặc tên mô hình cụ thể để dễ dàng theo dõi.

```
train:  
    output_dir: exps/vivos/conformer  
    num_epoch: 50  
    warmup_steps: 15000  
    lr: 0.001  
    weight_decay: 0  
    acc_steps: 2  
    pretrained_path: null  
    wandb_config:  
        project: vivos  
        tag: conformer
```

Hình 4. 7 Cấu trúc huấn luyện mô hình

Cấu trúc huấn luyện:

- Mô hình sẽ huấn luyện trong 50 epoch, mỗi epoch gồm nhiều batch với kích thước 16.
- Tốc độ học sẽ tăng dần trong 15,000 bước đầu và ổn định ở mức 0.001.
- Các kết quả và mô hình sẽ được lưu trong exps/vivos/conformer.

Bắt đầu train với 1 epoch

```
ch:118 - [TRAIN] EPOCH 1 | BATCH 100/647 | loss=139.1802978515625 | ctc_loss=550.15087890625 | decoder_loss=6.570314884185791
ch:121 - + Label : tám mươi tám mươi mốt
ch:122 - + Predict:
ch:118 - [TRAIN] EPOCH 1 | BATCH 200/647 | loss=69.02816009521484 | ctc_loss=269.595458984375 | decoder_loss=6.517172336578369
ch:121 - + Label : có thể khiến người bị sa thải cảm thấy ngót ngọt
ch:122 - + Predict:
ch:118 - [TRAIN] EPOCH 1 | BATCH 300/647 | loss=32.72514724731445 | ctc_loss=124.58273315429688 | decoder_loss=6.3178606033325195
ch:121 - + Label : từ khi họ còn là trẻ con
ch:122 - + Predict:
ch:118 - [TRAIN] EPOCH 1 | BATCH 400/647 | loss=32.46689978182617 | ctc_loss=123.69579315185547 | decoder_loss=6.171804904937744
ch:121 - + Label : vùn ngó lơ khách qua đường
ch:122 - + Predict:
ch:118 - [TRAIN] EPOCH 1 | BATCH 500/647 | loss=27.10342788696289 | ctc_loss=102.52003479003906 | decoder_loss=5.893679141998291
ch:121 - + Label : chúng sở hữu tiếng thết chói tai và rất thích nhảy nhót
ch:122 - + Predict:
ch:118 - [TRAIN] EPOCH 1 | BATCH 600/647 | loss=24.930028915485273 | ctc_loss=94.15489196777344 | decoder_loss=5.565227031707764
ch:121 - + Label : những đêm như vậy khi chị ngủ lung cũng là lúc trời gần sáng
ch:122 - + Predict:
- [TRAIN] STATS: {'train_loss': 63.81443129809231, 'train_ctc_loss': 249.0348334806227, 'train_decoder_loss': 6.222891910735754}
- [TRAIN] EPOCH 1/50 DONE, Save checkpoint to: exps/vivos/conformer_v1/checkpoint_epoch_1.pt
- [VALID] EPOCH 1/50 START
- [VALID] STATS: {'valid_loss': 55.88507339689467, 'valid_ctc_loss': 106.43483776516385, 'valid_decoder_loss': 5.33530941274431, 'valid_wer': 100.0, 'valid_cer': 100.0}
- saved best model to exps/vivos/conformer_v1/valid_loss_best.pt
- [VALID] EPOCH 1/50 DONE
```

