

DOI 10.36074/logos-20.09.2024.028

КЛАСИФІКАЦІЯ МУЗИЧНИХ КОМПОЗИЦІЙ ЗА ДОПОМОГОЮ МЕРЕЖ CNN

Колос Надія Мирославівна¹

1. канд. фіз.-мат. наук,

доцент кафедри дискретного аналізу та інтелектуальних систем

Львівський національний університет ім. І. Франка, УКРАЇНА

доцент кафедри управління інформаційною безпекою

Львівський державний університет безпеки життєдіяльності, УКРАЇНА

ORCID ID: 0000-0001-9710-9667

ВСТУП

У сучасному світі обробка та аналіз музики стали важливими аспектами різних стрибків в технологіях. Одним із найперспективніших напрямків є використання штучних нейронних мереж (ШНМ) для розпізнавання та класифікації музичних треків. ШНМ здатні автоматично виділяти ознаки з музичних даних, що дозволяє отримати вищі результати в порівнянні з традиційними методами.

Історія використання ШНМ у музичній індустрії почалася на початку 21 століття. Перші спроби класифікації музичних треків були здійснені з використанням простих алгоритмів. Однак з розвитком технологій та зростанням потужностей комп'ютерів, ШНМ стали доступними для більш складних завдань.

ШНМ працюють за принципом навчання на основі зразків, де музичний трек перетворюється на вектори ознак, які потім обробляються вхідними нейронами мережі. На основі тренування мережа навчається класифікувати нові треки, аналізуючи їх за різними критеріями, такими як темп, тональність та ритм [1].

ШНМ активно використовуються в різних сферах музичної індустрії. Одним з популярних застосувань є також створення рекомендаційних систем, які пропонують користувачам нову музику на основі їх попередніх вподобань [2]. Інші можливі застосування включають автоматичну класифікацію музичних жанрів, аналіз настроїв треків та навіть створення нової музики на основі заданих параметрів.

SECTION 12.

TECHNOLOGIES ET SYSTÈMES D'INFORMATION

Попри численні досягнення, існують деякі виклики, з якими стикаються дослідники та розробники. Наприклад, якість даних, на основі яких тренуються моделі, може суттєво вплинути на результати розпізнавання.

МЕТОДИ ТА АЛГОРИТМИ

Серед найбільш популярних методів розпізнавання та класифікації музичних треків можна виділити:

1. Згорткові нейронні мережі (CNN): Вони ефективно використовуються для обробки звуків завдяки своїй здатності виявляти просторові ознаки у даних.

2. Рекурентні нейронні мережі (RNN): Ці мережі особливо корисні для обробки послідовностей даних, таких як музичні треки, які містять часову залежність [3].

3. Автоенкодера: Використовуються для зменшення розмірності даних та виділення ключових ознак для подальшого аналізу [4].

Є досить багато робіт (наприклад, [5] та [6]), які описують застосування CNN для класифікації музичних треків за жанрами, використовуючи спектрограми як вхідні дані. Ці роботи детально описують методи і підходи для обробки аудіо даних, таких як спектрограми, за допомогою згорткових нейронних мереж, і можуть бути корисними для розробки власних моделей класифікації музики.

У даній роботі описано класифікацію музичних композицій за виконавцями з використанням згорткової нейронної мережі та спектрограми. Реалізацію зроблено у середовищі MATLAB.

СЕРЕДОВИЩЕ MATLAB

Реалізація, що наведена нище, базується на загальних знаннях і практиках роботи з MATLAB та нейронними мережами. MATLAB Documentation: Neural Network Toolbox (тепер відомий як Deep Learning Toolbox) [7] надає інформацію про функції та інструменти для роботи з нейронними мережами.

Книга [8] надає детальну інформацію про створення та навчання нейронних мереж в MATLAB. Онлайн-курси та ресурси: MathWorks Academia [9] та MATLAB Central [10] містять безкоштовні туторіали та приклади для роботи з нейронними мережами в MATLAB. Курс [11] надає глибоке розуміння нейронних мереж, яке може бути корисним для розробки CNN. Різні наукові статті про застосування CNN для обробки аудіо та спектрограм, такі як наприклад, [12] забезпечують основи та деталі для створення та тренування нейронних мереж, а також для обробки аудіо даних у MATLAB.

РЕАЛІЗАЦІЯ

1. Підготовка даних

–Збір даних:

У даному дослідженні використано 67 композицій з домашньої бібліотеки ліцензійних музичних дисків. Проте, також можна завантажити базу музичних треків з інтернету. Можна скорстатись одним з популярних джерел для навчання ШНМ, наприклад, Free Music Archive або GTZAN Genre Collection.

