

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM
KHOA: CÔNG NGHỆ THÔNG TIN



PHẠM THANH HẢI
NGUYỄN VĂN HẢI

SÁNG TÁC GIAI ĐIỆU MỚI TRONG ÂM NHẠC BẰNG
KỸ THUẬT HỌC SÂU

Ngành: Khoa Học Máy Tính

Giảng viên hướng dẫn: TS. Đặng Thị Phúc

TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2023

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY
FACULTY OF INFORMATION TECHNOLOGY



**COMPOSING NEW MELODIES IN MUSIC USING
DEEP LEARNING**

Major: Computer Science

Instructor: Dr. Dang Thi Phuc

Name	ID	Class
Pham Thanh Hai	19501521	DHKHMT15A
Nguyen Van Hai	18056541	DHKHMT14

HO CHI MINH CITY, MAY 2023

ABSTRACT

Reason for choosing the topic:

Music has been present for a very long time in the history of mankind. Over time, music has gone through many different stages of development, from folk songs and classical music to modern genres. With the development of music, it has become a large industry and has influenced many aspects of human life, from the economy to culture and entertainment. With the development of artificial intelligence, applying deep learning models to create new melodies will be very interesting and helpful for music producers to access new aspects of music. This technique can help reduce the time and cost of composing and producing new music, as well as open up many creative possibilities in the field of music.

Problems:

- Encode note point data into a time series.
- Learn melodies using deep learning.
- Composing new melodies using deep learning.

Methods:

- Research on data encode techniques, Models of learning and composing melodies.
- Optimizing the models to improve results.

LỜI CẢM ƠN

Kính gửi Quý thầy cô và các bạn, em xin gửi đến quý thầy cô báo cáo đề án với hi vọng đánh giá cao và hữu ích cho cộng đồng học thuật.

Báo cáo này được thực hiện với mục đích trình bày kết quả của quá trình nghiên cứu và thực hiện đề án của em. Nó bao gồm các phần trình bày về đối tượng nghiên cứu, phương pháp và quá trình thực hiện đề án, kết quả và phân tích, cùng với những kết luận và đề xuất đối với vấn đề được nghiên cứu.

Em mong rằng báo cáo này sẽ cung cấp cho Quý thầy/cô và các bạn cái nhìn tổng quan về quá trình thực hiện đề án của em và đóng góp cho việc tăng cường hiểu biết và nghiên cứu trong lĩnh vực tương tự.

Trước tiên, em xin gửi lời cảm ơn đến giảng viên hướng dẫn của em, TS. Đặng Thị Phúc. Với sự hướng dẫn tận tình và kiến thức chuyên môn sâu sắc của cô, em đã có thể nắm bắt được nội dung và phương pháp nghiên cứu của đề án. Những góp ý, lời khuyên và hướng dẫn của cô đã giúp em hoàn thiện đề án một cách chuyên nghiệp và có giá trị.

Em rất biết ơn vì đã có cơ hội hoàn thành khóa luận của mình. Tuy nhiên, em hiểu rằng trong quá trình viết và hoàn thiện, em có thể đã mắc một số thiếu sót và chưa đạt đến yêu cầu của quý Thầy/Cô. Vì vậy, em mong muốn nhận được những góp ý, phê bình của quý Thầy/Cô về khóa luận của mình, để em có thể cải thiện và hoàn thiện hơn. Em sẽ rất trân trọng mọi đánh giá và ý kiến xây dựng của quý Thầy/Cô, để từ đó có thể phát triển và cải thiện khả năng của mình trong tương lai. Một lần nữa, em xin chân thành cảm ơn quý Thầy/Cô đã dành thời gian để xem xét khóa luận của em và sẵn sàng chia sẻ những kiến đánh giá của mình.

TP. Hồ Chí Minh, ngày tháng năm 2022

NGƯỜI THỰC HIỆN

Phạm Thanh Hải

Nguyễn Văn Hải

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

TP. Hồ Chí Minh, ngày tháng năm 2023

GIẢNG VIÊN HƯỚNG DẪN

TS. Đặng Thị Phúc

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN 1

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

TP. Hồ Chí Minh, ngày tháng năm 2023

GIẢNG VIÊN PHẢN BIỆN 1

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN 2

This image shows a full page of primary-ruled paper. It features approximately 20 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins or additional markings.

TP. Hồ Chí Minh, ngày tháng năm 2023

GIẢNG VIÊN PHẢN BIỆN 2

MỤC LỤC

DANH MỤC THUẬT NGỮ CHỮ VIẾT TẮT	ix
DANH MỤC ĐỒ THỊ, HÌNH ẢNH	x
DANH MỤC BẢNG BIỂU	xi
CHƯƠNG 1: TỔNG QUAN	12
1.1. Giới thiệu đề tài	12
1.2. Lý do chọn đề tài.....	12
1.3. Mục tiêu nghiên cứu	12
1.4. Giới thiệu đầu vào, đầu ra bài toán	13
1.5. Một số nghiên cứu gần đây	13
1.6. Đề xuất mô hình mới	13
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	15
2.1. Long Short-Term Memory network (LSTM)	15
2.1.1. Tổng quan về LSTM	15
2.1.2. Ý tưởng cốt lõi LSTM	16
2.2. Attention	17
2.3. Bidirectional Long Short-Term Memory	19
2.4. Mô hình Transformer	20
2.4.1. Giới thiệu chung	20
2.4.2. Tổng quan mô hình.....	20
2.4.3. Embedding layer with Position Encoding	22
2.4.4. Encoder	23
2.4.5. Scale Dot-Product Attention	24
2.4.6. Multi-head Attention	25
2.4.7. Feed Forward.....	26
2.4.8. Decoder.....	26

2.5. Các hàm đánh giá chuỗi đầu ra.....	27
2.5.1. METEOR score	27
2.5.2. Perplexity	28
2.5.3. MIDI Pitch Accuracy	28
CHƯƠNG 3: MÔ HÌNH ĐỀ XUẤT	29
3.1. Mô hình LSTM cho bài toán sáng tác âm nhạc.....	29
3.2. LSTM kết hợp Attention.....	30
3.3. Bidirectional Long Short-Term Memory	31
3.4. Mô hình transformer cho bài toán sáng tác giai điệu	32
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM	34
4.1. Dữ liệu.....	34
4.1.1. Nguồn dữ liệu	34
4.1.2. Đầu vào và đầu ra	34
4.1.3. Tăng cường dữ liệu.....	38
4.2. Kết quả huấn luyện mô hình	38
4.2.1. Đánh giá mô hình dựa vào độ chính xác và giá trị hàm mất mát.....	38
4.2.2. Đánh giá thời gian huấn luyện và kích thước bộ trọng số.....	40
4.2.3. Đánh giá mô hình bằng cách đánh giá chuỗi đầu ra.....	41
4.2.4. Đánh giá giai điệu bằng thính giác của con người	42
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	46
5.1. Kết luận	46
5.1.1. Ưu điểm	46
5.1.2. Hạn chế.....	46
5.2. Hướng phát triển	46
TÀI LIỆU THAM KHẢO.....	47
PHỤ LỤC	49

DANH MỤC THUẬT NGỮ CHỮ VIẾT TẮT

Từ viết tắt	Từ đầy đủ	Nghĩa
RNN	Recurrent Neural Network	Mạng nơ ron hồi quy
LSTM	Long short-term memory	Bộ nhớ đệm dài và ngắn
Bi-LSTM	Bidirectional long short-term memory	Bộ nhớ đệm dài và ngắn hai chiều
MIDI	Musical Instrument Digital Interface	Giao diện kỹ thuật số nhạc cụ
XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng
NLP	Natural language processing	Xử lý ngôn ngữ tự nhiên
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
RAM	Random Access Memory	Bộ nhớ khả biến
CPU	Central Processing Unit	Bộ xử lý trung tâm
AI	Artificial Intelligence	Trí tuệ nhân tạo
METEOR	Metric for Evaluation of Translation with Explicit Ordering	Metric để đánh giá bản dịch với thứ tự rõ ràng
FFN	Feed-Forward Networks	Mạng lan truyền tiếp dữ liệu

DANH MỤC ĐỒ THỊ, HÌNH ẢNH

Hình 2.1. Mạng RNN truyền thống	15
Hình 2.2. Mạng LSTM	15
Hình 2.3. ký hiệu được sử dụng bên trong mỗi kiến trúc mạng.....	16
Hình 2.4. Cell state LSTM	16
Hình 2.5. Gate với hàm kích hoạt sigmoid trong LSTM.....	17
Hình 2.6. mô hình sequence to sequence khi sử dụng cơ chế attention	18
Hình 2.7. Mạng Bi-LSTM	19
Hình 2.8. kiến trúc mô hình Transformer.....	20
Hình 2.9. Ví dụ cho mô hình Transformer	21
Hình 2.10. So sánh cách xử lý của mô hình Transformer với mô hình LSTM.....	22
Hình 2.11. Cách sử dụng Position Encoding	22
Hình 2.12. Tầng encoder của mô hình Transformer	23
Hình 2.13. Ví dụ về self-attention	24
Hình 2.14. Ví dụ Scaled Dot-Product Attention	25
Hình 2.15. Cơ chế Mutil-head attention	26
Hình 2.16. Quá trình từ encoder đến decoder	27
Hình 3.1. Mô hình LSTM cho bài toán sáng tác giai điệu	29
Hình 3.2. Mô hình LSTM + attention	30
Hình 3.3. Mô hình Bi-LSTM + Attention + LSTM	31
Hình 3.4. Mô hình Transformer sáng tác nhạc	33
Hình 4.1. Minh họa sheet music	34
Hình 4.2. Minh họa Music XML dạng chữ	35
Hình 4.3. Minh họa Music XML dạng Piano roll	35
Hình 4.4. Minh họa mã hóa MIDI sang chuỗi ký tự.	36
Hình 4.5. Biểu đồ Categorical Cross Entropy Loss	38
Hình 4.6. Giai điệu đầu vào.....	42
Hình 4.7. Giai điệu sinh ra từ mô hình LSTM	43
Hình 4.8. Giai điệu sinh ra từ mô hình LSTM +Attention.....	43
Hình 4.9. Giai điệu sinh ra từ mô hình Bi-LSTM + Attention + LSTM.....	44
Hình 4.10. Giai điệu sinh ra từ mô hình Transformer	45

DANH MỤC BẢNG BIỂU

Bảng 4.1. Chi tiết trường độ nốt nhạc và dấu lặng	37
Bảng 4.2. MIDI note number.	37
Bảng 4.3. Chi tiết lịch sử của Accuracy và Loss khi huấn luyện	39
Bảng 4.4. Hiệu suất các mô hình	39
Bảng 4.5. Hiệu suất các mô hình tạo giai điệu	41
Bảng 4.6. Phân cứng colab	41
Bảng 4.7. Điểm số đánh giá chuỗi đầu ra (giai điệu sáng tác)	42

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu đề tài

Âm nhạc là một phần không thể thiếu trong cuộc sống, nó có thể được sử dụng để giải trí, thúc đẩy sự sáng tạo, mang đến trải nghiệm tuyệt vời cho người nghe qua những giai điệu nốt nhạc. Chúng được tạo ra từ nhiều loại nhạc cụ khác nhau như piano, guitar, trống, kèn,

Để tạo ra những giai điệu âm nhạc đó, đòi hỏi nhà soạn nhạc phải có trí tưởng tượng khả năng sáng tạo, kinh nghiệm lâu năm trong lĩnh vực âm nhạc.

Trong thời đại 4.0, các phương pháp Học máy và Học sâu phát triển không ngừng, chúng được áp dụng vào nhiều lĩnh vực khác nhau trong cuộc sống. Trong âm nhạc, một số mô hình máy học cũng được sử dụng để tạo ra những giai điệu, bản nhạc mới. Các nghiên cứu về vấn đề sáng tạo giai điệu âm nhạc cũng ngày càng trở nên phổ biến.

1.2. Lý do chọn đề tài

Trong khoá luận, có một số lý do để chúng em chọn đề tài này vì:

- Muốn nghiên cứu về ứng dụng của học sâu trong âm nhạc. Mở rộng sáng tạo ra những tác phẩm âm nhạc mới lạ và độc đáo. Sử dụng trí tuệ nhân tạo (AI) để sáng tác nhạc có thể giúp tạo ra những bản nhạc mới và độc đáo một cách nhanh chóng và hiệu quả hơn. Ngoài ra, đề tài này còn cung cấp cho người nghiên cứu cơ hội để thử nghiệm và áp dụng các kỹ thuật và phương pháp mới của AI vào lĩnh vực âm nhạc.
- Thông thường, nhà soạn nhạc phải tốn nhiều thời gian để tìm ra giai điệu thích hợp cho những nốt tiếp theo. Việc sử dụng học sâu sẽ giúp giảm thiểu công sức và thời gian sáng tác giai điệu.

1.3. Mục tiêu nghiên cứu

Mô hình sáng tác nhạc cần đáp ứng các tiêu chí:

- Giai điệu sáng tác phải dài, đúng tông, phần hợp âm đệm và giai điệu phải phù hợp.
- Giai điệu sáng tác phải phù hợp với giai điệu đầu vào để ghép nối thành một bài hát
- Thời gian sáng tác nhanh.

Đáp ứng tiêu chí trên, chúng em nghiên cứu những kỹ thuật học sâu để xây dựng mô hình:

- Sử dụng những mô hình xử lý ngôn ngữ tự nhiên.
- Sử dụng các thư viện numpy, music2, ... để xử lý dữ liệu.

1.4. Giới thiệu đầu vào, đầu ra bài toán

Trong đề tài này, đầu vào của hệ thống là một tập tin lưu trữ giai điệu (MIDI, xml, krn, mscz, ...). Đầu ra của hệ thống là một file MIDI mới, chứa các bản nhạc được tạo ra bởi mô hình AI.

Bài toán mà đề tài này tập trung giải quyết là bài toán NLP (Natural Language Processing), được áp dụng vào việc dự đoán các nốt tiếp theo của bản nhạc. Hệ thống sẽ phân tích các mẫu dữ liệu từ các bản nhạc đã sáng tác trước đó, sau đó sử dụng các thuật toán học máy để dự đoán các nốt tiếp theo phù hợp với phong cách và âm nhạc của bài hát được sáng tác

1.5. Một số nghiên cứu gần đây

Bài báo “LSTM Based Music Generation System” [1] xuất hiện năm 2019 đề xuất sử dụng mô hình LSTM (Long Short-Term Memory Network) với tập dữ liệu huấn luyện là các bản nhạc MIDI để tạo ra các bản nhạc mới. Sau khi được huấn luyện, mô hình LSTM của tác giả tạo bản nhạc mới bằng cách tạo ra một chuỗi các nốt nhạc mới dựa trên cấu trúc đã học được từ tập dữ liệu huấn luyện.

