UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU
WYDZIAŁ MATEMATYKI I INFORMATYKI

Cezary Adam Pukownik

Kierunek: Analiza i przetwarzanie danych
Numer albumu: 444337

Generowanie muzyki przy pomocy głębokiego uczenia

Music generation with deep learning

Praca licencjacka
napisana pod kierunkiem
dr hab. Tomasza Góreckiego

POZNAŃ 2020

Poznań, dnia .....

OŚWIADCZENIE

Ja, niżej podpisany Cezary Pukownik, student Wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu oświadczam, że przedkładaną pracę dyplomową pt: "Generowanie muzyki przy pomocy głębokiego uczenia", napisałem samodzielnie. Oznacza to, że przy pisaniu pracy, poza niezbędnymi konsultacjami, nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opracowania rozprawy lub jej części innym osobom, ani nie odpisywałem tej rozprawy lub jej części od innych osób.

Oświadczam również, że egzemplarz pracy dyplomowej w wersji drukowanej jest całkowicie zgodny z egzemplarzem pracy dyplomowej w wersji elektronicznej.

Jednocześnie przyjmuję do wiadomości, że przypisanie sobie, w pracy dyplomowej, autorstwa istotnego fragmentu lub innych elementów cudzego utworu lub ustalenia naukowego stanowi podstawę stwierdzenia nieważności postępowania w sprawie nadania tytułu zawodowego.

$[TAK]^\star$ - wyrażam zgodę na udostępnianie mojej pracy w czytelni Archiwum UAM

$[TAK]^\star$ - wyrażam zgodę na udostępnianie mojej pracy w zakresie koniecznym do ochrony mojego prawa do autorstwa lub praw osób trzecich

*Należy wpisać TAK w przypadku wyrażenia zgody na udostępnianie pracy w czytelni Archiwum UAM, NIE w przypadku braku zgody. Niewypełnienie pola oznacza brak zgody na udostępnianie pracy.

.....

Spis treści

Streszczenie

Abstract

Wstęp

Uczenie maszynowe w ostatnich latach mocno zyskało na popularności. Zastosowania i możliwości różnych algorytmów Mashine Learning czasami przekraczają nasze wyobrażenie o tym co komputer może zrobić. Niektóre aplikacje potrafią wręcz zaskoczyć użytkowników tym co potrafią zrobić. Wśród takich aplikacji znajdują się takie, które potrafią przewidywać następne wartości akcji giełdowych, rozpoznawać na filmie obiekty w czasie rzeczywistym, czy nawet prowadzić samochód. Algorytmy wyuczone proponują nam spersonalizowane reklamy, czy produkty na podstawie naszych upodobań. Najczęstsze zastosowania dotyczą przetwarzania obrazów lub tekstu, natomiast zastosowania w przetwarzaniu muzyki są niszowe i rzadko spotykane.

Celem tej pracy jest stworzenie modelu sieci neurowej, którego zadaniem bedzie generowanie krótkich multiinstrumentalnych klipów muzycznych.

W pierwszej częsci swojej pracy przedstawię podstawowe koncepcje związane z muzyką oraz sposobami jej reprezentacji. Następnie opiszę w jaki sposób działają sieci neuronowe, jak się uczą oraz podstawowe architektury sieci, które pomogą zrozumieć model który wykorzystałem.

Następnie przedstawię koncepcję działania modelu, jakie idee stały za wyborami które podjąłem w projektowaniu sieci. W szczegółowy sposób opiszę sposób ekstrakcji danych, tak aby mogły być one wykorzystane przez model. Opiszę architekturę którą wybrałem oraz przedstawię i opisze fragmenty kodu w języku python.

W kolejnym rozdziale skupimy się na rezultatach pracy, przedstawie zalety i wady modelu. Przeprowadze analizę jakie muzyczne koncepcje model się nauczył na podstawie danych oraz doprowadzę do ostatecznej konkluzji czy wygenerowana muzyka może być przyjemna dla odbiorcy.

Reprezentacja muzyki

W tym rozdziale przedstawię podstawowe koncepcje muzyczne, sposoby reprezentacji muzyki oraz omówie podstawy działania protokołu MIDI.

