Recunoașterea automată a acordurilor muzicale acustice

Linca Răzvan Cosmin

Universitatea Babeș-Bolyai
Facultatea de Matematică și Informatică

2020
24 Aprilie

1 Abstract

Acest articol urmărește prezentarea celor mai importante noțiuni de învățare automată, cu aplicabilitate pentru problema recunoașterii automate a acordurilor muzicale acustice, prin învățarea caracteristicilor extrase din semnalul audio corespunzător. Pentru fiecare metodă descrisă, se vor prezenta cele mai bune rezultate, pe baza cercetărilor conexe în domeniu. Astfel, se vor descrie cronologic algoritmi potriviți, pornind de la modelele probabilistice, ca modelele Markov cu stări ascunse, fiind prima abordare a problemei, până la construirea și optimizarea unor rețele neuronale convoluționale, utilizate recent în acest domeniu, acestă metodă fiind și cea aleasă pentru implementarea unei soluții, rezultatele obținute fiind prezentate în ultima secțiune.

2 Introducere

Recunoașterea automată a acordurilor muzicale, indiferent de instrumentele vizate, este un domeniu cercetat în mod activ în domeniul obținerii informațiilor muzicale[1], în ultimii 20 de ani. Această categorie de algoritmi este o parte esențială a multor aplicații muzicale, cum ar fi sisteme de transcriere automată pentru diverse instrumente, aplicații de învățare în cadrul educației muzicale sau algoritmi de recomandare a muzicii sau a genurilor muzicale.

Studiul acestui domeniu are la bază o motivație care s-a născut din evoluția muzicii contemporane, cu focus pe muzica acustică. S-a observat că, odată cu trecerea timpului, a început un proces de depărtare a interpretului de partitură, acesta fiind un artist profesionist care învată un cântec prin simpla ascultare sau vizualizare a unei înregistrări, sau un amator pasionat care învață prin urmărirea repetată a unor tutoriale aflate pe platformele online. De altfel, este cunoscut faptul că mulți compozitori și chitariști renumiți nu puteau să citească sau să scrie partituri muzicale, bazându-se pe talentul muzical în a memora fragmente sau a improviza pe loc diverse ritmuri. Printre ei se numără Jimi Hendrix, Eric Clapton, Elvis Presley sau interpreții trupei The Beatles [13].

Astfel, motivarea principală în a aborda acest subiect are la bază dorința de a automatiza procesul de transcriere a conținutului audio direct într-o reprezentare simplă și sugestivă.

Soluția propusă urmează, în general, o paradigmă precisă, formată din două etape:

1. Extragerea trăsăturilor muzicale: etapă ce urmărește analiza și utilizarea algoritmilor de procesare a semnalului audio, algoritmi ce vor fi folosiți într-un proces complex ce are ca finalitate obținerea unei reprezentări a caracteristicilor, ce va fi folosită apoi ca intrare pentru problemele de clasificare. Pentru soluția propusă, reprezentarea aleasă este de tip chromagramă[2].

2. Aplicarea unei metode de clasificare: etapă ce explorează diferite metode de clasificare, cu aplicabilitate pentru problema enunțată. Această fază va fi abordată în detaliu în acestă lucrare, prezentând în primă fază teoretic, iar apoi practic, cu parametrii și rezultate experimentale din alte lucrări științiifice, 3 metode de învățare automată supervizată. Se va pune accent pe descrierea rețelelor neuronale convoluționale, fiind utilizate în elaborarea unei soluții pentru această problemă.

[1] Music Information Retrieval (MIR), este o știință interdisciplinară care se ocupă cu extragerea și prelucrarea informațiilor muzicale.

[2] Chromagrama este formată dintr-o secvență de vectori caracteristici, fiecare având rolul de a măsura intensitatea relativă a fiecărei note muzicale, raportată la fiecare frame, într-un interval de timp.

3 Studiu de caz

3.1 Modele Markov cu stări ascunse

Modelele Markov cu stări ascunse (HMM), se bazează pe determinarea unui lanț Markov. Un lanț Markov este un model care ne oferă probabilitățile secvențelor unor variabile aleatorii, numite stări, fiecare putând prelua valori din cadrul unui set. Aceste seturi pot reprezenta orice, de la cuvinte până la diverse simboluri (de exemplu, simboluri care definesc vremea).