Bài báo “Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders” [2] xuất hiện năm 2019 đề xuất phương pháp mới là sử dụng mạng học sâu WaveNet Autoencoders với tập dữ liệu gồm các bản ghi nốt nhạc từ piano và guitar. Sau khi huấn luyện, kết quả cho thấy mạng WaveNet tạo ra âm thanh có chất lượng tốt.

Dự án MusicAutobot [3] là một dự án nghiên cứu trong lĩnh vực trí tuệ nhân tạo, tập trung vào việc sáng tác nhạc bằng máy tính thông qua mô hình học sâu. Được thực hiện và viết bài nghiên cứu tại towardsdatascience.com. Nhóm người này đã nghiên cứu và xây dựng mô hình sáng tác nhạc Pop với mô hình Transformer.

1.6. Đề xuất mô hình mới

Báo cáo này nghiên cứu một số kỹ thuật Học sâu cho bài toán sáng tác giai điệu:

- Xây dựng bộ dữ liệu mới đa dạng hơn, đặc biệt có thêm những giai điệu bài hát ở Việt Nam.
- Xây dựng mô hình LSTM.
- Xây dựng mô hình LSTM kết hợp với Attention.

- Xây dựng mô hình Bi-LSTM kết hợp Attention và LSTM.
- Xây dựng mô hình Transformer.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

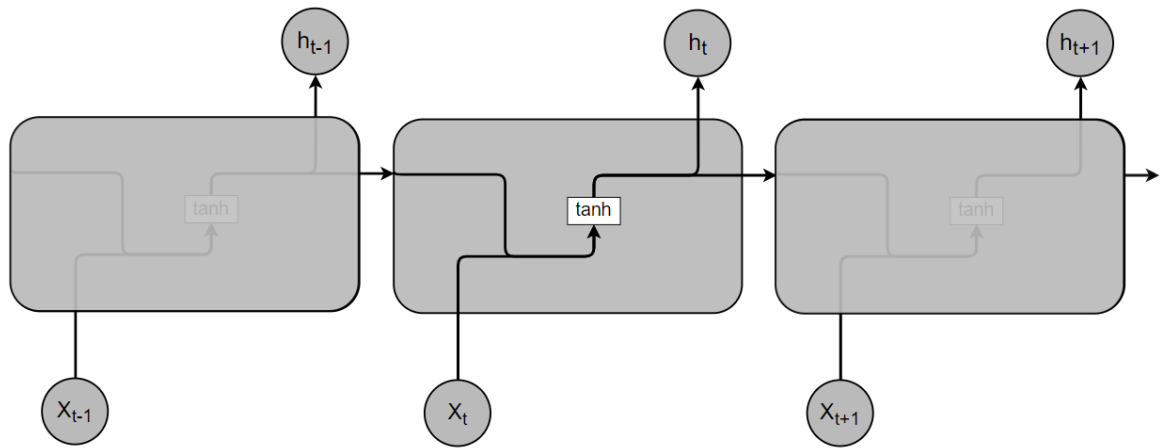
Ở chương này, chúng em sẽ đưa ra lý thuyết của các mô hình sử dụng để phục vụ cho quá trình nghiên cứu.

2.1. Long Short-Term Memory network (LSTM)

2.1.1. Tổng quan về LSTM

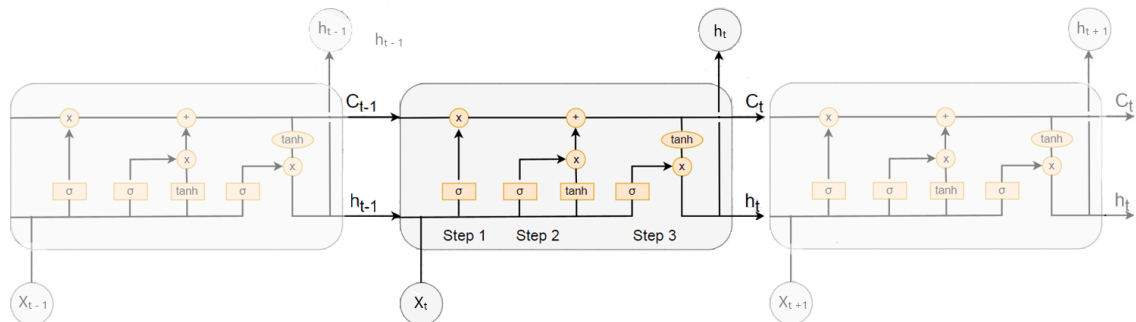
LSTM (Long Short-Term Memory Network) [4] là một kiến trúc đặc biệt của RNN (Recurrent Neural network) có khả năng học được sự phụ thuộc trong dài hạn (long-term dependencies). Kiến trúc này được nhiều nhà nghiên cứu sử dụng cho đến nay. LSTM đã khắc phục được hạn chế về triệt tiêu đạo hàm của RNN, đổi lại cấu trúc của nó cũng phức tạp hơn.

RNN truyền thống sẽ có kiến trúc đơn giản là một tầng ẩn là hàm tanh như hình dưới:



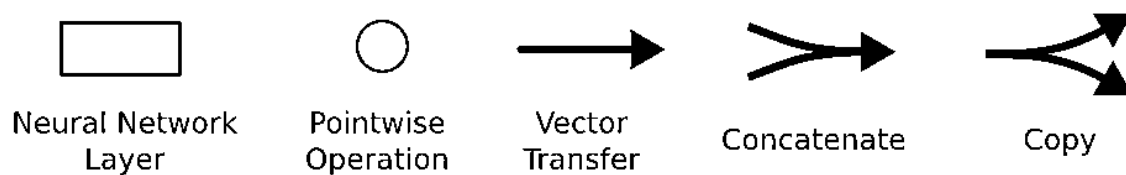
Hình 2.1. Mạng RNN truyền thống

LSTM cũng sử dụng kiến trúc chuỗi, nhưng các mô-đun trong kiến trúc lặp lại khác biệt so với mạng RNN truyền thống. Thay vì chỉ có một tầng ẩn đơn, LSTM sử dụng 4 tầng ẩn bao gồm 3 hàm sigmoid và 1 hàm tanh để tương tác với nhau.



Hình 2.2. Mạng LSTM

Các kí hiệu được sử dụng bên trong mỗi kiến trúc mạng:

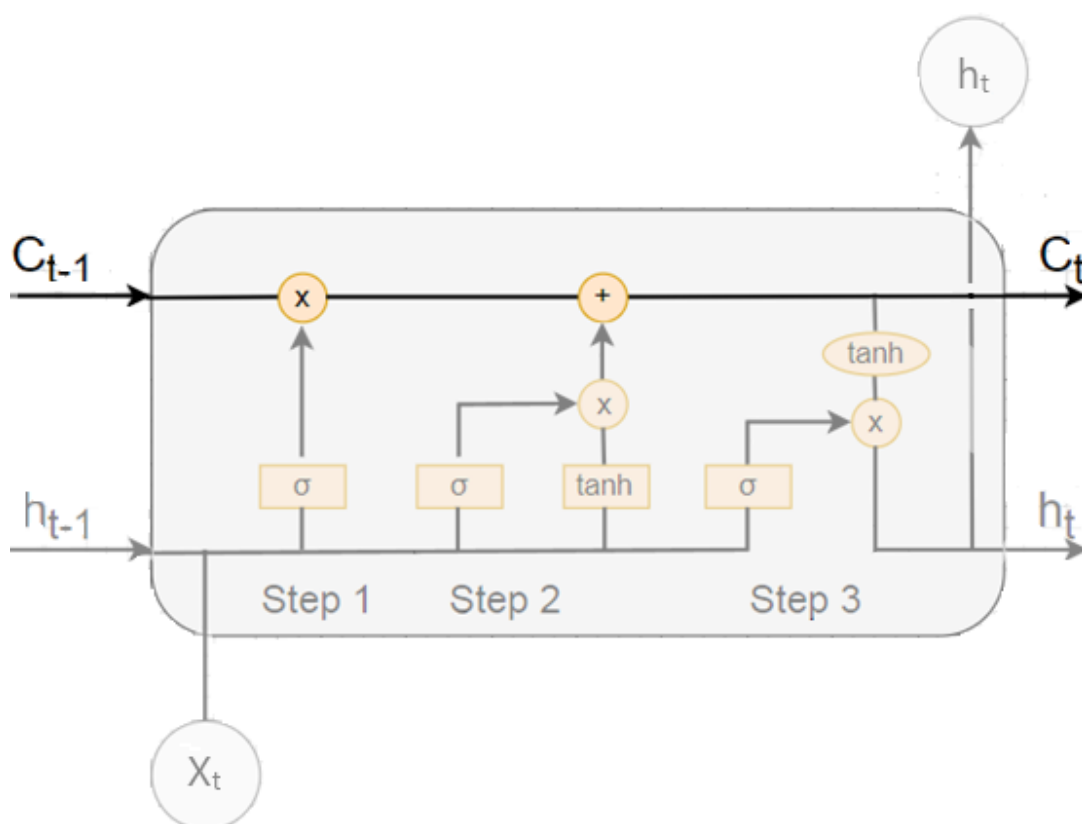


Hình 2.3. ký hiệu được sử dụng bên trong mỗi kiến trúc mạng

Trong sơ đồ, mỗi đường diễn hình dẫn từ đầu ra của một nút đến đầu vào của một nút khác. Các hình chữ nhật đại diện cho các hàm được sử dụng trong tầng ẩn của mạng nơron, thường là hàm sigmoid và tanh. Các hình tròn biểu thị cho các phép toán như cộng véc-tơ và tích vô hướng của các véc-tơ. Việc kết hợp được biểu thị bởi các đường giao nhau, trong khi các đường rẽ nhánh cho thấy véc-tơ được sao chép sang một phần khác.

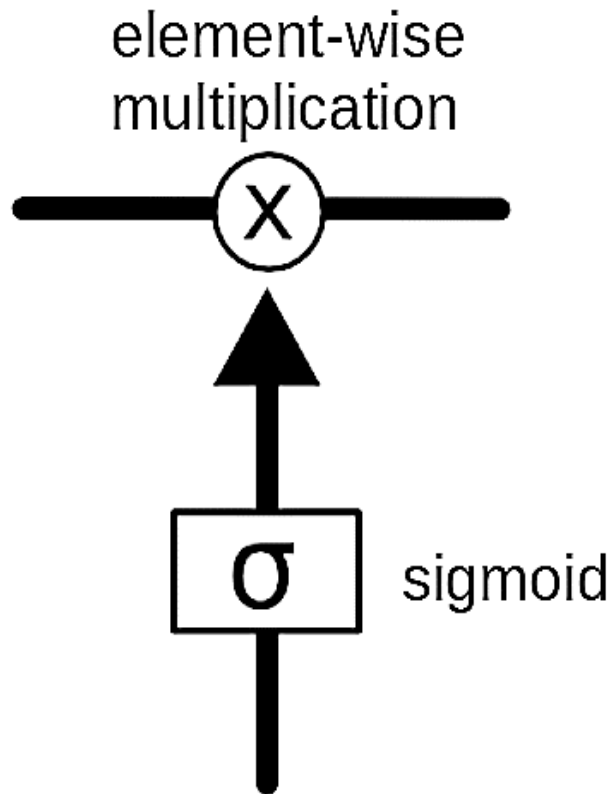
2.1.2. Ý tưởng cốt lõi LSTM

Điểm chính của LSTM là trạng thái tế bào (cell state) – chính là đường ngang phía trên của sơ đồ hình vẽ có tác dụng lưu trữ thông tin.



Hình 2.4. Cell state LSTM

- + Trạng thái tế bào chạy qua các mắt xích (nút mạng) và tương tác tuyến tính đôi chút. Điều này cho phép thông tin được truyền đi một cách dễ dàng và không bị thay đổi.
- + LSTM sử dụng các cổng để điều chỉnh thông tin cần thiết cho trạng thái tế bào. Các cổng này có khả năng loại bỏ hoặc bổ sung thông tin, và được kết hợp bằng một tầng mạng sigmoid và một phép nhân.

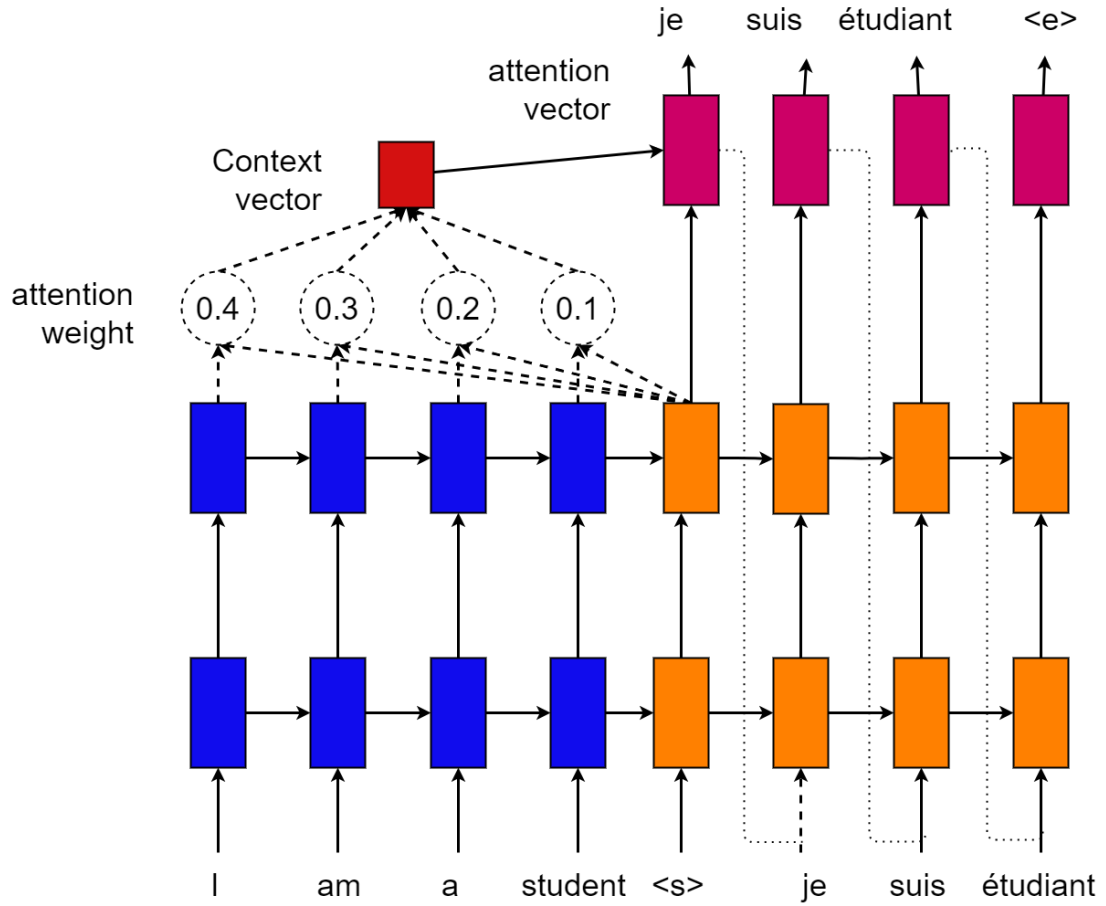


Hình 2.5. Gate với hàm kích hoạt sigmoid trong LSTM

Hàm sigmoid tạo ra giá trị xác suất từ 0 đến 1, biểu thị mức độ thông tin. Khi giá trị là 0, không có thông tin nào đi qua, còn khi là 1, toàn bộ thông tin được truyền qua. LSTM sử dụng ba cổng để duy trì và điều khiển trạng thái của tế bào.

