Sáng tạo âm nhạc ứng dụng Deep Learning

1st Nguyễn Ngọc Lương
UIT 21522311
KHDL2021
21522311@gm.uit.edu.vn

***Tóm tắt nội dung*—Sáng tạo âm nhạc bằng Deep Learning là một chủ đề tiềm năng đã thu hút được nhiều chú ý trong những năm gần đây. Các nhà nghiên cứu đã phát triển nhiều mô hình có kết quả rất thuyết phục, tuy nhiên chúng thường thiếu cấu trúc và sự sáng tạo. Trong nội dung bài báo cáo này, nhóm sẽ tìm hiểu và trình bày cách cài đặt mô hình LSTM cho bài toán 'Sáng tạo âm nhạc bằng Deep Learning'. Nhóm đã tiến hành xử lí dữ liệu, sau đó tối ưu mô hình. Cuối cùng, tiến hành xây dựng hàm sáng tạo âm nhạc từ dữ liệu dự đoán.**

***Index Terms*—Preprocessing, Deep Learning, Data Analysis, Music Generation**

I. Giới thiệu

A. Giới thiệu đề

Âm nhạc là một hình thức nghệ thuật âm thanh có sắc thái và cấu trúc, thường được tạo ra thông qua sự kết hợp của các yếu tố như âm sắc, nhịp, độ cao, độ thấp và cấu trúc nhạc lý. Nó không chỉ là một phần quan trọng của văn hóa và lịch sử con người mà còn là một phương tiện mạnh mẽ để truyền đạt cảm xúc, ý nghĩa và trải nghiệm.

Âm nhạc có thể mang nhiều hình dạng khác nhau, bao gồm âm nhạc cổ điển, nhạc dân gian, nhạc pop, rock, hip-hop, jazz, blues và nhiều thể loại khác. Mỗi thể loại đều có đặc điểm và phong cách riêng, thường phản ánh sự đa dạng và sự phát triển của xã hội và văn hóa tại mỗi thời kỳ lịch sử.

Sáng tạo âm nhạc bằng deep learning là một lĩnh vực nghiên cứu và phát triển đầy thú vị, nơi mà công nghệ học sâu được áp dụng để tạo ra và thúc đẩy quá trình sáng tác âm nhạc. Thay vì phụ thuộc hoàn toàn vào sự sáng tạo của con người, các mô hình Deep Learning trong lĩnh vực này có khả năng học từ dữ liệu âm nhạc có sẵn và tạo ra các tác phẩm mới, độc đáo. Các nhà nghiên cứu sử dụng mô hình deep learning để hiểu và mô phỏng các quy ước âm nhạc, từ các yếu tố cơ bản như nhịp và âm sắc đến các khía cạnh phức tạp hơn như phong cách biểu diễn và cấu trúc nhạc lý. Các mô hình này có thể học từ một lượng lớn dữ liệu âm nhạc, bao gồm cả các tác phẩm nổi tiếng và đa dạng.

Tuy nhiên, nói chung, các mô hình này cũng đối mặt với những thách thức, bao gồm việc thiếu sự sáng tạo và cấu trúc trong các tác phẩm được tạo ra. Các nhà nghiên cứu đang liên tục nỗ lực để cải thiện hiệu suất của mô hình và tăng cường khả năng tương tác của con người trong quá trình sáng tạo âm nhạc bằng deep learning. Điều này thường đưa ra những khám phá đáng chú ý trong lĩnh vực học sâu và âm nhạc, mở ra nhiều khả năng mới trong sự đổi mới âm nhạc.

Trong báo cáo này, trước tiên nhóm trình bày bộ dữ liệu trong Phần 2. Hướng tiếp cận của hệ thống sẽ được mô tả chi tiết ở Phần 3. Ở Phần 4, nhóm sẽ tiến hành thực nghiệm với mô hình và đánh giá kết quả. Cuối cùng, nhóm đưa ra kết luận và hướng phát triển ở Phần 5.