Un lanț Markov face o presupunere foarte puternică în ceea ce privește prezicerea viitorului într-o succesiune, starea actuală fiind singura care contează. Stările anterioare stării curente nu au niciun impact asupra previziunii din viitor. Un astfel de lanț este util atunci când trebuie calculată o probabilitate pentru o secvență de evenimente observabile. În multe cazuri însă, evenimentele care sunt de interes sunt ascunse/invizibile și nu pot fi observate în mod direct. Un model Markov cu stări ascunse permite abordarea ambelor tipuri de evenimente, observabile și ascunse, considerate drept factori cauzali în modelul probabilistic.

Modelele Markov cu stări ascunse sunt caracterizate de trei probleme fundamentale:

1. Probabilitatea (*Likelihood*): Fiind dat un HMM $\lambda = (A, B)$ și un set de observații O, să se determine probabilitatea $P(O|\lambda)$;

2. Decodarea (*Decoding*): Fiind dat un set de observații O și un HMM $\lambda = (A, B)$ să se determine cea mai bună secvență de stări ascunse Q;
   Cel mai comun algoritm de decodare este algoritmul lui Viterbi. Este un algoritm de programare dinamică pentru a găsi cea mai probabilă secvență de stări ascunse, numită calea Viterbi. Rezultatul este o succesiune de evenimente observate, mai ales în contextul modelelor Markov cu stări ascunse.

3. Învățarea (*Learning*): Fiind dat un set de observații O și un set de stări într-un HMM, să se determine prin învățare parametrii A și B.
   Algoritmul standard de antrenare a unui HMM este cunoscut ca algoritmul Baum-Welch, fiind un caz special derivat din algoritmul *Expectation-Maximization*[3] Algoritmul permite antrenarea atât a paramentrilor de tranziție(A), cât și a celor de emisie(B) [9].

Una din primele lucrări care a stat la baza abordărilor ulterioare pentru o perioadă de timp este lucrarea propusă de către Sheh și Ellis [5]. Metoda are la bază învățare statistică utilizând modele Markov cu stări ascunse. Antrenarea modelului s-a realizat folosind algoritmul EM, tratând etichetele pentru acorduri ca valori ascunse pentru construcția EM. Tot pentru antrenarea modelului au folosit doar secvența de acorduri (*without chord boudaries:* neclasificate, în

[3]Expectation-Maximization (EM) este un algoritm iterativ care calculează estimări inițiale pentru probabilități, apoi folosește aceste estimări pentru a calcula o estimare mai bună, procesul continuând în acest fel, îmbunătățind iterativ probabilitățile pe care le învață.

raport cu intervalele de timp în care se găsește un anumit acord) ca input pentru modelul Markov, aplicând astfel algoritmul Baum-Welch pentru a estima parametrii modelului. Algoritmul garanta că estimările se vor îmbunătăți de la o etapă la alta, ajungând într-un optim local, în ceea ce privește setul de parametrii. După determinarea parametrilor, asupra modelului se aplică algoritmul lui Viterbi, pentru a afla cea mai optimă cale (cea mai probabilă secvență de acorduri pentru un semnal audio de intrare).

Rezultatele obținute au fost: 76% pentru precizia segmentării acordurilor din piese și 22% pentru precizia în recunoașterea acordurilor. Performanța scăzută pentru recunoaștere se datorează faptului că datele de antrenare au fost neclasificate, dar și faptului că acestea au fost insuficiente (20 de melodii pentru 147 de acorduri).

Plecând de la această abordare, 3 ani mai târziu, Lee și Slaney au enunțat o metodă îmbunătățită [12], având un set de date adnotat și mai numeros, compus din 140 de melodii, obținând o precizie de 93.35% în recunoașterea acordurilor, din cadrul melodiilor testate.