1.1. Podstawowe koncepcje

1.1.1. Dźwięk muzyczny

Drgania powietrza z otoczenia człowieka są przetwarzane w mózgu i rozumiane jako dźwięki. Takie drgania nazywamy falą dźwiękową. Dzwięk muzyczny jest to fala dźwiękowa, którą wytwarza intrument muzyczny. Dzwięk muzyczny charakteryzuje się trzema podstawowymi parametrami:

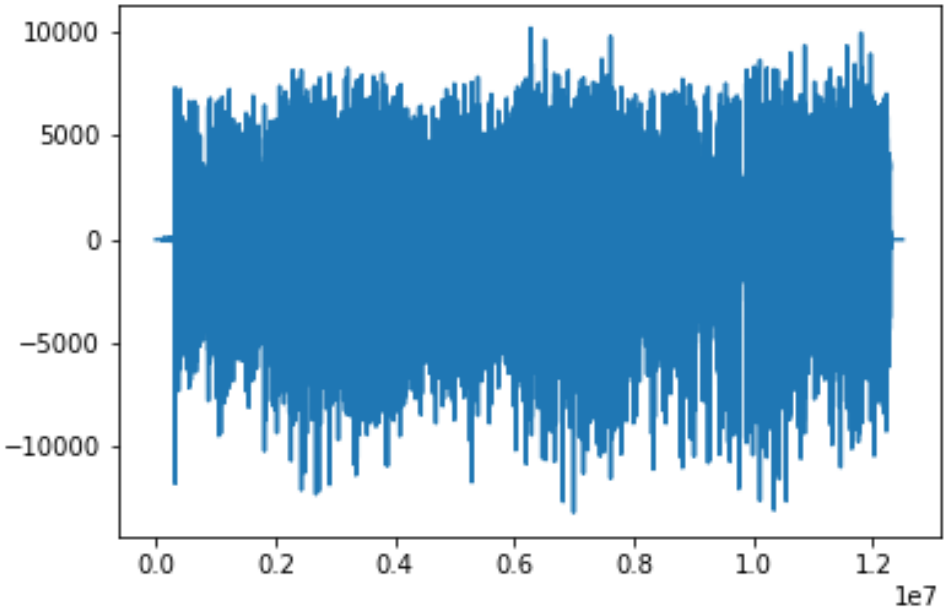
— wysokość (ang. pitch) - jest to częstotliwość drgań wyrażona w hercach. Im większa częstotliwość tym dźźięk jest rozumiany jako wyższy, Zakres słyszalny dla człowieka wynosi od 20Hz do 20kHz.
— głośność (ang. velocity) - jest to amplituda drgań fali dźwiękowej. Im większa aplituda, tym dźwięk jest odczuwany jako głosniejszy,
— długość (ang. duration) - jest to czas z jakim dźwięk wybrzmiewa, np. 2 sekundy.

1.1.2. Sygnał dźwiękowy

W rzeczywistości, utwór muzyczny jest zazwyczaj kombinacją wielu fal dźwiękowych, o różnych charaketystykach i nazwywany jest sygnałem dźwiękowym. Wizualizację sygnalu dźwiękowego przedstawiono na Rysunku 1.2

1.1.3. Zapis nutowy

Reprezentacja muzyki jako sygnału dźwiękowego przechowuje informacje o dokładnym brzmieniu danego utworu tzn. jakie drgania należy wytworzyć, aby móc odtwozyć dźwięki. Taki zapis nie informuje nas bezpośrednio jakie instrumenty zostały użyte, jakie wysokości i długości dźwięków zostały wykorzystane. Dlatego ludzkość na przestrzeni wieków opracowała abstrakcyjne objekty, które reprezentują utwór w czytelny dla człowieka sposób.