Bộ dữ liệu được nhóm thu thập tại trang web Classical Piano **Classical Piano** - Một trang web chuyên lưu trữ các file piano cổ điện dưới dạng midi.

B. Vấn đề nghiên cứu

Trong lĩnh vực sáng tạo âm nhạc bằng deep learning, những vấn đề nghiên cứu quan trọng bao gồm sự đa dạng và sáng tạo của âm nhạc phân tích và mô phỏng phong cách biểu diễn, giải quyết vấn đề cấu trúc và logic âm nhạc, cùng với xử lý dữ liệu lớn và hiệu suất mô hình. Nghiên cứu trong những lĩnh vực này nhằm cải thiện khả năng tạo ra âm nhạc, giúp tạo ra những bản nhạc độc đáo, sáng tạo và có tính tương tác cao. Điều này có thể mở ra những triển vọng mới trong lĩnh vực sáng tạo âm nhạc, tạo ra những trải nghiệm âm nhạc mới mẻ và độc đáo.

C. Mục tiêu của bài báo cáo

Mục tiêu của đề tài này là xây dựng và triển khai một mô hình sáng tạo âm nhạc. Cụ thể, đề tài tập trung vào các mục tiêu sau:

- **Xử lý dữ liệu âm thanh**: từ dữ liệu âm thanh ban đầu, nhóm sẽ xây dựng hàm trích xuất đê trích xuất dữ liệu, sau đó sẽ tiền hành tiền xử lý và chia dữ liệu để huấn luyện mô hình.
- **Xây dựng mô hình**: Phát triển một mô hình Deep Learning tối ưu trên bộ dữ liệu mà nhóm thu thập
- **Sáng tạo âm nhạc**: Xây dựng hàm sáng tạo âm nhạc từ dữ liệu dự đoán của mô hình. Dữ liệu âm thanh được tạo ra có thể tương đông với bản gốc những vẫn sẽ có những điểm mới, khác lạ so với bản gốc.

D. Các công trình liên quan

Một cuộc khảo sát và phân tích toàn diện của Briot và đồng nghiệp về các kỹ thuật học sâu để tạo nội dung âm nhạc có sẵn trong cuốn sách [1]. Herremans và đồng nghiệp [2] đề xuất một mô hình phân loại hướng chức năng cho các loại hệ thống tạo ra âm nhạc khác nhau. Các cuộc khảo sát về các phương pháp AI dựa trên thuật toán âm nhạc được thực hiện bởi Papadopoulos và Wiggins [3] cũng như Fernandez và Vico [4], cũng như các sách của Cope [5] và Nierhaus [6]. Graves [7] phân tích ứng

dụng của kiến trúc mạng nơ-ron tái lặp để tạo ra chuỗi (văn bản và âm nhạc). Fiebrink và Caramiaux [8] đề cập đến vấn đề sử dụng máy học để tạo ra âm nhạc sáng tạo. Eck và Schmidhuber [9] đã sử dụng LSTM để giải quyết vấn đề thiếu sự nhất quán trong sáng tác theo thuật toán do chúng tốt hơn trong việc học so với RNN thông thường. Công trình của họ đã chứng minh khả năng của mạng LSTM trong việc học cấu trúc cục bộ và toàn cục, tái tạo các đoạn nhạc một các hoàn chỉnh. Tuy nhiên, mạng có xu hướng ràng buộc với các quy ước trong tập dữ liệu đào tạo, ngăn chặn khả năng khám phá và tạo ra các hình thức âm nhạc mới. Để cải thiện hiệu suất của mô hình LSTM trong nhiệm vụ tạo ra âm nhạc, Eck và Lapalme [10] đề xuất một bộ học chuỗi cụ thể cho âm nhạc có thể bắt kịp cấu trúc theo thời gian dài trong tác phẩm âm nhạc. Họ giới thiệu một độ chệch hướng về cấu trúc nhịp để đối mặt với vấn đề của mạng trong việc học các chuỗi âm nhạc lặp lại bằng cách cung cấp các bản sao trễ thời gian của đầu vào.

