STRESZCZENIE

SŁOWA KLUCZOWE

sieci neuronowe, transkrypcja, detekcja częstotliwości

ABSTRACT

KEYWORDS

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

Spis Treści

[Cel i zakres pracy 3](#_Toc49714117)

[Przegląd rozwiązań z literatury 5](#_Toc49714118)

[Metody analizy sygnałów muzycznych 8](#_Toc49714119)

[Analizaczasowa 8](#_Toc49714120)

[Analiza częstotliwościowa 8](#_Toc49714121)

[Analiza STFT 11](#_Toc49714122)

[Współczynniki mel cepstralne 14](#_Toc49714123)

[Analiza autokorelacyjna 15](#_Toc49714124)

[Analiza falkowa 16](#_Toc49714125)

[Dźwięki w formie muzyki 18](#_Toc49714126)

[Dźwięki 19](#_Toc49714127)

[Zapis nutowy 22](#_Toc49714128)

[Notacja abc 24](#_Toc49714129)

[Interfejs MIDI 26](#_Toc49714130)

[Sieci Neuronowe 27](#_Toc49714131)

[Wstęp 27](#_Toc49714132)

[Omówienie zasady działania 29](#_Toc49714133)

[Neuron 29](#_Toc49714134)

[Aktywacje 30](#_Toc49714135)

[Parametry - wagi 30](#_Toc49714136)

[Parametry - bias 31](#_Toc49714137)

[Przegląd funkcji aktywacji 31](#_Toc49714138)

[Wnioskowanie i propagacja wsteczna 34](#_Toc49714139)

[Hiperparametry 36](#_Toc49714140)

[Ilość warstw ukrytych 36](#_Toc49714141)

[Stosowana funkcja aktywacji 37](#_Toc49714142)

[Dropout 37](#_Toc49714143)

[Sposób inicjalizacji wag 37](#_Toc49714144)

[Współczynnik uczenia 38](#_Toc49714145)

[Liczba epok i rozmiar batch 39](#_Toc49714146)

[Optimizer 40](#_Toc49714147)

[Funkcje Straty 40](#_Toc49714148)

[Funkcja straty dla regresji 41](#_Toc49714149)

[Funkcja straty dla klasyfikacji binarnej 41](#_Toc49714150)

[Funkcja straty dla klasyfikacji wielo-klasowej 41](#_Toc49714151)

[Metryki 42](#_Toc49714152)

[Dokładność 42](#_Toc49714153)

[Precyzja 43](#_Toc49714154)

[Czułość 43](#_Toc49714155)

[Swoistość 43](#_Toc49714156)

[Miara F1 43](#_Toc49714157)

[Generowanie danych uczących 44](#_Toc49714158)

[Baza wiedzy 44](#_Toc49714159)

[Generator ciągów dzięków 44](#_Toc49714160)

[Sposób działania 45](#_Toc49714161)

[Przykładowe wygenerowane ciągi 46](#_Toc49714162)

[Konfiguracja generatora 46](#_Toc49714163)

[Format danych wejścia i wyjścia 48](#_Toc49714164)

[Wykrywanie dźwięków 50](#_Toc49714165)

[// TODO Przeprowadzone eksperymenty 50](#_Toc49714166)

[// TODO Model liniowy 50](#_Toc49714167)

[// TODO Jednokanałowy Model Resnet 50](#_Toc49714168)

[// TODO Jednokanałowy model Resnet z warstwami rekurencyjnymi 50](#_Toc49714169)

[// TODO Analiza wyników eksperymentów. 50](#_Toc49714170)

[Podsumowanie 50](#_Toc49714171)

[// TODO Wnioski 50](#_Toc49714172)

[// TODO Perspektywy dalszego rozwoju projektu. 50](#_Toc49714173)

[Wykaz rysunków 51](#_Toc49714174)

[Wykaz Tabel 53](#_Toc49714175)

[Bibliografia 54](#_Toc49714176)

Cel i zakres pracy

Celem pracy jest stworzenie programu komputerowego który umożliwia generowanie zapisu nutowego utworu granego na gitarze na podstawie podanego pliku dźwiękowego. Program będzie miał postać skryptu w języku python. Wykrywanie tonów oraz wielotonów realizowane będzie będzie przez głęboką splotowo-rekurencyjną sieć neuronową. Dane uczące do sieci generowane będą przy pomocy obszernej biblioteki dźwięków synth.

Problem wykrywania tonów w muzyce jest trudny. Nawet dla doświadczonych muzyków jest to proces czasochłonny i nieraz skomplikowany. Według autorów [[1]](#endnote-1), narzędzia automatycznej transkrypcji na chwilę obecną nie są w stanie konkurować z ludzką percepcją. Głównym problemem stojącym przed siecią neuronową jest rozpoznawanie wielu dźwięków jednocześnie, na przykład akordów. Rozwiązanie dostępne jako skrypt w języku Python pozwala połączyć prostotę obsługi z zaawansowanym rozwiązaniem technologicznym, które ma zastosowanie w życiu codziennym.

Badania dotyczące przetwarzania sygnałów dźwięku pozwoliły na szeroki rozwój branży muzycznej, systemów komunikacji głosowej i obsługi komunikatów głosowych. Firma Slate Digital opracowała System Mikrofonu Wirtualnego (ang. Virtual Microphone System), który pozwala na emulację wpływu charakterystyki kierunkowej oraz brzmieniowej na zarejestrowaną muzykę po faktycznym nagraniu[[2]](#endnote-2). Wymagająca wiedzy i doświadczenia korekcja utworów automatyzowana jest przez sieci neuronowe[[3]](#endnote-3). Dostarczane rozwiązania wykorzystywane są w systemach internetu rzeczy które coraz częściej napotykamy w codziennym życiu. Oprogramowanie pozwalające na łatwą transkrypcję muzyki na zapis nutowy jest kolejnym krokiem w kierunku rozwiązania trudnego problemu inżynierskiego jakim jest rozpoznawanie zdarzeń dźwiękowych, w tym szczególnym przypadku muzyki granej na gitarze, w zadanym sygnale dźwiękowym.

Głębokie sieci neuronowe mające szerokie zastosowanie w analizie zdjęć, a najnowocześniejsze rozwiązania pozwalają na klasyfikację obrazów na poziomie 99.3%[[4]](#endnote-4) oraz wykrywanie obiektów na obrazach na poziomie 71.9%[[5]](#endnote-5). Sieci neuronowe mają coraz większe zastosowanie w przetwarzaniu sygnałów, w tym w rozpoznawaniu dźwięków instrumentów w nagraniach. Metody oparte o analizę danych eliminują wiele słabości algorytmów opartych o analizę heurystyczną i klasyczne przetwarzanie sygnałów dźwięku[[6]](#endnote-6), takich jak pYIN czy SWIPE.

Operacje takie jak odszumianie oraz poprawa jakości sygnałów mowy przy pomocy głębokich sieci neuronowych są realizowane na szeroką skalę przez firmy dostarczające rozwiązania służące do przeprowadzania zdalnych rozmów, telekonferencji oraz rozpoznawania komend głosowych. Prace prowadzone w kierunku augmentacji sygnałów mowy, opisane przez autorów prac “Speech Enhancement In Multiple-Noise Conditions using Deep Neural Networks” oraz “Speech Enhancement Using Multiple Deep Neural Networks” pokazały, że sieci neuronowe coraz lepiej radzą sobie z tym zadaniem.

Sztuczna inteligencja ma coraz większe zastosowanie w wykrywaniu częstotliwości, analizie częstotliwościowej i problemach związanych z rozpoznawaniem zdarzeń dźwiękowych. Rdzeniem trzworznego programu, elementem mającym za zadanie ekstrakcję informacji o występujących dźwiękach w nagraniu, jest głęboka sieć neuronowa. Danymi uczącymi tej sieci są wygenerowane sztucznie melodie składające się z nagranych wcześniej dźwięków gitary ze zbioru nsynth. Architektura sieci neuronowej została wybrana poprzez przegląd literatury oraz przeprowadzone eksperymenty. Zastosowana została architektura rekurencyjno-splotowa typu wavenet której wyjście to macierz gdzie wymiar pionowy stanowią indeksy midi, a wymiar poziomy czas. Indeksy midi będą następnie mapowane wprost do notacji abc[[7]](#endnote-7) pozwalającej w prosty sposób używać notacji muzycznej. Docelowo używana sieć neuronowa ma być w stanie rozpoznawać pojedyncze dźwięki na poziomie większym niż 90%, a dźwięki do trzech jednocześnie na poziomie 60%.

W rozdziale “[Przegląd rozwiązań z literatury](#_qfob632jaa6t)” omówione zostały przykłady podejść do problemu rozpoznawania dźwięków. W rozdziale ‘Metody analizy sygnałów muzycznych” opisane są sposoby na wyznaczanie częstotliwości bazowej oraz sposoby na badanie sygnałów w dziedzinie czasu i częstotliwości. Następnie omówione zostały ...

// TODO

Przegląd rozwiązań z literatury

W niniejszym rozdziale omówione zostaną artykuły oraz publikacje związane z problemem transkrypcji muzyki oraz wykrywania zdarzeń dźwiękowych w nagraniach. Ze szczególną uwagą zostały przedstawione takie aspekty jak zaproponowana architektura, używany zbiór danych uczących oraz wyniki przeprowadzanych eksperymentów. Tematyka transkrypcji poruszona została w pracy [[[8]](#endnote-8)] w której autorzy przyglądają się problemowi ekstrakcji pojedynczych dźwięków z plików audio. Autorzy w celu analizy spektrum częstotliwościowego najpierw zamieniają dźwięk w formie amplitudowej na spektrogram przy pomocy Dyskretnej Transformaty Fouriera. W artykule wykazano, że najlepiej z problemem ekstrakcji pojedynczych dźwięków z sygnału radzi sobie warstwa splotowata po której następuje warstwa w pełni połączona. Pomimo bardzo dobrych wyników dla sieci rozpoznającej dany dźwięk poważne problemy pojawiają się przy próbach wykrywania wielu dźwięków granych jednocześnie. Autorzy zwracają uwagę na to, że do problemu transkrypcji muzyki wykorzystane mogą zostać inne architektury, które oryginalnie zaprojektowane zostały do rozpoznawania zdarzeń dźwiękowych takich jak na przykład szczekanie psa czy jadący samochód. Na podstawie eksperymentów autorzy stawiają konkluzję, że sieci neuronowe są w stanie nauczyć się rozpoznawać kombinacje wielu dźwięków jednocześnie, ale nie radzą sobie z tymi kombinacjami dźwięków których wcześniej nie widziały[[9]](#endnote-9). Dodatkowo autorzy [8] proponują szereg augmentacji, które można zastosować by zwiększyć osiągi sieci. Te augmentacje to: zwiększenie ilości wymiarów danych, transformacje symetryczne sygnału oraz dodawanie delikatnego szumu.

W 2017 roku autorzy [[[10]](#endnote-10)] pokazali, że architekturą sieci neuronowej która najlepiej radzi sobie z problemem rozpoznawania zdarzeń dźwiękowych jest architektura splotowo-rekurencyjna. Sieć składa się z czterech głównych warstw: konwolucyjnych, rekurencyjnych, jednokierunkowej oraz warstwy binaryzującej. Jako wejście sieć przyjmuje mel spektrogram w którym wymiar wysokości to częstotliwość w skali mel, a wymiar szerokości to czas. Warstwy konwolucyjne odpowiedzialne są za ekstrakcję cech z dostarczonego wektora. Warstwy rekurencyjne pozwalają na uczenie sieci zależności czasowych, które w muzyce są jak najbardziej wskazane. Warstwa jednokierunkowa jest aktywacją sigmoidalną. Ta funkcja aktywacji, ponieważ jej wartości należą do zbioru (0, 1), jest używana do wnioskowania opartego o prawdopodobieństwo[[11]](#endnote-11). Ostatnią warstwą jest warstwa binaryzująca, która w zależności od poszczególnych wartości wejściowych zmienia ich wartość na 1, jeżeli wartość wejściowa jest większa od 0.5 oraz 0 w przeciwnym wypadku. Sieć wykrywa pojedyncze oraz nakładające się zdarzenia dźwiękowe. Jej uczenie oraz testowanie odbyło się na czterech zbiorach danych w celu uzyskania większej niezależności otrzymanych wyników. Te zbiory to TUT-SED Synthetic 2016, TUT-SED 2009[[12]](#endnote-12), TUT SED 2016[[13]](#endnote-13) oraz CHiME-Home[[14]](#endnote-14) stworzone przez Politechnikę w Tampere.

Metryki wykorzystane podczas uczenia opisywanej sieci neuronowej to wynik F oraz poziom błędu (ER). U podstaw obu metryk leżą pojęcia opisujące ilość poprawnych i niepoprawnych prób wnioskowania:

* prawdziwe pozytywne oznaczające, że sieć poprawnie zidentyfikować obecność zdarzenia dźwiękowego;
* fałszywe pozytywne oznaczające, że sieć wskazała na obecność zdarzenia dźwiękowego które wcale nie miało miejsca;
* fałszywe negatywny oznaczające, że sieć nie wykazała obecności zdarzenia dźwiękowego pomimo, że faktycznie miało ono miejsce[[15]](#endnote-15).

Powyższe pojęcia będą określane kolejno jako TP, FP, FN w dalszej części tej pracy. F jest reprezentacją efektywności ekstrakcji cech której kalkulacja oparta jest o wartości TP, FP oraz FN. ER jest stosunkiem trzech sum: poprawnie zidentyfikowanych zdarzeń, zdarzeń zidentyfikowanych niepoprawnie i zdarzeń niezidentyfikowanych, do liczby wszystkich faktycznie znajdujących się w sygnale zdarzeń. Autorzy opisanego w poprzednim paragrafie rozwiązania uzyskali średni wynik F równy 55,3 oraz średni wynik ER wynoszący 0.63. Warto zwrócić uwagę na fakt, że wyniki uzyskane przez autorów dla zbioru TUT-SED 2016 są o około połowę gorszę od wyników uzyskanych dla zbiorów TUT-SED Synthetic 2016 oraz TUT-SED 2009.

Jako rozwinięcie opisanego w poprzednim paragrafie rozwiązania w pracy [[[16]](#endnote-16)] autorzy opisują rozszerzenie zaproponowanej architektury o ekstrakcję cech związanych z wielokanałowością dźwięków. Autorzy wykorzystują takie zjawiska związane z ludzką, binauralną, percepcją dźwięku. Dodatkowy wymiar pozwala na uwzględnienie zjawiska różnych poziomów dźwięku docierającego do każdego z uszu. Ponadto w trakcie wnioskowania możliwe jest wzięcie pod uwagę różnego czasu dotarcia dźwięku do uszu. Autorzy zwracają uwagę na to, że poprawę w rezultatach osiąganych przez sieci neuronowe podczas rozwiązywania problemu detekcji wydarzeń dźwiękowych dało uczenie sieci rozróżniania częstotliwości dominujących w przykładowych nagraniach[[17]](#endnote-17). Fakt ten świadczy o tym, że sieci neuronowe są w stanie uczyć się budowy akordów granych na gitarze. Cechy

opisujące trzy wymienione powyżej zjawiska ekstraktowane są poprzez leżące równolegle warstwy splotowe. W ten sposób wykonywane jest wnioskowanie dla poszczególnych zjawisk osobno, a wyjścia warstw splotowych są poddawane konkatenacji. Tak utworzona struktura podawana jest na wejście warstwy rekurencyjnej, która umożliwia ekstrakcję zależności czasowych występujących w sygnale. Ostatnią warstwą jest warstwa w pełni połączona. Użytymi przez autorów zbiorami danych są TUT-SED 2009 oraz TUT-SED 2016. Pozwala to na bezpośrednie i łatwe porównanie z rozwiązaniem przedstawionym w poprzednim paragrafie. Autorzy wykazują poprawę F-score o 2.7% dla zbioru TUT-SED 2009 oraz o 6.1% dla zbioru TUT-SED 2016

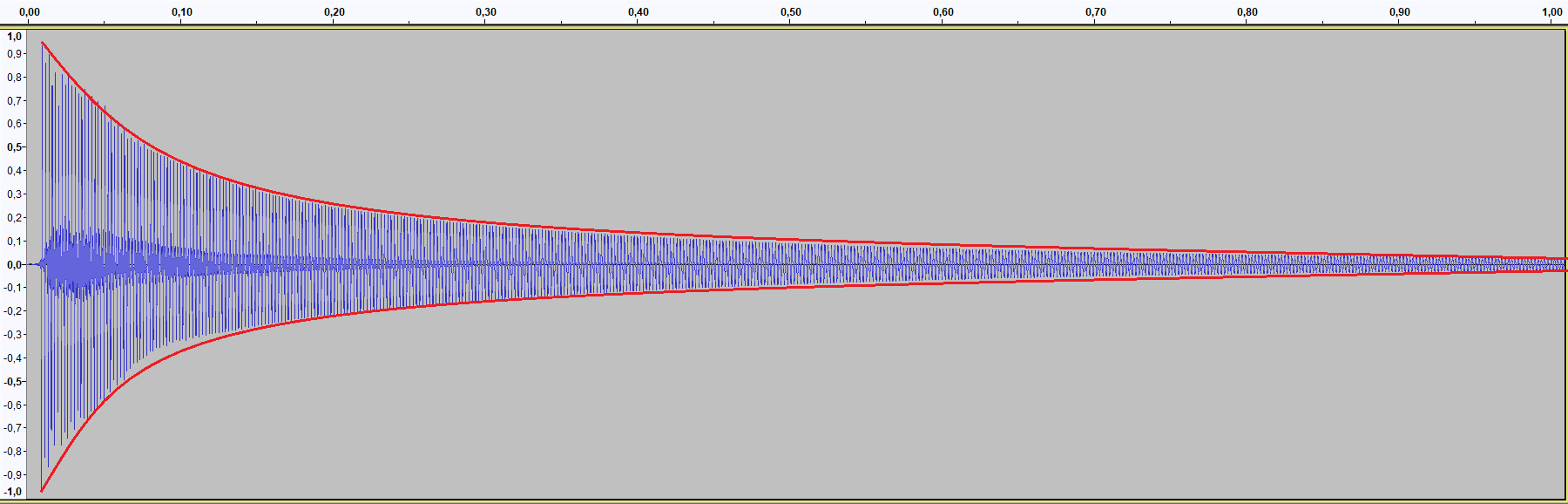
W 2020 roku zaproponowane zostało rozwiązanie oparte o warstwy konwolucyjne rozdzielne w wymiarze głębokości (ang. depthwise separable convolution) oraz warstw splotowych o dylatacyjnym jądrze (ang. dilated convolution)[[18]](#endnote-18). Pierwsze pozwalają na ekstrakcję oraz uczenie cech podobnie do stosowanych w [8] i [14] warstw splotowych przy znacznym obniżeniu ilości potrzebnych parametrów i wykonanych operacji. Podobne zalety daje zastąpienie warstw rekurencyjnych warstwami dylatacyjnymi, które pozwalają modelować zależności czasowe w sygnale. Badania przeprowadzone zostały przy użyciu bazy wiedzy TUT-SED Synthetic co pozwala na łatwe porównanie z opisanymi w dwóch poprzednich paragrafach rozwiązaniami.Autorzy tego rozwiązania opisują, że wykorzystanie obu tych technik jednocześnie może skutkować spadkiem wydajności całej architektury. Jednak wykorzystanie architektury składającej się z warstw splotowych rozdzielnych w wymiarze głębokości oraz warstwy rekurencyjnej pozwoliło na poprawę zarówno wyniku F jak i ER. W przypadku wykorzystania warstw splotowych dylatacyjnych o dylatacji większej od 50 odnotowano poważne spadki w detekcji wydarzeń dźwiękowych

Metody analizy sygnałów muzycznych

Podczas analizy sygnałów akustycznych należy brać pod uwagę sygnał w dziedzinie czasowej oraz dziedzinie częstotliwości. Sygnał przedstawiony w dziedzinie czasu przedstawia kształt fali. Dziedzina czasowa reprezentuje spektrum, sposób w jaki rozłożone są natężenia składowych częstotliwościowych danego sygnału.