Rysunek 1.1. Przykład przebiegu fali dźwiękowej

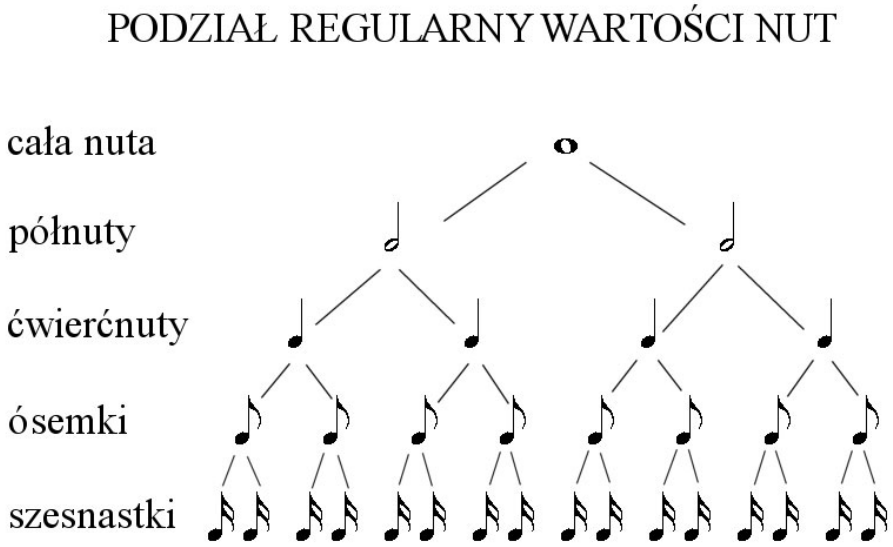
Tempo

W muzyce symbolicznej tempo informuje nas o prędkości utworu. W muzyce klasycznej stosowało się opisowy sposób dostosowywania tempa np. Allegro - Szybko lub Adagio - wolno. Jak można szybko stwierdzić są to zwroty subiektywne i nie wyznaczają tempa jednoznacznie. Obecnie wyraża się tempo w liczbie uderzeń na minutę (BPM ang. beats per minute). I tak Allegro jest to od 120 do 168 BPM a Adagio od 66 do 76 BPM. [1]

Nuta

Nuta jest to graficzna reprezentacja dźwięku muzycznego. Informuje nas ona o dwóch parametrach dźwięku, wysokości oraz długości dźwięku. Długość dźwięku nazywa się jej wartością. Podstawową wartością nuty jest ćwierćnuta, odpowiada ona jednemu uderzeniu. Dla przykładu w tempie 60 BPM w ciągu minuty zagramy dokładnie 60 ćwierć nut. Kolejne wartości tworzone poprzez sumowanie lub podział długości ćwierćnuty. Półnuta trwa tyle co dwie ćwierćnuty, cała nuta tyle co dwie półnuty, ósemka trwa połowę czasu ćwiercnuty, a szesnastka połowę ósemki itd.

[1] źródlo: http://www.classicalmusiccity.com/search/article.php?vars=446/Basic-Tempo-Markings.html 5 kwietnia 19:37



Rysunek 1.2. źródlo: https://www.infomusic.pl/poradnik/46934,poradnik-teoria-muzyki-rytm 5 kwietnia 2020 12:46

Tak jak pisałem wcześniej, wysokość dźwięku jest to częstotliwość drgań fali dźwiękowej wyrażona w hercach. W muzyce symbolicznej dla uproszczenia wybrane częstotliwości zostały nazwane literami alfabetu C, D, E, F, G, A, H. Każdej literze przypisana jest częstotliwość zgodnie z Tabelą 1.1

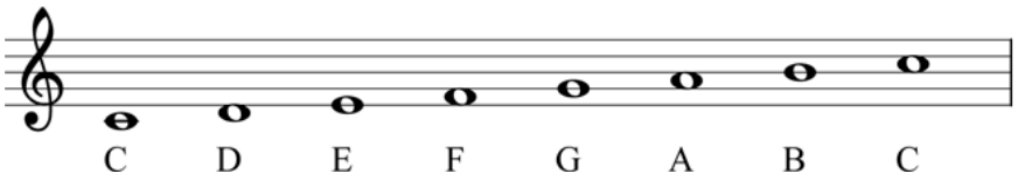
Dzwięk	Częstotliwość
C_4	261,6
D_4	293,7
E_4	329,6
F_4	349,2
G_4	391,9
A_4	440,0
H_4	493,9

