УДК 519.6

ГЕНЕРУВАННЯ МУЗИКИ ЗА ДОПОМОГОЮ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Ю. Калічун, Н. Колос

Львівський національний університет імені Івана Франка,
вул. Університеська, 1, Львів, 79000,
e-mail: yurii.kalichun@lnu.edu.ua nadiya.kolos@lnu.edu.ua

Розглянуто і проаналізовано передові методи генерування музики за допомогою штучних нейронних мереж. Виявлено переваги і недоліки цих методів і на підставі цієї інформації спроєктовано і натреновано штучний інтелект, здатний створювати музичні композиції. Під час дослідження варіантів вхідного і вихідного формату для нейронної мережі було визначено, що формат MIDI з нотами для фортепіано підходить для генерування музики, адже для нього вже створено багато бібліотек для обробки і він достатньо простий. На підставі цього формату і композицій з набору даних MAESTRO було побудовано алгоритм попередньої обробки музичних творів і створено тренувальний набір для нейронної мережі. Було спроєктовано модель штучного інтелекту на основі GRU та Self-Attention шарів. В статті описано результат роботи цієї моделі і проаналізовано Self-Attention шар після тренування. Також було зазначено можливі напрями покращення генерованих композицій і їх використання.

Ключові слова: генерування музики, генеративні змагальні мережі, навчання без вчителя, GRU, Self-Attention, MIDI.

1. Вступ

Однією із важливих задач штучного інтелекту є задача генерування різних даних (творчості, мови, рішень тощо). Зокрема, цікавою є задача генерування даних за допомогою навчання без вчителя. Наприклад, у сфері комп'ютерного зору є багато розробників, які досліджують передові методи створення зображень за допомогою Генеративних Змагальних Мереж (GAN). NVIDIA створює реалістичний генератор облич за допомогою GAN. Є також деякі дослідження щодо створення музики за допомогою GAN.

Якщо говорити про цінність генератора музики, то він може бути корисний музикантам як додаткове джерело натхнення у створенні власної музики. Такий продукт може підвищити креативність і продуктивність людей.

2. Короткий огляд відомих проектів

2.1. Magenta – 2016

Magenta – це проєкт, розроблений Google Brain. Він має на меті створити новий інструмент для виконавців, який буде використовуватись під час роботи над новими композиціями. В рамках цього проєкту Google Brain вже розробили декілька моделей для створення музики.

Наприкінці 2016 року вони опублікували модель LSTM, натреновану за допомогою навчання з підкріпленням. За допомогою цього методу вони мали на меті навчити модель дотримуватися певних правил, водночас даючи їй змогу запам'ятовувати певні закономірності в потоці даних.

Для цього команда Google визначила кілька метрик і визначила для них цільові значення. Чим більше відхилення від референтних значень матиме нейронна – тим більший штраф вона отримає. Отже, зменшуючи штраф за допомогою градієнтного спуску, ШНМ поліпшуватиме метрики, які їй визначили.

Ось ці метрики:

Ноти відсутні в ключі: чим більше таких нот, тим гірше звучить композиція

Середня автокореляція: метою є заохочення різноманітності, тому модель отримує штраф, якщо композиція сильно корелює з собою;

Ноти занадто часто повторюються: LSTM схильні повторювати ті ж самі моделі, тому за це дається штраф для досягнення більш креативного підходу.

Також за певні метрики, які розробники хочуть підвищити, дають нагороди ШНМ:

- композиція починається з тональної ноти;
- композиція має унікальну найвищу ноту або унікальну найнижчу ноту;
- композиція містить музичний "мотив": моделі винагороджуються за відтворення послідовності нот, які разом створюють коротку музичну "ідею".

Вибір метрики, а також ваги штрафу визначають форму музики, яка буде на виході.

Також зовсім недавно команда Magenta використала GAN разом з Transformers для створення музики з поліпшеною довгостроковою структурою.