Hình 4. 8 Train 1/50 epoch

Hoàn thành train 50 epoch

```
- [TRAIN] EPOCH 50/50 START
ch:118 - [TRAIN] EPOCH 50 | BATCH 100/647 | loss=2.7561097145080566 | ctc_loss=11.005488395690918 | decoder_loss=0.018950030207633972
ch:121 - + Label : bảy mươi sáu bảy mươi bảy
ch:122 - + Predict: bảy mươi sáu bảy mươi bảy
ch:118 - [TRAIN] EPOCH 50 | BATCH 200/647 | loss=3.296297788619995 | ctc_loss=13.1634750366212094 | decoder_loss=0.02171650156378746
ch:121 - + Label : chiếc sà lan di tiếp qua cầu bình điện và dừng lại tái một bãi cát ở xã tan kiên huyện bình chánh
ch:122 - + Predict: chiếc xe lan di tiếp qua cầu bến đê và dừng lại tái một sà lan bãi cát ở xã tan kiên nguyên bến tránh
ch:118 - [TRAIN] EPOCH 50 | BATCH 300/647 | loss=1.9610838890075684 | ctc_loss=7.824678897857666 | decoder_loss=0.01965652033686638
ch:121 - + Label : từ khỉ có hỏa định kiến đó đã được thay đổi có ánh như một người quản gia của chúng tôi ch chăm sóc mọi chuyện
ch:122 - + Predict: từ khỉ có hỏa định kiến nó đã được thay đổi có ánh như một người quản gia của chúng tôi ch chăm sóc mọi chuyện
ch:118 - [TRAIN] EPOCH 50 | BATCH 400/647 | loss=1.6585406064987183 | ctc_loss=6.613602638244629 | decoder_loss=0.020559942349791427
ch:121 - + Label : ông nói thêm
ch:122 - + Predict: ông nói thêm
ch:118 - [TRAIN] EPOCH 50 | BATCH 500/647 | loss=3.9902163410186768 | ctc_loss=15.582699238952637 | decoder_loss=0.018174832686781883
ch:121 - + Label : bói vì trong kĩ thuật đó có chứa bao nỗi lo to con về một công ty vừa mới thành lập
ch:122 - + Predict: bói vì trong kĩ thuật đó có chứa bao nỗi lo to con về một công ty vừa mới thành lập
ch:118 - [TRAIN] EPOCH 50 | BATCH 600/647 | loss=4.043429374694824 | ctc_loss=16.153804779052734 | decoder_loss=0.019911766052246094
ch:121 - + Label : đó không thông báo tạm ngừng hoạt động nhưng vẫn kinh doanh bình thường
ch:122 - + Predict: đó không dám ngừng hoạt động nhưng vẫn kinh doanh bình thường
- [TRAIN] STATS: {'train_loss': 2.776447252912474, 'train_ctc_loss': 11.085130677525374, 'train_decoder_loss': 0.02065833931762892}
- [TRAIN] EPOCH 50/50 DONE, Save checkpoint to: exps/vivos/conformer_v1/checkpoint_epoch_50.pt
- [VALID] EPOCH 50/50 START
- [VALID] STATS: {'valid_loss': 10.044294867250654, 'valid_ctc_loss': 28.066654827859665, 'valid_decoder_loss': 0.02193494617111153, 'valid_wer': 34.03012807751012, 'valid_cer': 18.48}
- [VALID] EPOCH 50/50 DONE
```

Hình 4. 9 Train 50/50 epoch

4.2.3. Đánh giá mô hình

Epoch 1/50:

Đây là giai đoạn đầu của quá trình huấn luyện. Loss ở mức cao train_loss= 63.81

Độ chính xác của kết quả dự đoán (predict) cũng thấp, thể hiện qua sự khác biệt lớn giữa label và predict.