–Перетворюємо аудіо у спектрограми:

Аудіо потрібно конвертувати у формат, який можна подати згортковій мережі (наприклад, спектрограми чи мел-спектрограми), оскільки CNN працюють з двовимірними масивами даних. В MATLAB використаємо функцію `audioread` для зчитування аудіофайлів: `[y, Fs] = audioread('your_audio_file.wav')`. Перетворимо аудіо сигнали в спектрограми за допомогою функції `spectrogram` (або `mfcc` - Мел-частотні кепстральні коефіцієнти) з бібліотеки Audio Toolbox: `[s, f, t] = spectrogram(y, window, noverlap, nfft, Fs)`. Наприклад, початкова розмірність може бути (128, 128, 1) для спектрограм 128x128 пікселів, якщо використовувати лише одноканальну версію (моно).

–Створення набору даних.

–Розбиваємо дані на навчальну та тестову вибірки.

2. Створення нейронної мережі

Створимо архітектуру CNN за допомогою MATLAB Neural Network Toolbox. Це базова модель, яку можна покращити шляхом налаштування гіперпараметрів та додавання/зміни шарів.

1. Шар 1 - Згортковий шар (Conv2D):

Кількість фільтрів: 32

Розмір ядра: 3x3

Функція активації: ReLU

Шар нормалізації партій (Batch Normalization) для стабілізації навчання.

Шар підвибірки (Pooling): MaxPooling 2x2

```
convolution2dLayer(3, 32, 'Padding', 'same', 'Name', 'conv_1') %створює шар згортки з ядром 3x3,
% 32 фільтрами і 'same' padding, щоб зберегти розмір вхідних даних.
batchNormalizationLayer('Name', 'norm_1') %додає шар нормалізації для стабілізації навчання.
reluLayer('Name', 'relu_1') %застосовує функцію активації ReLU.
maxPooling2dLayer(2, 'Stride', 2, 'Name', 'maxpool_1') %додає max-pooling шар з ядром 2x2 і кроком 2.
```

Рис. 1. Код для створення шару 1 у середовищі MATLAB

2. Шар 2 - Згортковий шар (Conv2D):

Кількість фільтрів: 64

Розмір ядра: 3x3

Функція активації: ReLU



SECTION 12.

TECHNOLOGIES ET SYSTÈMES D'INFORMATION

Шар нормалізації партій
Шар підвибірки: MaxPooling 2x2
3. Шар 3 - Згортковий шар (Conv2D):
Кількість фільтрів: 128
Розмір ядра: 3x3
Функція активації: ReLU
Шар нормалізації партій
Шар підвибірки: MaxPooling 2x2
4. Шар 4 - Згортковий шар (Conv2D):
Кількість фільтрів: 256
Розмір ядра: 3x3
Функція активації: ReLU
Шар нормалізації партій
Шар підвибірки: MaxPooling 2x2
5. Шар 5 - Шар згладжування (Flatten): Перетворення 2D виходу на вектор.
6. Шар 6 - Повнозв'язний шар (Dense/FC):
Кількість нейронів: 256
Функція активації: ReLU
Шар Dropout (ймовірність виключення 0.5) для уникнення перенавчання.

```
flattenLayer('Name','flatten_1') %перетворює багатовимірний вихід на вектор.  
fullyConnectedLayer(256,Name="fc_1") %створює повнозв'язний шар із 256 нейронами.  
reluLayer('Name','relu_5') %додає функцію активації ReLU.  
dropoutLayer(0.5,'Name','drop_1') %шар Dropout із ймовірністю виключення 50%  
% для запобігання перенавчання.
```

Рис. 2. Код для створення шару 6 у середовищі MATLAB

7. Шар 7 - Повнозв'язний шар (Dense/FC):
Кількість нейронів: 128
Функція активації: ReLU
Шар Dropout (ймовірність виключення 0.5)
8. Вихідний шар (Output layer):
Кількість нейронів: 4 (оскільки маємо 4 класи виконавців)
Функція активації: Softmax

```
fullyConnectedLayer(4) % 4 класи  
softmaxLayer %функція активації, що перетворює виходи в ймовірності  
classificationLayer]; %шар для обчислення втрат під час навчання та для класифікації  
%результатів на основі ймовірностей
```

Рис. 3. Код для створення вихідного шару в середовищі MATLAB

3. Тренування моделі

Гіперпараметри для навчання (рис. 1):

```
options = trainingOptions('adam', ... % Оптимізатор Adam
    'InitialLearnRate', 0.001, ... % Початкова швидкість навчання
    'MiniBatchSize', 32, ... % Розмір партії (batch size)
    'MaxEpochs', 50, ... % Кількість епох
    'Shuffle', 'every-epoch', ... % Перемішування даних на кожній епосі
    'Plots', 'training-progress', ... % Графік прогресу навчання
    'Verbose', false, ... % Відключення додаткових повідомлень
    'ExecutionEnvironment', 'auto'); % Автоматичний вибір середовища
```

Рис. 4. Налаштування гіперпараметрів для навчання мережі

Кількість епох можна варіювати 50-100 залежно від розміру даних.