II. Bộ dữ liệu

A. Mô tả dữ liệu và các thuộc tính

Bộ dữ liệu được nhóm thu thập tại trang web **Classical Piano** - Một trang web chuyên lưu trữ các file piano cổ điện dưới dạng midi. Bộ dữ liệu bao gồm 29 file piano của nhà soạn nhạc cổ điển nổi tiếng người Đức **Ludwig van Beethoven**. Ông là một hình tượng âm nhạc quan trọng trong giai đoạn giao thời từ thời kỳ âm nhạc cổ điển sang thời kỳ âm nhạc lãng mạn. Ông được coi là Người dọn đường (Wegbereiter) cho thời kỳ âm nhạc lãng mạn. Beethoven được khắp thế giới công nhận là nhà soạn nhạc vĩ đại, nổi tiếng và có ảnh hưởng nhất tới rất nhiều những nhà soạn nhạc, nhạc sĩ và khán giả về sau.

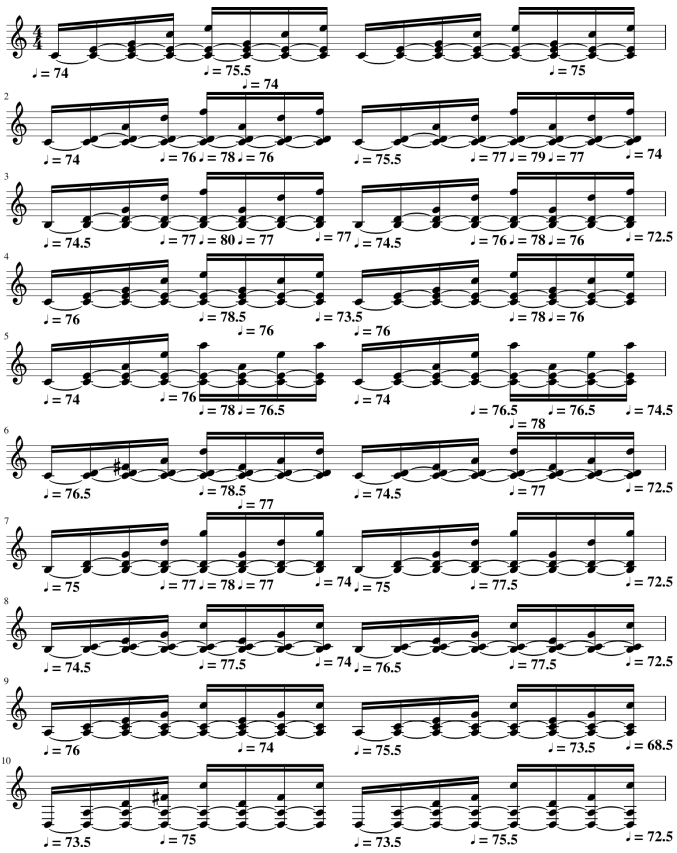
B. Khai phá dữ liệu

Trong ngôn ngữ âm nhạc, "note"(nốt nhạc) và "duration"(thời gian tồn tại) là hai khái niệm quan trọng liên quan đến việc biểu diễn âm nhạc và thời gian mỗi nốt được chơi. Dưới đây là mô tả chi tiết về cả hai khái niệm:

- **Note (Nốt nhạc)**: Là các biểu tượng trên bản nhạc, thường được đặt trên dải nền của bảng nhạc, đại diện cho âm thanh cụ thể. Các nốt có thể là nốt dài, nốt ngắn, nốt tròn, nốt vuông, và nhiều hình thức khác nhau tượng trưng cho thời gian và tần số của âm thanh.
- **Duration (Thời gian tồn tại)**: Là thời gian mà mỗi nốt được chơi hoặc giữ lại. Nó được đo bằng các đơn vị như nửa nốt, tám nốt, một nốt, và những đơn vị thời gian khác. Mỗi loại nốt có thời gian tồn tại khác nhau, và chúng xác định tốc độ và nhịp của bản nhạc.