У моделі Transformers використовується відносна самоувага(self-attention). Це модулює увагу, залежно від того, наскільки віддалені один від одного лексеми. Ця архітектура допомагає охопити різні рівні, на яких існують самореферентні явища в музиці.

2.2. MuseGAN – 2017

У цьому проєкті, щоб впоратися з групуванням нот, замість нот використовуються смуги як основна композиційна одиниця. Отже, музика генерується по смужках, використовуючи CNN, які добре підходять для пошуку локальних, інваріантних до перекладу шаблонів.

Цікавим підходом у цій роботі є використані показники оцінки. Автори проєкту хотіли досягти п'ять основних характеристик в музиці. Ці характеристики використовуються для навчання мережі і по них автори оцінювали, наскільки вдалий прогноз. Ось ці п'ять характеристик:

- співвідношення порожніх смуг;
- кількість використаних класів висоти тону в партії (від 0 до 12);
- співвідношення "кваліфікованих нот". Тут нота, довша за три часові кроки вважається кваліфікованою. Кваліфіковані ноти визначають, чи музика не надто фрагментована;
- шаблон барабана: співвідношення нот у 8 або 16 тактів;
- тональна відстань: вимірює гармонійність між парою доріжок. Більша тональна відстань передбачає слабкі міжколійні гармонійні залежності.

2.3. Wavenet – 2016

Це приклад проєкту, який використовує неперервне подання музики замість дискретного. Модель генерує необроблені звукові коливання, тому вона здатна генерувати будь-який звук, навіть голоси людей.

Модель побудована на базі CNN, де згорткові шари мають кілька факторів дилатації, і прогнози залежать лише від попередніх результатів. Цей проєкт застосовується для генерування творів на фортепіано та людського голосу.

2.4. MuseNet – 2019

Це модель генерування музики від OpenAI. Вона використовує найсучаснішу архітектуру NLP – масштабну модель трансформатора для прогнозування наступного токена в послідовності. Вона може поєднувати стилі різних відомих композиторів, а також різних музичних жанрів.

3. Проєктування

Візуалізація процесу тренування нейронної мережі зображена на рис. 1.

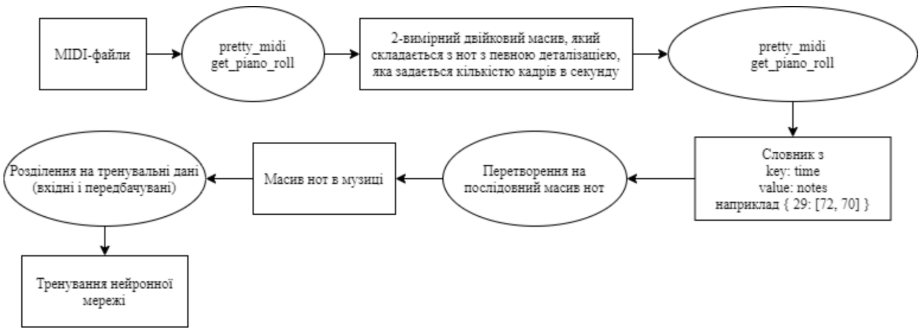

Рис. 1. Тренування нейронної мережі

Після тренування, використовуючи доволі схожі дії у зворотному порядку, можна отримати файли MIDI зі згенерованою музикою, як показано на рис. 2.

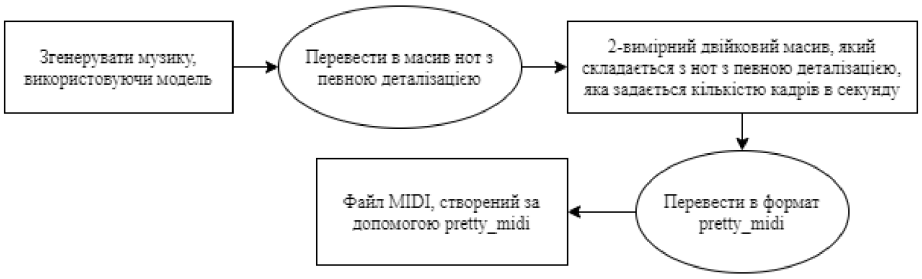

Рис. 2. Дії після тренування мережі

Для кращого розуміння процес проєктування був розділений на три основні підпроцеси:

– перетворення файлів MIDI в зручний для тренування формат;
– тренування моделі;
– отримання файлів MIDI з результатів роботи нейронної мережі.