- **Label:** tươi mòn mòn murót
- **Predict:** t

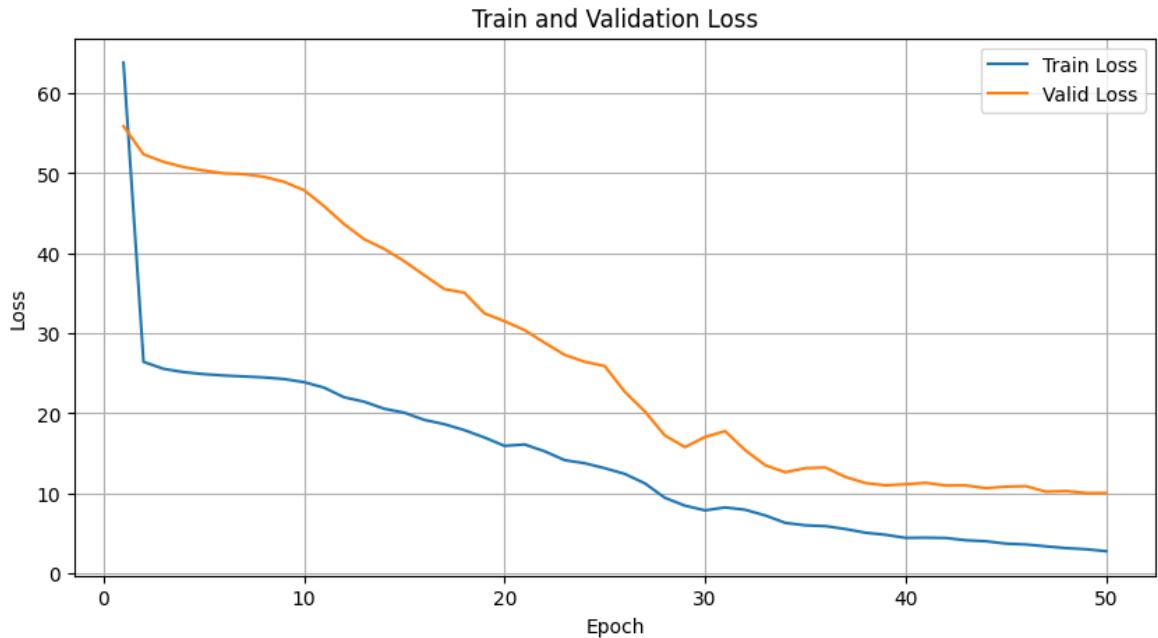
Epoch 50/50:

Đây là giai đoạn sau của quá trình huấn luyện. Loss đã giảm đáng kể so với epoch 1 train_loss= 2.77

Độ chính xác của kết quả dự đoán cũng được cải thiện đáng kể. Kết quả predict gần với label hơn.

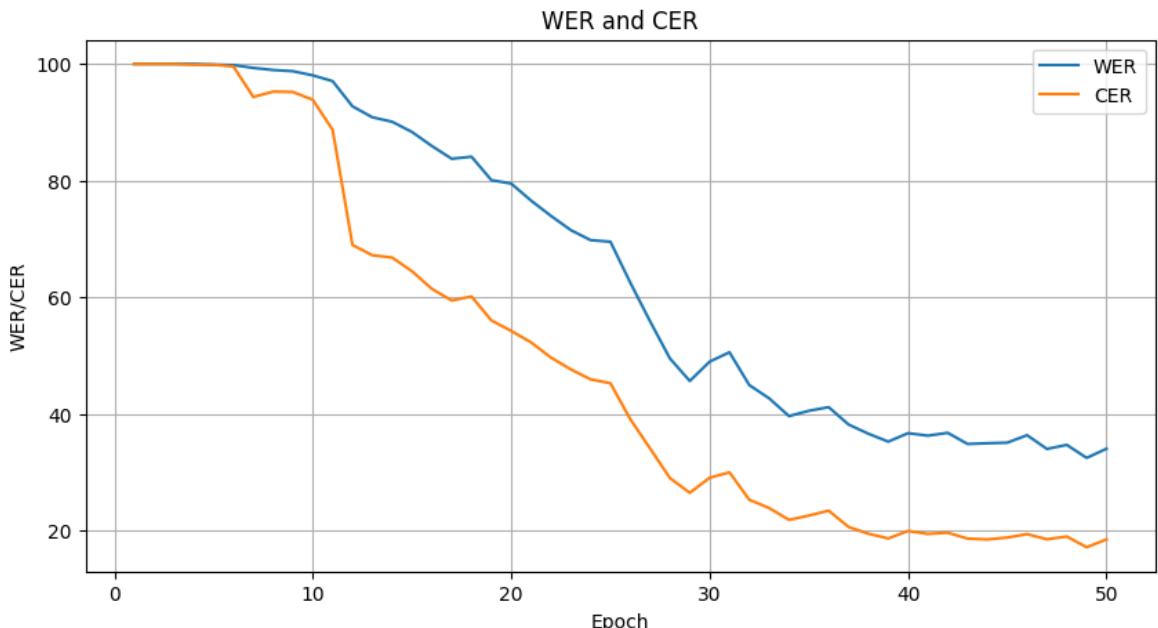
- **Label:** bảy mươi sáu bảy mươi bảy
- **Predict:** bảy mươi sáu bảy mươi bảy