Ngoài ra, còn một số khái niệm quan trọng nữa như 'pitch' - độ cao và 'chord' - hợp âm. Mỗi nốt nhạc sẽ có một độ cao nhất định, việc thay đổi độ cao của một nốt nhạc sẽ tạo ra một nốt nhạc khác. Hợp âm là sự kết hợp của ba nốt nhạc trở lên đồng thời được chơi hoặc hát cùng nhau. Nó là một nhóm các nốt nhạc được chơi cùng nhau để tạo ra một âm thanh thích hợp. Có nhiều loại hợp âm khác nhau, bao gồm hợp âm cơ bản, hợp âm 7, hợp âm sus4, và nhiều loại khác. Mỗi loại hợp âm mang đến một cảm giác âm nhạc khác nhau.

Note và duration cùng nhau tạo ra sự liền mạch và hấp dẫn trong âm nhạc. Sự sáng tạo trong việc sử dụng các loại nốt và thời gian tồn tại giúp âm nhạc trở thành một ngôn ngữ biểu đạt mạnh mẽ, chuyển đưa thông điệp và cảm xúc từ người sáng tác đến người nghe.



Hình 1. Cấu trúc của một bản nhạc

Sau khi thu thập dữ liệu, nhóm tiến hành xây dựng hàm trích xuất note và duration từ dữ liệu âm thanh ban đầu. Kết quả thu được hơn 60000 note và duration. Sau khi kiểm tra, nhóm nhận thấy rằng có một số nốt nhạc được chơi rất ít, cá biệt có những nốt chỉ được chơi 1 lần trên cả bộ dữ liệu, điều này có thể làm mô hình dự đoán không hiệu quả. Vì vậy, nhóm tiến hành loại các nốt có số lần chơi ít hơn 10. Sau khi loại bỏ những nốt hiếm, thu được bộ dữ liệu gồm 32319 mẫu.

Duration	**0.25, 0.5, 1.0, 1/6, 1/12, 5/12, 1/3, ...**
Note	**F1.F2, D2.D3, C#2.C#3.B4, E-2.B2.E-3.B3.F#4.B4, E2.E3.G#3.B3.E4, ...**

Hình 2. Dữ liệu sau khi được trích xuất

Tiếp theo nhóm tiến hành chia tập dữ liệu cho 32 nốt để tạo ra tập huấn luyện. Mục tiêu là nốt và dấu thời gian tiếp theo trong chuỗi. Cuối cùng, tiến hành chuyển dữ liệu nhãn thành dạng one-hot.

III. Hướng tiếp cận

Với đề tài này, chúng em sử dụng mô hình LSTM và GRU kết hợp với embedding để giải quyết bài. Mô hình LSTM và GRU trong báo cáo sử dụng 2 lớp embedding: embedding cho note và embedding cho duration. Nhóm sử dụng hàm loss categorical-crossentropy cho cả 2 mô hình.

Sau khi huấn luyện mô hình, nhóm sẽ tiến hành xây dựng một bản nhạc từ dữ liệu dự đoán và so sánh kết quả của 2 mô hình.

A. Long Short-Term Memory (LSTM)

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks) [11], thường được gọi là LSTM - là một dạng đặc biệt của RNN [12], nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter và Schmidhuber (1997) [13], và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.