4. Реалізація

4.1. Використані технології, методики та дані

Для створення ШНМ було використано такі технології:
- Tensorflow v 2.0 – фреймворк для створення і тренування моделей нейронних мереж;
- Python 3.7;
- Colaboratory – безплатне хмарне середовище з Jupyter notebook і відеокартами Nvidia Tesla;
- pretty_midi.py – бібліотека для створення і редагування MIDI-файлів.

Для досягнення цілі буде використано бібліотеку Tensorflow v2.0 як основу для створення і навчання нейронної мережі. Вона була обрана тому, що однією з особливостей Tensorflow v 2 є можливість прискорення навчання моделі, використовуючи AutoGraph.

Цей проєкт також використовує Self-Attention Layer. Для визначеної послідовності шар "самоуваги" повідомляє нам, наскільки сильно кожен елемент послідовності впливає на кожен інший.

Для тренувальних даних було використано композиції на фортепіано з набору даних MAESTRO (MIDI and Audio Edited for Synchronous Tracks and Organization) від Magenta.

4.2. Попереднє опрацювання файлів MIDI

Насамперед варто зауважити, що MIDI-файли містять музичні інструменти, в яких є ноти. Наприклад, поєднання фортепіано та гітари. Кожен з музичних інструментів зазвичай має різні ноти для гри. Для наших цілей потрібні лише партитури для фортепіано.

Для попередньої обробки файлів MIDI існує декілька Python-бібліотек. Одна з них це pretty_midi. Вона має функціонал для створення і маніпулювання файлами MIDI, тому її обрали для використання.

Ця бібліотека перетворює файли MIDI в масив об'єктів у форматі, який показано на рис. 3.

```
[Note(start=0.965625, end=1.104167, pitch=77, velocity=57),
 Note(start=1.088542, end=1.178125, pitch=73, velocity=62),
 Note(start=1.193750, end=1.307292, pitch=68, velocity=73),
```

Рис. 3. Масив об'єктів

Start це момент початку відтворення ноти, *End* - кінець. Зауважимо, що кілька нот можуть відтворюватись в один момент часу. *Pitch* це MIDI-номер відтворюваної ноти. *Velocity* це сила звуку, з якою відтворена ця нота. Для кращого розуміння – що більша кількість, то сильніше піаністу потрібно вдарити по клавіші.

Для тренувального набору нейронної мережі з файлу MIDI було отримано музичні ноти для фортепіано. Далі, час в композиції було дискретизовано з певною величиною кадрів в секунду(FPS). Бібліотека pretty_midi містить функцію get_piano_roll, яка обробляє файл MIDI і повертає ноти у двійковому двовимірному масиві розміру *(notes_cnt, time)*. Тут рядок позначає номер ноти, а стовпчик –