2.2. Attention

Cấu tạo attention tạo ra để có thể xử lý các vấn đề của mô hình seq2seq và được sử dụng thay thế cho mạng nơ-ron hồi tiếp. Ý tưởng của cơ chế attention là sử dụng một vector ngữ cảnh tương tác với toàn bộ vector trạng thái ẩn của Encode. Khi áp dụng cơ chế attention vào mô hình seq2seq, mô hình có cấu trúc như sau (khối màu xanh là encoder, khối màu đỏ là decoder):



Hình 2.6. mô hình sequence to sequence khi sử dụng cơ chế attention

Các bước chi tiết, mỗi bước thời gian t ở phía decoder:

- **Bước 1:** h_t, h_s được nhận lần lượt từ decoder và encoder
- **Bước 2:** Tính điểm attention. Với mỗi h_t, h_s sẽ được tính toán điểm attention để có sự liên kết giữa khối encoder và decoder $score(h_t, h_s)$
- **Bước 3:** Sử dụng hàm softmax để tính weight attention từ điểm attention

$$\alpha_{ts} = \frac{\exp(\text{score}(h_t, \bar{h}_s))}{\exp(\sum_{s'=1}^S \text{score}(h_t, \bar{h}_{s'}))}$$

- **Bước 4:** tổng của các trọng số attention nhân với vector trạng thái ẩn của decoder tại bước thời gian tương ứng ta sẽ có vector bối cảnh

$$c_t = \sum_{s'=1}^S \alpha_{ts} \bar{h}_{s'}$$

Cuối cùng, các vector attention α_{ts} được sử dụng để cung cấp đầu ra được tính toán dựa trên vector ngữ cảnh c_t và vector trạng thái ẩn tại bộ giải mã h_t .

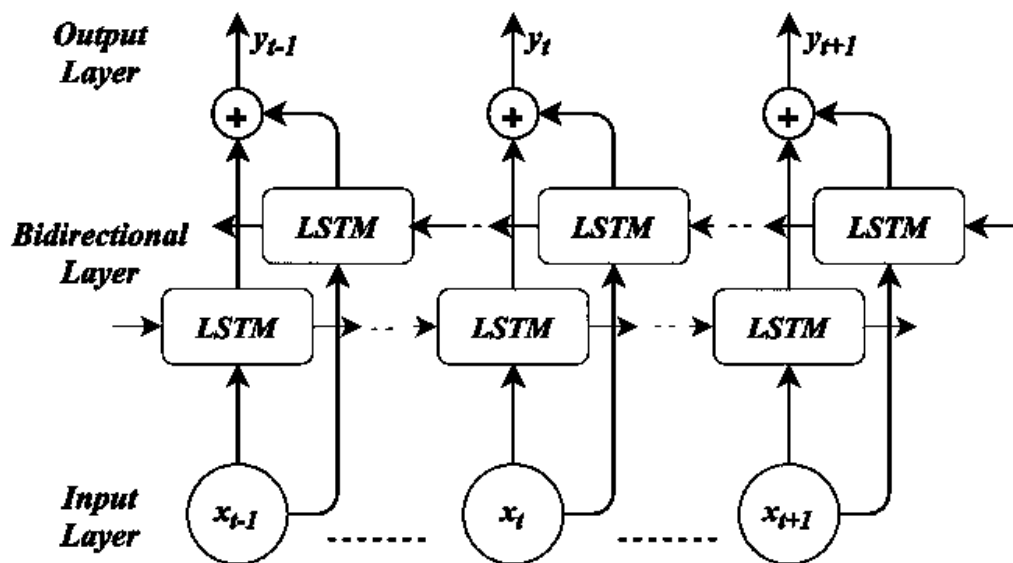
Cơ chế này là tạo ra một mạng của thể hiểu được rằng tại mỗi time-step thì phần nào của đầu vào là quan trọng. Có thể huấn luyện end-to-end cho mô hình Mô hình với cơ chế attention, chính mô hình sẽ tự học để làm điều đó.

2.3. Bidirectional Long Short-Term Memory

Bi-LSTM (Bidirectional Long Short-Term Memory) [5] là một kiến trúc mô hình RNN (Recurrent Neural Network) sử dụng hai lớp LSTM song song để xử lý dữ liệu đầu vào. Bi-LSTM sử dụng cả thông tin đầu vào về quá khứ và tương lai để tính toán đầu ra tại mỗi thời điểm, giúp mô hình có thể học được các phụ thuộc phức tạp và chuỗi dữ liệu có thể có.

Cơ chế hoạt động của Bi-LSTM như sau:

- + Sử dụng một chuỗi dữ liệu đầu vào: Bi-LSTM nhận vào một chuỗi dữ liệu đầu vào và chuyển nó thành các vector đặc trưng
- + Xử lý với lớp LSTM: Chuỗi đặc trưng được đưa vào hai lớp LSTM song song: một lớp xử lý chuỗi từ trái sang phải và lớp còn lại xử lý chuỗi từ phải sang trái. Mỗi lớp LSTM tính toán các trạng thái ẩn tại mỗi thời điểm và truyền chúng sang thời điểm tiếp theo.
- + Kết hợp các trạng thái ẩn: Tại mỗi thời điểm, Bi-LSTM kết hợp các trạng thái ẩn từ hai lớp LSTM để tạo ra một vector đặc trưng tổng hợp của đầu vào tại thời điểm đó.
- + Tính toán đầu ra: Các vector đặc trưng được tính toán tại mỗi thời điểm được đưa vào một hoặc nhiều lớp fully connected để tính toán đầu ra



Hình 2.7. Mạng Bi-LSTM

Bi-LSTM được sử dụng phổ biến trong các bài toán xử lý ngôn ngữ tự nhiên và cũng ứng dụng vào bài toán sáng tác giai điệu mới trong âm nhạc. Bi-LSTM giúp cho mô hình có khả năng học được các phụ thuộc phức tạp trong các chuỗi dữ liệu đầu vào, đồng thời tăng cường khả năng dự đoán và cải thiện độ chính xác của kết quả dự đoán.

2.4. Mô hình Transformer

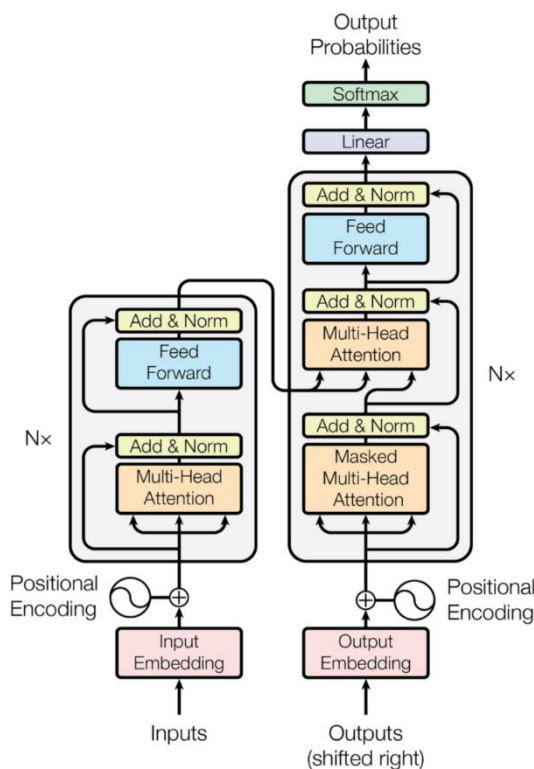
2.4.1. Giới thiệu chung

Trước đây, hầu hết các tác vụ xử lý ngôn ngữ tự nhiên, đặc biệt trong lĩnh vực dịch máy, sử dụng kiến trúc Recurrent Neural Networks (RNNs). Tuy nhiên, RNNs có nhược điểm về tốc độ xử lý chậm và hạn chế trong việc biểu diễn sự phụ thuộc xa giữa các từ trong câu do phải xử lý theo thứ tự. Trái lại, Transformer là một mô hình hoàn toàn dựa vào self-attention để tính toán biểu diễn của chuỗi đầu vào và đầu ra mà không cần sử dụng tuần tự như RNN, LSTM hay CNN.

Năm 2017, bài báo “Attention Is All You Need” [6] đã mang đến một bước tiến quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, giải quyết triệt để các vấn đề mà RNNs trước đây gặp phải.

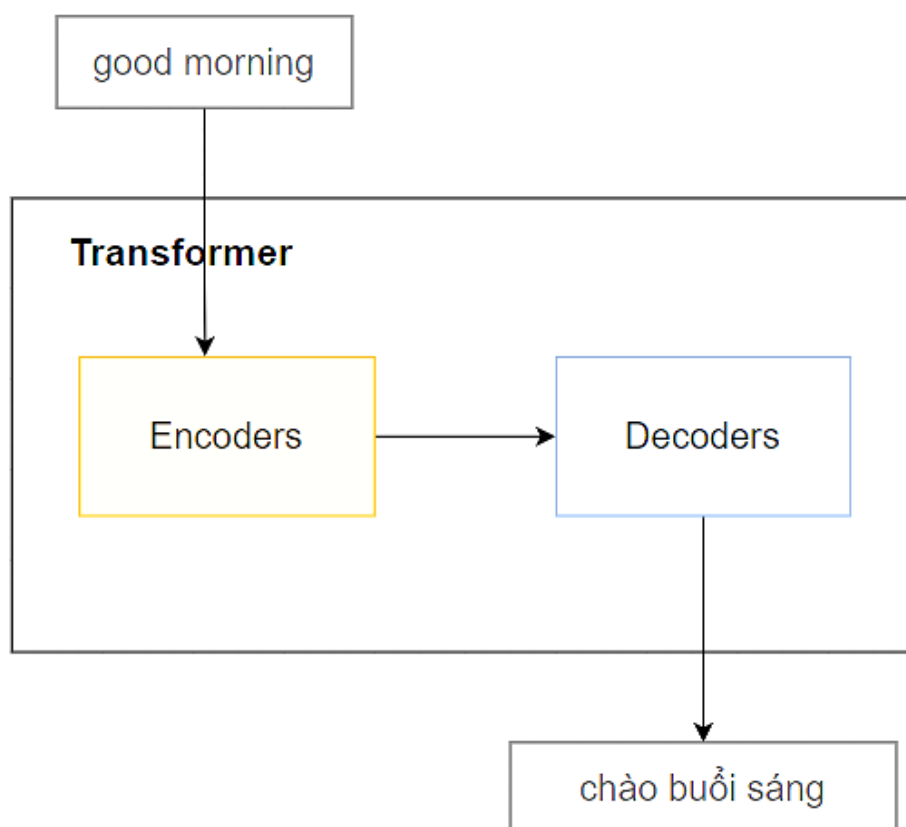
2.4.2. Tổng quan mô hình

Hình vẽ dưới đây mô tả mô hình transformer:



Hình 2.8. kiến trúc mô hình Transformer

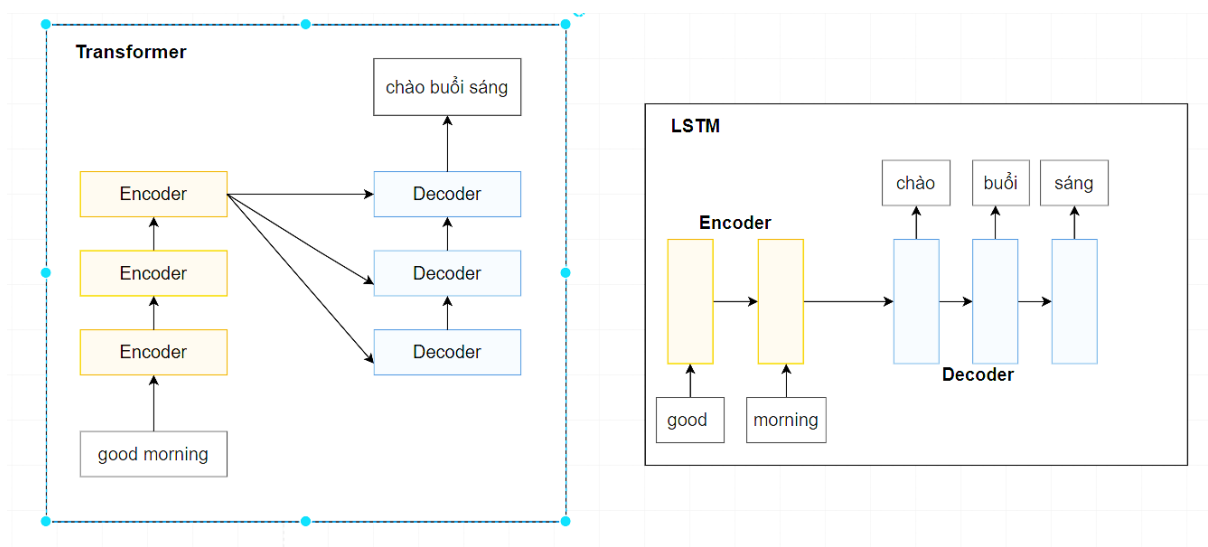
Mô hình transformer có cấu trúc gồm hai phần là encoder và decoder, tương tự như các mô hình dịch máy khác. Encoder được sử dụng để học các vector biểu diễn của câu, với hy vọng rằng vector này chứa đựng thông tin đầy đủ về câu đó. Trong khi đó, decoder có nhiệm vụ chuyển đổi vector biểu diễn đó thành ngôn ngữ mục tiêu.