**Analiza czasowa**

Wykres poniżej przedstawia zarejestrowany dźwięk C grany na gitarze akustycznej.



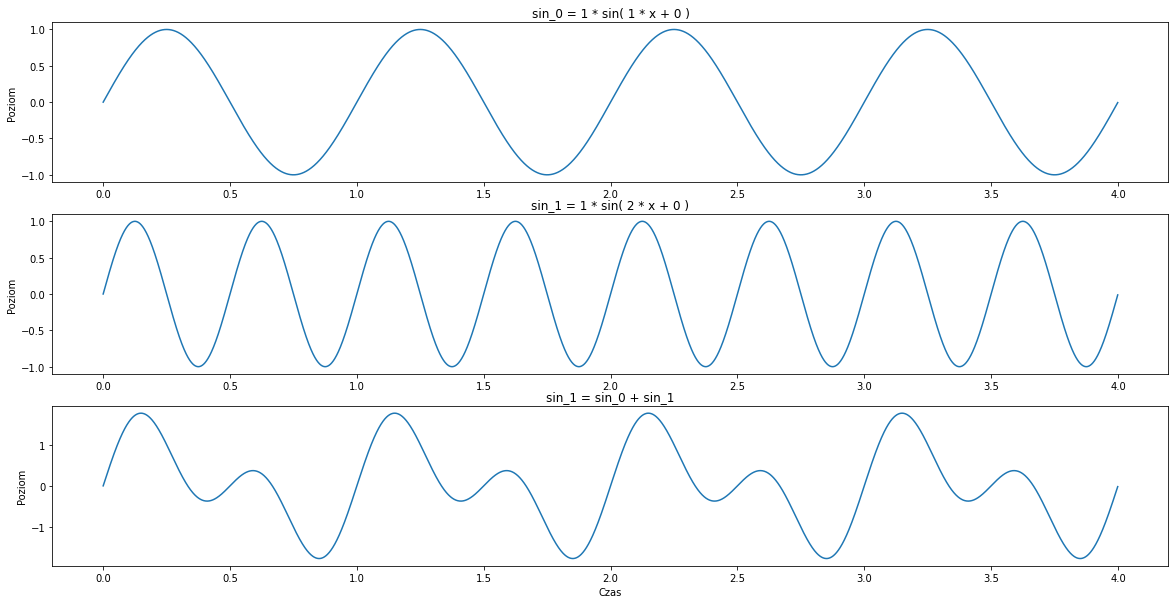
Rys 1. Dźwięk C grany na gitarze akustycznej.

Na analizowanym wykresie wyraźnie widać, że dźwięk na początku jest głośny a następnie następuje jego wyciszenie. Zmiany głośności sygnału tym jak głośny jest sygnał informuje obwiednia sygnału. Obwiednia jest linią podążającą za wierzchołkami kształtu fali zaznaczoną na wykresie czerwonym kolorem. Najprostszą metodą wyznaczania obwiedni jest podniesienie sygnału do kwadratu, następnie podzielenie go na krótkie fragmenty. Powstałe w ten sposób fragmenty należy uśredniać[[19]](#endnote-19).

Analiza częstotliwościowa

Sposób w jaki rozłożone są natężenia składowych fali akustycznej w zależności od częstotliwości w danej fali nazywa się widmem. Widmo zawsze wyznacza się dla określonej ramki czasowej dźwięku. Widmo takiej jednej ramki nazywa się widmem statycznym.

Podczas wyznaczania widma sygnału używana jest Transformata Fouriera. Złożone sygnały muzyczne są de facto sumą określonej ilości fal sinusoidalnych o różnych częstotliwościach. Ideę dobrze pokazuje zamieszczony poniżej rysunek. Są na nim trzy wykresy. Pierwsze dwa przedstawiają sygnały o kształcie sinusoidy o częstotliwościach wynoszących kolejno jedną i dwie jednostki czasu. Trzeci wykres to sygnał złożony, wynikowa sinusoida będąca sumą dwóch pozostałych sygnałów prostych.



Rys. 2 fala akustyczna jako suma sinusów.

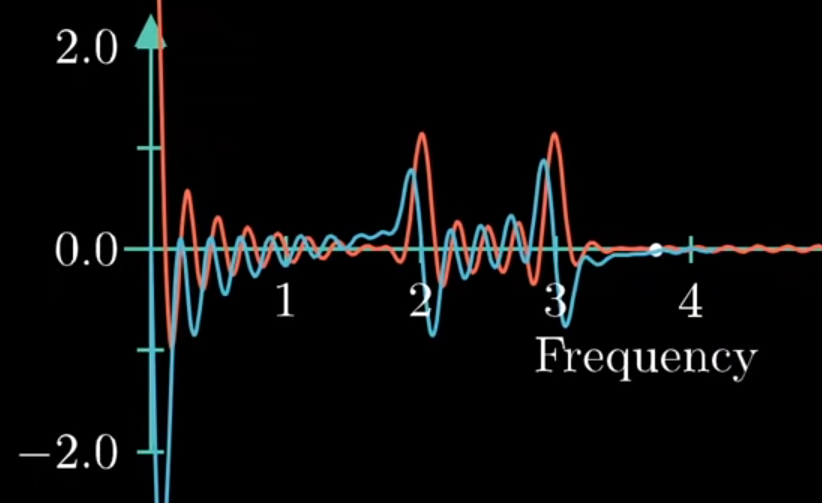
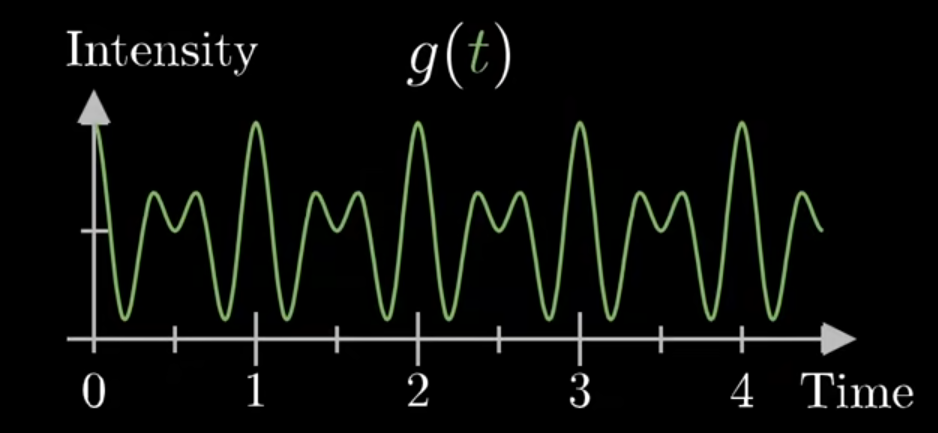
Transformata fouriera pozwala na dekompozycję złożonego sygnału na sygnały proste. Jest ona dana wzorem:

,

gdzie t - oznacza czas, a f - częstotliwość. Wynikowa funkcja *G(f)* opisuje to jak duży udział w sygnale g(t) ma dana częstotliwość f. Aby z postaci częstotliwościowej uzyskać postać czasową należy zastosować Odwrotną Transformatę Fouriera (IFT) daną wzorem[[20]](#endnote-20):

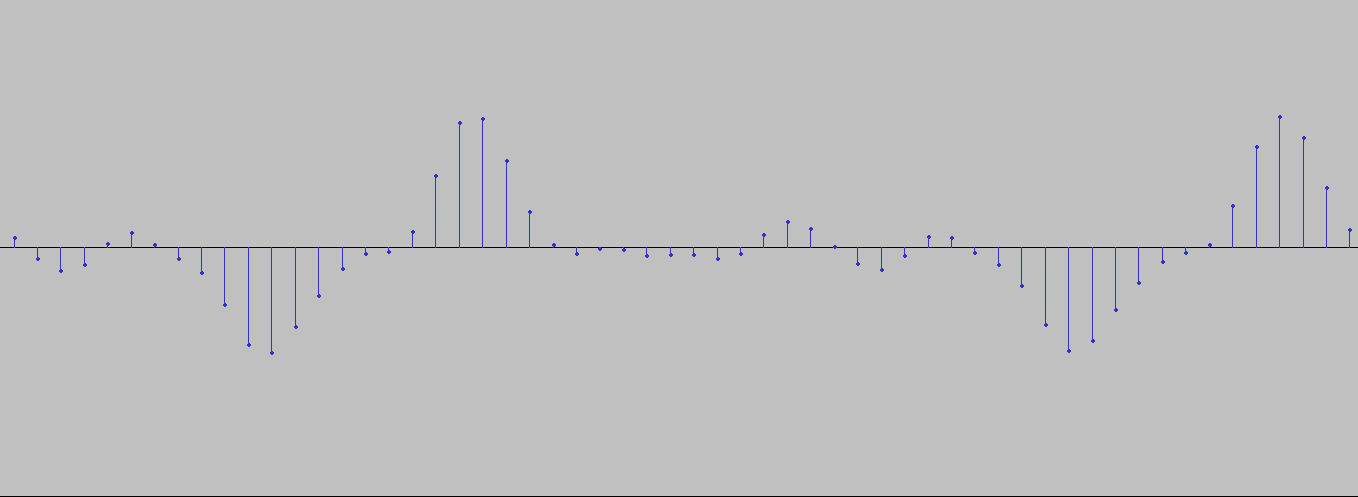
.

Wynik transformaty fouriera przedstawiony jest na wykresach poniżej. Rys 3 przedstawia postać czasową fali złożonej z dwóch fal sinusoidalnych. Rys. 4 przedstawia jego reprezentację w dziedzinie częstotliwości. Na wykresie czerwoną linią zaznaczone są wartości rzeczywiste, zaś linią niebieską zaznaczone są wartości urojone. Z rys. 3 trudno odczytać jakie częstotliwości składowe znajdują się w sygnale. Na rys. 4 widoczne są maksima lokalne dla częstotliwości składowych sygnału tj. w tym wypadku 2 i 3.



Rys. 3 postać czasowa sygnału złożonego Rys. 4 postać częstotliwościowa[[21]](#endnote-21)

Transformata Fouriera aplikowana jest na nieskończonym sygnale analogowym. Rodzi to problemy natury praktycznej. Po pierwsze komputery przechowują skwantyzowany sygnał bitowy nie ciągły. Po drugie sygnały muzyczne są skończone. Na wykresie poniżej przedstawiony jest skwantowany sygnał w formie w jakiej przechowywany jest faktycznie w pamięci komputera.



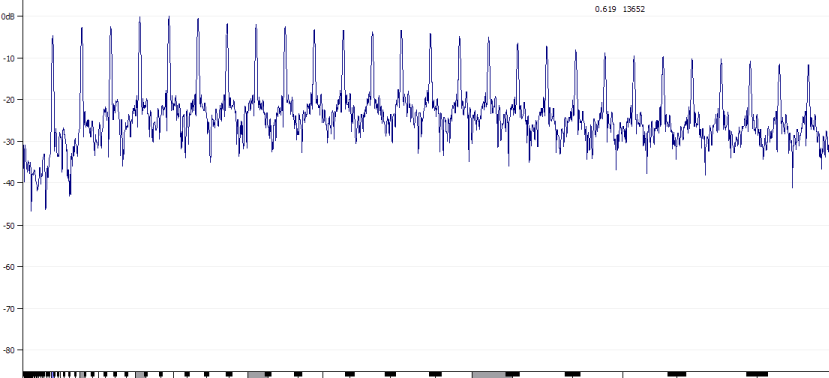
Rys 5. Skwantyzowany sygnał muzyczny

Natura takich sygnałów akustycznych wymaga wykorzystania dyskretnej transformaty fouriera (DFT). Kolejnym problemem jest złożoność obliczeniowa. Algorytm wynikający bezpośrednio z definicji ma złożoność obliczeniową . Z tego powodu stosuje się Szybką Transformatę Fouriera również implementuje obliczanie Transformaty Fouriera, której złożoność obliczeniowa wynosi [[22]](#endnote-22)

Aby obliczyć widmo dźwięku należy:

* wybrać ramkę czasową sygnału do analizy
* wybrać funkcję okna
* obliczyć FFT
* wyznaczyć widmo amplitudowe jako wartości bezwzględnej wyniku FFT
* wyznaczyć widmo fazowe sygnału jako kąta fazowego wyniku FFT

Na rysunku poniżej znajduje się widmo statyczne częstotliwościowe nagrania trąbki. Na wykresie zaobserwować można charakterystyczne prążki. Są to częstotliwości harmoniczne. Nie można ich zaobserwować dla instrumentów perkusyjnych.



Rys 6. Widmo statyczne dźwięku trąbki (w stanie ustalonym) poziom widma [dB] w funkcji częstotliwości19

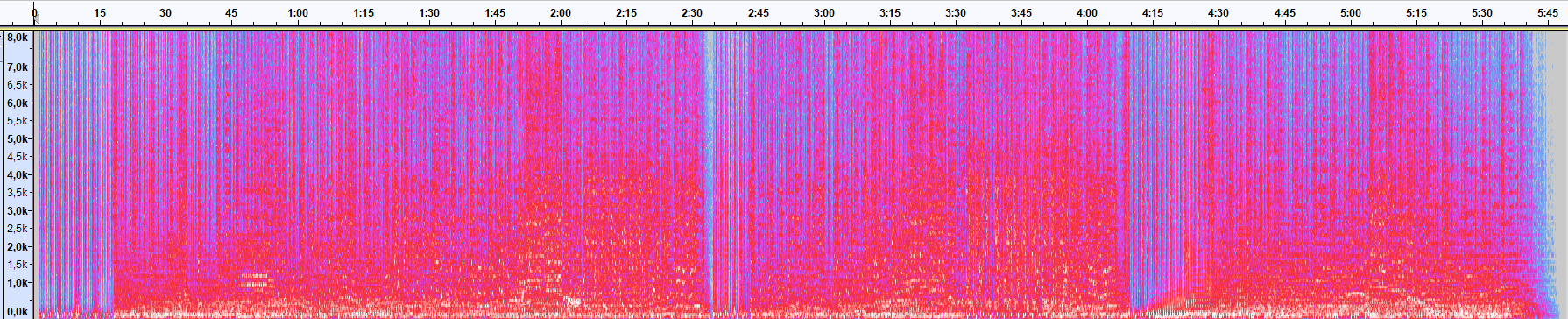
W praktyce widmo częstotliwościowe zmienia się w czasie. Z tego powodu analiza pojedynczych ramek jest niewystarczająca i stosuje się analizę czasowo-częstotliwościową.

Analiza STFT

Widmo dźwięku z czasem zmienia się. Z tego powodu należy zbadać widmo dźwięku w różnych punktach na osi czasu. Wykorzystuje się do tego analizę czasowo częstotliwościową STFT (eng. Short-term Fourier Transform). Polega ona na:

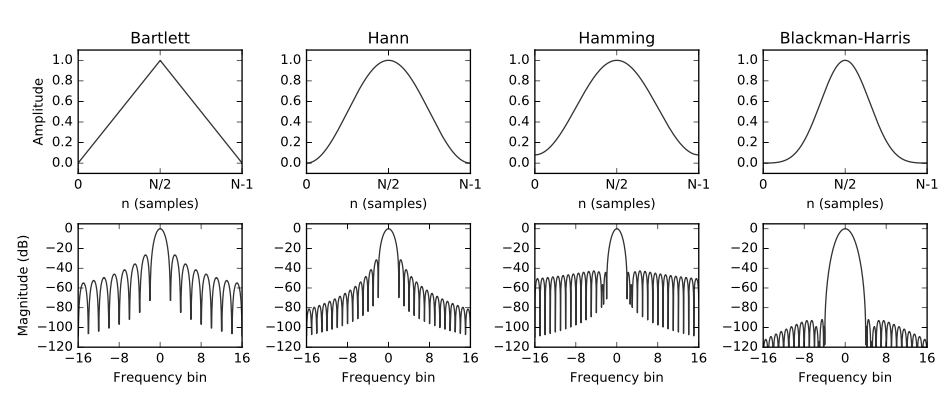
* Podzielenie sygnału na okna
* wyznaczenie widma statycznego dla okien
* połączenie wyników analizy poszczególnych okien.

Najczęściej spotykaną reprezentacją graficzną wyniku analizy jest spektrogramy, gdzie na osi poziomej oznaczony jest czas trwania sygnału zaś na osi pionowej oznacza się częstotliwość. Nasilenie danej barwy określa amplitudę widma. Na spektrogramie poniżej przedstawiony jest wynik analizy STFT.



Rys 7. Analiza STFT utworu “Radio gaga” wydanego w 1986 roku przez zespół Queen.

Podczas dzielenia sygnału na ramki o oknie prostokątnym powstaje zjawisko nazywane przeciekami widma. Aby temu zapobiec stosuje się różne funkcje okna. Nie ma arbitralnie lepszych czy gorszych funkcji okna, odpowiedni wybór to kompromis między szerokością prążka, a tłumieniem listków bocznych. Na załączonym poniżej wykresie przedstawione są niektóre z okien stosowanych podczas analizy czasowo częstotliwościowej.

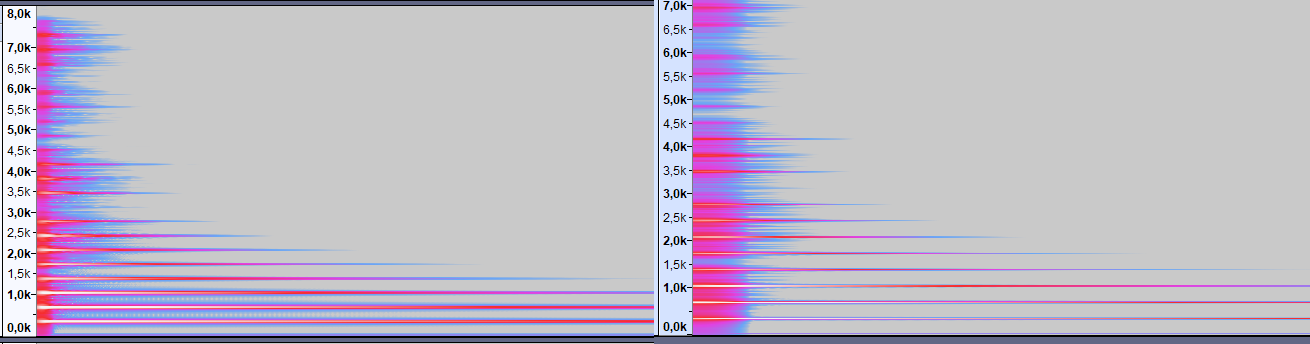


Rys. 8 Cztery często używane funkcje okna (na górze) i odpowiadające im Transformaty Fouriera (na dole)[[23]](#endnote-23).

Rozdzielczość częstotliwościowa STFT określa minimalną rozróżnialną różnicę między dwoma składowymi widma. Innymi słowy dla okna o rozmiarze wynoszącym 1024 próbek i częstotliwości próbkowania wynoszącej 48 kHz rozdzielczość częstotliwościowa wyniesie:

Oznacza to, że dwie składowe widma o częstotliwościach odległych od siebie o mniej niż =46.875 będą nieodróżnialne. Zwiększenie okna celem zwiększenia rozdzielczości częstotliwościowej skutkuje zmniejszeniem częstotliwości czasowej analizy. Dla okna wielkości 4096 próbek rozdzielczość częstotliwościowa wyniesie:

Pomimo znacznego zmniejszenia rozdzielczości częstotliwościowej pamiętać należy o fakcie, że ramka szerokości 4096 pokrywa odstęp czasowy wynoszący 85,33 milisekundy. Oznacza to, że nie rozróżnialne będą zjawiska czasowe w odstępnie 85,33 od siebie. Rozdzielczość czasowa dla ramki wielkości 1024 wynosi 21.3325. Należy zatem mądrze dobierać szerokość ramki tak by uzyskać najlepszy pożądany efekt. Na rysunku poniżej znajdują się: po lewej sygnał poddany analizie STFT o rozmiarze okna wynoszącym 1024 próbki; po prawej sygnał poddany analizie STFT o rozmiarze okna wynoszącym 4096 próbek. Szczególnie na początku sygnału gdzie jest on jeszcze mocny widać wyraźną różnicę w rozdzielczości czasowo-częstotliwościowej dla obu ramek analizy.