Тренуємо модель на навчальному наборі даних:

```
net = trainNetwork(trainData, layers, options);
```

4. Тестування та оцінка моделі

Оцінимо модель на тестовій вибірці:

```
predictions = classify(net, testData);
```

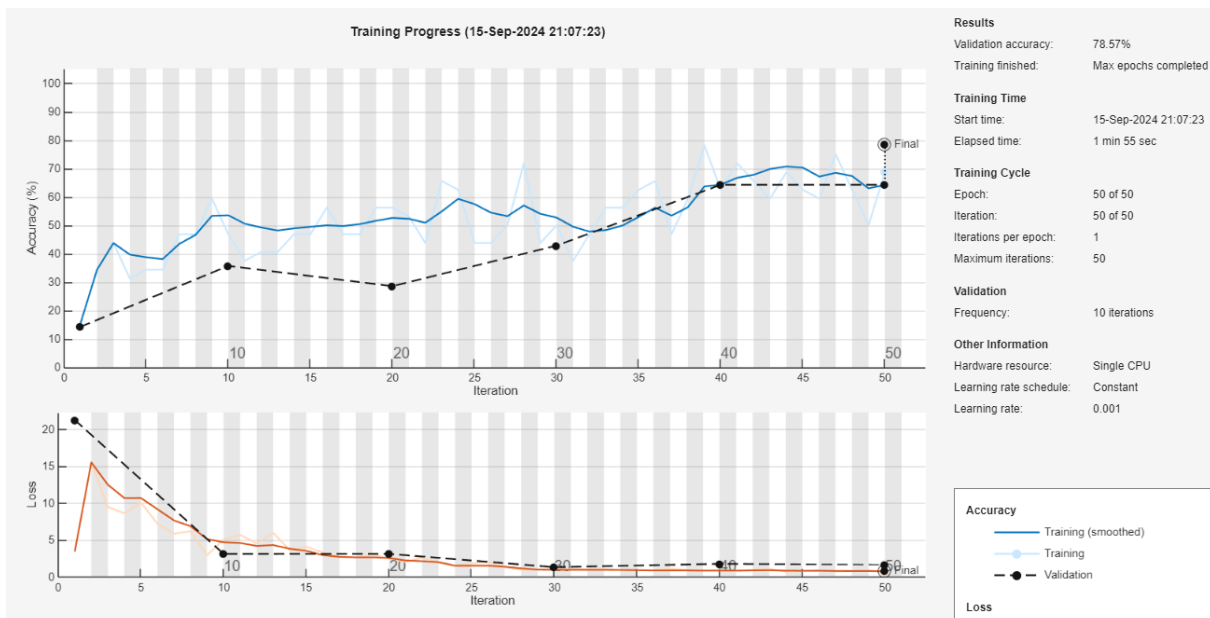


Рис. 5. Один із процесів навчання CNN. Графіки показують зростання точності (Accuracy) та спадання похибки (Loss) при навчанні. В даному випадку досягнуто точності 78,57%



SECTION 12.

TECHNOLOGIES ET SYSTÈMES D'INFORMATION

Створимо матрицю плутанини (confusion matrix) для оцінки якості класифікації:

```
actualLabels = testData.Labels;  
confMat = confusionmat(actualLabels, predictions);  
confusionChart = confusionchart(confMat,classNames);
```

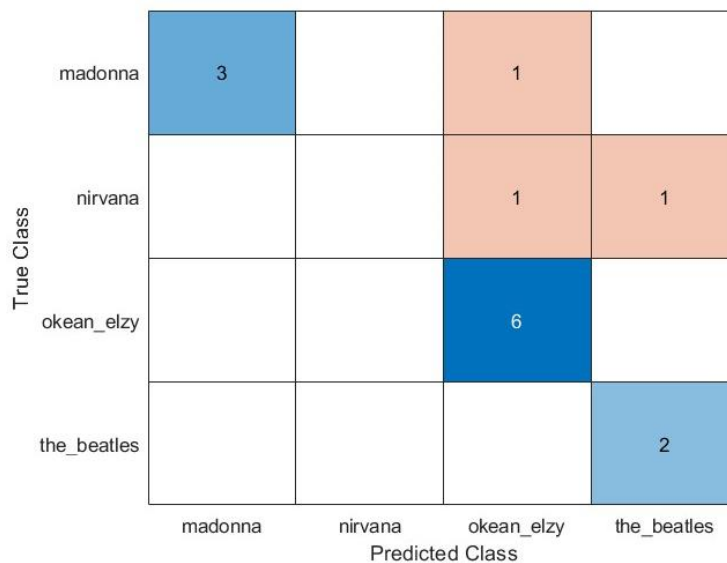


Рис. 6. Матриця плутанини, що показує якість класифікації на тестовій вибірці, що складалась із 14 музичних треків