Biểu đồ Loss cho thấy cả train loss và valid loss đều giảm dần theo epoch. Điều này cho thấy mô hình đang học tốt và không bị overfitting.



Hình 4. 10 Biểu đồ loss của 50 epoch

Biểu đồ WER/CER cũng cho thấy Tỷ lệ lỗi từ (WER - Word Error Rate) và tỷ lệ lỗi ký tự (CER - Character Error Rate) đều giảm dần theo số epoch



Hình 4. 11 Biểu đồ tỉ lệ lỗi từ và lỗi ký tự theo 50 epoch

- Epoch (1–10):

Mô hình học nhanh, loss giảm mạnh, nhưng WER và CER không cải thiện rõ rệt.

- Epoch (10–30):

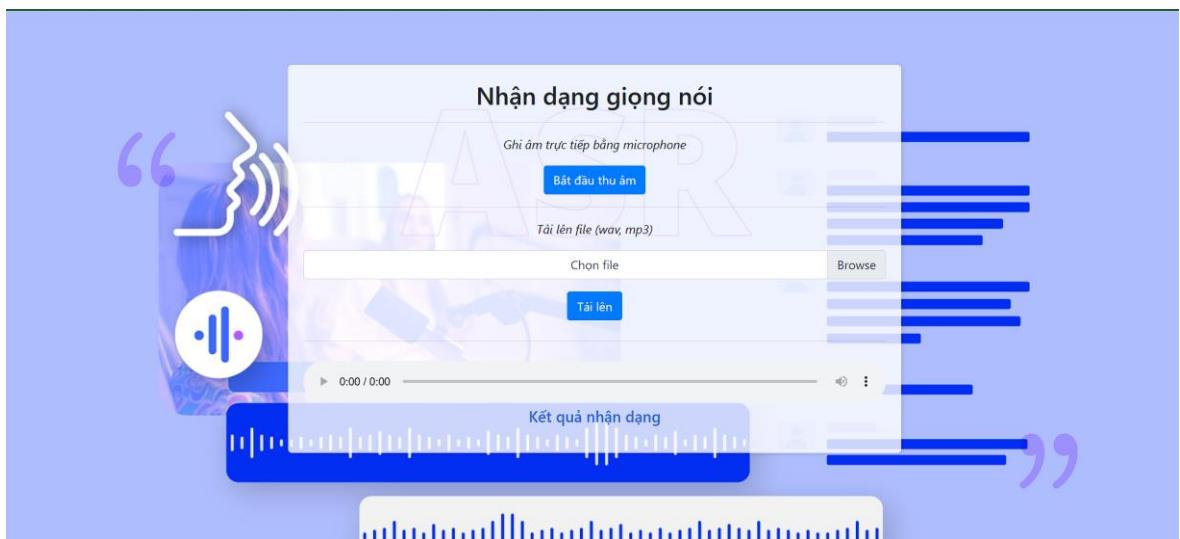
Loss giảm ổn định, WER và CER bắt đầu giảm, chứng minh mô hình đã học được mối quan hệ tốt hơn giữa dữ liệu đầu vào và đầu ra.