Tabela 1.1. Dźwięki symboliczne oraz ich częstosliwości

W zapisie nutowym aby nucie nadać wysokość, umieszcza się ją w odpowiednim miejscu na pięciolonii. Przedstawione powyżej dźwięki zapisalibyśmy w taki sposób jak przedstawiono na Rysunku 1.3

Oktawy

Oktawą nazywamy zestaw ośmiu nut od C do H. Podane w Tabeli 1.1 częstotliwości nut odpowiadają dźwiękom w oktawie czwartej. Dlatego w indeksie



Rysunek 1.3. źródło: https://amplitudaschool.weebly.com/lekcja-11.html 5 kwietnia 2020 13:24

dolnym nuty widnieje liczba 4. Aby utworzyć dźwięk, np. A_5 należy pomnożyć częstotliwość dźwięku A_4 razy dwa, natomiast aby utworzyć dźwięk A_3, należy tę częstotliwość podzielić przez dwa.

$$A_5 = 440Hz * 2 = 880Hz$$

$$A_3 = 440Hz/2 = 220Hz$$

W ten sposób możemy utworzyć nieskończenie wiele oktaw, jednak w rzeczywistości używa się nut od C0 do C8.

Akord

Gdy w jednym momencie zabrzmią dwie lub więcej różnych nut, wtedy mówimy o akordzie. Akord potrafi dodać emocje do brzmienia całego utworu.

1.2. Cyfrowa reprezentacja muzyki symbolicznej

1.2.1. Standard MIDI

Standrd MIDI (ang. Musical Instrument Digital Interface) został stworzony w 1983 aby umożliwić synchronizację i wymianę informacji między elektronicznymi urządzeniami muzycznymi takimi jak syntezatory, keyboardy czy sekwencery. W późniejszych latach został on zaadaptowany do środowiska komputerowego jako cyfrowa reprezentacja muzyki symbolicznej.

Wiadomości

Plik MIDI zawiera zestaw wiadomości przesyłanych w czasie rzeczywistym o każdej nucie w utworze. Dwie wiadomości, które są dla nas szczególnie istotne to:

— note_on, który sygnalizuje aby rozpocząć grać nutę,
— note_off, który sygnalizuje aby zakończyć grać nutę.

Dla przykładu wiadomość:

```
note_on channel=0 note=48 velocity=100 time=0
note_on channel=0 note=53 velocity=100 time=0
note_on channel=0 note=60 velocity=100 time=0
note_on channel=0 note=48 velocity=0 time=220
note_on channel=0 note=48 velocity=100 time=0
note_on channel=0 note=53 velocity=0 time=0
note_on channel=0 note=55 velocity=100 time=0
note_on channel=0 note=60 velocity=0 time=0
```

Rysunek 1.4. Fragment protokołu MIDI

note_on, channel 0, note 48, velocity 100, time 0.

oznacza aby na kanele 0 zagrać dźwięk nr 48 z głośnością 100 w momencie 0 utworu. Nie informuje nas on jednak o długości trwania dźwęku. Aby zakończyć dźwęk, należy wysłac wiadomość:

note_off, channel 0, note 48, velocity 100, time 24.

Zwróćmy uwagę że aby ustalić wartość nuty, potrzebujemy odebrać dwie wiadomości. Różnica między parametrami time, informuje nas o długości nuty. W tym przypadku jest to 24. Co oznacza ćwierćnutę.

Rozdzielczość

Czas w MIDI jest reprezentowany jako liczba naturalna i jest on zależny od ustalonego tempa utworu. Standardowa rozdzielczość pliku MIDI to 24. Oznacza to, że jedna jednostka czasu odpowiada jednej dwudziestejczwartej jednego udeżenia.

Kanały

Plik MIDI posiada 16 kanałów numerowanych od 0 do 15. Każdy kanał odpowiada intrumentowi lub ścieżce. Kanał 9 jest kanałem zarezerwowanym na intrumenty perkusyjne.

Nuty

Nuty w formacie MIDI opisane są kolejnymi cyframi naturalnymi w przedziale od 0 do 127. Odpowiada to dźwękom od C_0 do C_8. Dla przykładu nuta 69 odpowiada A_4, a nuta 47 odpowiada B_2.