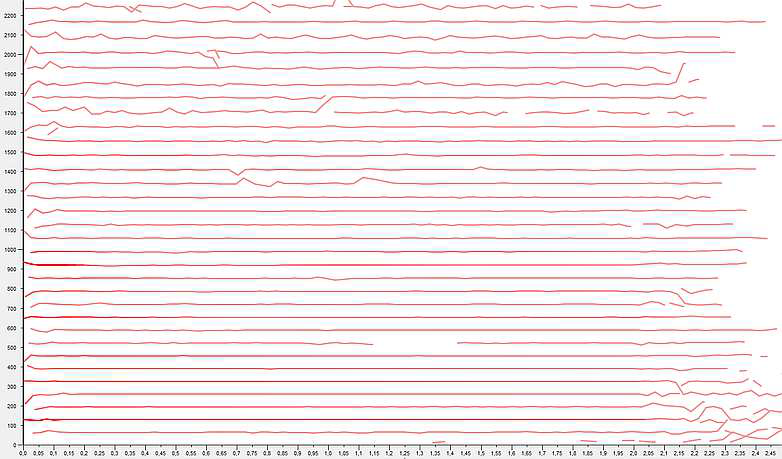


Rys 9. Rozdzielczość czasowo-częstotliwościowa. Rozmiar okna analizy: po lewej 1025; po prawej 4096

Podczas dzielenia sygnału na okna często stosuje się zakładkowanie. Jest to technika polegająca na przesuwaniu okna wzdłuż osi czasu nie o całą długość okna, a na przykład 25% czy 50% jego szerokości. Pozwala to uzyskać lepszą rozdzielczość czasową i zmniejszyć zniekształcenia spowodowane mnożeniem przez funkcję okna. Wadą jest wydłużenie czasu trwania algorytmu jednak przy mocy obliczeniowej współczesnych komputerów nie stanowi to realnego problemu. Aby zachować rozdzielczość częstotliwościową przy jednoczesnym poprawieniu rozdzielczości czasowej stosuje się również uzupełnianie zerami. Jest to nic więcej jak dodanie do końca sygnału określonej ilości zer.[[24]](#endnote-24) Inną techniką jest analiza widma progowego, gdzie uwzględnia się tylko wartości widma których amplituda jest większa niż określony próg. Jeszcze inną metodą na uwydatnienie wartości widma a dużych amplitudach jest analiza maksimów lokalnych. Pozwala to na obserwację tego jak częstotliwości się zmieniają oraz w jaki sposób układają się prążki amplitud składowych harmonicznych. Rozwinięciem opisanych metod analizy jest Analiza McAulay-Quatieri (MQ) której algorytm jest następujący:

1. Oblicz STFT
2. Zastosuj analizę widma progowego
3. połącz wyniki kolejnych ramek
4. odrzuć ścieżki które się urywają, są krótkotrwałe.

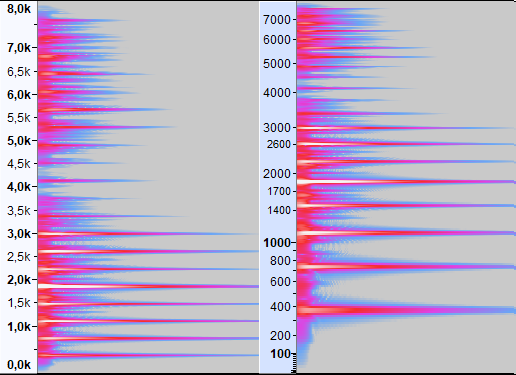
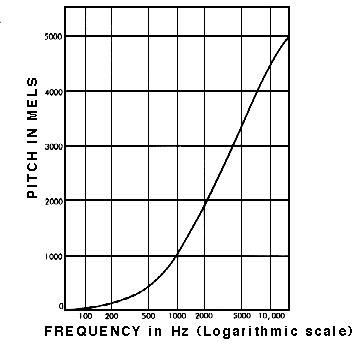
Wynik przykładowej analizy MQ



Rys. 10. Analiza McAulay-Quatieri (MQ) 19

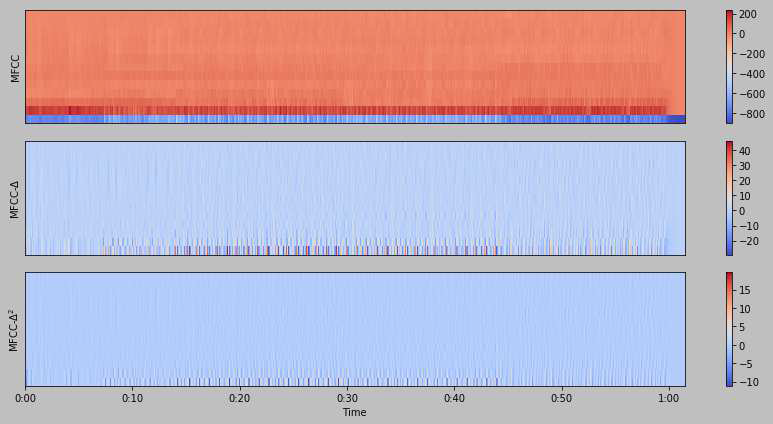
Współczynniki mel cepstralne

Ludzka percepcja skonstruowana jest w taki sposób, że łatwiej jest ludziom dostrzegać różnice pomiędzy sygnałami o niższych częstotliwościach częstotliwościach niż różnice pomiędzy sygnałami o wyższych częstotliwościach. Dla przykładu subiektywna różnica pomiędzy sygnałem o częstotliwości 500 Hz, a 1000 Hz jest o wiele większa niż dla sygnałów o częstotliwościach 1000 Hz, a 1500 Hz pomimo, że faktyczny interwał wynosi 500 Hz[[25]](#endnote-25). Zaproponowana w 1937 roku skala melowa określa relację między wysokością tonów prostych, a częstotliwością określając subiektywnie równo odległe od siebie wysokości dźwięków. Jako punkt odniesienia przyrównuje się częstotliwość 1000 Hz, o poziomie 42 dB do 1000 Meli. Na rysunku poniżej znajdują się trzy wykresy. Pierwszy od lewej przedstawia skalę melową w funkcji częstotliwości. Środkowy przedstawia spektrogram dźwięku F#4 zaś pierwszy od prawej to mel-spektrogram, innymi słowy spektrogram w skali melowej, dźwięku F#4.



Rys. 11. Od lewej: Skala melowa w funkcji częstotliwości [[26]](#endnote-26) [[27]](#endnote-27), spektrogram dźwięku F#4, mel-spektrogram dźwięku F#4.

Często wykorzystywane są współczynniki mel-cepstralne(MFCC). Aby je wyznaczyć należy obliczyć FFT sygnału oraz wyznaczyć jego kwadrat modułu widma. Następnie należy zastosować skalę melową. Wynik tej operacji należy zlogarytmować, a następnie zastosować odwrotną transformatę kosinusową. Otrzymane parametry to właśnie MFCC. Różnica pomiędzy obecną ramką, a ramką poprzednią nazywana jest MFCC-Δ, pierwszą pochodną MFCC. Jako MFCC-Δ2, drugą pochodną MFCC, oblicza się różnicę pierwszych pochodnych dla dwóch ramek, obecnej i poprzedniej.



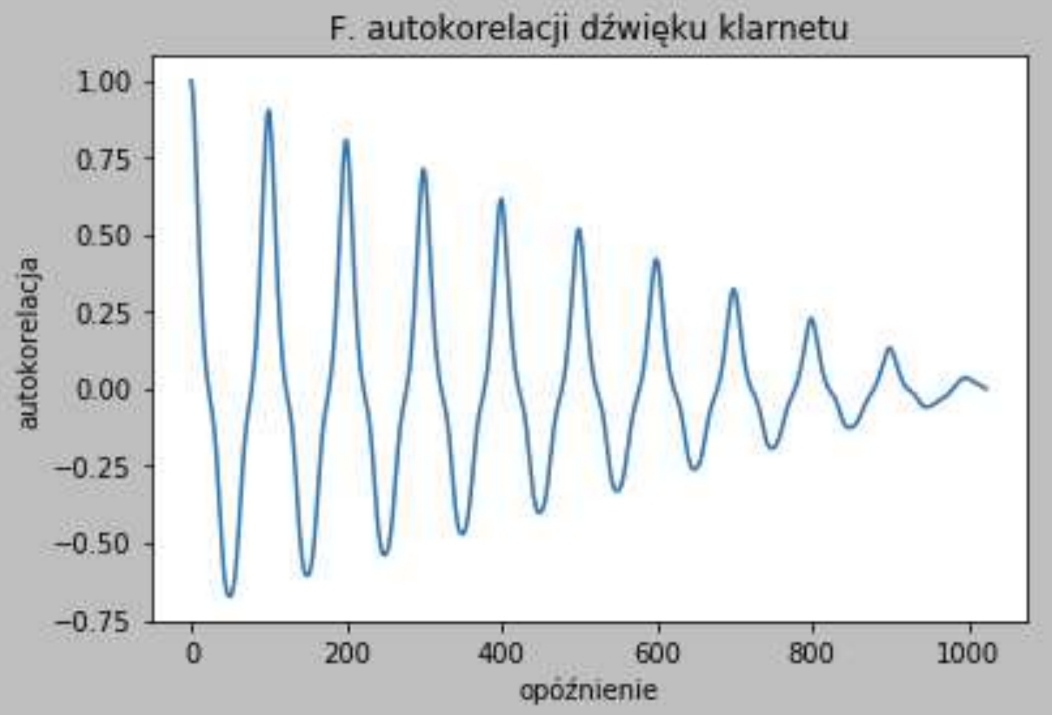
Rys. 12. Wykres MFCC i pochodnych. Współczynniki MFCC są używane m.in. do parametryzacji barwy dźwięków muzycznych.19

Analiza autokorelacyjna

Oprócz opisanych powyżej metod istnieje również metoda analizy autokorelacyjnej. Może ona posłużyć na przykład do wykrywania częstotliwości podstawowej sygnału. Opiera się ona o wyznaczanie korelacji między sygnałem oryginalnym, a sygnałem przesuniętym. Wynik analizy autokorelacyjnej przedstawiany jest na autokorelogramach, które podobnie do spektrogramów pozwalają na analizę składowych harmonicznych sygnałów. Autokorelacje sygnału oblicza się w następujący sposób:

1. Należy pobrać próbkę sygnału o długości dłuższej od okresu sygnału.
2. Obliczenie widma sygnału, szybkiej Transformaty Fouriera sygnału.
3. Wyznaczenie modułu z widma
4. Wyznaczenie odwrotnej Transformaty Fouriera z części rzeczywistej modułu widma
5. Ucięcie symetrycznej połowy sygnału
6. Normalizacja sygnału do wartości z zakresu (-1, 1)

W ten sposób wyznaczyć można częstotliwość podstawową sygnału. Na wykresie poniżej znajduje się funkcja autokorelacji dla dźwięku granego na klarnecie. Pierwsze maksimum lokalne funkcji autokorelacji, poza punktem zerowym przypada dla x = 100. Dla częstotliwości próbkowania wynoszącej 44100, częstotliwość bazowa sygnału wynosi 44100/100 = 441 Hz. Jest to zatem w przybliżeniu dźwięk "A" w oktawie razkreślnej.



Rys. 13. Funkcja autokorelacji dla dźwięku klarnetu

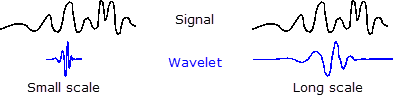
W porównaniu do FFT wyznaczanie częstotliwości podstawowej metodą analizy funkcji autokorelacji daje dokładniejsze wyniki.

Analiza falkowa

Stosowaną również metodą jest analiza falkowa. Falka to sygnał który jest skończony oraz ma zerową wartość średnią. Istnieje wiele rodzajów kształtów falek, wybór kształtu zależy od specyfiki i celu analizy. Podczas analizy wyznacza się parametry: translację i skalę dla których dana falka dopasowana jest do kształtu analizowanego sygnału. Parametr translacji, nazywany też w literaturze przesunięciem, odpowiada za przesuwanie określonej falki wzdłuż badanego sygnału. Skala zaś pozwala na zmianę “częstotliwości” falki tak aby mogła ona zostać dopasowana do sygnału. Wartość tego parametru jest odwrotnie proporcjonalna do “częstotliwości” falki:

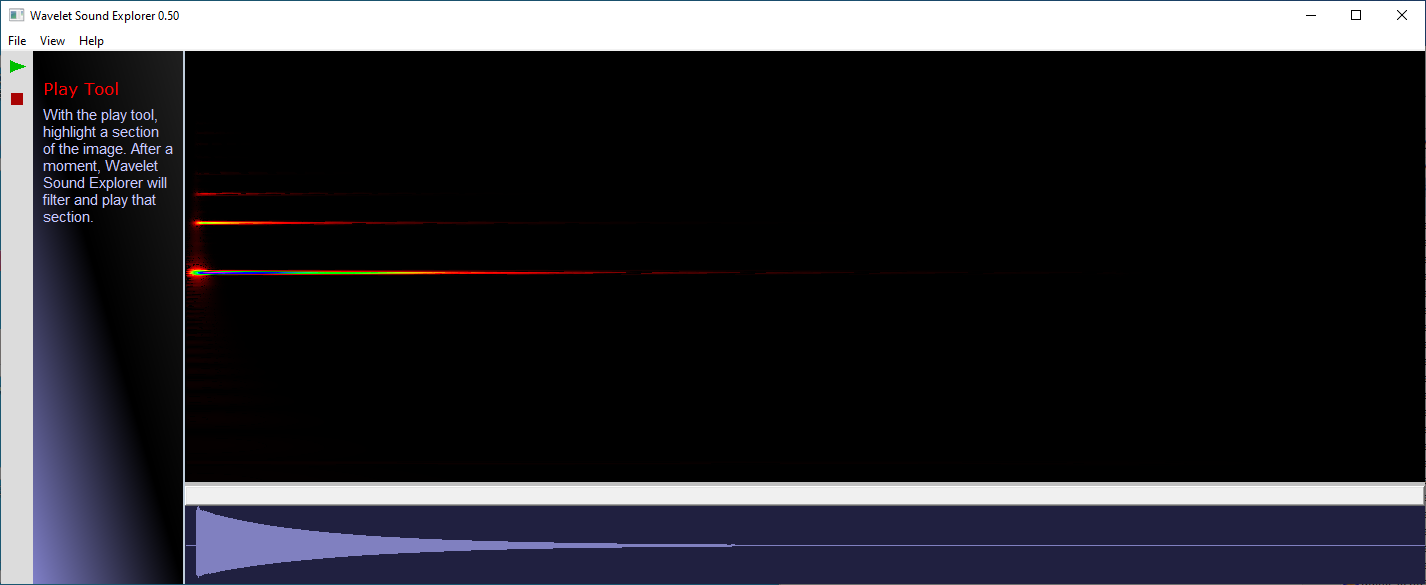
* mała skala - falka jest “ściśnięta” - szybkie zmiany w sygnale - wysokie częstotliwości
* duża skala - falka jest “rozciągnięta” - wolne zmiany w sygnale - niskie częstotliwości.

Opisaną ideę przedstawia rysunek numer 14 poniżej.



Rys 14. Związek pomiędzy skalą, a częstotliwością w sygnale[[28]](#endnote-28)

Pozwala to uzyskać dużo lepszą rozdzielczość czasową niż analiza z wykorzystaniem transformaty Fouriera. Na rysunku przedstawiony jest wynik analizy falkowej dźwięku F#4 przeprowadzonej przy pomocy oprogramowania Wavelet Sound Explorer[[29]](#endnote-29)



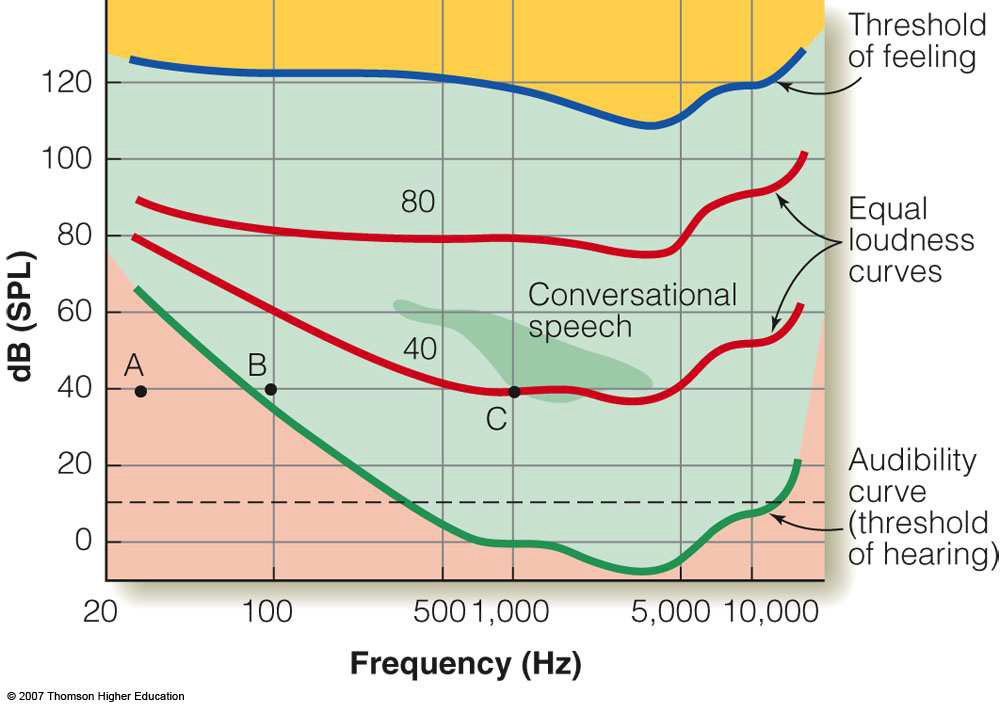
Rys 15. Wynik analizy falkowej dla dźwięku F#4

Zestawienie wyników analizy falkowej z rysunku 15 wynikami analizy STFT przedstawionymi na rysunku 11 identycznego sygnału pozwala na porównanie wyników obu analiz. Przede wszystkim zwrócić uwagę należy na sam początek sygnału. Wyniki analizy STFT sugeruje, że na początku sygnału znajduje się dużo częstotliwości których de facto w sygnale nie ma. Dopiero w dalszym oknie czasowym analizy STFT uwydatniają poszczególne składowe harmoniczne dźwięku. Na rysunku 15 brak jest tego szumu na początku sygnału, składowe są wyraźnie oddzielone i rozpoznawalne już w pierwszych interwałach czasu.

Dźwięki w formie muzyki

Muzyka jako sztuka operuje na specjalnie dobranych dźwiękach które ułożone są odpowiednio w czasie. Muzyk operując dźwiękami o określonych częstotliwościach jest w stanie stworzyć dzieło które oddziałuje na odbiorców, wpływa na nich, podoba się im lub nie.