- Epoch (30–50):

Train loss gần như ổn định, valid loss giảm chậm, WER và CER giảm rõ rệt, đặc biệt sau epoch 40.

4.3. Xây dựng giao diện và tích hợp mô hình

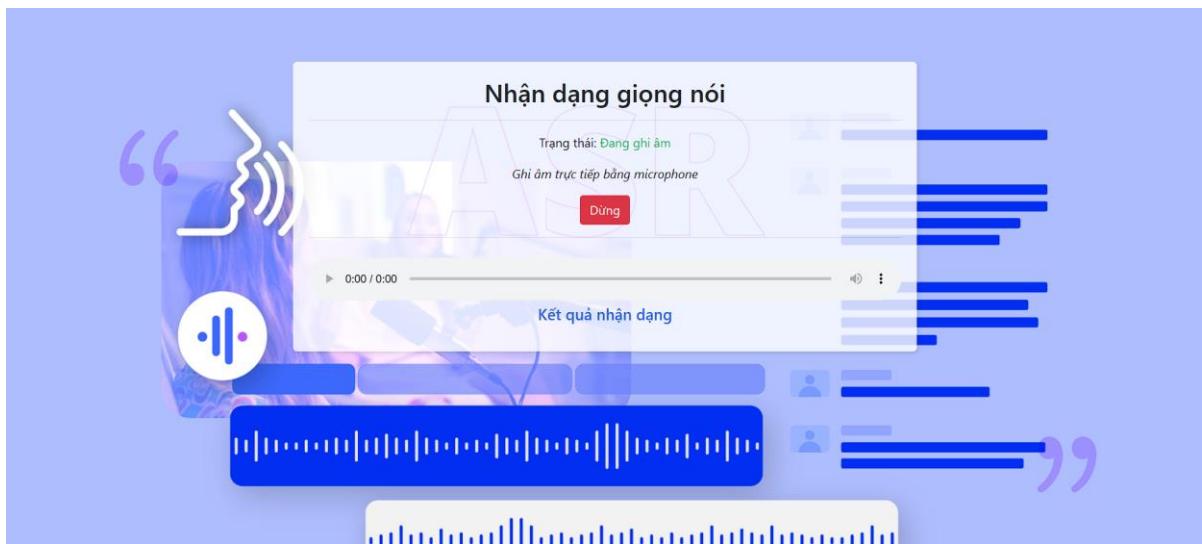
Giao diện chính



Hình 4. 12 Giao diện chính

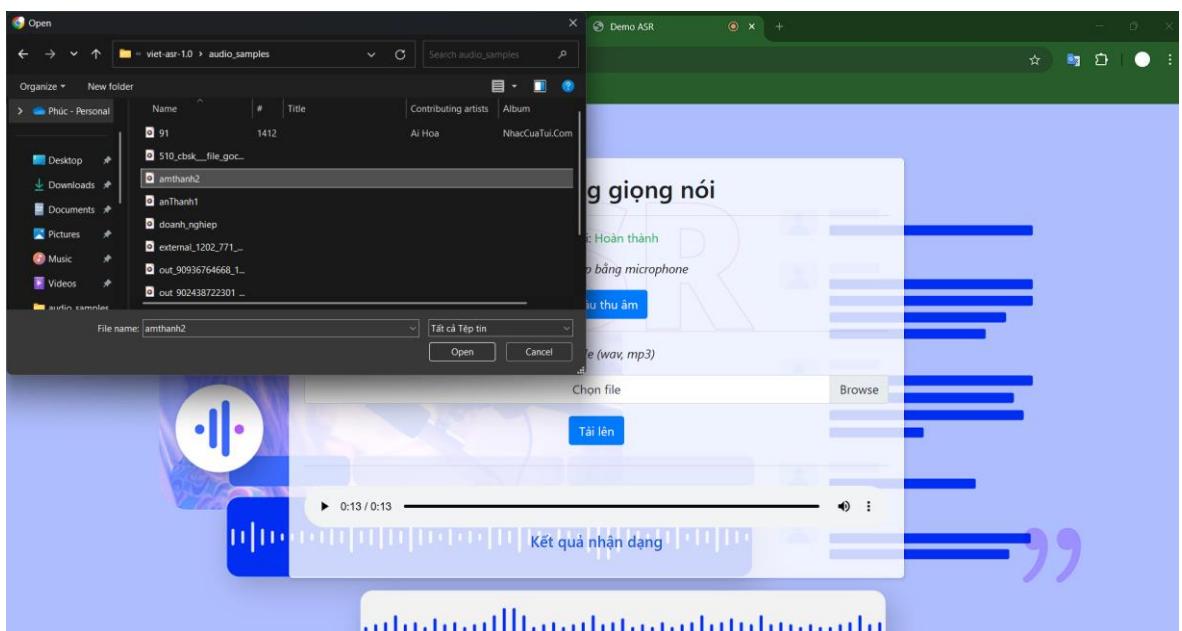
Cách hoạt động: Có thể lấy âm thanh qua việc thu âm hoặc tải file lên từ máy tính