Wyjątkiem są nuty z kanału dziewiątego, gdzie istnieją tylko nuty z zakresu od 35 do 81 i każda nuta odpowiada innemu elementowi perkusyjnemu np. 35 to stopa, a 37 to werbel.

Głośność

Za głośność dźwęku odpowiada parametr velocity, który jest liczbą z przedziału od 0 do 127. Im większa jest wartość tym głośniej wybrzmi dźwęk.

Program

Program w kontekscie standardu MIDI oznacza intrument który ma zagrać nuty. W standardzie GM (ang. General MIDI), jest 16 grup intrumenów a w każdej z nich znajduje się po 8 intrumentów. Są to piania, chromatyczne perkusje, organy, gitary, basy, intrumenty smyczkowe, zestawy intrumentów, intrumenty dmuchane blaszane, intrumenty dmuchane drewniane, flety, syntezatory prowadzące, syntezatory uzupełniające, efekty syntetyczne, instrumenty etniczne, perkusjonalia i efekty dzwiekowe.

Ścieżka

Ścieżka (ang. Track) grupuje nuty aby podzielić utwór muzyczny na różne intrumenty. Protokół MIDI pozwala aby grać wiele ścieżek dzwiękowych jednocześnie, wtedy mówimy o muzyce polifonicznej lub multiintrumentalnej.

Wprowadzenie do sieci neuronowych

W tym rozdziale opiszę podstawy działania sieci neuronowych.

2.0.1. Regresja liniowa

Podstawą wszystkich sieci neuronowych, jest regresja liniowa. W statystyce wykorzystywana aby wyjaśnić liniowe zalezności między zmiennymi.

Prosty model regresji liniowej dla jednej zmiennej można opisać wzorem.

$$\hat{y} = ax + b$$

gdzie,
— $\hat{y}$ jest to estymacja zmiennej objaśnianej,
— x jest to zmienna objaśniająca,
— a jest parametrem modelu,
— b jest wyrazem wolnym.

Zadaniem jest znalezienie takiego parametru $a \in \mathbb{R}$ oraz wyrazu wolnego $b \in \mathbb{R}$, aby dla znanych wartości $x \in \mathbb{R}$ estymacja zmiennej objasnianej $\hat{y} \in \mathbb{R}$ najlepiej opisywała zmienną objasnaną $y \in \mathbb{R}$

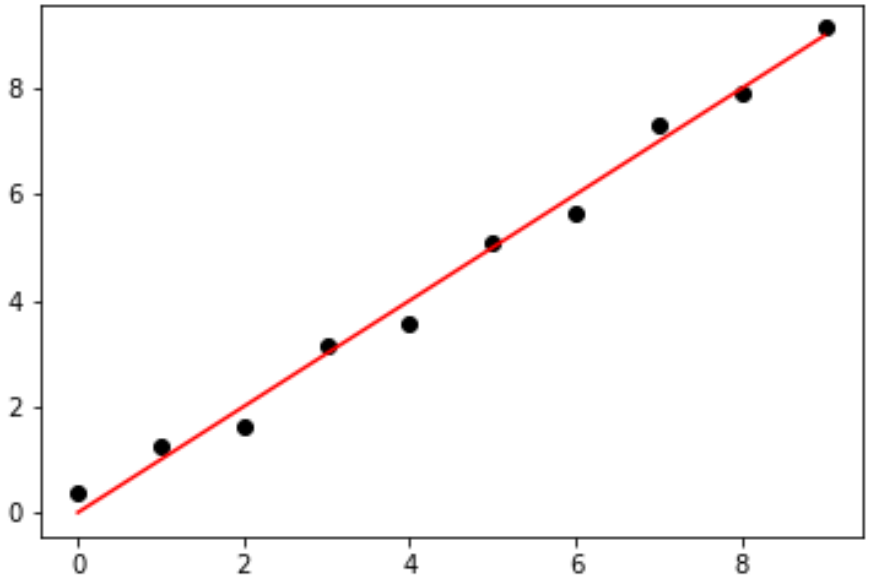
Wartość zmiennej objaśnianej y można również opisać za pomocą wielu zmiennych objaśniających. Wtedy dla zmiennych objaśniającej $x_1, x_2, ..., x_n \in \mathbb{R}$ szukamy parametrów $\theta_1, \theta_2, ..., \theta_n \in \mathbb{R}$. Otrzymany w ten sposób model nazywany jest również hipotezą i oznaczamy go $h(x)$.