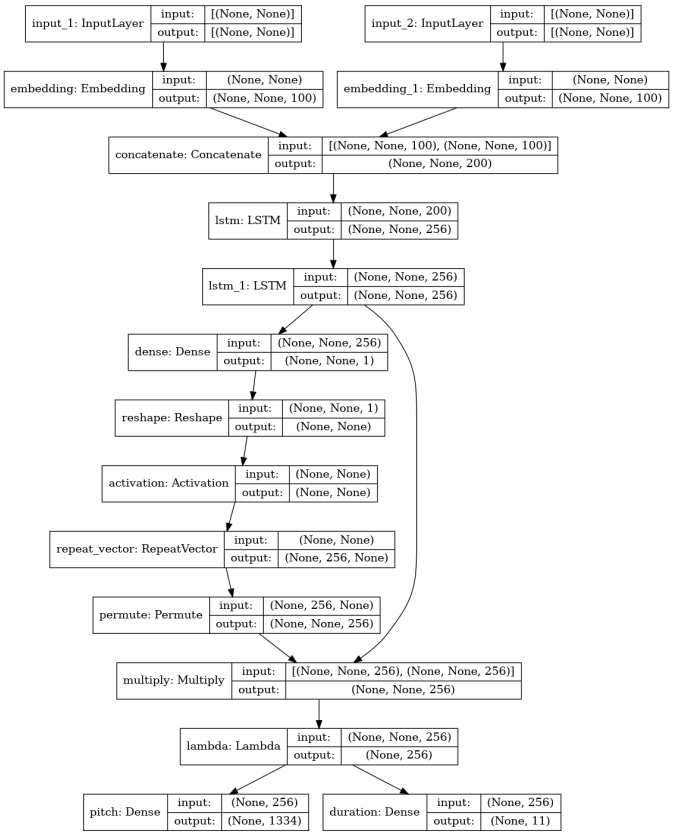
So với RNN, LSTM có khả năng xử lý tốt hơn các bài toán với dữ liệu chuỗi thời gian dài, vì nó có thể giữ lại những thông tin quan trọng và loại bỏ những thông tin không cần thiết. So với GRU [14], LSTM có cấu trúc phức tạp hơn và yêu cầu nhiều tài nguyên hơn để huấn luyện, nhưng nó có khả năng xử lý tốt hơn một số bài toán NLP với dữ liệu chuỗi thời gian. LSTM có cấu trúc gồm:

- **Memory cell**: là phần chứa thông tin quan trọng trong thời gian dài.
- **Input gate**: điều khiển việc cập nhật thông tin vào memory cell.
- **Forget gate**: điều khiển việc xóa thông tin không cần thiết từ memory cell.
- **Output gate**: điều khiển việc truyền thông tin từ memory cell ra ngoài mô hình.

Các phần trên hoạt động đồng thời và những giá trị tính toán từ các neural network đều được sử dụng để cập nhật, xóa hoặc truyền thông tin từ memory cell ra ngoài mô hình. Điều này cho phép mô hình LSTM giữ lại những thông tin quan trọng trong thời gian dài và loại bỏ những thông tin không cần thiết.

B. Gated Recurrent Unit

Recurrent Neural Network (RNN) [12] là một loại neural network hồi quy dựa trên việc sử dụng cùng một hàm tính toán để xử lý các đầu vào liên tục. Trong RNN, trạng thái hiện tại được sử dụng để dự báo trạng thái tiếp theo và quá trình này được lặp đi lặp lại cho mỗi đầu vào trong chuỗi. Gated Recurrent Unit (GRU) [14] là một phiên bản cải tiến của RNN, được sử dụng trong học sâu cho các bài toán như phân loại văn bản, dự đoán chuỗi thời gian và giải quyết các bài toán Natural Language Processing (NLP) khác. GRU có cấu trúc gồm 2 cổng (gate): cổng cập nhật trạng thái (update gate) và cổng xóa trạng thái (reset gate). Các cổng này dùng để quản lý và giữ nguyên thông tin trong một chuỗi và cho phép model tự động chọn việc cập nhật hay loại bỏ thông tin cũ. Cụ thể, update gate sẽ xác định mức độ cập nhật thông tin mới vào hidden state, trong khi reset gate sẽ xác định mức độ loại bỏ thông tin cũ để chuẩn bị cho việc nhập thông tin mới. Sau đó, một giá trị mới của hidden state được tính toán dựa trên giá trị cũ và thông tin mới. Kết quả cuối cùng của GRU là giá trị của hidden state đó, sẽ được dùng để dự đoán kết quả. GRU có thể hoạt động tốt với dữ liệu time-series bằng cách giữ nguyên những thông tin quan trọng và loại bỏ những thông tin không cần thiết.