Giao diện khi thu âm



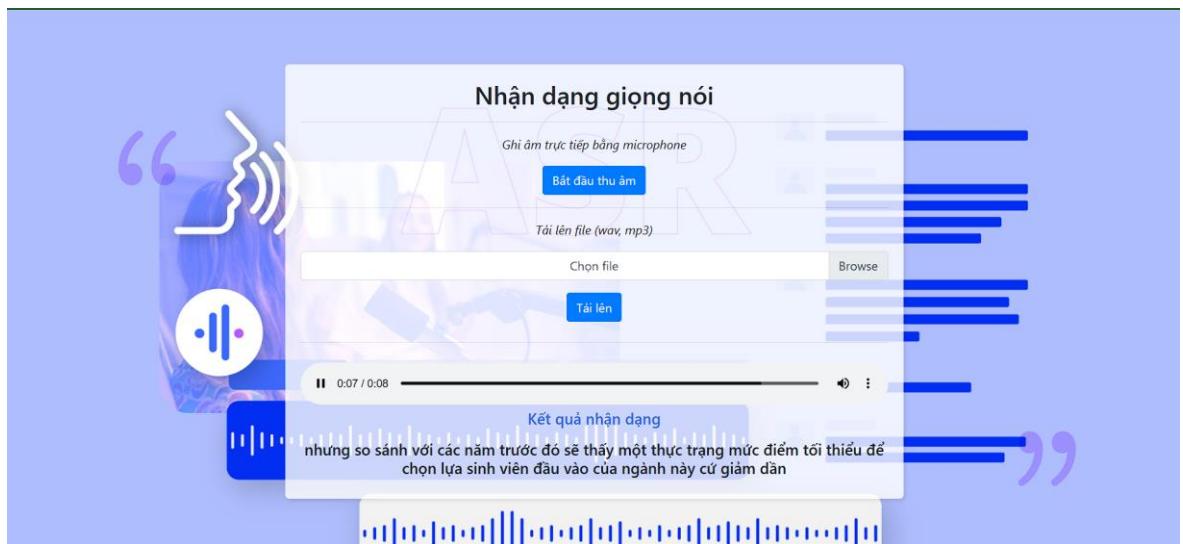
Hình 4. 13 Giao diện khi thu âm

Giao diện khi chọn file từ máy tính



Hình 4. 14 Giao diện khi chọn file từ máy tính

Trả về kết quả sau khi xử lý



Hình 4. 15 Trả về kết quả sau khi xử lý

CHƯƠNG 5 KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết luận

Nghiên cứu này tập trung vào việc xây dựng một hệ thống chuyển đổi âm thanh tiếng Việt sang văn bản, giải quyết các thách thức trong xử lý ngôn ngữ tự nhiên. Những vấn đề như môi trường tạp và sự thiếu hụt dữ liệu huấn luyện chất lượng cao đã được nhận diện và xử lý trong phạm vi nghiên cứu.