3.2 Rețele neuronale artificiale

"Birds inspired us to fly, burdock plants inspired velcro, and countless more inventions were inspired by nature. It seems only logical, then, to look at the brain's architecture for inspiration on how to build an intelligent machine. [6]"

Rețelele neuronale artificiale (ANN) se definesc ca structuri artificiale care încearcă să reproducă modul de funcţionare a creierului uman. Sunt construite din mai multe unități de procesare sau *neuroni artificiali* grupaţi în straturi, fiecare strat având un număr variabil de elemente. Fiecare neuron poate primi informații de la alți neuroni, fiind acceptată chiar primirea de informații de la el însuși.

Un neuron artificial modelează comportamentul unui neuron real. Astfel conexiunile dintre neuroni, numite ponderi sinaptice, sunt folosite în stocarea informației. După o procesare locală a semnalului de intrare pe baza informaţiei stocată în ponderile sinaptice (multiplicarea acesteia cu valorile informaţionale stocate) se produce o integrare (sumare) globală a rezultatelor obţinute (proces similar cu cel ce are loc în corpul celular al unui neuron biologic real). Dacă răspunsul obţinut depăşeşte un anumit prag, informaţia este transmisă mai departe (utilizarea unei funcții de activare) [4].

Valorile stocate prin intermediul ponderilor își schimbă valorile folosind algoritmi de învățare. Astfel, în cadrul unui ANN, în general, cel care construiește rețeaua nu trebuie să specifice valori pentru parametrii sistemului. Parametrii sunt extrași, în mod automat, prin intermediul algoritmilor de antrenare sau adaptare.

Odată ce rețeaua a fost proiectată, procesul de antrenare poate începe. Acest proces urmărește obținerea unor valori optime pentru ponderile aflate între oricare două noduri ale rețelei, prin *minimizarea erorii*. Eroarea se calculează determinând diferența dintre rezultatul real și rezultatul calculat de rețea.

Începând cu anul 2009, odată cu apariția unui set de date compus din piese de pe diverse albumuri apaținând trupelor Beatles, Zweieck sau Queen, complet adnotate cu acorduri, strategia în rezolvarea acestei probleme s-a schimbat, și foarte multe lucrări au utilizat acest set de date, aplicând diverse metode și comparând rezultate. Dimensiunea acestuia era de aproximativ 170 de piese.

O primă lucrare care a abordat mai multe strategii de clasificare, dar și de procesare sonoră, a fost lucrarea de masterat a lui Alessandro Bonvini, din anul universitar 2012-2013. Una din strategii a vizat crearea unei rețele neuronale artificiale, cu un singur strat ascuns, cunoscută ca *Single Hidden Multilayer Perceptron (MPL)* [3].

Bonvini a prezentat rezultatele comparativ, aplicând mai multe strategii de normalizare a datelor. Se vor prezenta rezultatele obținute prin aplicarea strategiei de normalizare $L - \infty$ [4]. În ceea ce privește metricile de evaluare a performanței, a introdus două valori specifice numite Weighted Average Overlap Ratio (WAOR), indicând durata de timp în care segmentele clasificate în mod corect rămân suprapuse celor din realitate, și Segmentation Quality (SQ), sugerând calitatea segmentării acordurilor.

Rezultatele obținute pentru cele două metrici de calcul au fost 69.6% pentru WAOR și 74.9% pentru SQ.

3.3 Rețele neuronale profunde

Nimic din natură sau din evoluția tehnologiei, până în ziua de azi, nu se compară cu abilitățile complexe de procesare a informației și de recunoaștere a diferitelor modele complexe pe care le are creierul uman. Încercarea tehnologiei este a avansa în această direcție, dezvoltând algoritmi care imită rețeaua creierului uman, acestea fiind numite rețele neuronale profunde (*deep neural networks*).

Rețelele neuronale profunde au o structură unică, deoarece au o componentă ascunsă (formată din straturi ascunse) relativ mare și complexă între straturile de intrare și ieșire. Pentru a fi considerată o rețea neuronală profundă, această componentă ascunsă trebuie să conțină cel puțin două straturi. Datorită structurii lor, rețelele neuronale profunde au o capacitate mai mare de a recunoaște tipare decât rețelele superficiale.