Dźwięki są odbierane przez człowieka wskutek rozchodzącej się fali akustycznej w danym ośrodku, na przykład powietrzu. Zakres słyszenia ludzkiego słuchu obejmuje częstotliwości od 20 Hz to 20 kHz - zakres ten nazywany jest zakresem słyszalnym. Dźwięki z zakresu poniżej 20 Hz nazywane są infradźwiękami zaś powyżej 20 kHz nazywane są ultradźwiękami. Człowiek w różny sposób odbiera głośność dźwięków o takim samym poziomie ciśnienia akustycznego lecz różnych częstotliwościach. Krzywą słyszenia przedstawia rys. 1



Rys. 16 Zakres słyszenia[[30]](#endnote-30) Oś pozioma przedstawia częstotliwość, oś pionowa zaś to ciśnienie akustyczne. Niebieską linią zaznaczony jest próg bólu, zieloną zaś próg słyszenia. Linie czerwone łączą częstotliwości które dla człowiek odbiera.

Dźwięki

Każdy dźwięk w kontekście muzyki ma określone parametry, są to [[31]](#endnote-31):

* Wysokość
* Głośność
* Czas trwania
* Barwa

O wysokości dźwięku decyduje to jaką częstotliwość podstawową dana fala akustyczna ma. Za dźwięki niższe odpowiedzialne są fale o niższych częstotliwościach, zaś dźwięki wyższe wytwarzane są przez fale o częstotliwościach wyższych. Najmniejsza odległość, interwał między dwoma dźwiękami to półton i ma on wartość . Dwa półtony stanowią cały ton. W muzyce dźwiękowi o nazwie a1przypisuje się częstotliwość 440 Hz, dźwiękowi a2  przypisuje się wysokość 880 Hz. Stosunek wysokości tych dwóch dźwięków wynosi 2:1, i nazywany jest oktawą. W każdej oktawie znajduje się dwanaście półtonów. Literowe nazwy dźwięków przedstawione są w poniższej tabeli 1.

|  |  |  |
| --- | --- | --- |
| Literowa nazwa  dźwięku naturalnego | alterowany dźwięk  obniżony o półton | alterowany dźwięk podwyższony o półton |
| c | ces | cis |
| d | des | dis |
| e | es | eis |
| f | fes | fis |
| g | ges | gis |
| a | as | ais |
| h | b | his |

Tabela 1. Literowe nazwy dźwięków naturalnych i alterowanych.

Interwał pomiędzy dźwiękami c - d - e oraz f - g - a - h wynosi cały ton, a interwał między dźwiękami e-f oraz h-c wynosi pół tonu. Z uwagi na ten fakt każdy dźwięk może mieć wiele nazw. Dla przykładu dźwięk c obniżony o półton daje dźwięk ces. Ponieważ interwał między dźwiękami h oraz c wynosi półton, dźwięk ces oraz dźwięk h to dokładnie te same dźwięki.\

W tabeli poniżej znajduje się wykaz częstotliwości dla poszczególnych dźwięków w poszczególnych oktawach.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| subkontra | 4 186,01 | 4 434,92 | 4 698,64 | 4 978,03 | 5 274,04 | 5 587,65 | 5 919,91 | 6 271,93 | 6 644,88 | 7 040,00 | 7 458,62 | 7 902,13 |
| kontra | 2 093,00 | 2 217,46 | 2 349,32 | 2 489,02 | 2 637,02 | 2 793,83 | 2 959,96 | 3 135,96 | 3 322,44 | 3 520,00 | 3 729,31 | 3 951,07 |
| wielka | 1 046,50 | 1 108,73 | 1 174,66 | 1 244,51 | 1 318,51 | 1 396,91 | 1 479,98 | 1 567,98 | 1 661,22 | 1 760,00 | 1 864,66 | 1 975,53 |
| mała | 523,25 | 554,37 | 587,33 | 622,25 | 659,26 | 698,46 | 739,99 | 783,99 | 830,61 | 880,00 | 932,33 | 987,77 |
| razkreślna | 261,63 | 277,18 | 293,66 | 311,13 | 329,63 | 349,23 | 369,99 | 392,00 | 415,30 | 440,00 | 466,16 | 493,88 |
| dwukreślna | 130,81 | 138,59 | 146,83 | 155,56 | 164,81 | 174,61 | 185,00 | 196,00 | 207,65 | 220,00 | 233,08 | 246,94 |
| trzykreślna | 65,41 | 69,30 | 73,42 | 77,78 | 82,41 | 87,31 | 92,50 | 98,00 | 103,83 | 110,00 | 116,54 | 123,47 |
| czterokreślna | 32,70 | 34,65 | 36,71 | 38,89 | 41,20 | 43,65 | 46,25 | 49,00 | 51,91 | 55,00 | 58,27 | 61,74 |
| pięciokreślna | 16,35 | 17,32 | 18,35 | 19,45 | 20,60 | 21,83 | 23,12 | 24,50 | 25,96 | 27,50 | 29,14 | 30,87 |
|  | C | Cis/Des | D | Dis/Es | E | F | Fis/Ges | G | Gis/As | A | Ais/B | H |

Tabela 2 .Dźwięki w oktawach w kontekście częstotliwości.

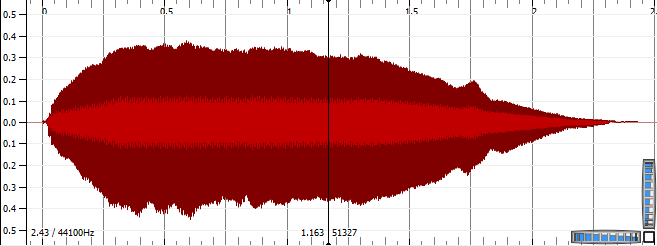
Głośność powiązana jest z natężeniem dźwięku, danym wzorem:

gdzie: N oznacza moc akustyczną, S oznacza pole powierzchni prostopadłej do kierunku w jakim rozchodzi się fala. Głośność jest subiektywnym wrażeniem, każdej osoby z osobna odczuwa wspomnianą właśnie miarę fizyczną. Faktycznie używaną miarą jest nie natężenie dźwięku, a ciśnienie akustyczne. Poziom ciśnienia akustycznego jest dane wzorem:

Czas trwania określa odstęp czasu od początku wystąpienia dźwięku do jego zakończenia. Czas trwania dźwięku podzielić można na fazy:

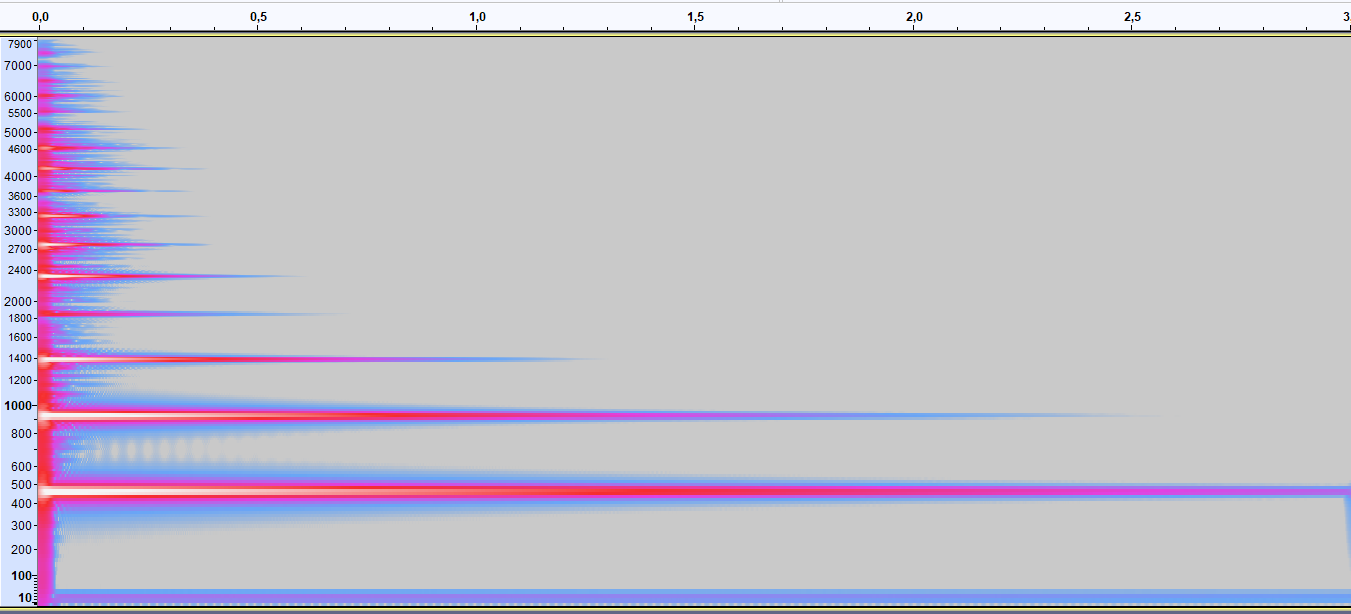
1. Faza budowania dźwięku, atak
2. Faza podtrzymania
3. Faza wybrzmiewania

Artykulacja z jaką muzyk wydobywa dźwięk oraz charakter instrumentu mają wpływ na czas trwania poszczególnych faz dźwięku.19 Na wykresie poniżej znajduje się czasowa reprezentacja dźwięku granego na trąbce. W przedziale czasu od 0 do około 0.4 sekundy zaobserwować można fazę budowania dźwięku. W przedziale czasu od około 0.4 do około 1.4 sekundy widoczna jest faza podtrzymywania, zaś od około 1.4 do 2 sekundy następuje długa faza wybrzmiewania.



Rys. 17. Przykłady obwiedni dźwięków muzycznych. Trąbka.

Barwa dźwięku ma duży wpływ na ekspresję danego instrumentu i to w jaki sposób instrument właściwie brzmi. To właśnie ta charakterystyka pozwala rozróżniać dźwięk instrumentów czy głosy. Innym wyrazem określającym barwę dźwięku jest tembr. Na tembr dźwięku wpływ ma liczba składowych tonów w dźwięku, liczba obecnych harmonicznych tonu podstawowego oraz wzajemny poziom tych właśnie harmonicznych[[32]](#endnote-32). Na rysunku poniżej znajduje się mel-spektrogram przedstawiający spektrum częstotliwości obecnych w sygnale w kontekście czasu. W tym wypadku jest to dźwięk c grany na gitarze akustycznej. Zaobserwować można wyraźne prążki ciągnące się wzdłuż osi poziomej - linii czasu.



Rys. 18. Składowe częstotliwościowe dźwięku C granego na gitarze, w tym ton podstawowy oraz składowe harmoniczne dźwięku.

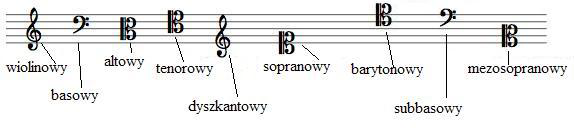
Barwa dźwięku jest wytłumiona gdy przewagę w sygnale mają niższe harmoniczne. Jeżeli mocniejsze od niskich są wyższe harmoniczne dźwięk jest ostry, narzucający się. Przykładowo dźwięk wydawany przez flet ma przeważającą część energii w paśmie głównej harmonicznej. Z tego powodu dźwięk ten jest wyjątkowo czysty i cienki. Dla porównania dźwięk klarnetu przenosi znaczną część energii w składowej głównej, jednak ze względu na podłużny i zamknięty kształt tego wytłumione są parzyste harmoniczne co daje efekt mocnych średnich tonów. W przeciwieństwie do klarnetu i fletu, fagot ma małe pokłady energii w częstotliwości głównej oraz niższych harmonicznych[[33]](#endnote-33).

Zapis nutowy

Aby zapisać i utrwalić utwory muzyczne stosuje się notację nutową. We współczesnej formie jest ona znana od przełomu XVIII i XIX. Dźwięki w postaci nut zapisuje się na pięciolinii, gdzie każda linia i przerwa między liniami oznacza kolejną wysokość dźwięku. W przypadku gdy pięć linii to za mało stosuje się dodatkowe linie dodane. Pięciolinia sama w sobie nie informuje o tym jaki dźwięk ma zostać zagrany. Aby to określić niezbędne jest wykorzystanie klucza, który skazuje na której linii pięciolinii znajduje się określony dźwięk. Najczęściej używane klucze to:

* Klucz wiolinowy - G - g1
* Klucz basowy - F - f
* klucz altowy - C - c1

Istnieje dużo więcej rodzajów kluczy, jednak trzy powyżej są najczęściej spotykanymi i najczęściej używanymi. Na rysunku 19 poniżej znajdują się wymienione powyżej klucze oraz inne mniej popularne klucze na przykład klucz sopranowy.



Rys. 19. Klucz (ang. clef) określa wysokość dźwięku na pięciolinii[[34]](#endnote-34)

Przy kluczu stoją często znaki przykluczowe określające to w jakiej tonacji dany utwór ma być grany. Znaki przykluczowe, zawsze zapisywane w takiej samej kolejności, modyfikują wysokość dźwięków podstawowych zapisanych na pięciolinii.

Przy pomocy koloru i kształtu nuty określa się jej wartość rytmiczną. Używane wartości rytmiczne to:

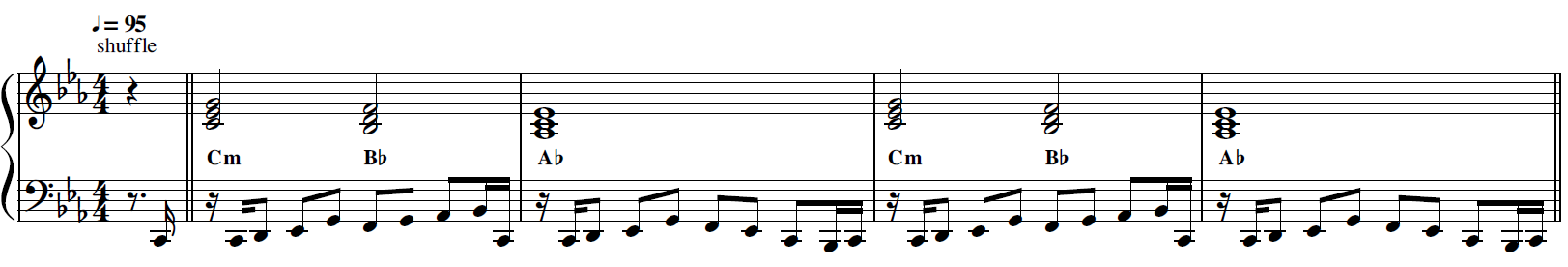
* Cała nut
* półnuta
* ćwierćnuta
* ósemka
* szesnastka
* trzydziestodwójka

Całą nutę liczy się do czterech, można ją podzielić na dwie półnuty, bądź cztery ćwierćnuty, osiem ósemek szesnaście szesnastek i trzydzieści dwie trzydziestodwójki. Z nut stworzone są takty czyli powtarzane odcinki których długość definiuje metrum. Metrum określa ilość miar danej długości w takcie oraz reguluje to w jakich miejscach utworu pojawiają się akcenty. Oprócz nut stosuje się również znaki pauzy oznaczające czasową przerwę w graniu. Podobnie jak nuty pauzy również mają określone długości. Na rysunku poniżej przedstawione są nuty oraz pauzy używane w współczesnej notacji muzycznej.



Rys 20. Od lewej cała nuta, półnuta, ćwierćnuta, ósemka, szesnastka, trzydziestodwójka i pauzy całonutowa, półnutowa, ćwierćnutowa, ósemka, szesnastka, trzydziestodwójkowa, sześćdziesięcio czwórkowa

Zapis nutowy przewiduje dużą ilość znaków która pozwala zapisać znaczną ilość charakterystyk dźwięków w utworze, takich jak tempo, akcenty czy melodia. Poniżej znajdują się dwa pierwsze takty partii klawiszowej utworu “Byłaś serca biciem” skomponowanego przez Jerzego Dobrzyńskiego, wykonywanego przez Andrzeja Zauchę.



Rys 21. Utwór “Byłaś serca biciem” skomponowany przez Jerzego Dobrzyńskiego

Pierwszą informacją którą muzyk może wyczytać z powyższych nut jest tempo utworu wynoszące 95 uderzeń na minutę (BPM).Kolejnym znakiem są dwa klucze wiolinowy i basowy. Oznacza to, że na górnej pięciolinii klucz wiolinowy wskazuje na to, że dźwięk g1 umieszczony jest na drugiej linii od dołu górnej pięciolinii. Będą to dźwięki grane przez muzyka na instrumencie klawiszowym prawą ręką. Klucz basowy na dolnej pięciolinii oznacza dźwięk F na czwartej linii od dołu. Najczęściej te nuty wykonywane są przez muzyka lewą ręką. Następnie przy obu kluczach stoją trzy znaki chromatyczne. Znaki chromatyczne są ogólnie trzy:

* bemole - obniżają dźwięk o pół tonu.
* krzyżyki - podwyższają dźwięk o pół tonu
* kasowniki - kasują nałożone wcześniej alternacje dźwięków.

Stać one mogą bezpośrednio przy kluczu utworu, jako znaki przykluczowe, obowiązują wtedy przez cały cały utwór bądź do momentu w którym następuje zmiana jego tonacji. Oprócz tego znaki chromatyczne mogą pojawiać się w przygodnych miejscach utworu. Obowiązują wtedy do końca taktu w którym wystąpiły. Z tego powodu jeżeli w obrębie jednego taktu zagrany ma być dźwięk zalterowany, a po nim oryginalny zastosowany musi zostać kasownik. W analizowanym utworze “Byłaś serca biciem” przy kluczach znajdują się po trzy bemole co wskazuje na to, że utwór jest w tonacji c-mol lub es-dur. Kolejną ważną informacją jest metrum, w tym wypadku wynoszące cztery czwarte. Oznacza to, że na jeden takt utworu przypadają cztery ćwierćnuty. Dla innego przykładu utwór zespołu Gorillaz o nazwie “5/4” ma metrum oznaczane właśnie jako . Oznacza to, że na jeden takt przypada pięć ćwierćnut. Po oznaczeniu metrum następuje ciąg nut, granych przez muzyka.

Notacja abc

W celu ułatwienia zapisu nutowego powstała notacja ABC. Jest to uproszczony sposób zapisu informacji o dźwiękach w utworze muzycznym. Ważną cechą notacji midi jest to, że w przeciwieństwie do na przykład tabulatur, oparta jest tylko o znaki z kodu ASCII. Pozwala to na łatwe generowanie zapisu przy pomocy komputera7. Dodatkowo Istnieje wiele narzędzi pozwalających generować pliki graficzne z nutami przedstawionymi w standardowej notacji[[35]](#endnote-35). Pierwszy takt utworu “Byłaś serca biciem” granego przy pomocy lewej ręki muzyka, zapisany w notacji abc wygląda następująco:

X: 1

T: Byłaś serca biciem

Q: 95

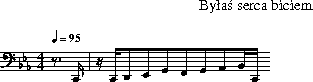
M: 4/4

L: 1/8

K: Cm clef=bass

z>C,,|z/2C,,/2D,,E,,G,,F,,G,,A,,B,,/2C,,/2

Przy pomocy narzędzia abcConverter[[36]](#endnote-36) na podstawie zamieszczonego powyżej kodu został wygenerowany poniższy zapis nutowy.



Rys. 22. Zapis utworu “Byłaś serca biciem” wygenerowany na podstawie notacji abc

Odpowiednie pola pozwalają ustawić właściwości zapisywanego utworu i w ten sposób:

* X – identyfikator utworu
* T - tytuł
* Q – tempo
* M – metrum
* L - domyślna długość nuty
* K – tonacja, wraz z określeniem używanego klucza
* Znaki reprezentujące kolejne dźwięki:
  + z – pauza
  + CDEFGAB - dźwięki z oktawy razkreślnej
  + cdefgab - dźwięki z oktawy dwukreślnej
  + , - korespondujący dźwięk z oktawy niższej
  + ‘ - korespondujący dźwięk z okrawy wyższej
  + | - rozgraniczenie między taktami

Domyślnie dźwięki mają długość określoną przez parametr L. Jeżeli dźwięk jest dwukrotnie dłuższy stosujemy mnożenie dźwięku przez dwa, w notacji midi na przykład: “C2”. Jeżeli dźwięk ma być dwukrotnie krótszy od dźwięku domyślnego stosujemy dzielenie, w notacji midi na przykład “C/2”. Znak kropki wydłużający daną nutę o połowę czasu jej trwania zaznaczany jest jako “>”.