Hình 3. Mô hình LSTM nhóm sử dụng

IV. Thực nghiệm và kết quả

A. Cài đặt thực nghiệm

Embedding của nhóm gồm có 2 phần: embedding cho note và embedding cho duration với kích thước là 100. Nhóm chọn RMSprop làm optimizer với learning rate 0.001, sử dụng hàm mất mát 'categorical crossentropy', số epoch là 40 để huấn luyện mô hình.

B. Độ đo đánh giá

Trong bài báo cáo này, nhóm sử dụng 3 độ đo đánh giá là Accuracy, Precision và Recall.

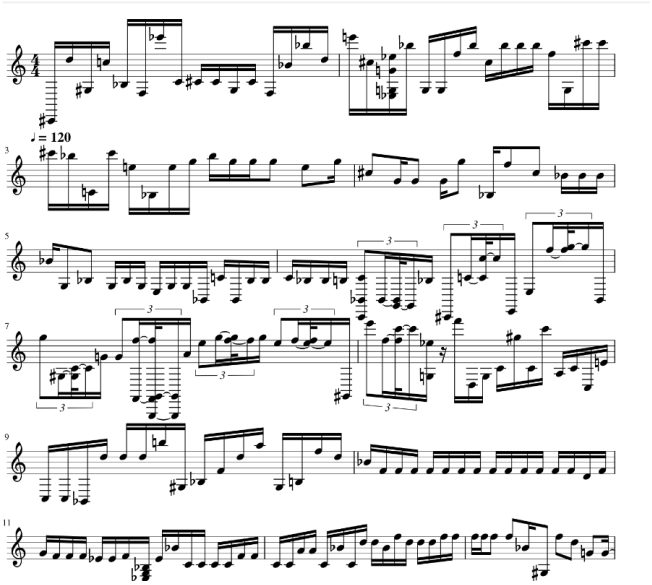
C. Đánh giá kết quả

1) Thử nghiệm lần 1: Đầu tiên nhóm đặt batch-size = 32, learning-rate = 0.001 và tiến hành huấn luyện, kết quả được biểu diễn ở bảng 1:

	Accuracy		Precision		Recall	
	note	duration	note	duration	note	duration
GRU-32	0.0705	0.7012	0.9255	0.7802	0.0921	0.6312
LSTM-32	0.0824	0.6617	0.9121	0.7651	0.0872	0.5688

Bảng I
KẾT QUẢ CÁC MÔ HÌNH VỚI BATCH-SIZE = 32

Cả 2 mô hình đều có accuracy và recall cho biến 'note' rất thấp, chỉ dưới 10%, tuy nhiên lại có precision trên 90%. Điều này có thể lý giải bởi trong toàn bộ bộ dữ liệu, có rất nhiều nốt khác nhau, điều này gây trở ngại cho việc phân loại đúng các nốt nhạc. Tuy nhiên, xét về mục đích ban đầu của bài toán là sáng tạo ra một bản nhạc mới, chứ không phải là sao chép lại bản nhạc ban đầu, điều này có thể chấp nhận được.