Există câteva arhitecturi neuronale profunde consacrate, cum ar fi:

- Rețele neuronale convoluționale (CNN)
- Rețele neuronale recurente (RNN)
- Rețele de convingeri profunde (DBN)

[4] Normalizarea $L - \infty$ este o strategie de normalizare prin care toate valorile unui vector de caracteristici sunt aduse în intervalul $[0, 1]$, prin împărțirea fiecărei valori la maximul acelui vector.

3.3.1 Rețele neuronale convoluționale

Rețelele neuronale convoluționale (CNN) sunt foarte similare cu rețelele neuronale obișnuite. Diferența constă în faptul că rețeaua face o presupunerea explicită că valorile de input sunt imagini, ceea ce îi permite să codifice anumite propietăți în cadrul arhitecturii. De asemenea, spre deosebire de o rețea obișnuită, straturile unui CNN au neuronii aranjați în 3 dimensiuni: lățime, înălțime și adâncime(de precizat, adâncimea se referă la a treia dimensiune de activare, nu la adâncimea rețelei neuronale complete, valoare care este egală cu numărul total de straturi dintr-o rețea).

În general, rețelele neuronale convoluționale, cunoscute și ca ConvNets, utilizează 3 tipuri de straturi pentru a construi arhitectura, și anume: convolutional layer, pooling layer și fully-connected layer.

Un strat de convoluție *(Convolutional Layer)*, este responsabil de scanarea unei reprezentări sursă de tip imagine, aplicând un filtru de o anumită dimensiune, cu scopul de a extrage caracteristici care pot fi importante pentru clasificare. Acest filtru mai este numit nucleu de convoluție *(convolution kernel)*. Nucleul conține parametrii care pot fi ajustați pentru a atinge cele mai precise predicții.

După încheierea convoluției, caracteristicile sunt comprimate *(downsampled)*, urmând ca aceeași structură convoluțională să se repete. La început, convoluția identifică trăsături din cadrul imaginii originale, apoi identifică subcaracteristici în părți mai mici ale imaginii. În cele din urmă, acest proces este menit să identifice caracteristicile esențiale care pot ajuta la clasificarea imaginii. Straturile de convoluție produc astfel unul sau mai multe imagini numite hărți de trăsături *(feature maps)*, imagini care conțini caracteristici ce apațin imaginii originale, înainte de punerea în aplicare a nucleului de convoluție.

Stratul de agregare *(Pooling layer)*, are rolul de a reduce/micșora dimensiunea imaginii de intrare, comprimând ieșirile unui strat convoluțional. Se consideră, ca exemplu, un filtru de dimensiunea 2 x 2 care urmează a fi aplicat folosind agregarea asupra unei imagini de dimensiunea 4 x 4. Folosind acest filtru, există două strategii care pot fi aplicate.

- Calculul mediei valorilor din regiunea acoperită de filtru *(mean pooling)*.
- Determinarea maximului din regiunea acoperită de filtru *(max pooling)*.

Straturile de convoluție și agregare sunt supuse unei funcții de activare cu rolul de a asigura comportamentul neliniar al rețelei. Ca funcție de activare, de cele mai multe ori se folosește ReLU *(Rectified Linear Unit)*. Mai este cunoscută ca operația de rectificare. ReLu este funcția de activare cel mai frecvent folosită în construirea modelelor de învățare profundă. Funcția returnează 0 dacă primește o intrare negativă, iar pentru orice valoare pozitivă x, se returnează aceea valoare. Pe scurt, se definește ca:

$$f_{ReLu}(x) = max(0, x)$$

O componentă complet conectată *(Fully connected)*, are rolul de a realiza clasificarea propriu zisă. Această componentă este o rețea neuronală clasică, la intrarea căreia se furnizează feature map-urile, și la ieșirea căreia se aplică funcția de activare *softmax*[5].

Astfel, ieșirea acestei componente este un vector de probabilități cu un număr de componente egal cu numărul de clase. Fiecare componentă a vectorului reprezintă probabilitatea ca imaginea dată ca input să se încadreze în clasa corespunzătoare.