Interfejs MIDI

Standard Midi jest protokołem komunikacyjnym używanym do transmisji informacji o granej muzyce pomiędzy użądzeniami. Określa takie paratemetry jak tempo, metrun, początki i końce dancyh dźwięków oraz to z jaką mocą naciśnięty zostałl przycisk urządzenia co wpływa na zsyntezowany ostatecznie dźwięk. Standard ten umożliwia również przesyłąnie tak zwanych wiadomości midi (eng. midi messeges). Pozwala to na ogromną wszechstronność zastosowania nie tylko w świecie audio ale takrze naprzykład podczas sterowania oświetleniem.[[37]](#endnote-37)[[38]](#endnote-38) Pliki midi mogą przechowywać informacje o wielu ścieżkach audio na raz. Format midi przypisuje wszystkim dźwiękom określony numer. Wykaz wszystkich numerów dźwięków znajduje się w tabeli 3. poniżej.

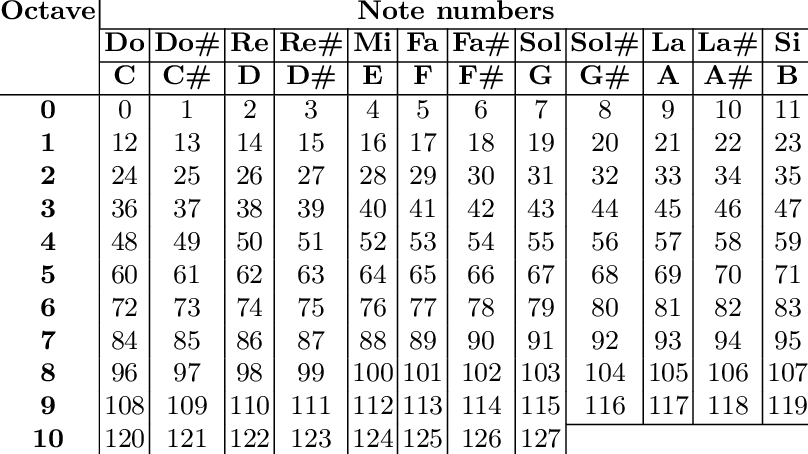


Tabela 3. Reprezentacja nut w standardzie MIDI[[39]](#endnote-39)

Powyższa reprezentacja jest dobrą metodą na zmapowanie niezrozumiałego dla komputera formatu jakim jest zapis nutowy na liczby w zakresie od 0 do 127.

Sieci Neuronowe

Wstęp

Sieci neuronowe są gałęzią sztucznej inteligencji zainspirowaną sposobem działania neuronów w ludźkich mózgach[[40]](#endnote-40). Na rysunku poniżej przestawiony jest stosunek uczenia gębokiego do uczenia maszynowego oraz do ogólnie pojętej sztucznej inteligencji.

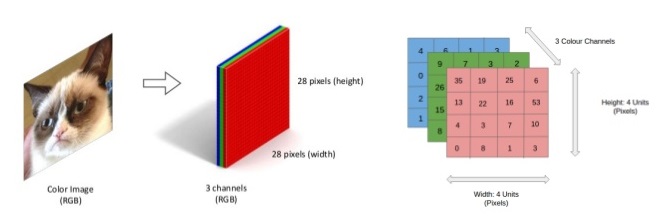


Rys. 23. Uczenie głębokie jako poddziedzina uczenia maszynowego, uczenie maszynowe wjako poddziedzina sztucznej inteligencji.[[41]](#endnote-41)

Uczenie głębokie odróżnia od uczenia maszynowego fakt, że proces uczenia głębokiego przebiegać może bez ingerencji człowieka – innymi słowy jest to proces nienadzorowany. Porównując uczenie głębokie z uczeniem maszynowym szybko nasuwającym się wnioskiem jest to, że sieci głębokie uzyskuja dużo lepsze wyniki okupione jednak ogromnie większym nakładem danych, czasem twania treningu oraz nakładem sprzętowym. Można powiedzieć, że aby móc skutecznie tworzyć rozwiązania o tą technologię niezbędny jest dostęp do dobrych, nowoczesnych kart graficznych.

Sieci neuronowe są w stanie rozpoznawać zależności i wzory w dostarczonych danych wejsciowych. Algorytmy uczenia maszynowego operują na tensorach - wielowymiarowych tablicach, gdzie dane wymiary odpowiadają określonym informacjom. Na rysunku 24. poniżej zdjęcie RGB o rozmiarze 28x28 pikseli przedstawione jako trzywymiarowy tensor – tensor trzeciego rzędu. Wymary tablicy reprezentują kolejno:

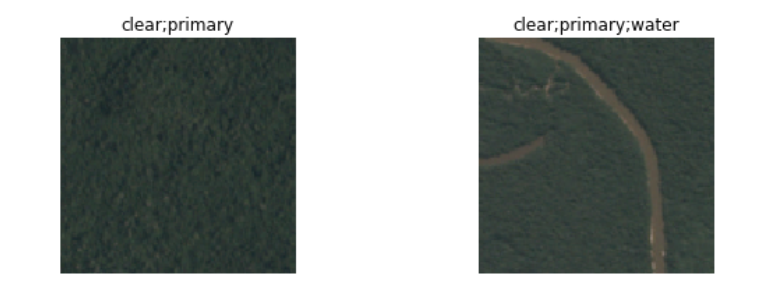
* Pierwszy wymiar – szerokość
* Drugi wymiar – wysokość
* Trzeci wymiar – kanały: czerwony, zielony i niebieski



Rys. 24 Kolorowe zdjęcie jest tensorem trzeciego rzędu.[[42]](#endnote-42)

Sieci neuronowe pozwalają na rozwiązywanie takich problemów jak:

* Klasyfikacja, Rozpoznawanie, indentyfikacja – wykrywanie obecności obiektów określonych wcześniej klas w dostarczonych danych. Umożliwia to implementację algorytmu który jest w stanie rozpoznawać na przykład ludzi na podstawie zdjeć lub oznaczać poruszające się samochody na dostarczonym materiale wideo. Technika umożliwia przypisanie do badanych danych jednej klasy bądź w rozwiniętym przypadku wielu klas. W ten sposób możliwe jest przypisywanie wielu klas do jednej danej wejściowej. Na rysunku 25 poniżej znajdują się zdjęcia satelitarne poddane klasyfikacji mającej na celu określić jakie elementy krajobrazu zjadują się na danych obrazkach.



Rys. 25 Przypisywanie wielu klas określających to co znajduje się na danych zdjęciach. [[43]](#endnote-43)

Metody klasyfikacji stosowane mogą być również identyfikacji e-maili będących spamem. Wraz z mijającym czasem sieci potrafiące dokonywać klasyfikacji znajdują coraz szersze rozwiązania w tym w technice audio, od detekcji obecności głosu w nagraniu, przez wyznaczanie ilości mówców po identyfikację głosową użytkowników. Problem transkrypcji muzyki na nuty również jest problemem klasyfikacyjnym. Z tego powodu w dalszej części niniejszego tekstu ta technika zostanie szczegółowo omówiona.

* Klastrowanie, grupowanie – znajdowanie podobieństw i różnic pomiędzy dostarczonymi danymi. Ta technika do treningu nie wymaga bezpośrednio skorelowanych par (dana, etykieta). Oznacza to, że proces trenowania może być długi, a co za tym idzie wyniki dokładniejsze. Klastrowanie wykorzystywane jest do porównywania ze sobą informacji. Na przykład znajdowanie utworów które powinny spodobać się słuchaczowi ze względu na podobieństwa między tym czego słuchać słucha, a być może chce posłuchać. Ponieważ klastorowanie pozwala na detekcję podobieństw, znajduje ono również zastosowanie podczas wykrywania anomalii, błędów w dostarczonych danych. Pozwala to na przykłąd na wykrywanie „Fake News” [[44]](#endnote-44).
* Regresja – dziedzina sieci neuronowych zajmująca się wyznaczanie zależności pomiędzy zdarzeniami które miały miesjce w przeszłości, a tymi które wydarzą się w przyszłości. Pozwala to między innymi na wyznaczanie kiedy dojdzie do usterki jakieś sprzętu lub tego czy klient zostanie na stronie www lub nie. W 2019 roku przedstawione zostały modele będące w stanie przewidywać jakość sera Gouda podczas procesu jego przechowywania[[45]](#endnote-45) oparte właśnie o regresję.

Omówienie zasady działania

Sieci neuronowe składają się z trzech wysokopoziomowych elementów nazywanych warstwami:

* wejściową
* ukrytymi
* wyjściową

Warstwa wejściowa jest pierwszą warstwą w modelu, to na nią podawane są cechy wejściowe otrzymane na podstawie dancyh treningowych. Wastwy ukryte są warstwami w których dokonywane są nieliniowe transformacje cech wejściowych. Ta warstwa znajduje się po warstwie wejściowej i przed warstwą wyjściową. Wastwa wyjściowa jest warstwą wynikową, przedstawiającą wynik wyznaczony przez sieć. Warstwy składają się z neuronów.

Neuron

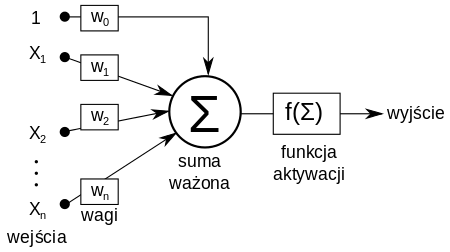
W każdej warstw sieci znajdują się neurony. Neurony na podstawie wartości pobudzenia wejściowego są aktywowane bądź nie. Harakteryzuje je fakt, że posiadają one jedno bądź więcej wejść i tylko jedno wyjście. Formalnie ralizowana przez neuron funkcja ma postać:

,

Gdzie:

* f – odpowiednia funkcja aktywacji
* – parametry / wagi dla kolejnych wejść tensora
* – aktywacje / dane wejściowe
* – parametry dla wejścia bias[[46]](#endnote-46).

Graficzny schemat neuronu przedstawiony jest na rysunku poniżej. Nazywany jest neuronem McCullocha-Pittsa i przedstawia zarówno etap sumy ważonej obliczanej dla wejść sieci, jak i etap aplijacji funkcji aktywacji dla otrzymanej wcześniej wartości.



Rys. 26 Schemat neuronu McCullocha-Pittsa

Aktywacje

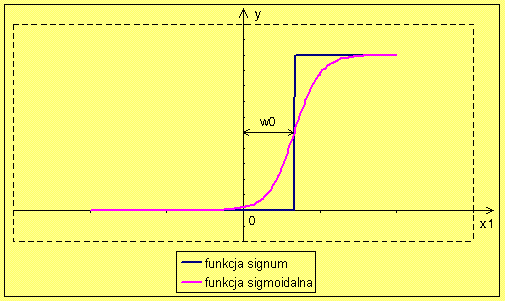
Aktywacje są to wartości które nie są howywane na stałe w modelu. Są to dane wejściowe, wyniki poszczególnych operacji czy warst. To te dane „przepływają” przez model od początku do jego końca.

Parametry - wagi

Parametry nayzwane wagami są wszelkimi informacjami przechowywanymi w modelu. Są to macierze wykorzystywane do multiplikacji przychodzących aktywacji, w ten sposób kontrolowana jest przepustowość danego połączenia między neuronami. W procesie uczenia wagi są wielokronie uaktualniane w celu znalezienia najlepszej aproksymacji końcowych wyników.

Parametry - bias

Oprócz wag parametrami sieci neuronowych są również biasy. Pozwalają one na przesówanie aproksymującej funkcji wzdłóż osi co pozwala na znalezienie jeszcze lepszego przybliżenia poszukiwanej funkcji. Autorzy „Neurony z biasem i bez biasu”46 wyjaśniają działanie biasu na podstawie neuronu o jednym wejściu i jednym wyjściu dla dwóch funkcji aktywacji: signum i sigmoidalnej, widocznych na rysunku poniżej.



Rys. 27. Funkcje aktywacji neuronu z biasem. Funkcja signum i funkcja sigmoidalna

Zastosowanie biasu pozwala na przesówanie progu aktywacji, co za tym idzie sieć jest w stanie dokłądniej przybliżyć oczekiwaną funkcję co było by nie możliwe bez zastosowania dodatkowego parametru jakim jest bias. Kosztem dodania dodatkowego parametru do modelu (co oczywiście zwiększa ,jesgo rozmiar) można uzyskać znaczny skok w uzyskiwanych rezultatach.

Przegląd funkcji aktywacji

Podczas procesu uczenia sieci neuronowej celem zwiększenia ilości szczegółów których sieć moze się nauczyć wprowadza się tak zwane nieliniowości. Są one powodowane przez funkcje aktywacje których obecność w modelu pozwala na rozwiązywanie bardziej złożonych i zaawansowanych problemów. Funkcje aktywacji działają w taki sposób aby na podstawie danych wchodzących do neuronu określić czy okreslony neuron jest aktywowany czy nie. Sieć neuronowa bez funkcji aktywacji jest de facto modelem wykonującym zwykłą regresję liniową[[47]](#endnote-47). Najczęściej stosowanymi funkcjami aktywacji są:

* Funkcja progowa unipolarna (BSF) – pierwsza funkcja przychodząca na myśl, aktywująca dany neuron po przekroczeniu określonej wartości progowej. Neuron przyjmuje dwa stany – aktywny, nieaktywny – bez stanów pośrednich. Takie podejście jest proste w implementacji i zostało wykorzystane w pierwszym modelu perceptronu, sztucznego neuronu, zaproponowanym przez Mcullocha oraz Pittsa. Funkcja znajduje zastosowanie podczas rozwiązaywania problemu podzialu danych na dwie kategorie. Z uwagi na naturę tej funkcji nie znajduje ona zastosowania jeżeli problem wymaga dzielenia dancyh na wiele klas, co jest poważnym ograniczeniem. Kolejnym ograniczeniem BSF jest zerowa wartość jej pochodnej. Wpływ pochodnych, tej i innych, funkcji aktywacji na działanie sieci neuronowej opisany jest w sekcji „Wnioskowanie i propagacja wsteczna” niniejszego tekstu.
* Funkcja liniowa (Linear) – Celem uniknięcia problemu zerowej pochodnej zastosować można funkcję liniową której pochodna jest wartością stałą. Fakt ten ma wpływ na propagację wsteczną, podczas której parametry modelu będą aktualizowane w sposób liniowy.
* Sigmoidalna funkcja unipolarna (Sigmoid) – Funkcja, o krztalcie gładkiej litery „S”, która zwraca wartości z zakresu od 0 do 1. Jej najważniejszą cechą jest fakt, że jest ona nieliniowa. Oznacza to, że wyjście z danego neuronu również będzie nieliniowe. Funkcja sigmoidalna dana jest wzorem:

Powyższa funkcja jest ciągła dla wszystkich wartości co za tym idzie jest ona całkowalna. Funkcja ta nie jest symetryczna względem osi odciętych, z tego powodu często stosowaną funkcją aktywacji jest tangens hiperboliczny. Funkcja często stosowana w klasyfikatorach. Podczas treningu może wystąpić problem zanikającego gradientu.

* Sigmoidalna funkcja bipolarna (Tanh) – bliźniacza do funkcji sigmoidalnej posiadająca takie same własności, zwracająca wartości z zakresu od -1 do 1. Fakt, że jest ona symetryczna wokół osi odciętych sprawia, że wejście do kolejnej warsty może być różnego znaku. Podobnie jak funkcja sigmoidalna Tanh jest ciągły w całym zakresie co pozwala na wyznaczenie jej pochodnej. Tanh dany jest wzorem:
* Jednostronnie obcięta funkcja liniowa (ReLU) – funkcja dana wzorem:

,

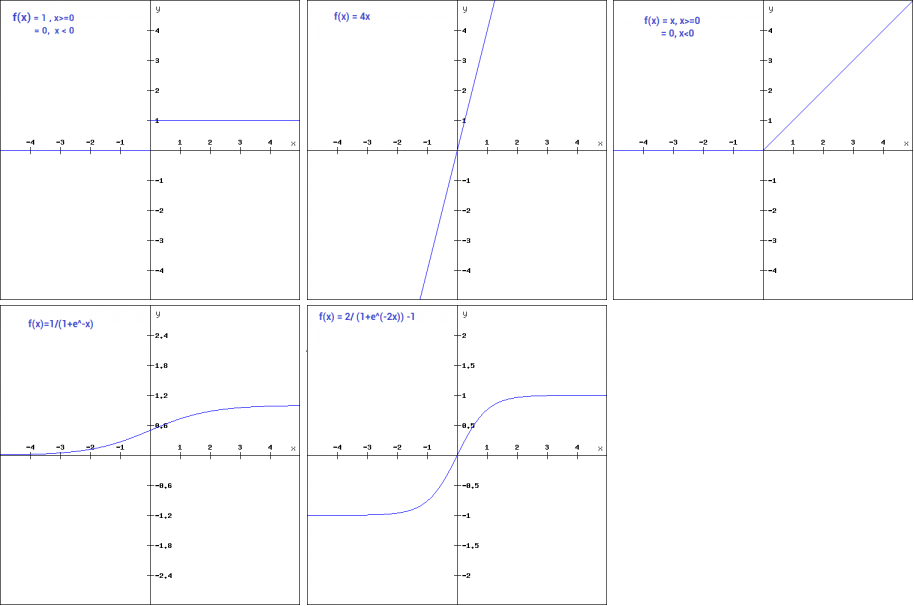
jest to nielinowa funkcja która dla waartości mniejszych niż zero nie akrywuje neuronu. Wpływa to na większą wydajność obliczeniową w porównaniu do funkcji sigmoidalnej oraz tanh. Jest to obecnie najczęsciej wykorzystywana funkcja aktywacji, powinna ona być wykorzystywana jedynie w warstwach ukrytych modelu. Podczas wykorzystania tej funkcji dochodzić może do powstawania tak zwanych martwych neuronów – takich które przestają się uczyć podczas treningu. Aby temu zapobiec stosuje się wariację funkcji ReLU o nazwie Leaky ReLU.

* Softmax - w przeciwieństwie do opisywanych wcześniej funkcji aktywacji Softmax używany jest do rozwiązywania problemów w których występuje wiele klas kandydatów. Operacja dana jest wzorem:

Dla i = 1,..,K

Ilość neuronów w warstwie musi być taka sama jak ilość klas z których następuje wybór. Funkcja softmax pozwala wyznaczyć poziom aktywacji tych neuronów na podstawie ich wartości wejściowych. Poziom aktywacji neuronów po zaaplikowaniu funkcji sigmoidalnej zawsze sumuje się do 1. Wynik softmax mówi o tym z jakim prawdopodobieństwem określona dana wejściowa przynależy do danej klasy[[48]](#endnote-48).

Opisane powyżej funkcje aktywacji przedstawione są na poniższym rysunku numer 28.



Rys. 28 Wykresy wybranych funkcji aktywacji. U góry od lewej: BSF, Funkcja liniowa, ReLU.