Hình 2.9. Ví dụ cho mô hình Transformer

Trong ví dụ trên, mô hình Transformer sử dụng một encoder để biểu nghĩa của một câu tiếng Anh "good morning" thành một vector. Tiếp đó, phần decoder sử dụng vector này để dịch câu thành tiếng Việt "chào buổi sáng". Một lợi ích của Transformer là khả năng xử lý các từ song song.

Encoders trong mô hình transformer sử dụng các feedforward neural nets, bao gồm nhiều encoder layer khác nhau để xử lý đồng thời các từ. Trong khi đó, mô hình LSTM xử lý các từ tuần tự. Mô hình Transformer cũng xử lý đầu vào theo cả hai hướng mà không cần sử dụng LSTM hai chiều như kiến trúc Bidirectional LSTM.



Hình 2.10. So sánh cách xử lý của mô hình Transformer với mô hình LSTM

Để có thể hiểu rõ hơn về mô hình, chúng em sẽ tìm hiểu các phần quan trọng như multi head attention của encoder, position encoding, ...

2.4.3. Embedding layer with Position Encoding

Các từ được mã hóa bằng vector có kích thước tương đương với word embedding và được thêm trực tiếp vào word embedding.



Hình 2.11. Cách sử dụng Position Encoding

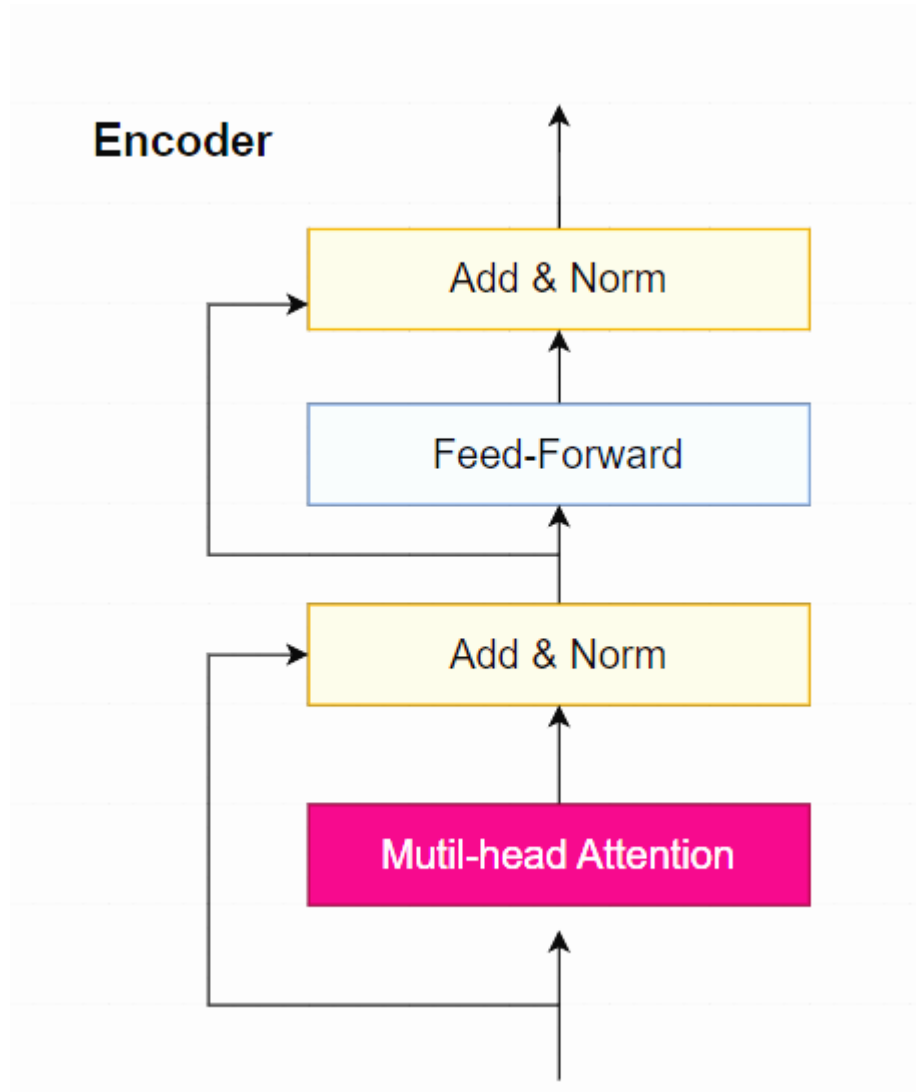
Các từ được mã hóa bằng vector có kích thước tương đương với word embedding và được thêm trực tiếp vào word embedding.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

2.4.4. Encoder

Mô hình transformer có thể có nhiều lớp mã hóa tương tự nhau. Mỗi lớp mã hóa trong transformer gồm hai phần chính là multi head attention và feedforward network. Ngoài ra, còn có skip connection và normalization layer.



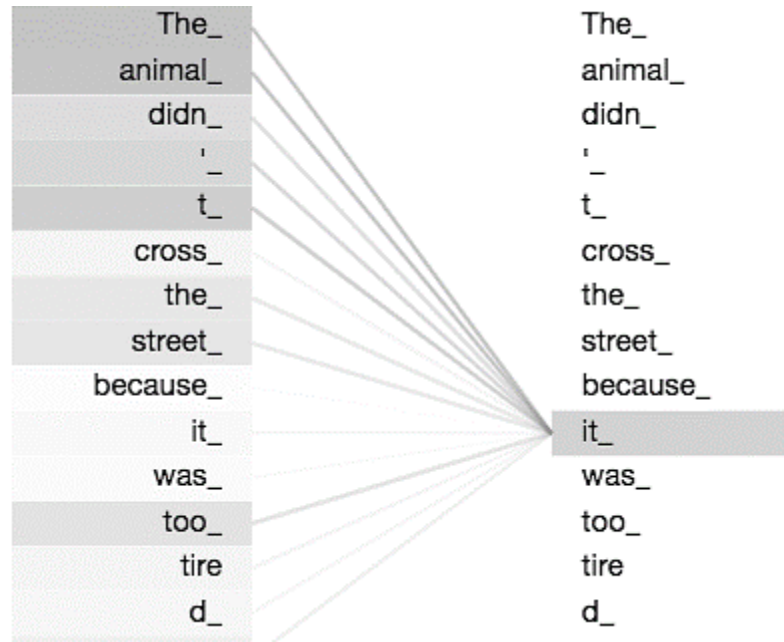
Hình 2.12. Tầng encoder của mô hình Transformer

Encoder đầu tiên tiếp nhận ma trận biểu diễn từng từ đã được kết hợp với thông tin vị trí bằng positional encoding. Tiếp theo, ma trận này được xử lý thông qua Multi Head Attention, một phương pháp sử dụng nhiều self-attention để cho phép mô hình chú ý đến nhiều mẫu khác nhau.

2.4.5. Scale Dot-Product Attention

Cơ chế tự chú ý (Self Attention), mô hình có thể sử dụng thông tin của những từ liên quan tới nó khi mô hình mã hóa một từ.

Cho câu sau: “The animal didn’t cross the street because it was too tired.”. Thì "it" trong câu gốc biểu diễn cho "the animal". Trong việc sử dụng Self-Attention, mô hình sẽ tính toán mức độ chú ý của từ "it" với các từ khác trong câu để hiểu được ý nghĩa của nó. Điều này cho phép mô hình hiểu được rằng "it" trong trường hợp này là đề cập đến "the animal" chứ không phải là "the street" hoặc bất kỳ từ nào khác.

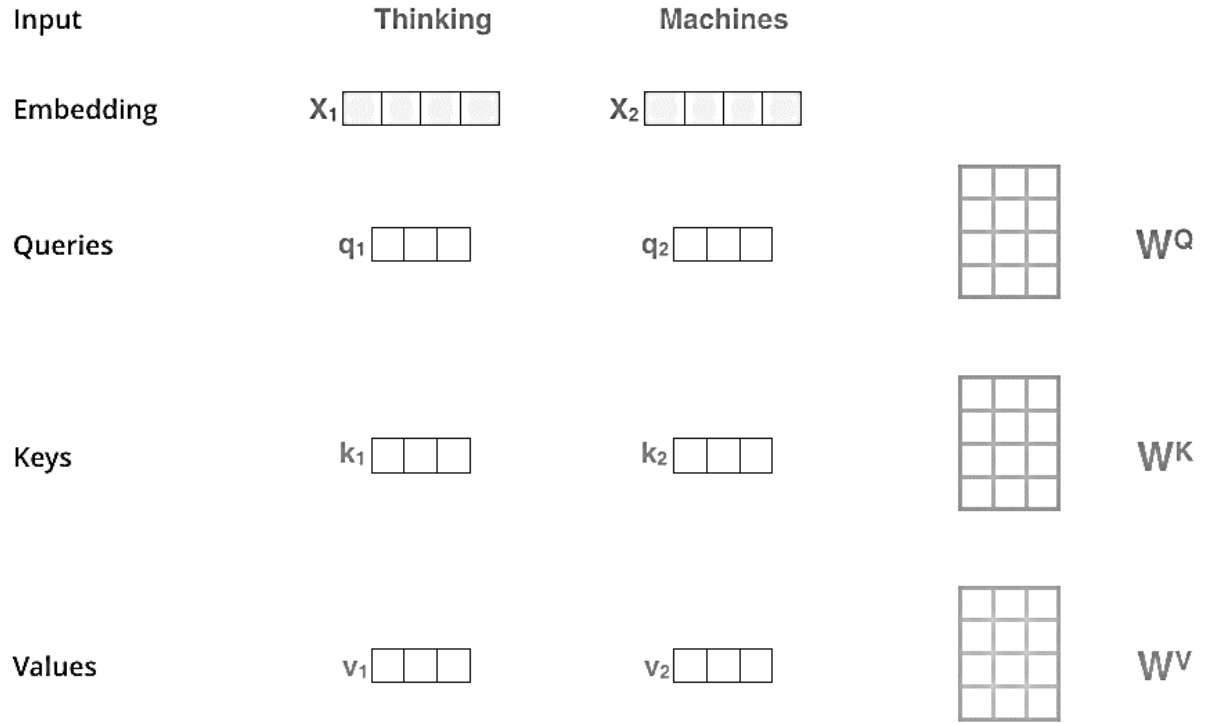


Hình 2.13. Ví dụ về self-attention

Ở đây, ta có thể nhận thấy rằng "it_" liên kết mạnh với "The animal" và cũng liên kết với các từ khác, tuy nhiên sức mạnh của liên kết này không cao. Self attention tương tự như cơ chế tìm kiếm, nó cho phép mô hình tìm kiếm các từ "giống" với từ được cho trước trong cách từ còn lại, sau đó mã hóa thông tin dựa trên tất cả các từ đó.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V [6]$$

Với mỗi đầu vào là vector x_i ta sẽ có 3 ma trận W_Q , W_K , W_V tương ứng với (Q, K, V) Lấy x_i nhân với từng ma trận trên sẽ ra 3 vector q_i, k_i, v_i tương ứng.



Hình 2.14. Ví dụ Scaled Dot-Product Attention

2.4.6. Mutil-head Attention

Sau khi thực hiện quá trình Scale dot production, chúng ta sẽ có một ma trận attention. Các tham số cần được điều chỉnh trong mô hình bao gồm ma trận W_q , W_k , W_v .. Mỗi quá trình như vậy được gọi là một head của attention. Khi lặp lại quá trình này nhiều lần (trong bài báo là 3 heads), ta thu được quá trình Multi-head Attention.

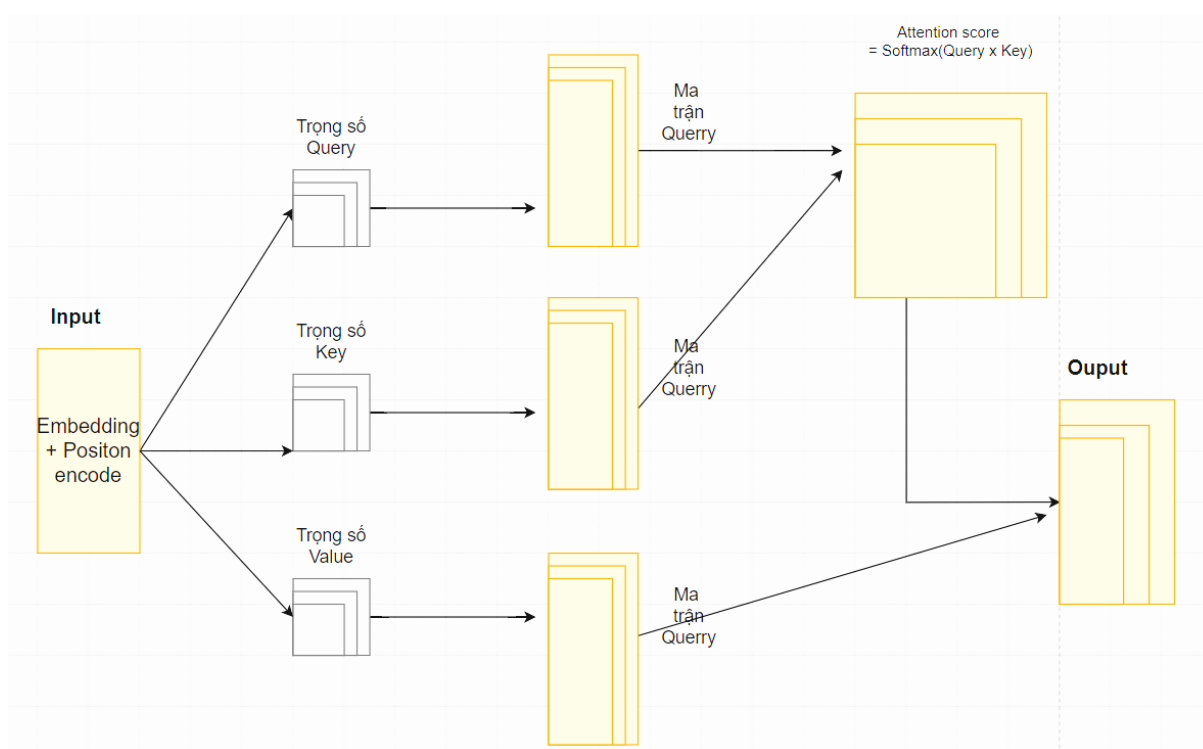
Sau khi thu được 3 matrix attention ở đầu jra chúng ta sẽ concatenate các matrix này theo các cột để thu được ma trận tổng hợp multi-head matrix có chiều cao trùng với chiều cao của ma trận input :

$$MultiHead(Q, K, V) = Concat_{i=1,2,...h}(head_i)W^0$$

Trong đó:

$$head_i(Q, K, V) = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Để trả về output có cùng kích thước với ma trận input chúng ta chỉ cần nhân với ma trận W_0 chiều rộng bằng với chiều rộng của ma trận input.