$$h(x) = b + \theta_1 x_2 + \theta_2 x_2 + ... + \theta_n x_n = b + \sum_{i=1}^{n} \theta_i x_i$$

2.0.2. Uczenie modelu

Funkcja kosztu

Celem uczenia modelu jest znalecienie ogólnych parametrów, aby model dla każdej pary x, y zwracał wartości $\hat{y}$ najlepiej opisujące całe zjawisko według pewnego kryterium. W ten sposób jesteśmy wstanie znaleźć przybliżenie funkcji $h(x)$.



Rysunek 2.1. Regresja liniowa jednej zmiennej

W tym celu używa się funcji $J_\theta(h) \to \mathbb{R}$, która zwraca odległość między wartościami $h(x)$ oraz y dla wszystkich obserwacji. Taka fuckcja nazywana jest funkcją kosztu.

Dla przykładu regresji liniowej, funkcją kosztu może być odchylenie średnio kwadratowe. Wtedy funkcja kosztu przyjmuje postać:

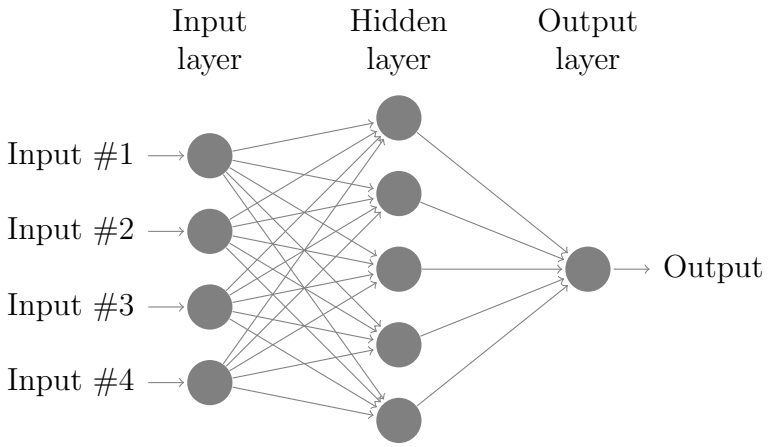
$$J_\theta(h) = \frac{1}{m}\sum_{i=1}^{m}(y_i - h(x_i))^2$$

gdzie $m \in \mathbb{N}$ jest liczbą obserwacji.

Przy zdefiniowanej funcji kosztu, proces uczenia sprowadza się do znalezienia takich parametrów funcji $h(x)$ aby funkcja kosztu była najmniejsza. Jest to problem optymalizacyjny sprowadzający się do znalezienia globalnego minimum funcji.

Metoda gradientu prostego

Jednym z algorytmów stosowanych do rozwiązania powyższego problemu optymalizacji jest metoda gradientu prostego (ang. gradient descent).



Rysunek 2.2. źródlo: Przykład sieci neuronowej

Rozdział 3

Projekt

W tym rozdzialę opiszę w jaki sposób zbudowałem swój własny geneator muzyki, jak przechodził procesz uczenia, jakie próbki udało mi się wygenrować. Opis kodu który napisałem.

Podsumowanie

Ostateczne wnioski, czy muzyka generowana komputerowa da się lubić? Czy to pozytywnie wpłynie na przemysł muzyczny? Tak i nie. Może złużyć jako inspiracja dla muzyków, proces wspierający. Z drugiej strony może obnizy koszty produkowania muzyki pop, która i tak jest już bardzo powtarzalna. Czy sieci neuronowe nauczą się produkować Hity?

Bibliografia

[1] Briot, J.P., Hadjeres, G., Pachet, F.D. (2019): *Deep Learning Techniques for Music Generation - A Survey. arXiv:1709.01620v3*

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016): *Deep Learning. MIT Press.*

[3] Zocca, V., Spacagna, G., Slater, D., Roelants, P. (2018): *Deep Learning. Uczenie głębokie z językiem Python. Helion.*