Na dole od lewej: Sigmoid, Tanh

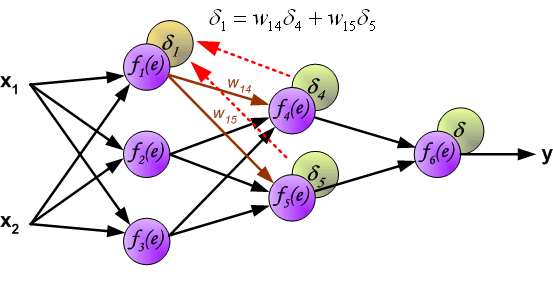
Wnioskowanie i propagacja wsteczna

Proces trenowania sieci neuronowej polega na iteracyjnej aktualizacji parametrów we wszystkich neuronach, włącznie z tymi w warstwach ukrytych modelu, w celu minimalizacji funkcji straty – błędu aproksymacji modelu. Algorytm umożliwiający umożliwiający na zautomatywzowane wyznaczanie wag w neuronach nazywany jest popagacją wsteczną. Aby rozpocząć trening niezbędny jest zbiór uczący składający się z wielu próbek skłądających się z dwóch tensorów: wejściowego i wyjściowego (ang. Target), będącego faktyczną wartością jaką przybliżać powinna trenowana sieć. Podczaj jednej iteracji treningu sieci neuronowej należy:

* Załadować tensor wejściowy oraz wzorcowy do pamięci
* Wyznaczyć tensor wyjściowy, odpowiedź sieci dla określonego tenosora wejściowego.

Każdy neuron wykonuje operację mnożenia aktywacji przez wagi, sumowania oraz wyznacza wyjście tensora przy pomocy funkcji aktywacji. W ten sposób następuje propagacja „do przodu” (ang. Feed-forward)

* Porównać tensor wyjściowy ze znanym wcześniej tensorem wzorcowym i obliczenie wartości funkcji straty.
* Dokonać propagacji wstecznej obliczonego błędu. Dla każdego neurona obliczany jest iloczyn przychodzącego „z przodu” błędu z wychodzącymi z niego wagami. Jeżeli wartość błędu „przychodzi” do neuronu z kilku miejsc wyznaczone iloczyny są sumowane, przykład takiej operacji przedstawiony jest na rysunku 29 poniżej.



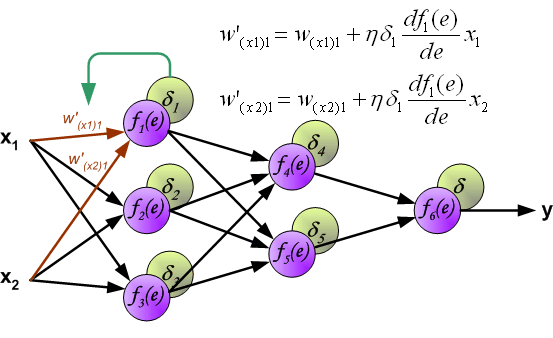
Rys. 29 Jeśli błędy propagowane pochodzą z kilku neuronów, są dodawane.

* Nowe wartości wag obliczane są w oparciu o poprzednią wartość wag, gradient oraz współczynnik uczenia. Operacja ta opisywana jest wzorem:

Gdzie:

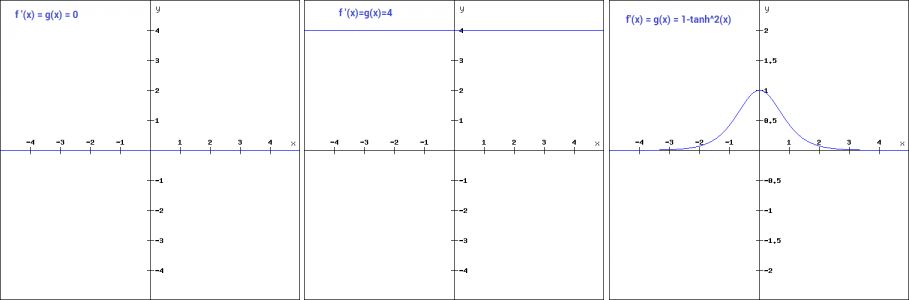
* + w’ – nowa zakutalizowana waga
  + w – poprzednia wartość wagi
  + - współczynnik uczenia (ang. Learnig Rate)
  + – pochodna funkcji aktywacji

Aktualizacja wag w modelu przedstawiona jest na rysunku numer 30 poniżej.



Rys. 30. Propagacja wsteczna, obliczanie uaktualnionych wartości parametrów[[49]](#endnote-49).

Z uwagi na to, że podczas propagacji wstecznej wykorzystywana jest pochodna funkcji aktywacji ważne jest aby unikać takich funkcji aktywacji które mają zerową bądź stałą pochodną. W przypadku gdy pochodna funkcji jest zerowa, na przykład dla BSF, obliczony gradient również będzie zerowy, a co za tym idzie parametry danego tensora nie będą aktualizowane. Aby uniknąć tego problemu zastosować można funkcję liniową której pochodna jest stałą. W takim wypadku wagi będą aktualizowane zawsze o taką samą wartość – pozwala to na trenowanie modeli do prostych zadań jednak uniemożliwia rozpoznawanie bardziej zaawansowanych i złożonych zależności między danymi. W celu uniknięcia dwóch opisanych powyżej problemów stosyje się takie funkcje aktywacji których pochodna zależna jest od wartości wchodzącej do neuronu. Na przykład nieliniowe funkcje sigmoid i tanh mają również nieliniowe pochodne których wartość zależna jest od danych wejściwych. Takie podejście pozwala na uczenie modelu zaawansowanych prawidłowości. Wykresy wybranych funkcji aktywacji przedstawione zostały na rysunku 31.



Rys. 31. Pochodne wybranych funkcji aktywacji, od lewej: pochodna BSF, pochodna funkcji liniowej f(x) = 4x, pochodna funkcji aktywacji tanh.

Hiperparametry

Hiperparametry to wartości które nie są wypracowywane w procesie treningu, lecz ich wartości ustawiane są przed treningiem. Dobór hiperparametrów ma wpływ między innymi na szybkość uczenia, dokładność uzyskanego przybliżenia. W zależności od tego jak bardzo zaawansowany jest lagorytm uczenia i model hiperparametrów może być zero, kilka i Hiperparametry podzielić można na dwa rodzaje. Te dotyczące bezposrednio architektury modelu:

* Ilość warstw ukrytych
* Stosowana funkcja aktywacji
* Dropout
* Sposób inicjalizacji wag

oraz te związane z procesem uczenia, treningu modelu:

* Współczynnik uczenia
* Liczba epok
* Rozmiar batch

Ilość warstw ukrytych

Ważnym jest aby odpowieednio dobrać ilość warstw z których zbudowany jest model. W przeszłości, w oparciu o „Uniwersalne twierdzenie o aproksymacji”, stosowano tylko jedną warstwę ukrytą. Zgodnie z teorią taki model jest w stanie rozwiązywać dowolnie złożone problemu. Niestety w praktyce uzyskanie dobrej aproksymacji zaawansowanych problemów przy wykorzystaniu tylko jedej warstwy jest zadaniem trudnym i obecnie nieosiągalnym. Model który składa się z jednej warstwy ukrytej będzie w stanie aprksymować jedynie proste funkcje liniowe. Jedna warstwa ukyrta umożliwia aproksymację funkcji która jest rzutowaniem z ciąłej skończonej przestrzeni do drugiej skończonej przestrzeni. Dwie warstwy ukryte w połączeniu z odpowiednimi funkcjami aktywacji pozwalają na przybliżanie dowolnie gładkich odwzorowań z dowolnie dobrą dokładnością. Więcej jak dwie warstwy ukyte pozwalają na uczenie modelu zaawansowanych reprezentacji na podstawie cech – takie podejście można nazwać zautomatyzowaną inżynierią cech. [[50]](#endnote-50)

Stosowana funkcja aktywacji

Stosowana w modelu funkcja aktywacji może pozytywnie wpłynąc na wszystkie etapy uczenia maszynowego, zarówno propagacji „do przodu” ale również propagacji wstecznej. Współczesne funkcje aktywacji wprowadzające nieliniowości pozwalają modelom na uczenie się bardziej zaawansowanych i skomplikowanych zależności między danymi. Zastosowanie funkcji o nieliniowej pochodnej pozwala na efektywniejszą aktualizację wag modelu, a co za tym idzie dokładniejszą aproksymację.

Dropout

Podczas trenowania sieci neuronowej występuje tak zwane zjawisko overfittingu. Występuje ono wtedy gdy stosunkowo duży model uczony jest na stosunkowo małym zbiorze uczącym. Wtedy pomimo dobrych wyników treningu model nie radzi sobie z danymi z zewnątrz zbioru uczącego ponieważ nauczył się dokładnie aproksymować zjawiska zachodzące w zbiorze uczącym jednak na za małym poziomie abstrakcji. Aby uniknąć tego zjawiska stosuje się tak zwany dropout. Jest to tymczasowe usunięcie losowo wybranych neuronów z modelu wraz ze wszystkimi połączeniami jakie dane neurony miały. W ten sposób przy każdej kolejnej iteracji treningu wykorzystywany jest delikatnie zmodyfikowany model. Dzięki takiemu podejściu model uczy się korygować decyzje wypracowane w poprzednich warstwach modelu. Jest to rówież wprowadzenie kolejnego szumu do procesu treningu, a zwięszkenie nieliniowości modelu ma pozytywny wpływ na jego zdolność do uczenia.

Dropout zastosowany może być w większości warstw z których budowane są wpółczesne modele, to znaczy warstw liniowych, splotowych, rekurencyjnych i innych. Stosuje się go dla poszczególnych warstw i wprowadza hiperparametr nazywany w języku angielskim „dropout rate”. Najczęściej jego wartość ustawiana jest jako 0.5 i oznacza prawdopodobieństwo odrzucenia określonego neuronu.[[51]](#endnote-51)

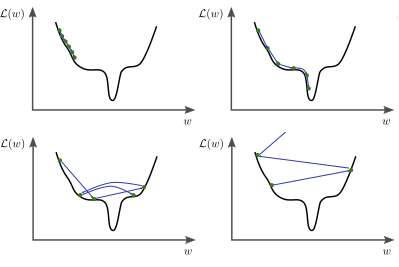
Sposób inicjalizacji wag

Istenieje co najmniej kilka sposobów inicjalizacji wag. Najprostrzy z nich to inicjalizacja paramaterów samymi zerami. Problemy które rodzą się w tym przypadku są dwa. Pierwszy z nich to, że wagi zainicjalizowane zerami w kolejnych krokach będą aktualizowane takimi samymi wartościmi. Oznacza to, że taki model będzie zachowywał się jak model liniowy. Drugim problemem jest to, że zaincjalizowany w ten sposób nie będzie miał żadnego, ani pozytywnego, ani negatywnego, wpływu na proces uczenia. Aby polepszyć ogólny wynik sieci stosuje się inicjalizację losową parametrów. Takie podejście pozwala uniknąć dwóch problemów opisanych przed chwilą. Mimo to, podejście to nie ma swoich wad. Zainicjalizowanie parametrów wartościami skrajnie małymi i dużymi może skutkować bardzo wolnym treningiem. W wypadku wykorzystania funkcji sigmoidalnej dla dużych i małych aktywacji wyznaczony gradient będzie mały, więc stopień zmiany wag będzie znikomy. Aby zapobiec problemowi zanikającego gradientu stosuje się tak zwane inicjalizacje He i Xaviera. Metoda He polega na zastosowaniu mnożnika o wartości:

gdzie n – ilosć wag w warstwie[[52]](#endnote-52). Zainicjalizowane losowo parametry mnoży się przez opisany powyżej mnożnik. Powoduje to, że wagi po inicjalizacji nie przyjmują skrajnie małych i dużych wartości. Zastosowanie takiego podejścia wyraźnie poprawia ogólny wynik uzyskiwany przez trenowane modele oraz przyspiesza czas trwania treningu. Inizjalizacja Xaviera stosowana jest gdy używaną funkcją aktywacji jest tangens hiperboliczny. Mnożnik wtedy przyjmuje wartość:

Współczynnik uczenia

O sieci neuronowej można myśleć jak o bardzo dużym wzorze który ma bardzo dużą ilość parametrów. Wzór ten po dobraniu odpowiednich parametrów potrafi określać na przykład czy na danym zdjęciu jest pies. Proces trenowania polega na dobraniu takich parametrów modelu aby znaleźć jego minimum globablne. Pomaga w tym współczynnik uczenia który określa jak duży krok ma zostać wykonany w kolejnej iteracji algorytmu propagacji wstecznej. Gdt wpółczynnik uczenia jest za mały trening może trwać bardzo długo, a znalezione minimum może okazać się jedynie minimum lokalnym. W wypadku zbyt dużego współczynnika uczenia konwergencja do minimum może zupełnie nie nastąpić. Należy więc dobrać ten współczynnik tak by nie był mały. Różne przypadki dobrania współczynnika uczenia zaprezentowane są na rysunku numer 32 poniżej.

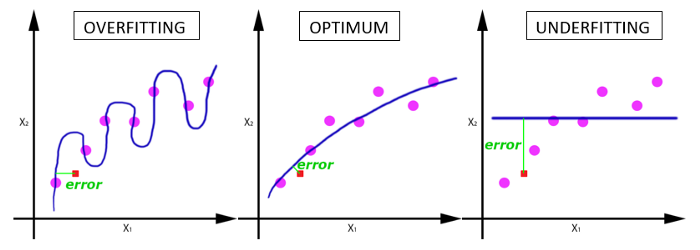


Rys. 32 Współczynnik uczenia. Od góry od lewej: Za mały współczynnik uczenia – wolny trening, dobry współczynnik uczenia; U dołu od lewej: duży współczynnik uczenia – problemy ze znalezieniem minium; Za duży współczynnik uczenia, eksplozja współczynników[[53]](#endnote-54).

Powrzechnie stosowaną techniką jest zmniejszanie współczynnika uczenia wraz z malejącą wartością funkcji straty. Pozwala to na jeszcze dokładniejsze określenie minimum funkcji i zapobiega ciągłemu omijaniu właściwiego minimum.

Liczba epok i rozmiar batch

Z uwagi na to, że proces uczenia sieci neuronowej jest procesem iteracyjnym nie wystarczy zaktyalizować wag modelu tylko raz – taka sytuacja nazywana jest underfittinigiem przedstawionym na rys.33 poniżej. Operację aktualizacji wag wykonuje sie wielokrotnie tak by uzyskać jaknajlepszy wynik. Niestety w wypadku zbyt dużej ilości epok możliwe jest doprowadzenie do tak zwanego overfittingu – stanu modelu w którym umie on dobrze aproksymować odpowiedź dla danych którymi był uczony, ale nie radzi sobie z innymi, zewnętrznymi danymi. Ważne jest by model został wyuczony w sposób optymalny. Na rystunku 33 poniżej przedstawione są trzy opisane stany.



Rys 33. Wraz ze wzrostem liczby epok zwiększa się liczba zmian wagi w sieci neuronowej, a krzywa przechodzi od underfittingu przez optymalną do overfittingu.

Optimizer

Optimizery są obiektami łączącymi funkcję straty z tym w jaki sposób powinny zostać zaktualizowane wagi. Przy ich pomocy wyznaczane jest minimum lokalne funkcji. Optimizery na podstawie wyników funkcji straty są w stanie określić w jakim kiernunku znjajduje się minimum funkcji i dobrać tak parametry modelu by bliżej tego minimum się znaleźć. Najbardziej popularnym optimizerem jest metoda gradientu prostego, sposób działania tego optimizera opisany został w niniejszej pracy w podrozdziale „Wnioskowanie i propagacja wsteczna”. Bardziej zaawansowanym optimizerem jest Adam (ang. Adaptive Momentum Estimation) wykorzystujący poprzednie gradienty do wyliczenia kolejnego gradientu. Oparty jest on o momentum, czyli parametr mający na celu zwiększanie skoku w kierunku minimum globalnego z pominięciem minimów lokalnych.

Funkcje Straty

Funkcje straty służą do określania błędu aproksymacji modelu.Na podstawie uzyskanego błędu w danej iteracji wagi aktualizowane są w taki sposób aby wynik funkcji straty się zmniejszył. Istnieje wiele rodzajów funkcji strat, należy starannie dobrać je do rozwiązywanego problemu – tak by była wstanie operować na tensorach odpowiednich krztałtów. Podobnie w zależności od wejścia jakie przyjmuje funkcja straty należy opdpowiednio dopasować format danych przychodzących z warstwy poprzedzającej. Poniżej opisane są niektóre z funkcji strat, często stosowane do rozwiązywania problemów związanych z regresją liniową, klasyfikacją binarną oraz klasyfikacją wieloklasową.

<https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>

Funkcja straty dla regresji

Najpopularniejszą funkcją wykorzystywaną do obliczania błędu modelu rozwiązującego problem regresji jest błąd średniokwadratowy (MSE). Jest on dany wzorem:

,

Gdzie: n – ilość próbek; ŷ – wartość aproksymowana; y – wartość rzeczywista.

Błąd średniowadratowy skonstuowany jest w taki sposób aby zwaracać dużą wartość błędu dla mocno różniących się wyników, nie jest ważne czy odchylenie było w „górę” czy w „dół” – dzięki podniesieniu wyniku do kwadratu znak jest ignorowany. MSE jest dobrą metryką błędu gdy dane mają rozkład gaussowski.

Innymi stosowanymi dla regresji funkcjami straty są: Logarytmiczny błąd średniokwadratowy (MSLE) oraz Średni Błąd Bezwzględny (MAE). Pierwszy z nich stosowany jest gdy aproksymowane wartości są mocno rozrzucone, a przybliżenie nie musi być aż tak dokładne. Jest on mniej „surowy” niż MSE. MAE stosowany jest gdy dane nie mają rozkładu w pełni gaussowskiego, a posiadają pewne wartości odstające.

Funkcja straty dla klasyfikacji binarnej

Klasyfikacja binarna opisana dokładnie wcześniej w tym tekście, koncentruje się na przypisaniu danych wejściowych do dwóch klas. Częstą praktyką jest przypisanie wartości „1” przynajleżności do pierwszej z klasy, zaś wartości „0” przynależności do drugiej klasy. W ten sposób okresla się prawdopodobieństwo przynależności do klasy „1”. Aby obliczyć to prawdopodobieństwo wykorzystuje się funkcję entropii skrośnej. Wykorzystanie tej funkcji wymaga, aby ostatnia warstwa była pojedynczym neuronem o sigmoidalnej funkcji aktywacji. Istnieją inne funkcje straty stosowane podczas rozwiązywania problemów klasyfikacji binarnej między innymi: funkcja straty Hinge’a i kwadratowa funkcja straty Hinge’a.

Funkcja straty dla klasyfikacji wielo-klasowej

Podczas rozwiązywania w których danym wejściowym przypisuje się wiele klas wykorzystuje się funkcję entropii skrośnej. Kolejnym klasom przypisuje się indeksy od „0” do „n. Entropia skrośna pozwala na wyznaczenie średniej odległości pomiędzy wektorem będącym aproksymacją, a wektorem wzorcowym.dla wszystkich klas. Aby zastosować entropię skrośną podczas treningu należy upewnić się, że ostatnia warstwa modelu ma taką ilość neuronów jaka jest ilość klas do przewidzenia oraz aby wykorzystywana funkcja aktywacji to sigmoid. Pomimo, że wieloklasowa entropia wskrośna jest najpopularniejszą i najszerzej stosowaną funkcją straty stosowane są również inne na przykład rzadka entropia wskrośna czy dywergencja Kullbacka-Leiblerta.[[54]](#endnote-55)

Metryki

Podczas treningu należy znaleźć sposób na ewaluację jego postępów. Wykorzystywane do tego mimary to:

* Precyzja (ang. Precission)
* Czułość (ang. Recall)
* Dokładność (ang. Accuracy)
* Swoistość (ang. Specifity)

Oparte są one o założenie, że stworzony model może dawać dodatnie oraz ujeme wyniki. W ten sposób jesteśmy w stanie podzielić odpowiedzi otrzymane przez model na cztery rodzaje:

* Prawidzie dodatni (TP)
* Prawdziwie ujemy (TN)
* Fałszywie dodatni (FP)
* Fałszywnie ujemny (FN)

Zastosowanie czterech przypadków zobrazowane jest w tabeli 4. przedstawiającej przykładowe zastosowanie dla problemów medycznych. Jest to tak zwana macierz pomyłek dla problemu klasyfikacji mającego określać czy dany pacjent jest chory czy zdrowy. Jako prawdziwie dodatni oznaczeni są ci pacjenci którzy są chorzy i tak zostali zdiagnozowani. Jako ludzie fałszywie dodatni oznaczeni są ci pacjenci którzy pomimo bycia zdrowym zdiagnozowani zostali jako chorzy - w statystce nazywane jest to błędem pierwszego rodzaju. Osoby negatywnie dodatnie to takie które są chore pomimo, że zdiagnozowane zostały jako zdrowe – w statystyce nazywane jest to błędem drugiego rodzaju. Osoby które są zdrowe i faktycznie tak zostały zdiagnozowane oznaczane sa jako prawdziwie ujemne.