Hình 2.15. Cơ chế Mutil-head attention

Multi-head attention giúp mô hình đồng thời chú ý đến các mẫu dễ quan sát như sau:

- Chú ý đến phía trước của một từ.
- Chú ý đến phía kế sau của một từ.
- Chú ý đến các từ liên quan của một từ.

2.4.7. Feed Forward

Ở tầng này, lớp Feed-Forward Networks bao gồm 2 tầng biến đổi thông tin và 1 hàm ReLU ở giữa. Sau khi các vector qua hàm ReLU dropout được áp dụng ở lần biến đổi thứ nhất.

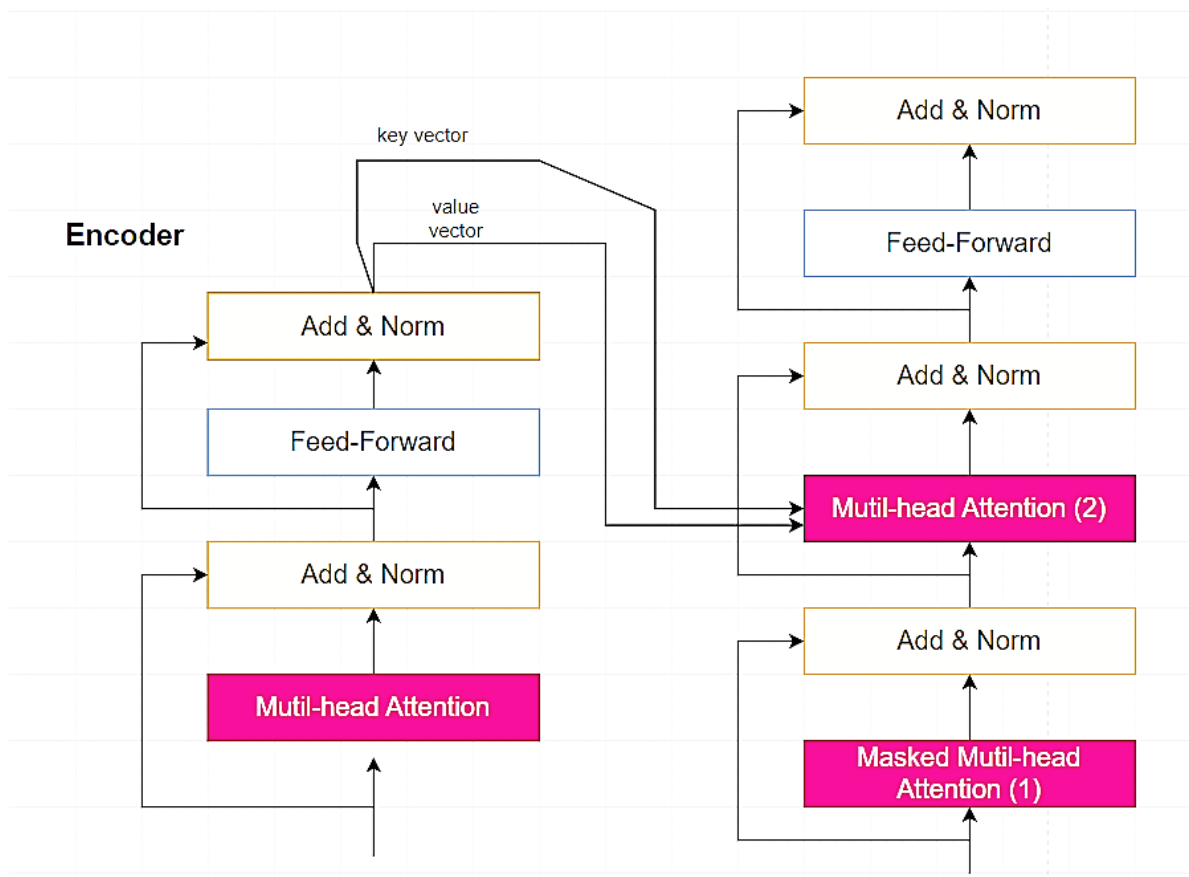
$$\text{FFN}(X) = W_2 \text{ReLU}(X, W_1)$$

Lớp Feed-Forward Networks có chức năng học mối quan hệ tiềm ẩn giữa các vector độc lập không rõ ràng. Lớp này sẽ học những mối quan hệ tiềm ẩn khác không thể diễn giải bằng công thức toán học, khác với mối quan hệ được khuếch đại qua lớp self-attention.

2.4.8. Decoder

Các vector sau khi đi qua lớp Feed-Forward Networks của khối encoder cuối cùng sẽ được nhân với hai ma trận trọng số K và V để tạo thành các cặp vector $\{(k_1, v_1), (k_2, v_2), \dots, (k_n, v_n)\}$ tương ứng với câu có n từ. Sau đó, đi đến decoder để decoder thực hiện phân tích giải mã vector của câu nguồn thành câu đích. Kiến trúc của bộ giải mã rất

giống với bộ mã hóa, 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.



Hình 2.16. Quá trình từ encoder đến decoder

2.5. Các hàm đánh giá chuỗi đầu ra

2.5.1. METEOR score

METEOR (Metric for Evaluation of Translation with Explicit ORdering) [7] là một metric đánh giá chất lượng dịch máy dựa trên sự tương đồng giữa hai đoạn văn bản, sử dụng phép đo từ vựng, cấu trúc ngữ pháp, cùng với một số phép đo khác để xác định độ tương đồng giữa hai đoạn văn bản. Phương pháp này được đề xuất bởi Banerjee và Lavie vào năm 2005.

$$\text{METEOR} = (1-\omega) \times \text{precision} + \omega \times \text{recall} \times \text{matching_score}$$

Trong đó:

- precision là tỉ lệ số từ chung được dự đoán đúng trên tổng số từ được dự đoán
- recall là tỉ lệ số từ chung được dự đoán đúng trên tổng số từ trong reference
- matching_score là điểm đánh giá sự tương đồng giữa các từ bằng cách sử dụng hàm wordnet của thư viện nltk.

- ω là hệ số trọng số của recall.

2.5.2. Perplexity

Perplexity [8] là một phương pháp đánh giá đơn giản và phổ biến nhất để đo lường chất lượng một mô hình ngôn ngữ. Perplexity là một số thực không âm, càng thấp thì mô hình càng tốt. Để tính perplexity, chúng ta cần tính toán tổng loss trên tập dữ liệu kiểm tra và số lượng từ trong corpus. Sau đó, tính giá trị exponent của tổng loss, chia cho số lượng từ trong tập kiểm tra, và lấy logarit tự nhiên của kết quả đó.

$$Perplexity = e^{\frac{1}{N} \sum_{i=1}^N -\log P(w_i)}$$

Trong đó:

- N là số lượng từ trong corpus.
- $P(w_i)$ là xác suất của từ w_i được tính bởi mô hình.
- $\log P(w_i)$ là loss được tính bằng hàm logarithm tự nhiên của xác suất $P(w_i)$.
- $\frac{1}{N} \sum_{i=1}^N -\log P(w_i)$ là giá trị trung bình của loss.

2.5.3. MIDI Pitch Accuracy

MIDI Pitch Accuracy [9] là độ đo để đánh giá độ chính xác về mặt âm sắc của một bản nhạc MIDI. Được tính toán bằng cách so sánh tần số của các nốt trong bản nhạc mới sáng tác với tần số của các note tương ứng trong bản nhạc gốc. Cho phép chúng ta đánh giá chính xác mức độ chính xác của một bản nhạc MIDI so với bản gốc. Điều này rất quan trọng trong việc đánh giá chất lượng của mô hình máy học tạo ra bản nhạc MIDI. Giả sử chúng ta có 2 tập MIDI sáng tác và gốc. Ta sẽ tính toán độ chính xác tần số như sau:

$$MIDI\ Pitch\ Accuracy = \frac{N_{sáng\ tác}}{N_{sáng\ tác + gốc}}$$

Trong đó:

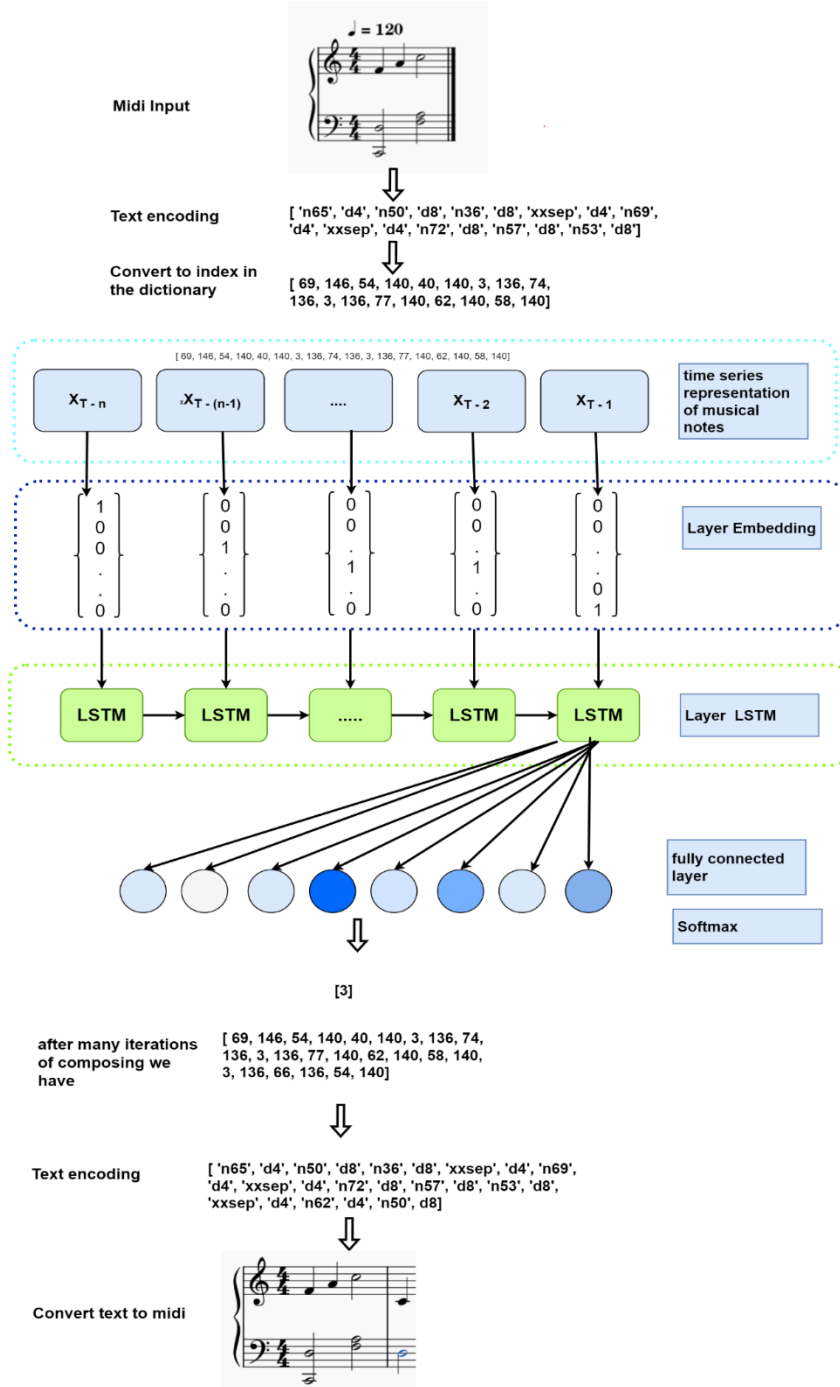
- $N_{sáng\ tác}$ số lượng các nốt nhạc trong tập MIDI sáng tác.
- $N_{sáng\ tác + gốc}$ tổng số lượng các nốt nhạc trong hai tập MIDI.

CHƯƠNG 3: MÔ HÌNH ĐỀ XUẤT

Sau đây là những mô hình em xây dựng và đem vào so sánh trong bài báo cáo này.

3.1. Mô hình LSTM cho bài toán sáng tác âm nhạc

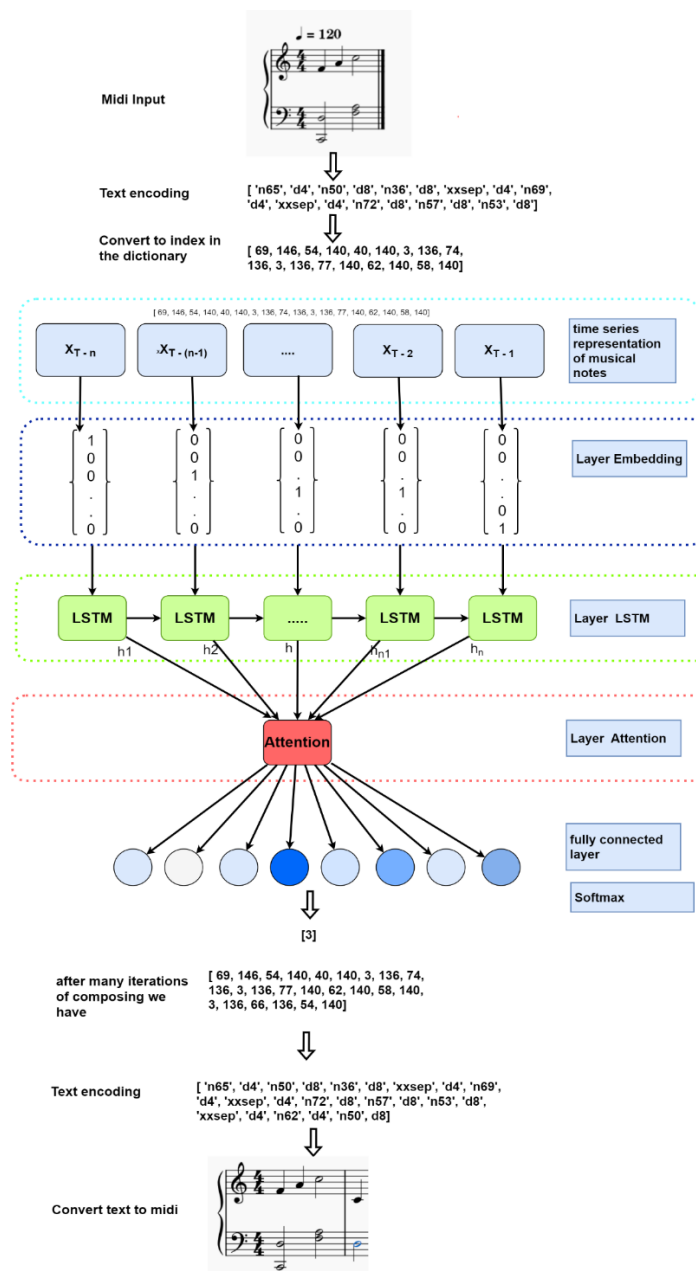
LSTM có thể xử lý các chuỗi văn bản dài và phức tạp. rất phù hợp với bài toán sáng tác giai điệu, khi mà các các đầu vào được mã hóa về dạng chuỗi các nốt nhạc (các chi tiết bước chuyển ở mục 4.1.2).