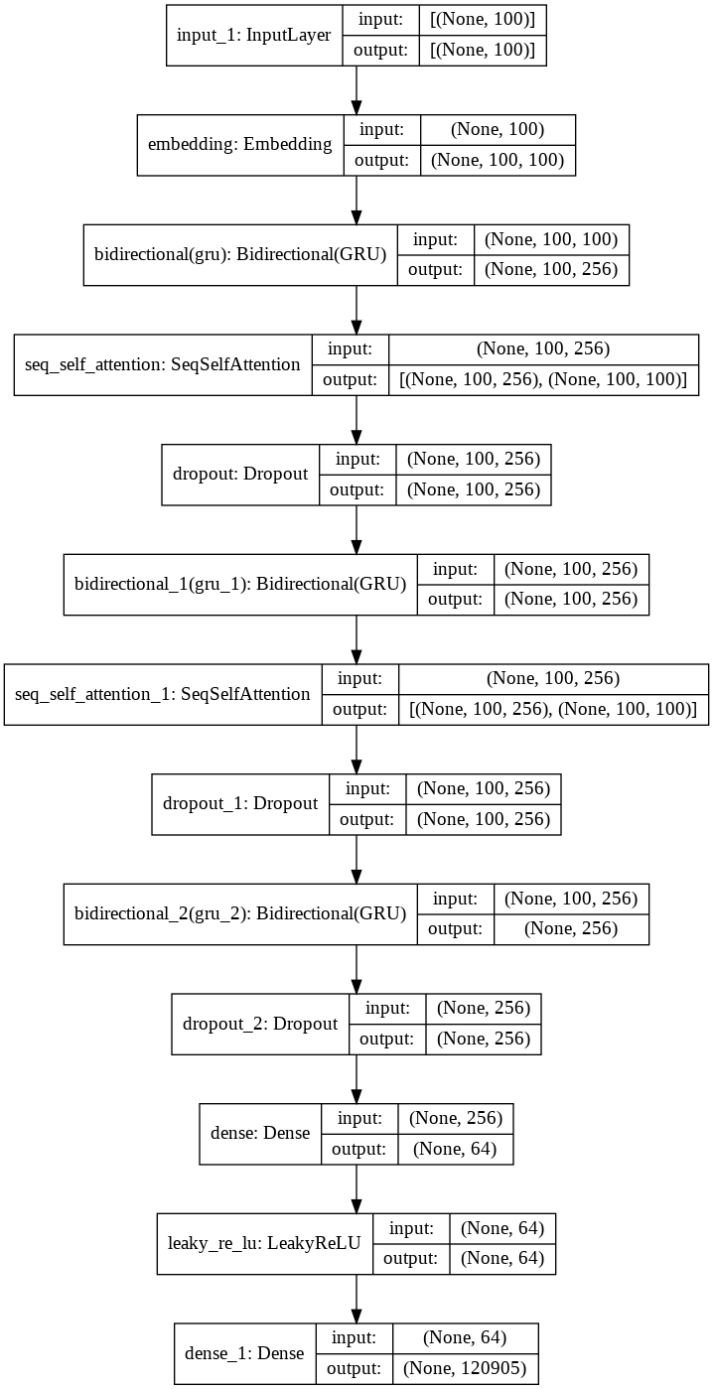

Рис. 4. Архітектура моделі

момент у фрагментованому часі з вибраним нами FPS. Тобто, якщо елемент на позиції *[i, j]* містить одиницю, то це означає, що в момент часу i відтворюється нота j.