Recapitulând întreg procesul, imaginea de la intrare conține o entitate care trebuie încadrată într-una din clasele preexistente. Input-ul este supus mai multor operații de convoluție, agregare și rectificare, de fiecare dată generându-se hărți de trăsături. Acestea conțin trăsături semnificative ale imaginii inițiale. Trăsăturile sunt apoi supuse unui proces de clasificare prin intermediul unei rețele neuronale complet conectate, rezultând o serie de probabilități ca imaginea să aparțină fiecărei categorii.

Arhitecturile tipice de CNN conțin câteva straturi convoluționale (fiecare având în continuarea lui câte un strat ReLu), apoi un strat de agregare, apoi alte câteva straturi convoluționale, apoi un alt strat de agregare, și așa mai departe. Imaginea devine din ce în ce mai mică pe măsură ce progresează prin rețea, devenind de asemenea din ce în ce mai profundă (adică, cu mai multe hărți de trăsături), datorită straturilor convoluționale. Structura unui astfel de model convoluțional este prezentat în Figura 1.

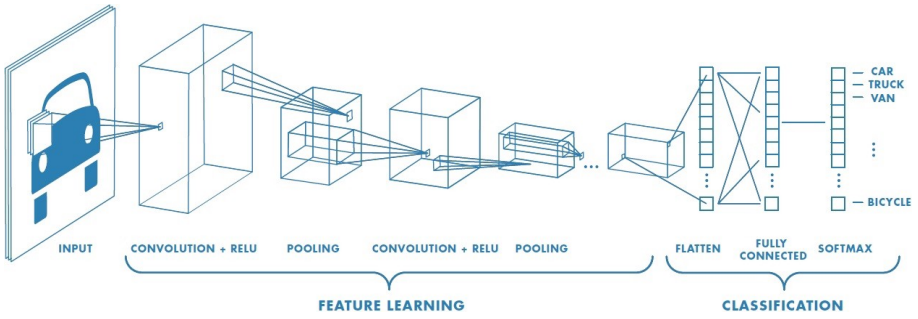

Figura 1: Arhitectură CNN. Se consideră o imagine ca input, care este supusă operațiilor corespunzătoare straturilor de convoluție și agregare. Rezultatul se propagă prin straturi complet conectate. Ieșirea rețelei constă într-un vector de probabilități ce încadrează imaginea într-o mulțime de clase preexistente [10].

[5]În matematică, funcția softmax, cunoscută și ca softargmax sau funcție exponențială normalizată, este o funcție care ia ca input un vector K de numere reale și îl normalizează într-o distribuție de probabilități, constând în probabilități K proporționale la valorile exponențiale ale numerelor din vectorul de intrare.

În jurul anului 2012, în cadrul conferinței internaționale de machine learning din Florida, SUA, s-a prezentat o lucrare care avea să schimbe încă odată direcția de a aborda problema recunoașterii automate a acordurilor, prezentând rețelele neuronale convoluționale ca o soluție cu eficiență ridicată, prin comparație cu abordările anterioare, în special prin referire la modelele Markov.

Lucrarea propusă de către Humphrey și Bello, [1], descrie două arhitecturi convoluționale, prezentând construcția strat cu strat și strategia antrenării. Setul de date utilizat conține 475 de înregistrări audio, însumând aproximativ 50000 de acorduri diferite.

Antrenând și testând cele două rețele convoluționale, s-a obținut 77.4%, respectiv 76.8% acuratețe la testare, demonstrând că această abordare performează într-un mod competitiv cu alte abordări considerate consacrate până în acel moment.

Începând din acel moment, majoritatea lucrărilor au propus metode pe baza rețelelor neuronale convoluționale. Cea mai concludentă lucrare analizată, prin raportare la soluția propusă, este cea prezentată de Zhou și Lerch, [8], de la centrul pentru tehnologia muzicii, din cadrul institutului de tehnologie din Georgia, SUA. Lucrarea prezintă construcția a două rețele profunde cu 6 straturi, în două arhitecturii diferite: o arhitectură clasică, cu același număr de neuroni (1024) în fiecare strat, și o arhitectură de tip *bottleneck (cu număr scăzut de neuroni în mijlocul rețelei, și în vecinătatea mijlocului)*. Setul de date constă în 317 piese adunate din discografiile unor trupe celebre, fiecare piesă fiind divizată în peste 1000 de cadre, pentru recunoașterea individuală a acodurilor. Algoritmul propus este capabil să recunoască acordurile majore și minore pentru fiecare notă de tip rădăcină, rezultând un dicționar de etichete pentru acorduri de dimensiunea 24+1, având 24 de acorduri cu etichete corespunzătoare cunoscute, și o etichetă pentru orice alt acord care nu este recunoscut.