Hệ thống nhận dạng giọng nói tự động (ASR) được phát triển dựa trên mô hình Conformer-Transformer, tận dụng sức mạnh của Convolutional Neural Networks (CNN) và cơ chế attention của Transformer để tối ưu hóa khả năng nhận dạng tiếng Việt. Kết quả thực nghiệm cho thấy hệ thống đạt độ chính xác cao trên tập dữ liệu kiểm thử, vượt trội so với một số phương pháp truyền thống.

Hệ thống đã chứng minh khả năng xử lý tốt giọng nói trong môi trường tạp âm, tuy nhiên vẫn còn gặp khó khăn với giọng nói vùng miền hoặc tốc độ nói nhanh.

5.2. Kiến nghị và hướng phát triển

Để nâng cao hiệu quả của hệ thống và mở rộng phạm vi ứng dụng, chúng tôi đề xuất một số hướng nghiên cứu tiếp theo:

- **Mở rộng dữ liệu huấn luyện:** Thu thập thêm dữ liệu âm thanh từ nhiều vùng miền, ngữ cảnh và phong cách nói khác nhau để cải thiện khả năng tổng quát hóa của mô hình. Đặc biệt, cần tập trung vào các dữ liệu khó như giọng nói trẻ em, người cao tuổi, và trong môi trường nhiều tạp âm.
- **Khám phá các kiến trúc mô hình mới:** Nghiên cứu và áp dụng các kiến trúc mạng nơ-ron tiên tiến hơn, ví dụ như các biến thể của Transformer hoặc Conformer, để nâng cao độ chính xác và hiệu suất của hệ thống.
- **Phát triển ứng dụng thực tế:** Ứng dụng hệ thống đã xây dựng vào các sản phẩm và dịch vụ cụ thể, ví dụ như phụ đề tự động, trợ lý ảo, hoặc hệ thống ghi chép y tế, để đánh giá hiệu quả thực tế và thu thập phản hồi từ người dùng.
- **Nghiên cứu xử lý giọng nói đa dạng:** Tập trung vào việc xử lý các trường hợp đặc biệt như giọng nói bị ảnh hưởng bởi cảm xúc, giọng nói trong môi trường ồn ào, hoặc giọng nói có âm sắc đặc biệt.

DANH MỤC TÀI LIỆU THAM KHẢO

1. L. Zhang, C. Li, F. Deng and X. Wang, "Multi-task audio source separation", 2021. <https://ieeexplore.ieee.org/abstract/document/9687922>
2. Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolutionaugmented transformer for speech recognition," , 2020. <https://arxiv.org/pdf/2005.08100>
3. Abel Chandra, Laura Tünnermann, Tommy Löfstedt, Regina Gratz," Transformer-based deep learning for predicting protein properties in the life sciences", 2022, <https://elifesciences.org/articles/82819>
4. Samik Sadhu, Ruizhi Li, Hynek Hermansky, "M-vectors: Sub-band Based Energy Modulation Features for Multi-stream Automatic Speech Recognition", 2019, <https://ieeexplore.ieee.org/document/8682710>
5. Kai Han, An Xiao, Enhua Wu, Jianyu Guo, Chunjing XU, Yunhe Wang, "Transformer in Transformer", 2021, https://proceedings.neurips.cc/paper_files/paper/2021/file/854d9fca60b4bd07f9bb215d59ef5561-Paper.pdf
6. SJ Arora, RP Singh, "Automatic Speech Recognition: A Review" 2012, <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=aac912cbdb4edd c2ac5a62c0d8938ec2f5a7dc6b>