Hình 3.1. Mô hình LSTM cho bài toán sáng tác giai điệu

3.2. LSTM kết hợp Attention

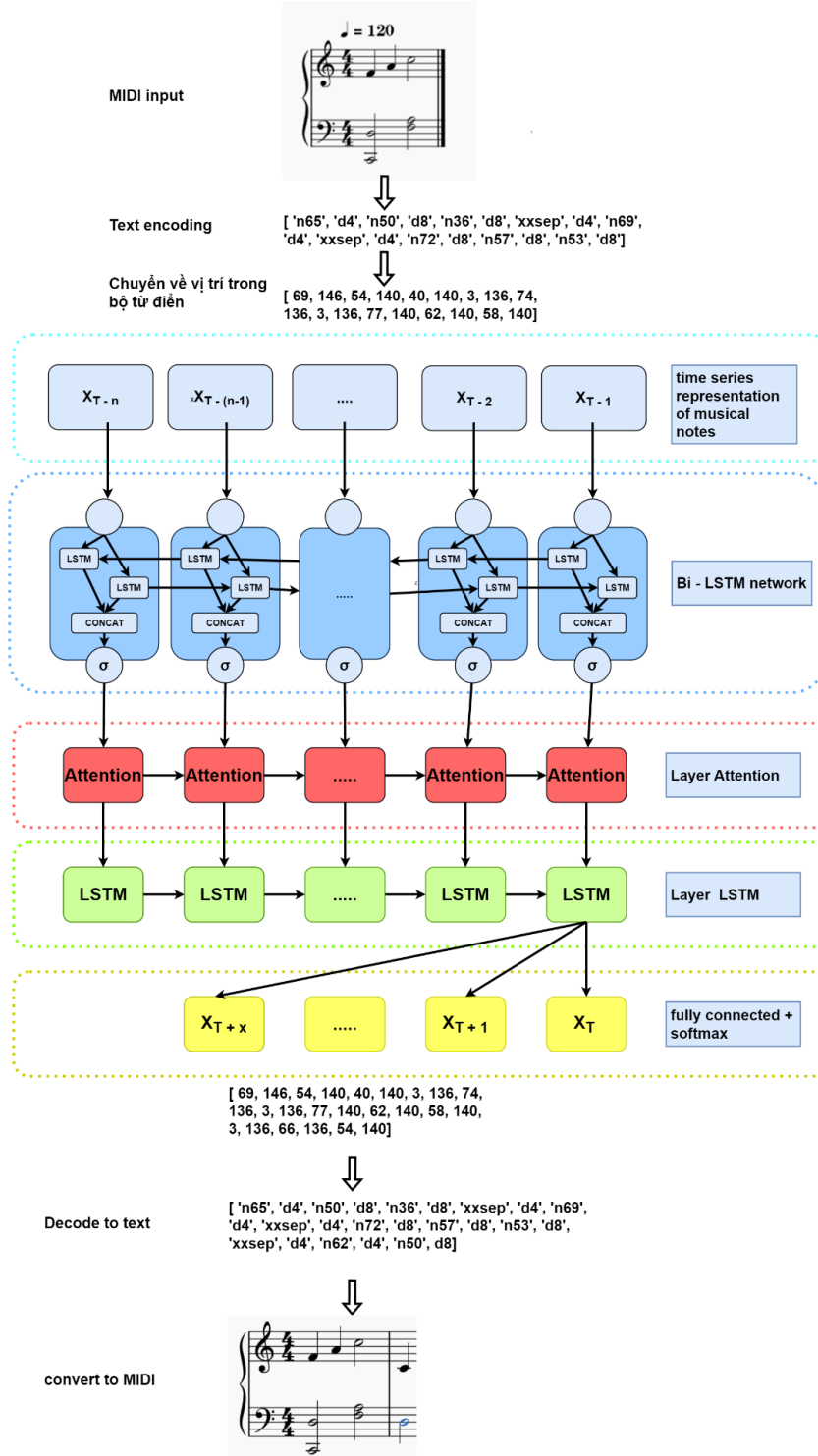
Attention là giải pháp cần thiết để giải quyết các vấn đề của LSTM là không đủ chú ý đủ vào các điểm đặc biệt quan trọng trong giai điệu. Cơ chế này cho phép mô hình tập trung vào các yếu tố quan trọng trong một chuỗi âm nhạc. Nó tạo ra một trọng số cho từng yếu tố đầu vào trong chuỗi và sau đó tính tổng có trọng số của các yếu tố này để tạo ra một vector biểu diễn chú ý (attention vector). Vector chú ý này được sử dụng để tạo ra đầu vào cho LSTM ở các bước tiếp theo, giúp mô hình nhận biết và tạo ra những phần tử quan trọng trong quá trình sáng tác nhạc.



Hình 3.2. Mô hình LSTM + attention

3.3. Bidirectional Long Short-Term Memory

LSTM hai chiều là sự mở rộng của LSTM thông thường có thể cải thiện việc thực thi mô hình đối với các vấn đề về trật tự sắp xếp. Trong các vấn đề mà tất cả các dấu thời gian của sự sắp xếp thông tin đều có thể truy cập được, LSTM hai chiều đào tạo hai thay vì một LSTM trên đoạn giai điệu.

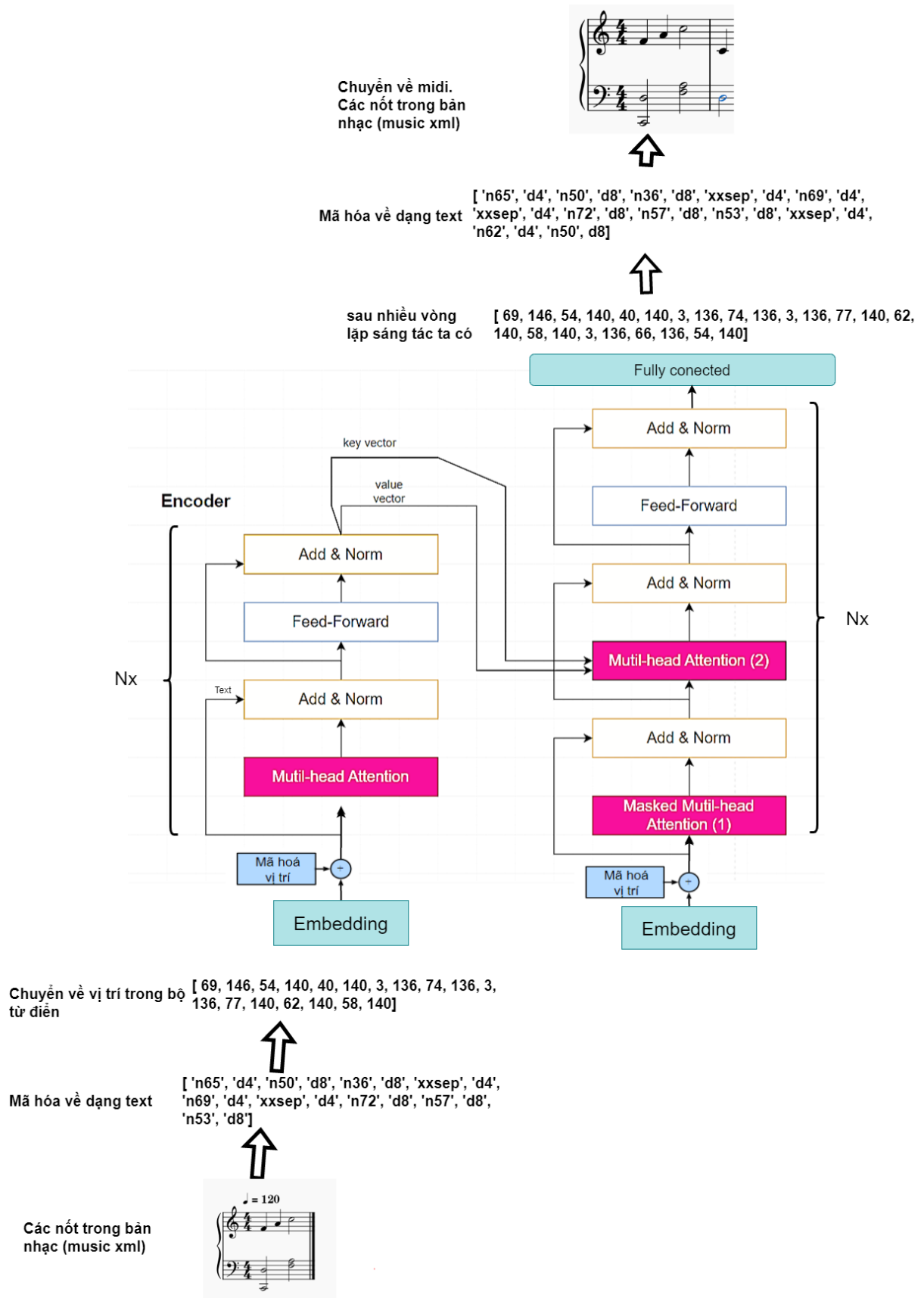


Hình 3.3. Mô hình Bi-LSTM + Attention + LSTM

3.4. Mô hình transformer cho bài toán sáng tác giai điệu

Trong những năm gần đây, transformer đã được đề xuất bởi Vaswani [6] đã nhận được kết quả tốt trong NLP, mặt khác còn được áp dụng rộng rãi trong các lĩnh vực khác. Trong nghiên cứu này em cũng áp dụng mô hình dựa trên transformer để giải quyết bài toán sáng tác giai điệu trong âm nhạc.

Bằng cách sử dụng bộ encode để nắm bắt thông tin chính từ giai điệu đầu vào, và bộ decode có thể tạo ra nốt nhạc thích hợp với các đặc trưng từ bộ encode. Lớp fully connected với chức năng chuyển đổi đầu ra từ decoder thành dạng phù hợp với số lớp cần dự đoán. Ở đây số lớp là kích thước của bộ từ điển gồm các nốt nhạc và các giá trị trường độ (được giải thích ở mục 4.1.2).



Hình 3.4. Mô hình Transformer sáng tác nhạc

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1. Dữ liệu

4.1.1. Nguồn dữ liệu

Tổng dữ liệu gồm 2000 bài hát, giai điệu dưới dạng MIDI. Dữ liệu được thu thập từ các trang mạng cung cấp các tệp midi như cungtapnhac.com, everyonepiano.com, bitmidi.com, ...

4.1.2. Đầu vào và đầu ra

Đầu vào của model AI sáng tác giai điệu từ tệp Music XML là một tệp âm nhạc được lưu dưới dạng tệp XML có thể có các định dạng (*.mid, *.xml, *.krn, *.mscz, ...). Tệp này chứa các thông tin về nốt nhạc, độ cao của nốt, nhịp điệu, hợp âm, phong cách âm nhạc, ... Thông qua các thuật toán học máy, model sẽ phân tích và học các thông tin này để tạo ra các giai điệu mới. Như vậy, đầu vào/ đầu ra của chương trình là :

Đầu vào : một chuỗi các nốt nhạc giai điệu đầu tiên (dưới dạng tệp Music XML)

Đầu ra : một chuỗi nốt tiếp theo sau giai điệu input (trả dưới dạng tệp Music XML)

Tệp Music XML có thể trình bày theo các dạng để thể hiện giai điệu như :

- Sheet Music



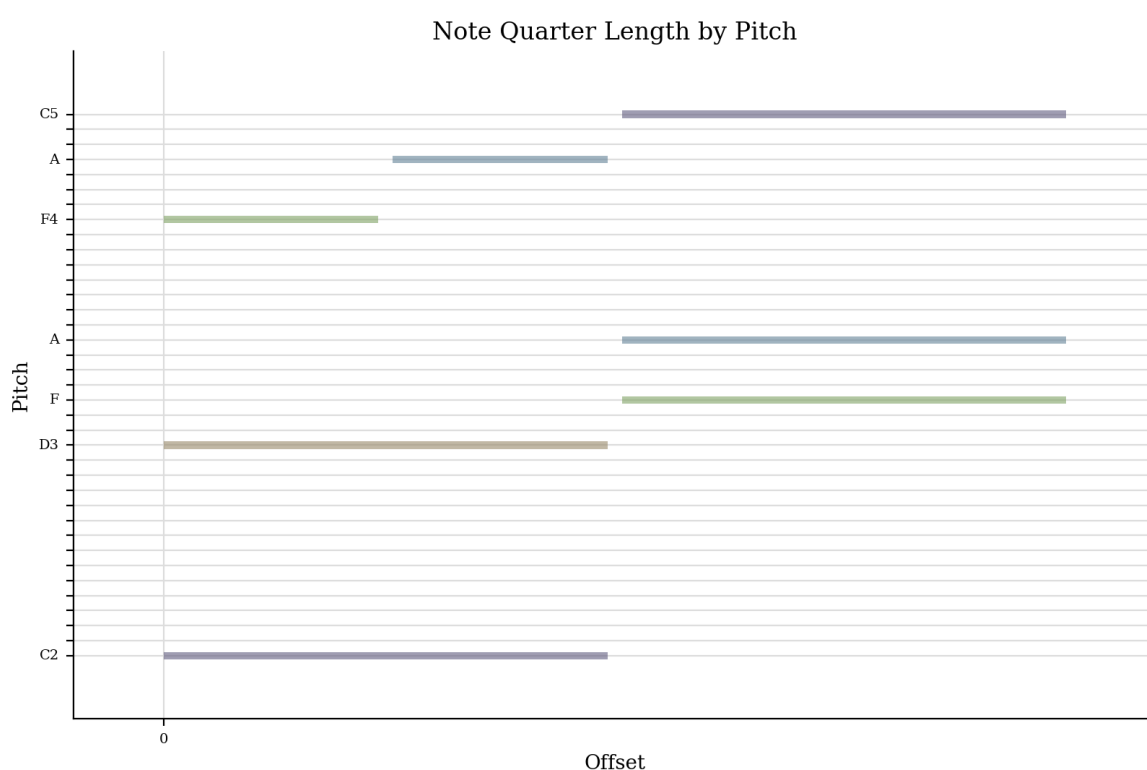
Hình 4.1. Minh họa sheet music

- Text theo từng part

```
{0.0} <music21.stream.Part 0x7f4e2c6ca2b0>
{0.0} <music21.instrument.Piano 'Melody'>
{0.0} <music21.note.Note F>
{1.0} <music21.note.Note A>
{2.0} <music21.note.Note C>
{0.0} <music21.stream.Part 0x7f4e2c6ca0d0>
{0.0} <music21.instrument.Piano 'Chords'>
{0.0} <music21.chord.Chord C2 D3>
{2.0} <music21.chord.Chord A3 F3>
```