Rezultatele obținute sunt prezentate pe ambele arhitecturi, aplicând diferite metode de pre-procesare a datelor. Cele mai bune rezultate au fost obținute folosind strategia *spliced filters*. Metrica de evaluare a perfomanței este durata totală a segmentelor cu predicție corectă. Astfel, rezultatele sunt:

- Arhitectura comună - 98.5% la antrenare și 87.6% la testare;
- Arhitectura bottleneck - 93.6% la antrenare și 91.9% la testare.

Se observă o îmbunătățire semnificativă a rezultatelor, construindu-se rețele convoluționale tot mai eficiente, creșterea fiind direct proporțională cu creșterea datelor etichetate corect și în detaliu.

Evoluția studiului în domeniu continuă până în preajma perioadei actuale, fiind prezentată o lucrare foarte importantă acum aproximativ un an de către Nadar, Abeßer și Grollmisch, [7], în cadrul unei conferințe de procesare sonoră, de la Malaga. Lucrarea urmărește extinderea dicționarului de etichete de la 24 la 84 de acorduri, extinzând aria claselor de acorduri care pot fi recunoscute. Dacă până acum se recunoșteau doar acordurile majore și minore ale notelor rădăcină, în această lucrare se dorește recunoașterea și a altor clase, printre care septacorduri sau acorduri diminuate. Setul de date este complex, fiind compus

atât din clasicele piese din cadrul discografiilor unor trupe celebre, cât și dintr-un set de date special pentru chitară, creat de Institutul de semantică muzicală, Fraunhofer, din cadrul universității tehnice Ilmenau, Germania.

Modelul convoluțional prezentat este alcătuit din mai multe straturi de tip convoluțional, intercalate de straturi de agregare de tip max pooling, rețeaua având un total de 12 straturi. Rețeaua a fost construită pentru a aborda două stragetii, una care propune recunoașterea pe baza dicționarului extins, dar care nu folosește întreg setul de date (S-84), și una care folosește dicționarul clasic, cu 24 de acorduri (S-24).

Metrica folosită este *f-score*[6]. Astfel, cele mai bune rezultate obținute sunt:

- S-84 - 76%, folosind doar setul de date al institutului Fraunhofer;
- S-24 - 91%, folosind întreg setul de date.

[6]Scorul F, denumit și scorul F1, este o măsură care ține de acuratețea unui test. Se definește ca media armonică ponderată a altor două metrici, anume precizia și recall-ul.

4 Rezultate experimentale

Pentru construcția modelului de recunoaștere acustică, am ales utilizarea unei rețele neuronale convoluționale. Detectarea acordurilor muzicale poate fi tratată în mod similar cu recunoașterea imaginilor, deoarece se pot crea imagini cu reprezentarea de tip chromagramă. Identificarea notelor sau a acordurile este mai simplă decât clasificarea imaginilor, întrucât nu implică texturi importante de învățat, rotiri sau scalări [11]. Cu toate acestea, recunoașterea acordurilor muzicale vine cu alte provocări. Notele muzicale din cadrul chromagramei nu sunt localizate într-o singură regiune în același mod în care sunt cele mai multe obiecte din imagini: o notă la o anumită frecvență fundamentală va fi compusă din armonice la multiplii acelei frecvențe.

În ciuda acestei provocări, CNN-urile au propietăți avantajoase care pot fi aplicate pentru problema recunoașterii acordurilor muzicale. Experimentele anterioare au sugerat că agregarea informațiilor considerând mai multe cadre muzicale din același sample, determină obținerea unei predicții cu o performanță mai mare. Astfel, convoluțiile determinate asupra datelor de intrare permit modelului creat să învețe caracteristici muzicale polifonice valoroase.