Hình 4. Bản nhạc sau được tạo từ kết quả dự đoán mô hình GRU-32

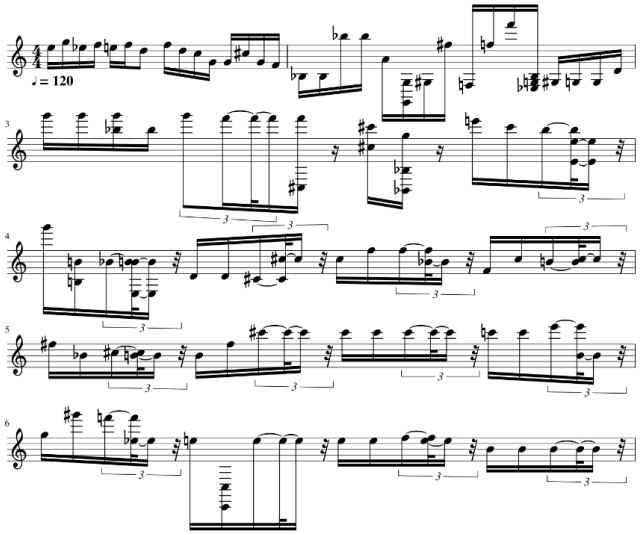
Từ hình 4 có thể thấy mô hình đã thành công tạo ra một bản nhạc với đầy đủ các nốt, hợp âm, tuy nhiên vẫn có vài vị trí những nốt nhạc bị trùng, lặp đi lặp lại.

2) Thử nghiệm lần 2: Tiếp theo nhóm đặt batch-size = 64, learning-rate = 0.001 và tiến hành huấn luyện, kết quả được biểu diễn ở bảng 2:

	Accuracy		Precision		Recall	
	note	duration	note	duration	note	duration
GRU-64	0.0842	0.6497	0.9268	0.7248	0.0042	0.5526
LSTM-64	0.0923	0.6353	0.9346	0.7808	0.0057	0.5739

Bảng II
KẾT QUẢ CÁC MÔ HÌNH VỚI BATCH-SIZE = 64

Nhìn chung cả 2 mô hình đều có kết quả tương tự so với lần thử đầu tiên. Tuy nhiên bản nhạc được tạo ra lại có tính liên kết và giống với bản gốc hơn.



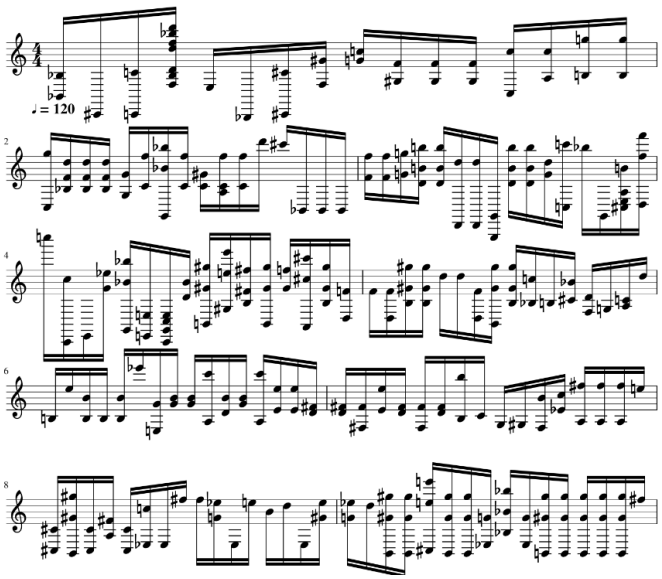
Hình 5. Bản nhạc sau được tạo từ kết quả dự đoán mô hình LSTM-64

3) Thử nghiệm lần 3: Cuối cùng nhóm đặt batch-size = 128, learning-rate = 0.001 và tiến hành huấn luyện, kết quả được biểu diễn ở bảng 3:

	Accuracy		Precision		Recall	
	note	duration	note	duration	note	duration
GRU	0.0722	0.6722	0.9126	0.7312	0.0424	0.5581
LSTM	0.1062	0.7072	0.9481	0.8032	0.0510	0.6015

Bảng III
KẾT QUẢ CÁC MÔ HÌNH VỚI BATCH-SIZE = 128

Sau 3 lần thử, nhóm nhận thấy mô hình LSTM với batch-size = 128 cho kết quả tốt nhất.