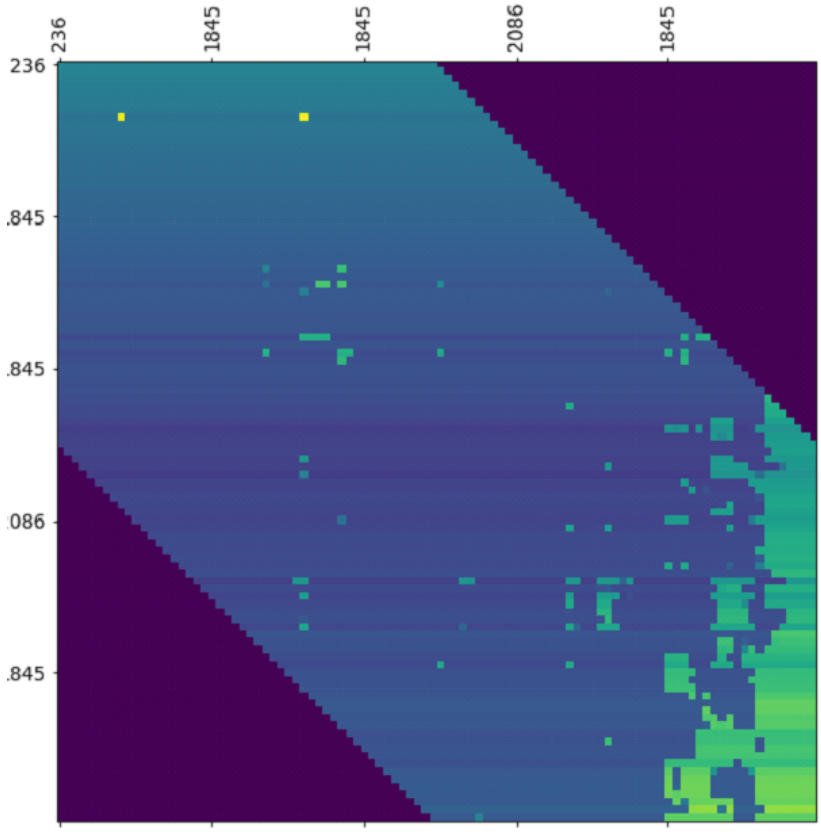

Рис. 5. Візуалізація шару

Після того, як отримано масив нот фортепіано, їх було перетворено у словник. Ключем словника є момент у часі.

Отримавши словник, перетворимо його на масив наборів нот, які будуть використовуватися для тренування нейронної мережі. Даючи нейронній мережі початкову послідовність, вона буде повертати набір нот, який відтворюватиметься у наступний момент часу.

Щоб довести дані до формату, який зможе отримувати нейронна мережа, нам потрібно токенізувати усі набори нот, тобто дати кожному набору певний індекс. Для цього було створено NoteTokenizer. Він конвертує послідовності нот у послідовності чисел, які більш прийнятні для тренування нейронної мережі.

4.3. Архітектура моделі

Архітектура нейронної мережі складається з 3 шарів Gated Recurrent Unit (GRU, вид рекурентної нейронної мережі) та кількох рівнів уваги. Прошарок Dropout

використовується для того, щоб не перетренувати нейронну мережу. Детальніша архітектура моделі зображена на рис. 4.

4.4. Тренування моделі

Для оновлення ваг нейронної мережі використовується GradientTape. Під час навчання спочатку обчислюється помилка(loss), а після цього застосовується метод зворотного поширення помилки, і оновлення ваг за допомогою функції apply_gradients.

Тренування однієї епохи займає від 20 хвилин до 1 години залежно від того, наскільки великий набір даних обирати для тренування і за якого значення кількості кадрів в секунду його обробляти. Для досягнення цілей було взято 4 кадри в секунду, 100 композицій і натреновано 14 епох. Загалом тренування зайняло близько 4 годин на відеокарті Nvidia Tesla K80. Може видатись, що це доволі швидко і кожен може зробити те ж саме в себе на ноутбуці, але на звичайній відеокарті ноутбука процес навчання міг би зайняти до 10 діб.

5. Результат

З отриманими мелодіями та кодом програми можна ознайомитись в [1].

Отримані на виході композиції більше схожі до структурованої музики, ніж до випадкового набору нот і часто в композиціях попадаються мотиви, які свідчать про те, що нейронна мережа все ж здатна генерувати музику.

Також варто звернути увагу на перший шар уваги (self-attention layer), який демонструє важливість одних нот щодо інших. Візуалізація цього шару зображена на рис. 5.

Зауважимо, що перший шар уваги визначає, на яку ноту треба зосередити увагу для кожної ноти в екземплярах послідовностей. Досить легко помітити, що для кожної ноти шар уваги зосереджується на певному проміжку, не надаючи важливості нотам поза ним. До того ж видно, що для певних нот є одинарні варіанти нот, які часто трапляються після визначеної і якщо йти по цій послідовності нот, то скоріш за все буде отримана коротка милозвучна мелодія.