Arhitectura rețelei convoluționale construite este prezentată în Figura 2.

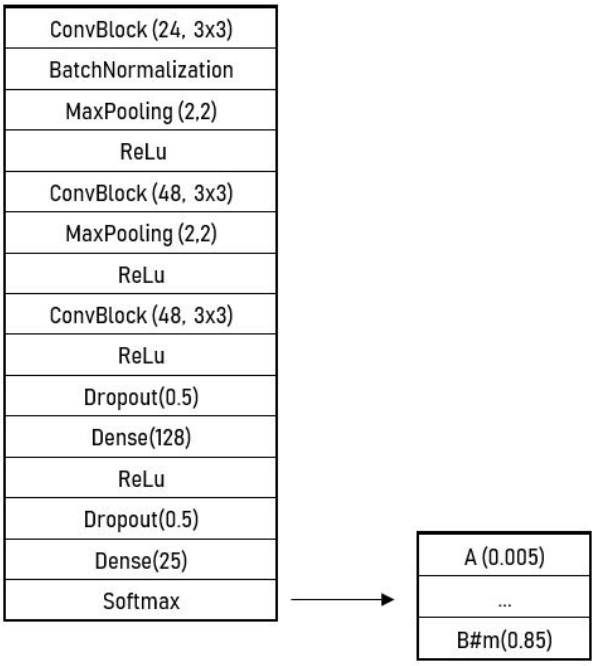

Figura 2: Arhitectura rețelei convoluționale construite. Numărul filtrelor și dimensiunea nucleului de convoluție sunt indicate între paranteze, pentru fiecare ConvBlock. Funcția de activare softmax este folosită în ultimul strat dens.

În ceea ce privește setul complet de date utilizat, s-au combinat 3 seturi de date pentru chitară, disponibile online, asupra cărora s-au aplicat o serie de algoritmi de augmentare. Setul de date este astfel compus din:

- 7398 de acorduri individuale, grupate în 16 fișiere audio de tip wav, ce aparțin Institutului de semantică muzicală, Fraunhofer, din cadrul Universității tehnice Ilmenau, Germania. Setul de date se numește IDMT-SMT-CHORDS;
- 6580 de înregistrări audio, fiecare având în compoziție un singur acord, aparținând laboratorului de cercetări muzicale al Universității din New York, SUA, numit GuitarSet[2];
- 200 de fișiere audio pentru 10 tipuri de acorduri, fiind colectat de grupul de cercetare Motefiore al Universității din Liège, Belgia.

După aplicarea algoritmilor de augmentare asupra setului de date complet, s-a obținut un total de 58.577 de înregistrări audio de tip wav, grupate în cele 24+1 clase (24 de acorduri cu etichete corespunzătoare cunoscute, și o etichetă pentru orice alt acord care nu se încadrează).

Cele 24 de acorduri recunoscute de către sistem, împreună cu valorile numerice asociate, sunt următoarele (perechi de forma acord muzical-valoare asociată): A-0, A#-1, A#m-2, Am-3, B-4, Bm-5, C-6, C#-7, C#m-8, Cm-9, D-10, D#-11, D#m-12, Dm-13, E-14, Em-15, F-16, F#-17, F#m-18, Fm-19, G-20, G#-21, G#m-22, Gm-23. Pentru orice alt acord care nu se încadrează în această înșiruire s-a creat perechea N-24 (acord necunoscut N, cu valoarea asociată 24).

Cu setul de date procesat și modelul convoluțional construit mai rămân de stabilit valorile unor parametrii înainte de startul procesului de învățare. Astfel, pentru determinarea erorii, s-a folosit strategia prin entropie încrucișată[7]. Numărul de epoci a fost de 25. În ceea ce privește algoritmul de optimizare, s-a ales algoritmul de optimizare Adam[8], cu dimensiunea unui bloc de 64 *(batch size)*.

Metricile utilizate au fost acuratețe, precizie, recall și scorul F1. Rezultatele obținute sunt prezentate în Tabelul 1.