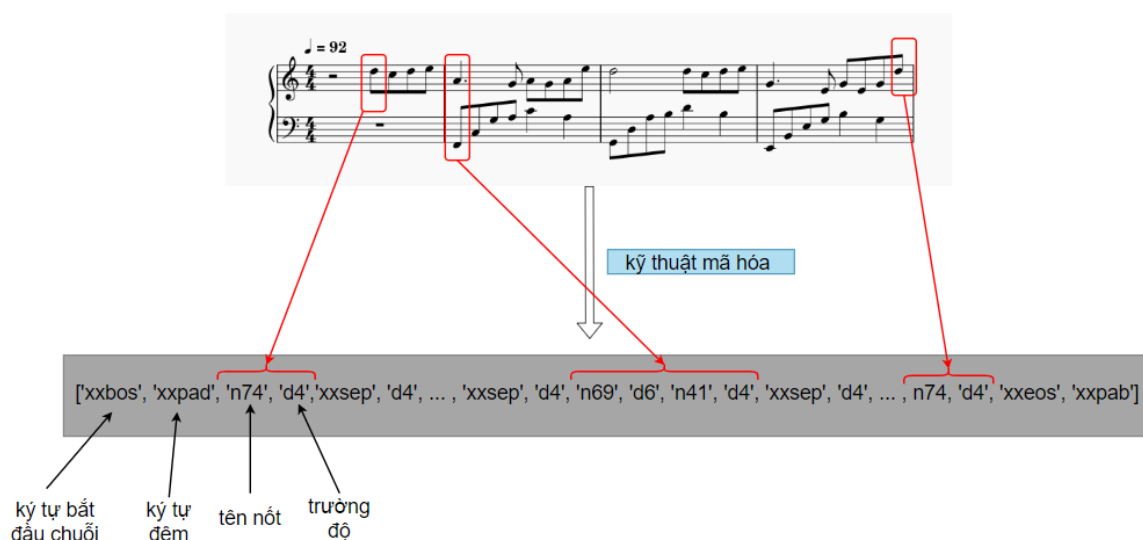
Hình 4.2. Minh họa Music XML dạng chữ

– Piano roll



Hình 4.3. Minh họa Music XML dạng Piano roll

Vì dữ liệu của chúng ta có dạng chuỗi thời gian nên sẽ mã hóa tệp music XML thành chuỗi các note theo từng bước thời gian. Trong âm nhạc đơn vị gần nhỏ nhất thường gặp là nốt có trường độ bằng 0.25 tương đương với 1 nốt móc đôi (vì độ dài thời gian là từ 0 đến đương vô cùng nên có thể tồn tại nốt có trường độ < 0.25 nhưng rất ít xuất hiện trong bản nhạc), nên đơn vị trường độ khi mã hóa một note sẽ là N số nốt móc đôi. Sau đây là ví dụ về mã hóa MIDI thành về dạng chuỗi ký tự:



Hình 4.4. Minh họa mã hóa MIDI sang chuỗi ký tự.

Trong đó:









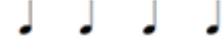






- ‘xxbos’, ‘xxpad’ được chèn vào đầu mỗi chuỗi để mô hình có thể phân biệt khi nhiều bài hát được kết hợp.
- xxsep để phân cách các nhóm nốt (hợp âm) ở một mốc thời gian khác.
- $n < \text{MIDI note number} >$ và $d < \text{thời gian của nốt} >$ để ghi lại thông tin cao độ và trường độ. Ví dụ như đoạn tokens trên ‘n60’, ‘d4’ thể hiện rằng note 60 có độ dài là 1 nốt đen, thông thường nốt nhạc có trường độ (duration) ngắn nhất trong các bản nhạc là nốt móc đôi. Nên để đơn vị thời gian là 1 nốt móc đôi thì duration của một nốt đen bằng 4 lần nốt móc đôi, do đó nốt C6 hoặc là n60 có duration là d4.

Sau đây là một chút lý thuyết cơ bản về nhạc lý và các ký hiệu của nốt nhạc:

Trong âm nhạc, ta sử dụng tên các dấu nhạc với cao độ khác nhau, gồm Do, Re, Mi, Fa, Sol, La, Si, được đặt theo thứ tự tiếng La-tin. Đây là 7 bậc cơ bản của hệ thống thất âm, được xếp từ âm thấp nhất đến âm cao nhất. Khi âm thanh có tần số cao hơn hoặc thấp hơn, chúng ta lặp lại các tên của các bậc này theo quãng 8. Chi tiết ở bảng 4.2.

Trường độ trong âm thanh trong giai điệu được thể hiện bằng các ký hiệu khác nhau (chi tiết ở bảng 4.1). Căn bản gồm dấu tròn, dấu trắng, dấu đen, dấu móc đơn, dấu móc đôi, dấu móc ba, Với trường độ nốt trước gấp 2 lần trường độ nốt sau theo thứ tự trên. Dấu lặng thể hiện khoảng nghỉ trong giai điệu. Ngoài ra còn có dấu chấm đôi là ký hiệu dấu chấm ngay sau nốt nhạc hoặc dấu lặng, làm tăng thêm $\frac{1}{2}$ trường độ nốt đó.

Bảng 4.1. Chi tiết trường độ nốt nhạc và dấu lặng

Tên	Nốt nhạc	Nốt lặng	Trường độ	Giá trị label	Số nốt trong 1 nhịp 4/4
Nốt tròn			4	d16	
Nốt trắng			2	d8	
Nốt đen			1	d4	
Nốt móc đơn			$\frac{1}{2}$	d2	
Nốt móc đôi			$\frac{1}{4}$	d1	

Bảng 4.2. MIDI note number.

Bậc	-1	0	1	2	3	4	5	6	7	8	9
Note											
C	0	12	24	36	48	60	72	84	96	108	120
C#	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
D#	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F#	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G#	8	20	32	44	56	68	80	92	104	116	
A	9	21	33	45	57	69	81	93	105	117	
A#	10	22	34	46	58	70	82	94	106	118	
B	11	23	35	47	59	71	83	95	107	119	

Từ bảng trên ta có thể chuyển nốt nhạc sang số MIDI hoặc ngược lại một cách dễ dàng. Với cột là tên nốt nhạc với cao độ cách nhau theo thứ tự là nửa cung và hàng là vị trí quãng tám trong số dãy số MIDI. Ví dụ: số MIDI = 59 là nốt **B3**, hoặc ngược lại nốt **G2** có số MIDI là 43.

Với các mã hóa midi về chuỗi các nốt theo số midi và trường độ thì số lớp cần phân lớp sẽ là: các ký hiệu đặc biệt [xxeos, xxsep], phạm vi số midi [0, ..., 127] và phạm vi trường độ [1, ..., 160]. Vậy số lớp cần phân lớp là 290 lớp.

4.1.3. Tăng cường dữ liệu

Trong âm nhạc, như chúng ta đã biết, gồm có 12 nốt nhạc chính như bảng 3.2:

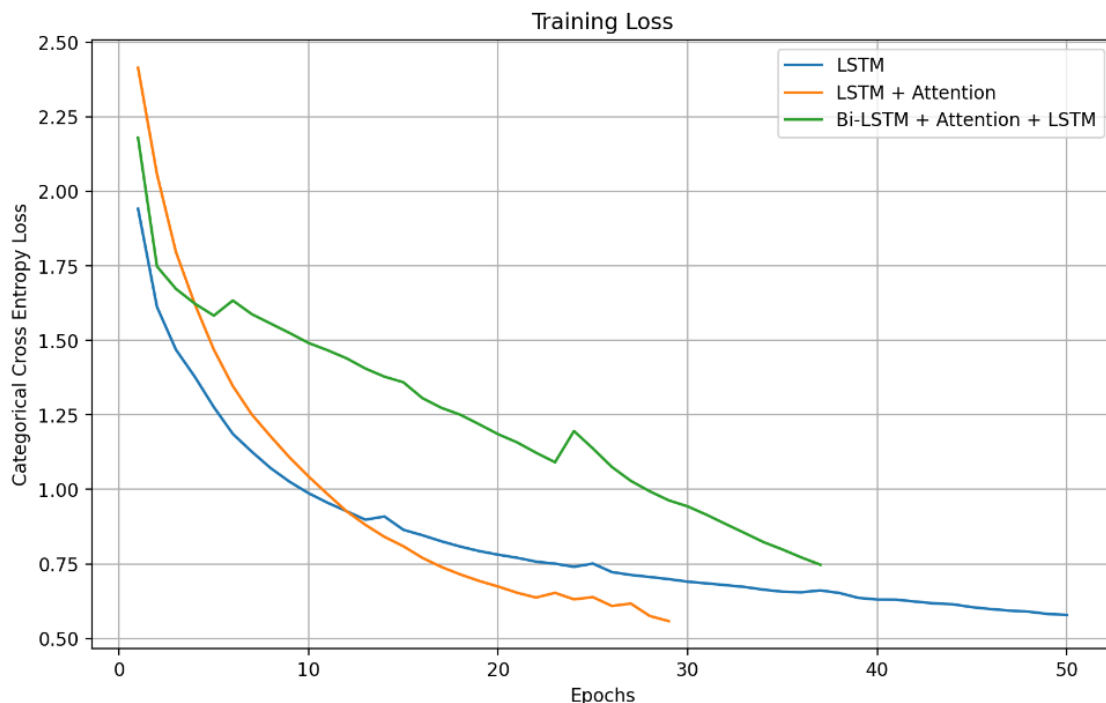
C C# D D# E F F# G G# A A# B.

Và có 2 loại cấu tạo phổ biến nhất của âm giai là âm giai Trưởng và âm giai Thứ. Vậy chúng ta có 12 âm giai Thứ và 12 âm giai Trưởng. Do đó mỗi bài hát hay mỗi giai điệu có thể chuyển thành một giai điệu mới trên một âm giai. Từ đó sử dụng công cụ transform của music21 để chuyển giai điệu lên tông và xuống tông trong khoảng cách với tông gốc là 6. Từ đó chúng ta có 12 giai điệu khác nhau từ một bài hát. Cách tăng cường dữ liệu này giúp cho mô hình đa dạng và bao phủ hết toàn bộ nốt nhạc trong phạm vi số midi.

4.2. Kết quả huấn luyện mô hình

Tiến hành huấn luyện các mô hình NLP, So sánh và đánh giá kết quả.

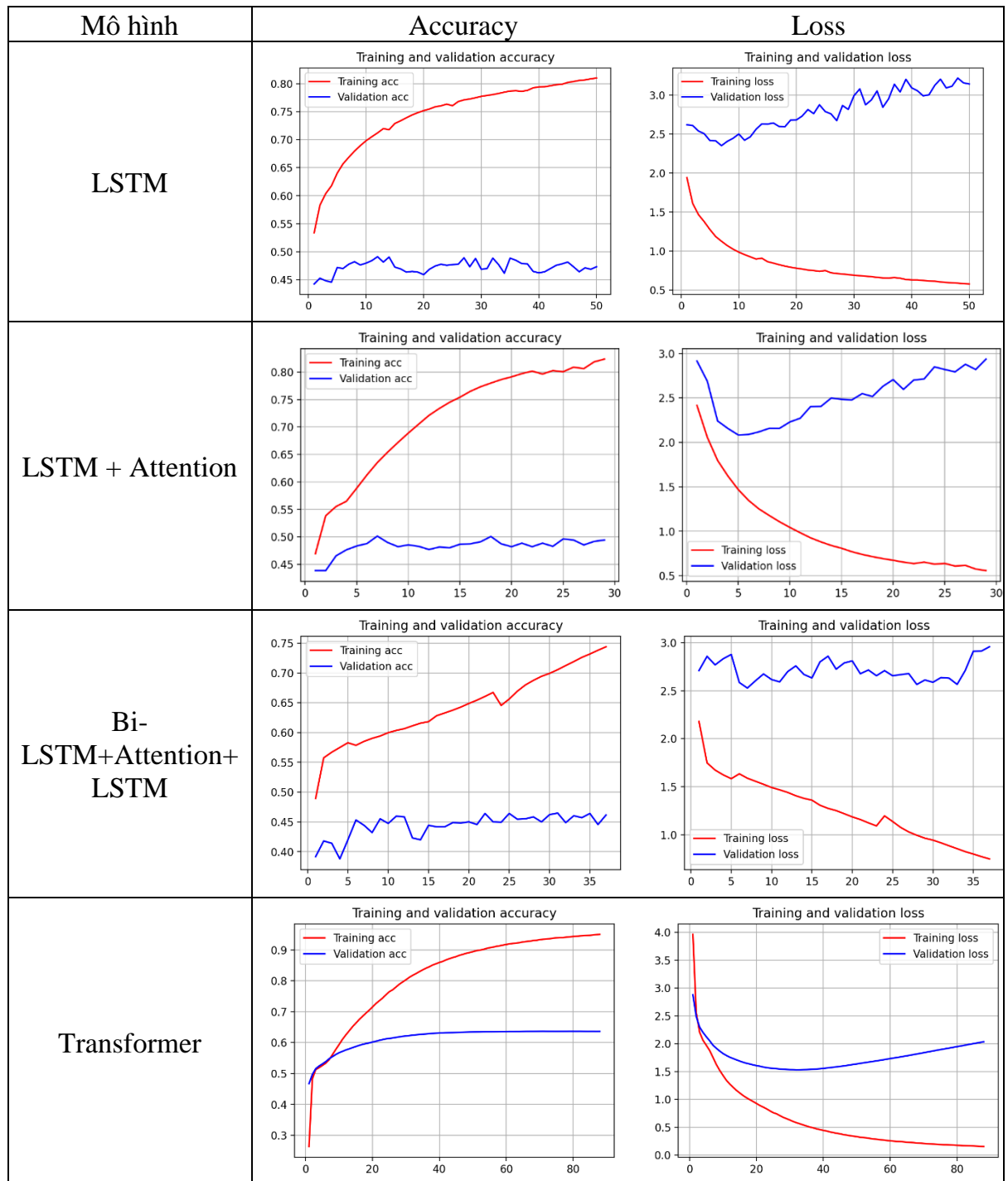
4.2.1. Đánh giá mô hình dựa vào độ chính xác và giá trị hàm mất mát



Hình 4.5. Biểu đồ Categorical Cross Entropy Loss

Sau đây là bảng chi tiết quá trình thay đổi Accuracy và Loss của tập dữ liệu huấn luyện (data train) và dữ liệu đánh giá (data validation) sau mỗi chu kỳ huấn luyện.