Tabela 4. Tablica pomyłek na przykładzie problemu medycznego

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Wynik predykowany | |
| osoba chora = tak | osoba chora = nie |
| Wynik rzeczywisty | osoba chora = tak | TP | FN |
| osoba chora = nie | FP | TN |

Dokładność

Dokładność jest stosunkiem wszystich poprawnie zaklasyfikowanych przypadków do wszystkich przypadków.

Precyzja

Jako prezycję oblicza się stosunek poprawnie zidentyfikowanych pozytywnych przypadków w stosuku do wszystkich pozytywnie zaklasyfikowanych przypadków.

Czułość

Czułość to miara poprawnie oznaczonych przypadków pozytywnych w stosunku do wszystkich pozytywnych przypadków.

Swoistość

Stosunek poprawnie ujemnych przypadków do ogólnej ilości ujmenych przypadków nazywana jest swoistością.

Miara F1

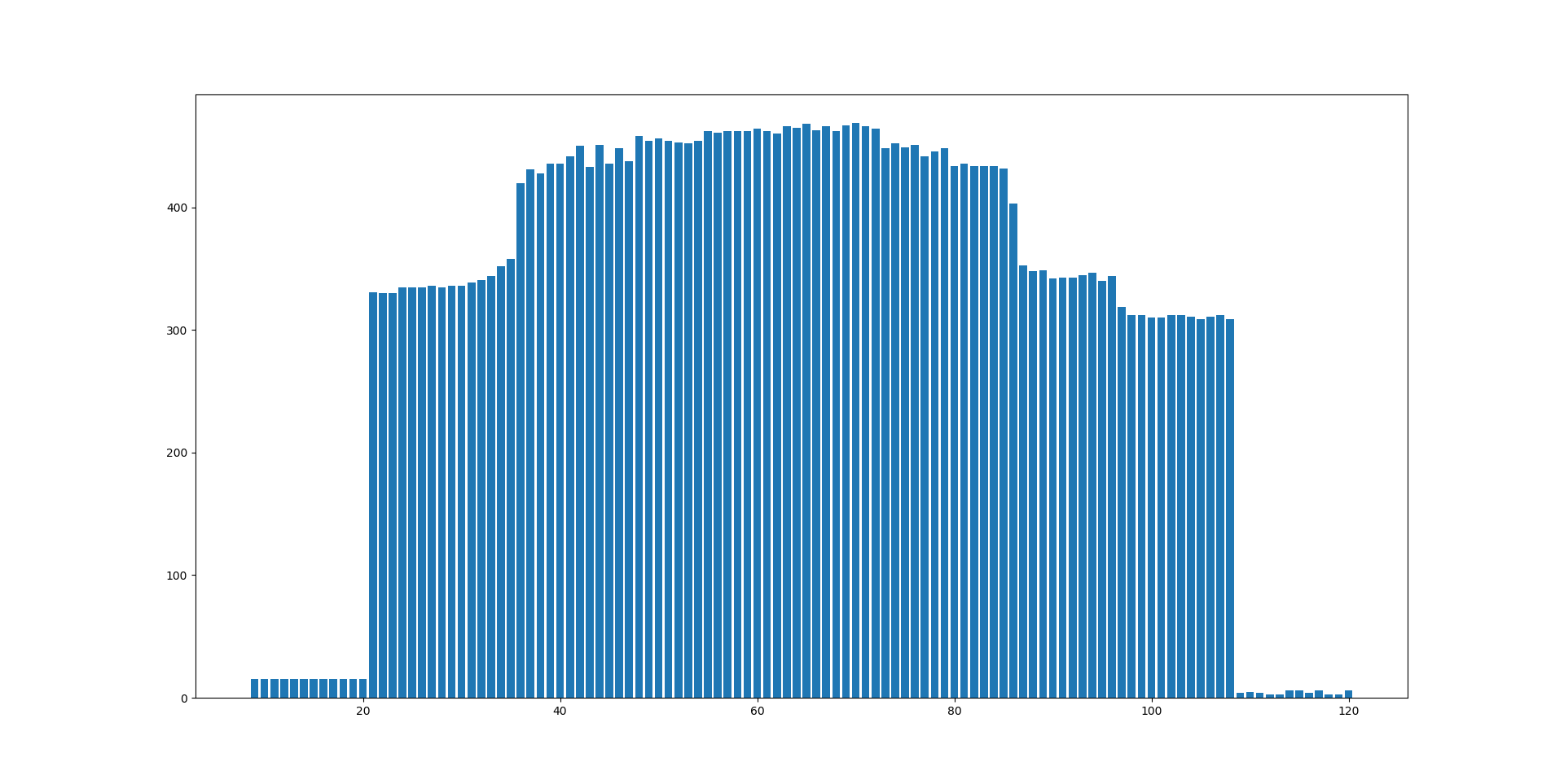
Miara F1 (ang. F1-score) - będąca naturalnym następstwem wprowadzenia opisanych w porzpednich sekcjach metryk jest średnią harmoniczną czułości oraz precyzji. Miara ta jest stosowana gdy dane nie są rozłożone w sposób niezbalansowany.

Najlepszą wartością miary F1 jest 1, zaś najgorszą jest 0.

Generowanie danych uczących

Baza wiedzy

Do przeprowadzenia treningu niezbędna jest odpowiednia baza wiedzy składająca się z dużej ilości dzwięków w tym wypadku granych na gitarze. Dźwięki te powinny być odpowiednio oznaczone w taki sposób aby móc dokładnie określić w jakich interwałach czasu dźwięk jest a w jakich go nie ma. Zbiór uczący Nsynth (ang. The NSynth Dataset) zawiera w sobie 305979 wygenerowanych dźwięków, dla instrumentów z 11 rodzin w tym gitar: akustycznej , elektrycznej oraz wygenerowanej w sposób syntetyczny[[55]](#endnote-56). Dla samych gitar ilość próbek to 35423 różnych dźwięków. Histogram przedstawiający rozkład próbek dla gitary względem klas midi przedstawiony jest na rysunku 34 poniżej



Rys. 34 Rozkład próbek dla dźwięków gitary wględem wysokości dźwięków.

Są one czterosekundowymi plikami muzycznymi, gdzie każdy plik to osobny dźwięk oznaczony indeksem midi. Na potrzeby transkrypcji muzyki wybrane zostały fragmenty grane na gitarze w zakresu midi od 40 do 84 włącznie. Pomijane są prawie niesłyszalne dźwięki, które po ocenie subiektywnej nie zdawały się brzmieć jak grane na gitarze, aby nie wprowadzać mylnych informacji na wejście sieci neuronowej.

Generator ciągów dzięków

Zaimplementowany został program który tworzy dłuższe ciągi dzwięków wykorzystując do tego krótkie fragmemty ze zbioru Nsynth. Program pozawala w prosty sposób wygenerować ciągi dźwięków będące następnie podstawą do wyekstrachowania cech i stworzenia macierzy wejściowej oraz docelowej. Generator pozwala na generowanie ciągów zarówno pojedynczych dźwięków jak i wielu nakładających jednocześnie, zaczynających na początek taktu bądź z odpowiednim przesunięciem.

Sposób działania

Działanie generatora sterowane jest przez plik konfiguracyjny który przygotować i dostarczyć musi użytkownik. Dokładny opis pliku konfiguracyjnego znajduje się w dalszej części niniejszego tekstu. Początkowo generator skanuje odpowiednie katalogi w poszukiwaniu krótkich próbek dźwięków. Ścieżki do wszystkich plików trzymane są w specjalnie przygotowanej strukturze danych będącej połączeniem słownika oraz listy opartej o mechanizm round-robin. Taki sposób przetrzymywania danych pozwala na łatwy dostęp do dźwięków z danej kategorii midi i zapobiega szybkiemu wyczerpaniu dostępnych fragmentów utworów. Schemat omawianej struktury danych przedstawiony jest poniżej:

sample\_dict = {

„midi\_key\_x” : [ round-robin list ],

„midi\_key\_y” : [ round-robin list ],

⋮

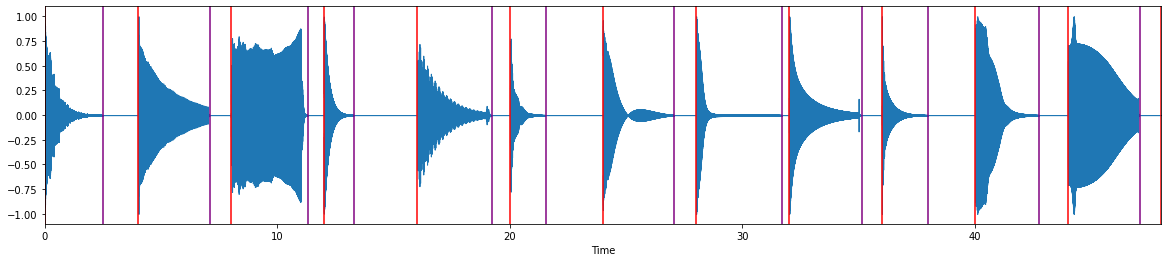
„midi\_key\_z” : [ round-robin list ]

}

Następnie ze struktury danych wybierane są losowe dźwięki, które są do siebie dodawane a następnie konkatenowane. W ten sposób tworzony jest ciąg o określonej w pliku konfiguracyjnym długości.

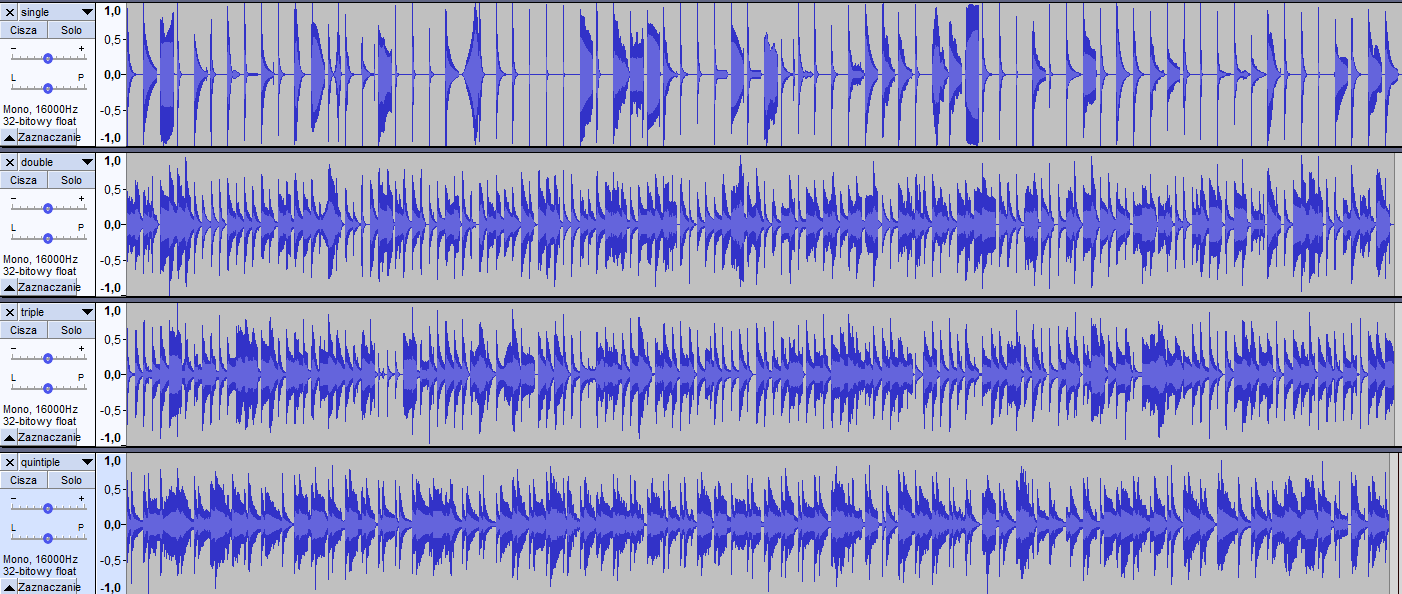
Ważne jest aby generator oprócz tworzenia sanych ciagów muzycznych śledził również precyzyjnie miejsca w których dane dźwięki się zaczynają i kończą. Dla każdej próbki która ma zostać dodana do ciągu dźwięków w sposób automatyczny wyznaczany jest początek oraz koniec trwania dźwięku. Informacja o tym w jakim punkcie czasu zaczyna się i kończy dany dźwięk przychowywana jest w pliku w formacie csv. Postać czasowa przykładowego pliku z zaznaczonymi punktami rozpoczęcia i zakończenia dźwięków przedstawiona jest na rysunku 35.

Rys. 35. Postać czasowa sygnału audio z zaznaczonymi punktami rozpoczęcia (kolor czerwony) oraz zakończenia (kolor fioletowy).



Przykładowe wygenerowane ciągi

Generator jest w stanie generować różnorodne ciągi składające się z losowo wybrancych dźwięków. Pozwala na nakładanie dźwięków na siebie oraz przesuwanie ich w czasie – tak aby uzyskać efekt nachodzących na siebie dźwięków tak jak ma to miejsce w faktycznej, rzeczywistej muzyce. Na rysunku 36 poniżej znajdują się różne ciągi dźwięków uzyskane przy pomocy opisywanego generatora.



Rys. 36 Postać czasowa sygnałów od góry: Pojedynczy dźwięk na początku taktu, podwójne dźwięki na początku oraz z połowie taktu, potrójny dźwięk na początku oraz w połowie taktu, pięć dźwięków jednocześniem na początku i w połowie taktu.

Konfiguracja generatora

Generator obsługiwany jest przy pomocy pliku konfiguracyjnego w formacie json podawanego jako parametr wejściowy programu. Poniżej przedstawiona jest lista wszystkich kluczy które powinny znaleźć się w pliku konfiguracyjnym aby generowanie ciagów dźwięków wykonane zostało poprawnie:

* dry\_data\_paths
  + ścieżka do fragmentów dźwięków wykorzystanych następnie do tworzenia dłuższych ciągów
  + przykładowa wartość: „E:/Dataset/nsynth/guitar/audio/\*”
* sr
  + częstotliwość próbkowania
  + przykładowa wartość: 16000
* folds
  + ilość ciągów która ma zostać wygenerowana
  + przykładowa wartość: 100
* midi\_range
  + zakres dźwięków z których tworzony ma być sygnał wyjściowy
  + przykładowa wartość: [40, 84]
* version
  + wersja
  + przykładowa wartość: 0.3
* chord\_name
  + nazwa generowanych ciągów muzycznych
  + przykładowa wartość: „quintiple”
* working\_directory
  + Folder docelowy
  + przykładowa wartość: „E:/Dataset/chords”
* audio\_filename
  + przedrostek dla wygenerowanych plików audio
  + przykładowa wartość: „data\_fold”
* meta\_filename
  + przedrostek dla wygenerowanych plików meta
  + przykładowa wartość: „meta\_fold”
* meta\_directory
  + Folder w którym przechowywane mają być pliki meta
  + Przykładowa wartość: “meta”
* audio\_directory
  + Folder w którym przechowywane mają być pliki audio
  + Przykładowa wartość: “audio”
* overlap\_list
  + Lista określająca ile dźwięków jednocześnie może być nałożonych na siebie. Dla każdego kolejnego taktu losowana jest jedna liczba z listy w oparciu o prawdopodobieństwo przechowywane w parametrze overlap\_prob
  + Przykładowa wartość: [3, 4, 5]
* overlap\_prob
  + lista określająca prawdopodobieństwo wylosowania danej ilości dzięków nałożonych na siebie w danym takcie
  + Przykładowa wartość: [5, 5, 90]
* zero\_padding
  + parameter określający ilość przesunięć w sygnale. Jeżeli będzie miał wartość zero, to wszystkie dźwięki rozpoczną się na początek taktu. Każda dodana do listy wartość to dodatkowa nałożona na sygmał warstwa dźwięków przesunięta o określoną ilość próbek
  + Przykładowa wartość: [0, 16000]
* max\_real\_time\_mins
  + Maksymalny czas trwania pojedynczego ciągu w minutach
  + Przykładowa wartość: 5

Sterowanie generatorem z poziomu pliku konfiguracyjnego ułatwia dokonywanie oraz śledzenie wszelkich zmian. Generator przechowuje konfigurację która została wykorzystana do wygenerowania określonego typu danych.

Dodatkowo opisany plik konfiguracyjny może zostać rozszerzony o dodatkowe klucze które pozwolą na ekstrakcję cech z plików audio, niezbędną do dalszego treningu.

Format danych wejścia i wyjścia

W celu ekstrakcji informacji o częstotliwościach w sygnale na wygenerowanych przez generator wykorzystana została transformata FFT o parametrach:

* szerokość ramki: 1024 próbki
* przesunięcie: 512 próbek
* funkcja okna: Hann’a
* rozmiar okna: 1024 próbki

Wynik tej operacji przemnarzany jest przez bank filtrów melowych o czterdziestu pasmach co daje spektrogram w skali melowej. Otrzymany melspektrogram jest macierzą o dwóch wymiarach. Pierwszym z tych wymiarów jest czas, drugi to częstotliwości w skali melowej, w danym momencie czasu. W ten sposób dla pliku o częstotliwości próbkowania wynoszącej 16000 Hz wiersze macierzy wynikowej odpowiadają energii w sygnale zgromadzonej w odcinku czasu wynoszącym około 64 milisekund. Tensor podawany na wejście składa się z szesnastu takich wierszy i ma wymiar (1, 16, 40), gdzie:

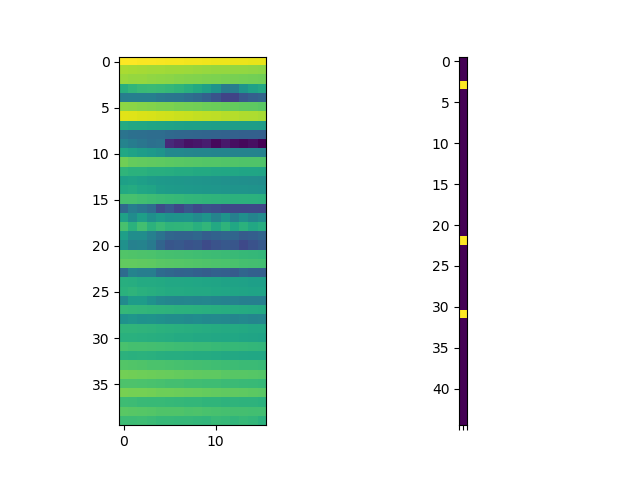
1. Ilość kanałów w sygnale audio
2. Wymiar czasu
3. Współczynniki melcepstralne

Oznacza to, że jeden tensor obejmuje zakres czasowy o długości:

Dla każdego z przetwarzanych plików dźwiękowych analizowany jest odpowiadający mu plik z metadanymi. Na podstawie informacji o momentach rozpoczęcia i zakończenia dźwięków w sygnale tworzona jest macierz składająca się z zer i jedynek o wymiarach: (wymiar\_czasu, ilość\_klas\_midi). Wymiar czasu takiej macierzy jest sumowany, a następnie aplikowana jest funkcja:

def apply\_time\_tresholding(meta\_vector, audible\_threshold):  
 audible\_threshold = meta\_vector.shape[0] / audible\_threshold  
  
 meta\_vector[meta\_vector < audible\_threshold] = 0  
 meta\_vector[meta\_vector >= audible\_threshold] = 1  
  
 return meta\_vector

Daje to wynikowy tensor docelowy o wymiarach (1, ilość\_klas\_midi). Indeksy midi mapowane są na kolejne ineksy od zera, aby ułatwić obliczenia. W ten sposób jeżeli zakres midi obejmuje klucze od 40 do 80, indeks zerowy w wektorze odpowiadać będzie indeksowi midi równemu 40. Indeks pierwszy w wektorze docelowym odpowiada klasie midi 41 i tak dalej. Mapy ciepła wygenerowane na podstawie tensorów wejściowego i docelowego przedstawione są na rysunku 37 poniżej.