Etapă	Acuratețe	Precizie	Recall	Scor F1
Antrenare	98.20%	99.05%	97.00%	97.99%
Testare	85.33%	90.28%	81.06%	85.66%

Tabelul 1: Rezultate obținute, în urma celor două etape: antrenare și testare.

[7]Entropia încrucișată compară predicția modelului cu valoarea etichetei corespunzătoare. Valoarea ei scade, pe măsură ce predicția devine din ce în ce mai precisă. Ea devine zero dacă predicția este perfectă. Astfel, entropia încrucișată este o funcție de pierdere pentru a forma un model de clasificare.

[8]Adam este o metodă de adaptare a ratei de învățare, ceea ce înseamnă că rata de învățare este calculată individual pentru diferiți parametri.

5 Concluzii

În acestă lucrare am prezentat cele mai adecvate și eficiente metode de învățare automată pentru recunoașterea și clasificarea acordurilor muzicale acustice. S-au abordat mai multe strategii, de la modele probabilistice, până la rețele convoluționale moderne, tratând fiecare subiect în parte, în primă fază la nivel teoretic, cu diverse definiții sau anumiți algoritmi specifici, prezentând în faza a doua rezultatele celor mai apreciate lucrări științiifice care au abordat aceeași problemă, folosind unul din metodele enunțate.

Pe baza cercertărilor conexe, am propus o soluție care utilizează o rețea neuronală convoluțională, prezentând arhitectura, setul de date folosit și rezultatele obținute la antrenare și la testare, în ultima secțiune a lucrării. Am demostrat astfel eficiența și versatilitatea acestui model neuronal, care poate fi aplicat cu succes și în acest domeniu atât de răspândit.

Bibliografie

[1] Eric J. Humphrey; Juan P. Bello. "Rethinking Automatic Chord Recognition with Convolutional Neural Networks". In: *Music and Audio Research Lab (MARL) New York University* (2012 11th International Conference on Machine Learning and Applications).

[2] Q.Xi; R.Bittner; J.Pauwels; X.Ye; J. P. Bello. "Guitarset, A Dataset for Guitar Transcription". In: *19th International Society for Music Information Retrieval Conference, Paris, France* (2018).

[3] Alessandro Bonvini. "Automatic chord recognition using Deep Learning techniques". In: *Journal: POLITECNICO DI MILANO Facoltà di Ingegneria dell'Informazione Corso di Laurea Magistrale in Ingegneria e Design del suono Dipartimento di Elettronica e Informazione* (2012-2013).

[4] Dan-Marius Dobrea. *Tehnici de inteligență computațională. Aplicații în electronică și biomedicină*. Editura Performantica (editură acreditată CNCSIS), Iaşi, România, ISBN 978-606-685-546-4, 2017.

[5] Alexander Sheh; Daniel P.W. Ellis. "Chord Segmentation and Recognition using EM-Trained Hidden Markov Models". In: *International Symposium on Music Information Retrieval* (2003).

[6] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow*. O'Reilly Media, ISBN-13: 9781492032649, 2019.

[7] Christon-Ragavan Nadar; Jakob Abeßer; Sascha Grollmisch. "Towards CNN-based Acoustic Modeling of Seventh Chords for Automatic Chord Recognition". In: *Sound and Music Computing Conference (SMC), Malaga* (2019).

[8] Xinquan Zhou; Alexander Lerch. "Chord detection using Deep Learning". In: *ISMIR 2015, International Conference on Music Information Retrieval* (2015).

[9] Daniel Jurafsky; James H. Martin. *Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Draft of October 16, ISBN-13: 9789332518414, 2019.

[10] *MathWorks. Convolutional Neural Network, 3 things you need to know.* http://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. Accesat în 2020-04-08.

[11] Michael Bereket; Karey Shi. "An AI Approach to Automatic Natural Music Transcription". In: *Stanford University* (2017).

[12] Kyogu Lee; Malcolm Slaney. "Automatic Chord Recognition from Audio Using an HMM with Supervised Learning". In: *ISMIR 2006, 7th International Conference on Music Information Retrieval* (2006).

[13] *The Music Studio*. http://www.themusicstudio.ca/blog/2017/11/909. Accesat în 2020-03-29.