Hình 6. Bản nhạc sau được tạo từ kết quả dự đoán mô hình LSTM-128

Ở lần thử nghiệm này ta có thể thấy hiện tượng lặp nốt đã không còn, tuy nhiên bản nhạc có tiết tấu rất nhanh và khá rời

rạc, những hợp âm chưa thực sự kết hợp với nhau một cách trơn tru.

V. Kết luận

Trong báo cáo này, nhóm đã xây dựng mô hình sáng tạo âm nhạc ứng dụng Deep Learning. Qua quá trình xử lý dữ liệu, lựa chọn phương pháp, và thực hiện các thử nghiệm, nhóm đã tìm ra kết quả tốt nhất. Mục tiêu là sáng tạo được một bản nhạc mới từ bản nhạc gốc, tuy vẫn có những điểm giống bản gốc nhưng bản mới sẽ có những điểm sáng tạo hơn.

Nhóm đã áp dụng cho 2 mô hình GRU và LSTM, để đánh giá hiệu suất, chúng tôi sử dụng độ đo Accuracy, Precision và Recall. Kết quả tốt nhất mà nhóm đạt được khi sử dụng mô hình LSTM với batch-size = 128.

Tuy nhiên, hệ thống của nhóm vẫn còn một số hạn chế, bao gồm kết quả dự đoán chưa đạt được mức cao mong muốn và hiệu suất vận hành chưa đạt được độ ổn định mong đợi.

Hướng phát triển trong tương lai:

- Áp dụng và tinh chỉnh các mô hình pre-trained cho bộ dữ liệu
- Xây dựng một trang web tạo ra âm nhạc từ dữ liệu người dùng tải lên
- Xây dựng được một bản nhạc hoàn chỉnh, có tiết tấu phù hợp và có sự kết hợp hài hòa giữa các hợp âm.

Tài liệu

[1] Jean-Pierre Briot, Gaetan Hadjeres, and Francois Pachet. Deep Learning Techniques for Music Generation. Computational Synthesis and Creative Systems. Springer Nature, 2018.

[2] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. ACM Computing Surveys (CSUR), 50(5), September 2017.

[3] George Papadopoulos and Geraint Wiggins. AI methods for algorithmic composition: A survey, a critical view and future prospects. In AISB 1999 Symposium on Musical Creativity, pages 110 117, April 1999.

[4] Jose David Fernandez and Francisco Vico. AI methods in algorithmic composition: A comprehensive survey. Journal of Arti cial Intelligence Research (JAIR), (48):513582, 2013.

[5] David Cope. The Algorithmic Composer. A-R Editions, 2000

[6] Gerhard Nierhaus. Algorithmic Composition: Paradigms of Automated Music Generation. Springer Nature, 2009.

[7] Alex Graves. Generating sequences with recurrent neural networks, June 2014. arXiv:1308.0850v5.

[8] Rebecca Fiebrink and Baptiste Caramiaux. The machine learning algorithm as creative musical tool, November 2016. arXiv:1611.00379v1.

[9] D. Eck and J. Schmidhuber, Finding temporal structure in music: Blues improvisation with LSTM recurrent networks, in Proc. 12th IEEE Workshop Neural Netw. Signal Process., Sep. 2002, pp. 747756, doi: 10.1109/NNSP.2002.1030094.

[10] D. Eck and J. Lapalme, Learning musical structure directly from sequences of music, Dept. Comput. Sci., CP, Univ. Montreal, Montreal, QC, Canada, 2008.

[11] Hua, Yuxiu, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. "Deep learning with long short-term memory for time series prediction."IEEE Communications Magazine 57, no. 6 (2019): 114-119.

[12] Husken, Michael, and Peter Stagge. "Recurrent neural networks for time series classification."Neurocomputing 50 (2003): 223-235

[13] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". December 1997Neural Computation 9(8):1735-80 , doi:10.1162/neco.1997.9.8.1735

[14] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling."arXiv preprint arXiv:1412.3555 (2014).