6. Висновок

У межах цього дослідження було згенеровано ноти для фортепіано нейронною мережею. Наразі згенеровані композиції важко назвати шедевром у музиці, але часто в них трапляються чудові поєднання звуків і стилів, наприклад, перехід зі стилю класичної музики XVIII-го століття в джаз. Позаяк для цієї роботи використовувався набір даних MAESTRO, то важко вловити, чий саме стиль наслідує ця модель, бо вона поєднує все, чому навчилась серед різних композицій різних виконавців. Для поліпшення результату можна використовувати набір даних з творами лише одного музиканта і, можливо, отримати “нову композицію Шопена”.

Також вибраний препроцесинг MIDI визначає обсяг та обмеження створеної моделі. Вибір дискретного перетворення з файлу MIDI призводить до неминучої втрати інформації з вихідного неперервного аудіофайлу.

Альтернатива – працювати безпосередньо зі звуковими хвилями. Це також дасть змогу передавати тембр звуку на виході, зберігаючи його висоту.

Список використаної літератури

1. Результати роботи нейронної мережі та код [Електронний ресурс] Режим доступу: https://drive.google.com/drive/folders/1AWLkrvFy8EUL9u-vAfcPyug_gd-cmmLP?usp = sharing
2. Tensorflow 2.0 Essential Documentation [Електронний ресурс] // Tensorflow. – 2021. – Режим доступу: https://www.tensorflow.org/guide
3. Karim R. Illustrated: Self-Attention [Електронний ресурс] / Raimi Karim // Towards Data Science. – 2019. – Режим доступу: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a
4. The MAESTRO Dataset [Електронний ресурс] // Tensorflow. – 2018. – Режим доступу: https://magenta.tensorflow.org/datasets/maestro
5. Magenta [Електронний ресурс] // Tensorflow. – Режим доступу: https://magenta.tensorflow.org/
6. Dong Hao-Wen MuseGAN [Електронний ресурс] / Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, Yi-Hsuan Yang. – Режим доступу: https://salu133445.github.io/musegan/
7. Aaron van den Oord. WaveNet: A generative model for raw audio [Електронний ресурс] / Aaron van den Oord, Sander Dieleman // DeepMind. – 2016. – Режим доступу: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio
8. MuseNet [Електронний ресурс] // OpenAI. – 2019. – Режим доступу: https://openai.com/blog/musenet/
9. McDonald Kyle Neural Nets for Generating Music [Електронний ресурс] / Kyle McDonald // Medium. – 2017. – Режим доступу: https://medium.com/artists-and-machine-intelligence/neural-nets-for-generating-music-f46dffac21c0
10. Sigurur Skúli How to Generate Music using a LSTM Neural Network in Keras [Електронний ресурс] / Skúli Sigurur // Towards Data Science. – 2017. – Режим доступу: https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5

Стаття: надійшла до редколегії 20.09.2021
доопрацьована 10.11.2021
прийнята до друку 24.11.2021

MUSIC GENERATION THROUGH NEURAL NETWORKS

Y. Kalichun, N. Kolos

Ivan Franko National University of Lviv,
Universytetska str., 1, Lviv, 79000,
e-mail: yurii.kalichun@lnu.edu.ua nadiya.kolos@lnu.edu.ua

The paper considers and analyzes advanced methods of music generation using generative neural networks. A neural network was created and trained based on the advantages and disadvantages of these methods. During the study of input and output format options for the neural network, it was determined that the MIDI format with notes for piano is suitable for generating music. It was chosen for this study because there are already plenty of libraries for processing it and it is pretty simple. Based on this format and musical compositions from the MAESTRO data set, an algorithm for pre-processing musical works was built and training set for the neural network was created. A model of artificial intelligence based on GRU and Self-Attention layers was designed. The article describes the result of this model and analyzes the Self-Attention layer after training. Possible ways to improve the generated compositions and their use were also indicated.

Key words: music generation, Generative adversarial networks, unsupervised learning, GRU, Self-Attention, MIDI.