ВИСНОВКИ

У результаті експериментів з різними архітектурами та гіперпараметрами моделі (кількість шарів, кількість фільтрів, функції активації та методи регуляризації), різні моделі давали точність у проміжку 60-80%. Крім того, на точність розпізнавання впливає ще кілька факторів:

– Кількість та якість даних. Якщо треків мало або вони містять шуми, це може суттєво знизити точність. Для того, щоб модель мала достатньо даних для навчання і, відповідно, підвищилась точність класифікації бажано збільшити кількість треків для кожного класу – мінімум до 100-200.

– Різноманітність виконавців. В даному дослідженні використано музичні твори досить різних за жанром та стилем виконавців. Проте, при розширенні моделі можуть траплятись виконавці, що мають схожі музичні елементи (наприклад, ритм, гармонія). Це ускладнить завдання для мережі.

– Попередня обробка. Обробка даних (нормалізація, видалення шумів, вирізання частин) має велике значення для результату.

З урахуванням цих факторів і за умови достатньої кількості треків у кожного виконавця, можна очікувати стабільну точність у межах 80-85% після навчання та оптимізації моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

- [1] M.Singh, S. K. Jha, B. Singh and B. Rajput (2021) Deep Learning Neural Networks for Music Information Retrieval, International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 2021, pp. 500-503, doi: 10.1109/ICCIKE51210.2021.9410732.
- [2] Velankar, M., Kulkarni, P. (2023). Music Recommendation Systems: Overview and Challenges. In: Biswas, A., Wennekes, E., Wieczorkowska, A., Laskar, R.H. (eds) Advances in Speech and Music Technology. Signals and Communication Technology. Springer, Cham., 51-69, https://doi.org/10.1007/978-3-031-18444-4_3.
- [3] Kakarla, C., Eshwarappa, V., Babu Saheer, L., Maktabdar Oghaz, M. (2022). Recurrent Neural Networks for Music Genre Classification. Artificial Intelligence XXXIX: 42nd SGAI International Conference on Artificial Intelligence, AI 2022, Cambridge, UK, December 13-15, 2022, Proceedings, (13652), 267-279. Springer, Cham. https://doi.org/10.1007/978-3-031-21441-7_19.
- [4] Jason Brownlee (2020) Autoencoder Feature Extraction for Classification. Вилучено з: <https://machinelearningmastery.com/autoencoder-for-classification/>.
- [5] Dhall, A., Srinivasa Murthy, Y.V., Koolagudi, S.G. (2021). Music Genre Classification with Convolutional Neural Networks and Comparison with F, Q, and Mel Spectrogram-Based Images. In: Biswas, A., Wennekes, E., Hong, TP., Wieczorkowska, A. (eds) Advances in Speech and Music Technology. Advances in Intelligent Systems and Computing, (1320), 235-248, Springer, Singapore. https://doi.org/10.1007/978-981-33-6881-1_20.
- [6] Bawitlung, A., Dash, S.K. (2024). Genre Classification in Music using Convolutional Neural Networks. In: Badioze Zaman, H., et al. Advances in Visual Informatics. IVIC 2023. Lecture Notes in Computer Science, (14322), 397-409. Springer, Singapore. https://doi.org/10.1007/978-981-99-7339-2_33.
- [7] Офіційна документація MATLAB: MATLAB Documentation: Neural Network Toolbox (тепер відомий як Deep Learning Toolbox). Онлайн-ресурс: <https://www.mathworks.com/help/deeplearning/>.
- [8] "Deep Learning with MATLAB" від MathWorks, Онлайн-ресурс: <https://www.mathworks.com/help/deeplearning/ug/deep-learning-in-matlab.html>.
- [9] MathWorks Онлайн-ресурс: Academia <https://www.mathworks.com/academia.html>.
- [10] MATLAB Central Онлайн-ресурс: <https://www.mathworks.com/matlabcentral/>.
- [11] Курс "Deep Learning Specialization" від Andrew Ng на Coursera <https://www.coursera.org/specializations/deep-learning?msoclid=1f2ff9c19ef76e0c1438ed6a9fd66f17>.
- [12] H.Purwins, B. Li, T. Virtanen, J. Schlüter, S. -Y. Chang and T. Sainath (2019) Deep Learning for Audio Signal Processing, IEEE Journal of Selected Topics in Signal Processing, vol. (13), no. 2, pp. 206-219, doi: 10.1109/JSTSP.2019.2908700.