Rys. 37 Po lewej tensor wejściowy: melspektrogram o czterdziestu pasmach obemujący zakres czasu około 1024 milisekund. Po prawej tensor docelowy: wektor którego wymiarem są kolejne obecne w sygnale dźwięki o danych indeksach midi. Przedstawiony przykład składa się z trzech dźwięków o indeksach midi: 43, 62, 71.

Wykrywanie dźwięków

W celu wykrywania dźwięków w nagraniach audio wykorzystane zostały opisane wcześniej sieci neuronowe. Celem jest uzyskanie miary f1 na poziomie 0.85 dla wygenerowanych wcześniej dźwięków dźwiękowych o złożoności do pięciu dźwięków granych jednocześnie. Jako, że wykrywanie dźwięków pojedynczych juest zedecydowanie prostsze od wykrywania wielu nałożonych dźwięków wykorzystana zostanie technika nazywana w jezyku angielskim „transfer learning”. Polega ona na wykorzystaniu do treningu modelu który wcześniej został przygotowany do jakiegoś problemu pokrewnego. Proces uczenia przeprowadzonego na potrzeby niniejszej pracy podzielony jest na sześć iteracji przedstawionych w Tabeli 5. poniżej.

Tabela 5. Iteracje treningu sieci neuronowej

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| numer iteracji | dźwięków jednocześnie | folds | max\_real\_time\_mins | midi\_range | overlap\_list | overlap\_prob | zero\_padding |
| 1 | 1 | 100 | 15 | [21, 108] | [1] | [100] | [0] |
| 2 | 2 | 100 | 15 | [21, 108] | [1, 2] | [20, 80] | [0] |
| 3 | 3 | 100 | 15 | [21, 108] | [2, 3] | [20, 80] | [0] |
| 4 | 4 | 100 | 15 | [21, 108] | [3, 4] | [20, 80] | [0] |
| 5 | 5 | 100 | 15 | [21, 108] | [4, 5] | [20, 80] | [0] |
| 6 | 5 | 100 | 15 | [21, 108] | [1, 2, 3, 4, 5] | [25, 25, 25, 15, 10] | [0, 32000] |

Podczas każdej kolejnej iteracji do treningu wykorzystywany jest model zainicjalizowany wagami wyznaczonymi podczas iteracji poprzedzającej. W ten sposób w kolejnych krokach wykorzystywany jest model operujący na takich samych danych wejściowych, radzący sobie z rozwiązywaniem podobnego, uproszczonego problemu. Technika ta nazywana jest w polskojęzycznej i zagranicznej literaturze Transfer Learningiem[[56]](#endnote-57).

Wykorzystane podczas treningu metryki to dokładność (ang. accuracy) oraz miara f1. Dokładność odzwierciedlać będzie ogólną ilość poprawnie zakwalifikowanych próbek. Wysoka wartość tej metryki może jednak wprowadzać w błąd będąc sztucznie zawyżany przez dużą ilość próbek zakwalifikowanych jako prawidziwie ujemne (TN). Z tego bardzoiej miarodajną miarą jest metryka f1 oparta o precyzję oraz czułość, a co za tym idzie niuwzględniane są Zastosowanie stałych metryk podczas ekpserymentów pozwoli na bezpośrednie porównywanie ich.

Dla każdej z iteracji wykonywana jest ewaluacja na losowo wybranym ciągu dźwiękóws

Przeprowadzone eksperymenty

Jednokanałowy Model Resnet

Wykorzystywaną z powodzeniem architekturą służącą do analizy zdjęć kolorowych jest architektura ResNet oparta o warstwy splotowe z połączeniami rezydualnymi. W celu wykrywania dźwięków przystosowany został model ResNet18. Zmieniona została ilość kanałów w tym modelu. W miejsce trzech kanałów odpowiadających kolorom, czerwonemu, zielonemu oraz niebieskiemu w zdjęciu, zastosowany został jeden kanał dla reprezentacji sygnału mono.

//opisać oroces treningu

Trening

// TODO Model liniowy

- pojedyncze dźwięki vs wiele dźwięków

// TODO Jednokanałowy model Resnet z warstwami rekurencyjnymi

// TODO Analiza wyników eksperymentów.

Podsumowanie

// TODO Wnioski

// TODO Perspektywy dalszego rozwoju projektu.

Wykaz rysunków

Rys. 1. Dźwięk C grany na gitarze akustycznej.

Rys. 2. Fala akustyczna jako suma sinusów.

Rys. 3. postać czasowa sygnału złożonego

Rys. 4. postać częstotliwościowa

Rys. 5. Skwantyzowany sygnał muzyczny

Rys. 6. Widmo statyczne dźwięku trąbki (w stanie ustalonym) poziom widma [dB] w funkcji częstotliwości

Rys. 7. Analiza STFT utworu “Radio gaga” wydanego w 1986 roku przez zespół Queen.

Rys. 8. Cztery często używane funkcje okna (na górze) i odpowiadające im Transformaty Fouriera (na dole)

Rys. 9. Rozdzielczość czasowo-częstotliwościowa. Rozmiar okna analizy: po lewej 1025; po prawej 4096

Rys. 10. Analiza McAulay-Quatieri (MQ)

Rys. 11. Od lewej: Skala melowa w funkcji częstotliwości , spektrogram dźwięku F#4, mel-spektrogram dźwięku F#4.

Rys. 12. Wykres MFCC i pochodnych. Współczynniki MFCC są używane m.in. do parametryzacji barwy dźwięków muzycznych.

Rys. 13. Funkcja autokorelacji dla dźwięku klarnetu

Rys. 14. Związek pomiędzy skalą, a częstotliwością w sygnale

Rys. 15. Wynik analizy falkowej dla dźwięku F#4

Rys. 16. Zakres słyszenia Oś pozioma przedstawia częstotliwość, oś pionowa zaś to ciśnienie akustyczne. Niebieską linią zaznaczony jest próg bólu, zieloną zaś próg słyszenia. Linie czerwone łączą częstotliwości które dla człowiek odbiera.

Rys. 17. Przykłady obwiedni dźwięków muzycznych. Trąbka.

Rys. 18. Składowe częstotliwościowe dźwięku C granego na gitarze, w tym ton podstawowy oraz składowe harmoniczne dźwięku.

Rys. 19. Klucz (ang. clef) określa wysokość dźwięku na pięciolinii

Rys. 20. Od lewej cała nuta, półnuta, ćwierćnuta, ósemka, szesnastka, trzydziestodwójka i pauzy całonutowa, półnutowa, ćwierćnutowa, ósemka, szesnastka, trzydziestodwójkowa, sześćdziesięcio czwórkowa

Rys. 21. Utwór “Byłaś serca biciem” skomponowany przez Jerzego Dobrzyńskiego

Rys. 22. Zapis utworu “Byłaś serca biciem” wygenerowany na podstawie notacji abc

Rys. 23. Uczenie głębokie jako poddziedzina uczenia maszynowego, uczenie maszynowe wjako poddziedzina sztucznej inteligencji.

Rys. 24 Kolorowe zdjęcie jest tensorem trzeciego rzędu.

Rys. 25 Przypisywanie wielu klas określających to co znajduje się na danych zdjęciach.

Rys. 26 Schemat neuronu McCullocha-Pittsa

Rys. 27. Funkcje aktywacji neuronu z biasem. Funkcja signum i funkcja sigmoidalna

Rys. 28 Wykresy wybranych funkcji aktywacji. U góry od lewej: BSF, Funkcja liniowa, ReLU.

Na dole od lewej: Sigmoid, Tanh

Rys. 29 Jeśli błędy propagowane pochodzą z kilku neuronów, są dodawane.

Rys. 30. Propagacja wsteczna, obliczanie uaktualnionych wartości parametrów .

Rys. 31. Pochodne wybranych funkcji aktywacji, od lewej: pochodna BSF, pochodna funkcji liniowej f(x) = 4x, pochodna funkcji aktywacji tanh.

Rys. 32 Współczynnik uczenia. Od góry od lewej: Za mały współczynnik uczenia – wolny trening, dobry współczynnik uczenia; U dołu od lewej: duży współczynnik uczenia – problemy ze znalezieniem minium; Za duży współczynnik uczenia, eksplozja współczynników.

Rys. 33 Wraz ze wzrostem liczby epok zwiększa się liczba zmian wagi w sieci neuronowej, a krzywa przechodzi od underfittingu przez optymalną do overfittingu.

Rys. 34 Rozkład próbek dla dźwięków gitary wględem wysokości dźwięków.

Rys. 35. Postać czasowa sygnału audio z zaznaczonymi punktami rozpoczęcia (kolor czerwony) oraz zakończenia (kolor fioletowy)

Rys. 36 Postać czasowa sygnałów od góry: Pojedynczy dźwięk na początku taktu, podwójne dźwięki na początku oraz z połowie taktu, potrójny dźwięk na początku oraz w połowi

Rys. 37 Po lewej tensor wejściowy: melspektrogram o czterdziestu pasmach obemujący zakres czasu około 1024 milisekund. Po prawej tensor docelowy: wektor którego wymiarem są kolejne obecne w sygnale dźwięki o danych indeksach midi. Przedstawiony przykład składa się z trzech dźwięków o indeksach midi: 43, 62, 71.

Wykaz tabel

Tabela 1. Literowe nazwy dźwięków naturalnych i alterowanych.

Tabela 2 .Dźwięki w oktawach w kontekście częstotliwości.

Tabela 3. Reprezentacja nut w standardzie MIDI

Tabela 4. Tablica pomyłek na przykładzie problemu medycznego

Tabela 5. Iteracje treningu sieci neuronowej

Bibliografia

1. B. S. Gowrishankar, N. U. Bhajantri, “An exhaustive review of automatic music transcription techniques: Survey of music transcription techniques”. 2016 [↑](#endnote-ref-1)
2. “Virtual Microphone System - User Guide” [Online] [Dostęp: 13.07.2020] http://download.slatedigital.com/vmr/Virtual%20Microphone%20System%20-%20User\_Guide.pdf [↑](#endnote-ref-2)
3. M.A. Martinez-Ramirez, J Reiss, “End-to-end equalization with convolutional neural networks”, 2018 [↑](#endnote-ref-3)
4. A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, “[Big Transfer (BiT): General Visual Representation Learning](https://paperswithcode.com/paper/large-scale-learning-of-general-visual)”, 2019 [↑](#endnote-ref-4)
5. Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, H. Ling, “[CBNet: A Novel Composite Backbone Network Architecture for Object Detection](https://paperswithcode.com/paper/cbnet-a-novel-composite-backbone-network)”, 2019 [↑](#endnote-ref-5)
6. J. W. Kim, J. Salamon, P. Li, J. P. Bello “CREPE: A CONVOLUTIONAL REPRESENTATION FOR PITCH ESTIMATION”, 2018 [↑](#endnote-ref-6)
7. “Abcnotation” [Online]. [Dostęp: 03.12.2019]. Dostępny w Internecie:

   http://abcnotation.com [↑](#endnote-ref-7)
8. V. Sarnatskyi, V. Ovcharenko, M. Tkachenko, S. Stirenko, Y. Gordienko, A. Rojbi, “Music Transcription by Deep Learning with Data and “Artificial Semantic” Augmentation”, 2017 [↑](#endnote-ref-8)
9. R. Kelz and G. Widmer. “An Experimental Analysis of the

   Entanglement Problem in Neural-Network-based Music Transcription

   Systems”. 2017. [↑](#endnote-ref-9)
10. E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, “Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection”, 2017 [↑](#endnote-ref-10)
11. C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall , “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, 2018 [↑](#endnote-ref-11)
12. T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, “Audio context recognition using audio event histograms” 2010 [↑](#endnote-ref-12)
13. A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection” 2016. [↑](#endnote-ref-13)
14. P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, “Chime-home: A dataset for sound source recognition in a domestic environment” 2015 [↑](#endnote-ref-14)
15. A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for Polyphonic Sound Event Detection”, 2016 [↑](#endnote-ref-15)
16. S. Adavanne, P. Pertila, T. Virtanen, “Single and multichannel sound event detection using convolutional recurrent neural network”, 2017 [↑](#endnote-ref-16)
17. “Detection and classification of acoustic scenes and events (DCASE),” 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/tasksound-event-detection-in-real-life-audio [↑](#endnote-ref-17)
18. K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, T. Virtanen, “Sound Event Detection with Depthwise Separable and Dilated Convolutions” 2020 [↑](#endnote-ref-18)
19. G. Szwoch „Analiza dźwięków muzycznych, Akustyka muzyczna” [↑](#endnote-ref-19)
20. „The Fourier Transform” [Online] [Dostęp: 13.07.2020] http://www.thefouriertransform.com/transform/fourier.php [↑](#endnote-ref-20)
21. G. Sanderson “Czym jest transformata Fouriera? Wizualne wprowadzenie.” [Online][Dostęp: 13.07.2020] https://www.youtube.com/watch?v=spUNpyF58BY [↑](#endnote-ref-21)
22. Z. Fortuna, B. Macukow, J. Wąsowski, „Metody numeryczne” 1993 [↑](#endnote-ref-22)
23. D. C. Price, "Spectrometers and Polyphase Filterbanks in Radio Astronomy”, 2018 [↑](#endnote-ref-23)
24. S. Hilbert, “FFT Zero Padding” [Online][Dostep: 13.07.2019] https://www.bitweenie.com/listings/fft-zero-padding/ [↑](#endnote-ref-24)
25. L. Roberts “Understanding the Mel Spectrogram” [Online] [Dostęp 13.07.2020] https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 [↑](#endnote-ref-25)
26. „MEL” [Online][Dostęp: 13.07.2020] https://www.sfu.ca/sonic-studio-webdav/handbook/Mel.html [↑](#endnote-ref-26)
27. O. Luening, A. Wayne-Slawson, G. Ciamaga, J. Chadabe, J. E. Rogers, G. Mumma, "The Development and Practice of Electronic Music", 1975 [↑](#endnote-ref-27)
28. “Continuous Wavelet Transform and Scale-Based Analysis” [Online] [Dostęp 13.07.2020] https://uk.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html [↑](#endnote-ref-28)
29. S. Hanov, “Wavelet Sound Explorer” [Online] [Dostęp: 13.07.2020] http://stevehanov.ca/wavelet/ [↑](#endnote-ref-29)
30. E. B. Goldstein, “Sensation and Perception” [↑](#endnote-ref-30)
31. J Habela, “Słowniczek muzyczny” 1968 [↑](#endnote-ref-31)
32. M. Velankar, “Study paper for Timbre identification in Sound”, 2013 [↑](#endnote-ref-32)
33. A. Zlatintsi, P. Maragos, “Multiscale Fractal Analysis of Musical Instrument Signals With Application to Recognition”, 2012 [↑](#endnote-ref-33)
34. K. Łopatka, „Podstawy notacji muzycznej i teorii muzyki” [↑](#endnote-ref-34)
35. „abctool” [Online][Dostęp 13.07.2020] www.github.com/attejensen/abctool [↑](#endnote-ref-35)
36. „abcConverter” [Onine][Dostęp 13.07.2020] <http://www.mandolintab.net/abcconverter.php> [↑](#endnote-ref-36)
37. “MIDI History:Chapter 6-MIDI Is Born 1980-1983” [Online] [Dostęp: 13.07.2020] https://www.midi.org/articles-old/midi-history-chapter-6-midi-is-born-1980-1983 [↑](#endnote-ref-37)
38. “Standard MIDI-File Format Spec. 1.1, updated” [Online] [Dostęp: 13.07.2020] http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html [↑](#endnote-ref-38)
39. D. Baccherini, D. Merlini, R. Sprugnoli, “Tablatures for Stringed Instruments and Generating Functions” [↑](#endnote-ref-39)
40. M. Roos “Deep Learning Neurons versus Biological Neurons” [Online] [Dostęp 27.07.2020]

    https://towardsdatascience.com/deep-learning-versus-biological-neurons-floating-point-numbers-spikes-and-neurotransmitters-6eebfa3390e9 [↑](#endnote-ref-40)
41. K. Ciomcia „Czym jest deep learning i sieci neuronowe” [Online] [Dostęp 27.07.2020] https://bulldogjob.pl/articles/1136-czym-jest-deep-learning-i-sieci-neuronowe [↑](#endnote-ref-41)
42. B. Earnshaw „A brief survey of tensors”, 2017 [↑](#endnote-ref-42)
43. J. Howard „Multi-label prediction with Planet Amazon dataset” [Online] [Dostęp 28.07.2020] https://nbviewer.jupyter.org/github/fastai/course-v3/blob/master/nbs/dl1/lesson3-planet.ipynb [↑](#endnote-ref-43)
44. S. Hosseinimotlagh, E.E. Papalexakis “Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles”, [↑](#endnote-ref-44)
45. J. Stangierski, D. Wiess, A.Kaczmarek, “Multiple regression models and Artificial Neural Network (ANN) as prediction tools of changes in overall quality during the storage of spreadable processed Gouda cheese”, 2019 [↑](#endnote-ref-45)
46. „Neurony z biasem i bez biasu” , [Online] [Dostęp 31.07.2020] http://galaxy.agh.edu.pl/~vlsi/AI/bias/bias.html [↑](#endnote-ref-46)
47. S. Tiwari „Activation functions in Neural Networks”, [Online] [Dostęp 04.08.2020] https://www.geeksforgeeks.org/activation-functions-neural-networks/ [↑](#endnote-ref-47)
48. D. Gupta “Fundamentals of Deep Learning – Activation Functions and When to Use Them?”, [Online] [Dostęp 09.08.2020] https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/ [↑](#endnote-ref-48)
49. M. Bernacki, P. Włodarczyk, “Principles of training multi-layer neural network using backpropagation” [Online] [Dostęp 09.08.2020] http://home.agh.edu.pl/~vlsi/AI/backp\_t\_en/backprop.html [↑](#endnote-ref-49)
50. J. Heaton „The Number of Hidden Layers” [Online] [Dostęp: 10.08.2020] https://www.heatonresearch.com/2017/06/01/hidden-layers.html [↑](#endnote-ref-50)
51. J. Browniee „A Gentle Introduction to Dropout for Regularizing Deep Neural Networks” [Online] [Dostęp 10.08.2020] https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/ [↑](#endnote-ref-51)
52. P. Sayak “The effects of weight initialization on neural nets” [Online] [Dostęp: 12.08.2020] https://www.wandb.com/articles/the-effects-of-weight-initialization-on-neural-nets [↑](#endnote-ref-52)
53. B. D. Hammel “What learning rate should I use?”, 23.03.2019 [Online] [ Dostęp: 12.08.2020] www.bdhammel.com/learning-rates/ [↑](#endnote-ref-54)
54. J. Browniee “How to Choose Loss Functions When Training Deep Learning Neural Networks” [Online] [Dostęp 16.08.2020] https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/ [↑](#endnote-ref-55)
55. „The NSynth Dataset” [Online] [Dostęp 24.08.2020] https://magenta.tensorflow.org/datasets/nsynth [↑](#endnote-ref-56)
56. J. Browniee “A Gentle Introduction to Transfer Learning for Deep Learning” [Online] [Dostęp 15.09.2020] https://machinelearningmastery.com/transfer-learning-for-deep-learning/ [↑](#endnote-ref-57)