Bảng 4.3. Chi tiết lịch sử của Accuracy và Loss khi huấn luyện



Bảng 4.4. Hiệu suất các mô hình

Stt	Mô hình	Categorical Cross Entropy Loss	Accuracy
1	LSTM	0.5786	0.81
2	LSTM + Attention	0.5578	0.82
3	Bi-LSTM+Attention+LSTM	0.6659	0.74
4	Transformer	0.21	0.95

Trong hình 3.4 và bảng 3.3, Categorical Cross Entropy Loss của 3 mô hình: LSTM, LSTM kết hợp Attention, Bi-LSTM kết hợp Attention và LSTM. Mô hình Bi-LSTM kết hợp Attention và LSTM có vẻ học rất là chậm, không ổn định và có giá trị thấp nhất. Mô hình LSTM có tốc độ giảm hàm mất mát là ổn trên tập huấn luyện nhưng trên tập đánh giá lại khác biệt, điều này cho thấy mô hình dễ bị overfitting. Để cải thiện khả năng học tập lớp Attention được thêm vào LSTM và kết quả cho thấy nó hoạt động tốt hơn mạng LSTM truyền thống. LSTM kết hợp Attention cho kết quả tốt nhất trong các sự kết hợp lớp LSTM.

Transformer nhờ các tầng chuẩn hóa ở cuối mô hình giúp cho mô hình học ổn định hơn trong quá trình huấn luyện. Đồ thị chi tiết huấn luyện mô hình transformer cho thấy độ chính xác của mô hình là cao nhất và giá trị lỗi cũng dần về giá trị thấp nhất trong 4 mô hình.

4.2.2. Đánh giá thời gian huấn luyện và kích thước bộ trọng số

Trong quá trình huấn luyện mô hình machine learning, thời gian huấn luyện và kích thước bộ trọng số là hai yếu tố quan trọng được sử dụng để đánh giá hiệu suất của mô hình.

Mô hình cần thời gian để học từ dữ liệu huấn luyện, gọi là thời gian huấn luyện. Thời gian này phụ thuộc vào kích thước dữ liệu huấn luyện, kích thước mô hình, phương pháp tối ưu hóa được sử dụng và cấu hình phần cứng sử dụng huấn luyện. Nếu mô hình có quá nhiều tham số hoặc kích thước dữ liệu lớn, thì thời gian huấn luyện có thể rất lâu.

Kích thước bộ trọng số là dung lượng bộ nhớ mà mô hình sử dụng để lưu trữ các tham số đã học. Kích thước này phụ thuộc vào số lượng tham số của mô hình và độ lớn của từng tham số. Vì vậy, các mô hình có số lượng tham số lớn và độ phức tạp cao sẽ có kích thước bộ trọng số lớn hơn.

Thời gian sáng tác là thời gian cần thiết để mô hình dự đoán ra các nốt nhạc. Thời gian này phụ thuộc vào kích thước mô hình hoặc kiến trúc của mô hình. Nếu thời gian dự đoán quá lâu thì mô hình sẽ không thực tế để sử dụng trong ứng dụng thực tế. Vì vậy, mô hình sáng tác giai điệu cần đảm bảo rằng thời gian dự đoán của mô hình là hợp lý và có thể đáp ứng được yêu cầu của ứng dụng.

Bảng 4.5. Hiệu suất các mô hình tạo giai điệu

Stt	Mô hình	Thời gian huấn luyện (s/epoch)	Thời gian sáng tác (s/256 nốt)	Size weight (MB)
1	LSTM	168	16	9
2	LSTM + Attention	484	20	22
3	Bi-LSTM+Attention+ LSTM	745	31	39
4	Transformer	87	4	38

Tất cả các model đều được huấn luyện trên một phần cứng. Cụ thể ở đây chúng em sử dụng máy ảo Colab của Google để chạy các tác vụ tính toán. Chi tiết phần cứng được mô tả ở bảng dưới đây:

Bảng 4.6. Phần cứng colab

Phần cứng	Mô tả	
CPU	Bộ xử lý:	Intel Xeon CPU @ 2.30GHz
	RAM:	12.7 GB
	Cache size :	46080 KB
	CPU cores :	1
GPU	Tên :	NVIDIA Tesla T4
	Kiến trúc :	Turing
	CUDA cores :	3200
	Tensor Cores :	320
	Bộ nhớ :	16 GB GDDR6 với băng thông 320 GB/s
	xung nhịp lõi:	1590 MHz (base clock) và 1770 MHz (boost clock)

Tuy mô hình transformer có kích thước bộ trọng số là khá lớn nhưng vẫn chiếm ưu thế về thời gian huấn luyện và thời gian tạo ra các nốt nhạc. Với tốc độ rất nhanh (thời gian huấn luyện 87 s/epoch và thời gian sáng tác 4 s/256 nốt) so với 3 mô hình kết hợp LSTM với Attention. So sánh trực tiếp mô hình transformer với mô hình Bi-LSTM + Attention + LSTM có kích thước bộ trọng số là gần bằng nhau (38MB – 39MB) nhưng transformer tạo giai điệu nhanh gấp 7 lần Bi-LSTM + Attention + LSTM.

4.2.3. Đánh giá mô hình bằng cách đánh giá chuỗi đầu ra

Tác dụng của đánh giá chuỗi đầu ra giúp phân tích độ tương đồng giữa giai điệu được sinh ra từ các mô hình máy học và giai điệu gốc và độ đa dạng của giai điệu mới. Tuy cách đánh giá dựa trên những số liệu từ những công thức nên không có kết quả cho giai điệu hay hay tệ, nhưng có thể cho ra một điểm số tổng quát về các khía cạnh mà độ đo thống kê và tính toán từ những nốt nhạc.

Bảng 4.7. Điểm số đánh giá chuỗi đầu ra (giai điệu sáng tác)

Stt	Mô hình	METEOR	Perplexity	Pitch Accuracy
1	LSTM	0.287	22.19	0.66
2	LSTM + Attention	0.365	16.44	0.61
3	Bi-LSTM+Attention+ LSTM	0.295	18.17	0.51
4	Transformer	0.366	8.16	0.76

4.2.4. Đánh giá giai điệu bằng thính giác của con người

Ngoài các đánh giá chung bằng điểm số cho các bản nhạc thì đánh giá bằng cảm nhận của con người là cách đánh giá chi tiết hơn và có tính khả thực tế hơn.

Sau đây là mô tả và đánh về các giai điệu mà các mô hình sáng tác ra. Vì chưa có kinh phí cũng như về thời gian để có thể nhờ các chuyên gia về âm nhạc hay các nhạc sĩ để đánh giá nên em sẽ đánh giá theo cảm nhận của em, là một người đam mê âm nhạc và chơi được 2 loại nhạc cụ là piano và guitar và nghe rất nhiều nhạc.



Hình 4.6. Giai điệu đầu vào

Melody of LSTM



Trong 3 ô nhạc đầu tiên thì giai điệu có chút màu sắc nhưng không có liên kết giữa giai điệu input.



3 ô nhạc tiếp theo giai lại bị lặp lại nhiều lần. đây là lỗi không nên có trong âm nhạc.

phần cuối giai điệu này cũng có chút thay đổi và thay đổi nhịp điệu. Giai điệu và hợp âm là phù hợp với nhau. Nhưng tổng quan giai điệu tạo ra có nhịp điệu và màu sắc không giống với giai điệu input



Hình 4.7. Giai điệu sinh ra từ mô hình LSTM

Melody of LSTM +Attention



3 ô nhạc đầu có vẻ kết nối với giai điệu input khá tốt



Giai điệu bị lặp đi lặp lại nhiều lần ở 1 nốt và hợp âm. không có dấu hiệu thay đổi

Hình 4.8. Giai điệu sinh ra từ mô hình LSTM +Attention

. Melody of Bi-LSTM



Hình 4.9. Giai điệu sinh ra từ mô hình Bi-LSTM + Attention + LSTM

Giai điệu từ mô hình Bi-LSTM + Attention + LSTM có rất nhiều vấn đề. Về giai điệu không có sự hòa hợp giữa giai điệu và hợp âm, không dùng tông với giai điệu input, và không có liên quan hay tiếp tục với giai điệu input.

Melody of Transformer

$\text{♩} = 120$

Từ đầu tới ô thứ 8 đã tạo ra giai điệu khá hay. sự liên kết hòa hợp của giai điệu với hợp âm rất tốt, tuy nhiên nó còn mang 1 chút giống với giai điệu gốc.

giai điệu ở khúc giữa bài đến khuôn nhạc thứ 17 có sự khác biệt với giai điệu gốc, nhưng nhịp độ có chút lệch và giai điệu chưa bắt tai.

Đoạn cuối này cho 1 giai điệu rất thú vị khi giai điệu về đúng nhịp và có các khoảng trống hay ngân dài nốt.

Hình 4.10. Giai điệu sinh ra từ mô hình Transformer

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Ưu điểm

Qua các thông số đánh giá ở chương 3 chúng ta có thể thấy mô hình tốt nhất là transformer. Mô hình có thời gian sáng tác giai điệu nhanh chóng. Đảm bảo được nhịp điệu ổn định, và các hợp âm và giai điệu hòa hợp cùng một âm giai. Có thể điều chỉnh độ sáng tạo khi sáng tác giai điệu.

5.1.2. Hạn chế

Giai điệu sáng tác chưa đa dạng về nhiều thể loại nhạc và có một chút đạo lại các giai điệu trong dữ liệu huấn luyện. Giai điệu mới tuy có chút sáng tạo nhưng không có điểm nhấn trong nghệ thuật hay cấu trúc của một bài hát mà con người sáng tác.

Vì dữ liệu các tệp MIDI ở các nước ngoài công khai nhiều hơn nên dữ liệu chủ yếu vẫn đang là âm nhạc nổi tiếng của các bài hát tiếng Anh, Trung, Nhật Bản, Việt Nam chỉ chiếm khoảng 20% dữ liệu.

5.2. Hướng phát triển

Nâng cao dữ liệu đa dạng hơn, cập nhật các bài hát nổi tiếng và các giai điệu mới hay hơn. Phân loại từng nhạc cụ để phù hợp với mô hình.

Nghiên cứu và xây dựng lại mô hình để giai điệu đúng theo cấu trúc bài hát. Tạo ra các mô hình đa nhiệm có khả năng tạo ra các đầu ra có tính đa dạng về âm sắc, lời bài hát, nhịp điệu, nhiều loại nhạc cụ riêng biệt. Kết hợp transformer và GANs để xây dựng mô hình có khả năng tránh đạo các giai điệu trong dữ liệu.

Xem xét các mô hình dựa trên học tăng cường (reinforcement learning) để tạo ra giai điệu phù hợp với sở thích của người dùng hoặc các phương pháp giao tiếp khác để tạo ra các bản nhạc dựa trên ý tưởng của người dùng.

TÀI LIỆU THAM KHẢO

- [1] A. Spiliopoulou, "A Topic Model for Melodic Sequences," 2012.
- [2] A. B. K. E. & B. T. Jansson, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*, 2019.
- [3] "GitHub repository," MusicAutobot, 2021. [Online]. Available: <https://github.com/bearpelican/musicautobot>. [Accessed 3 2023].
- [4] P. D. Khanh, "Lý thuyết về mạng LSTM part 2," 22 Apr 2019. [Online]. Available: https://phamdinhhkhanh.github.io/2019/04/22/Ly_thuyet_ve_mang_LSTM.html. [Accessed Feb 2023].
- [5] Y. Verma, "Complete Guide To Bidirectional LSTM (With Python Codes)," [Online]. Available: [https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=Bidirectional%20long-short%20term%20memory\(bi-lstm\)%20is,different%20from%20the%20regular%20LSTM..](https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=Bidirectional%20long-short%20term%20memory(bi-lstm)%20is,different%20from%20the%20regular%20LSTM..) [Accessed 4 2023].
- [6] "Attention Is All You Need," 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>. [Accessed March 2023].
- [7] S. B. a. A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, Michigan*, pp. 65-72, June 2005.
- [8] D. Y. L. Deng, "Deep Learning: Methods and Applications," *IEEE Signal Processing Magazine*, vol. 31, pp. 147-148, May 2014.
- [9] M. Cuthbert, "music21 documentation," 2021. [Online]. Available: web.mit.edu/music21/doc/moduleReference/moduleStreamCore.html. [Accessed January 2023].

- [10] R. M. P. J. Sanidhya Mangal, "LSTM Based Music Generation System," 2019.
- [11] Q. Pham, "Tim hieu mo hinh Transformer," 20 March 2020. [Online]. Available: <https://pbcquoc.github.io/transformer/>. [Accessed March 2023].
- [12] J. -. M. R. -. G. E. TY - JOURAU - Bosch, "Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music," *Journal of New Music ResearchER*, 2016.
- [13] R. M. v. P. J. S. Mangal, "LSTM Based Music Generation System," Bhubaneswar, India, 2018.
- [14] P. Mukherjee, "Attentional networks for music generation," 2020.
- [15] M. &. Jurafsky, "Speech and Language Processing," 2023, p. 52.
- [16] B. &. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," June 2005. [Online].

PHỤ LỤC

Hình ảnh ứng dụng sáng tác nhạc được viết bằng Flask. Tạo giao diện để dễ dàng sử dụng. Ứng dụng chưa được hoàn thiện nên em chỉ trình bày chạy trên localhost.

