Neurális háló alapú zenekategorizálás

Tudományos Diákköri Dolgozat

2014

Kristóf Tamás
VO9ZH6

Szabó Dávid
NW3L7F

Konzulens:
Kovács Viktor

Tartalom

1. Kivonat

Dolgozatunkban egy olyan tanuló algoritmust mutatunk be, amely képes különböző zeneszámokból információt kinyerni, és ez alapján szubjektív kategóriákba, műfajokba sorolni azokat, illetve egy számok közötti hasonlósági viszonyt is meghatározni. Ehhez a program először kiszámítja a Mel-frekvencia kepsztrum együtthatókat (MFCC) az egyes hangfájlokból, majd megállapítja a dalok tempóját (BPM) is. Az így kapott értékekből egy, az adott számot leíró jellegvektort képzünk, amit le is normálunk. Ezt a normált jellegvektort kapja meg a neurális háló bemenetként, így ezt minden egyes feldolgozandó számhoz legeneráljuk.

A további működéshez két halmazba szedjük a számainkat, az egyik felükkel a neurális háló tanítását végezzük, míg a másik felüket tesztelésre használjuk. A háló kimenete az általunk előre megszabott négy csoport valamelyikébe osztja be a számot. Ez a négy csoport jelenleg a klasszikus, a rock, a rap, és pop. A teszteléshez és a tanításhoz felhasznált számok listájának összeállításánál több különböző kombinációt is megpróbáltunk, ezzel próbálva az algoritmus hatékonyságát különböző körülmények között tesztelni. A program eredményessége az adott számlista változatosságától függően a legrosszabb esetben 60%, a legjobb esetben pedig 80% volt.

2. Bevezető

Manapság a különböző multimédiás weboldalak és alkalmazások egyre nagyobb térnyerésével több felhasználóban is felmerül az igény arra, hogy ne kizárólag a saját maguk által kiválasztott zeneszámokat hallgathassák csak meg, hanem ajánlásokat is kapjanak új, általuk még nem ismert előadókra és zeneszámokra. Ennek jól látható bizonyítékául szolgál az is, hogy az ezt a szolgáltatást implementáló programok - mint például a *Spotify*, *Pandora*, *Deezer*, *last.fm*, *iTunes*, *Youtube*, stb. - felhasználóinak a száma egyre csak növekszik. A felsorolt programok mindegyike képes arra, hogy a hallgató előző értékeléseinek, vagy akár egyszerűen hallgatási szokásainak figyelembe vételével egy olyan lejátszási listát állítsanak össze, ami jó eséllyel tetszik majd neki.

Ehhez természetesen szükség van egyrészt a felhasználók nyomon követésére, másrészt az így összegyűjtött adatokat megfelelően felhasználó algoritmusra, amely meg tudja becsülni a hallgató ízlését. Jelenleg két elterjedt megoldás létezik a feladatra, egy statisztikai, illetve egy ezt kiegészítő, zenei hasonlóságon alapuló módszer.

A *Spotify*, az *iTunes*, a *YouTube*, a *last.fm* működése főképp a statisztikát veszi alapul. Elég nagy felhasználói bázis esetén jó közelítést ad az egyes felhasználók ízlésére a *kollaboratív szűrés* [1]. Ennek a módszernek az alapja az, hogy az egyén ízlésére a közösség ízléséből lehet következtetni. Például, ha sok olyan felhasználó van, akiknek tudjuk, hogy tetszenek az *X, Y* és *Z* előadók, és jön egy új felhasználó, akinek tetszik az *Y* és *Z* előadó, akkor az algoritmus javasolni fogja ennek a felhasználónak az *X* előadót is hallgatásra. A módszer nagy előnye, hogy tartalomtól függetlenül bármilyen rendszerre alkalmazható, így például ezt alkalmazza a *Netflix* is filmekre, illetve az *Amazon* könyvekre. Az algoritmus hátránya pedig pont ugyanaz, mint ami az előnye: a tartalmat nem veszi figyelembe. Ez akkor jelent problémát, ha szeretnénk egy olyan rendszert megvalósítani, amelyben a felhasználó bizonyos sajátosságok, vagy egy másik számhoz való hasonlóság alapján kereshetne tartalmat.

A *Pandorán* például ez lehetséges, mert az nem ezt az algoritmust alkalmazza, hanem az ajánlásokhoz az úgynevezett *Music Genome Project*et veszi figyelembe [2]. Ez nem más, mint egy óriási adatbázis, melyet a Pandora megbízásából zeneszakértők hoztak létre. Ebben minden egyes számot először az általuk kinevezett öt fő műfaj (Pop/Rock, Hip-Hop/Elektronikus, Jazz, Világzene, Klasszikus zene) egyikébe sorolnak, majd az ehhez a kategóriához tartozó további jellemzők (*gének*), alapján tovább osztályoznak. Ilyen jellemzők lehetnek az énekes neme, a torzítás szintje, a szám hangulata, stb. Egy-egy számhoz 150-500 ilyen gén tartozik a kategóriától függően. Az adatbázis feltöltése és frissítése nem számítógépes algoritmussal, hanem valós zenészek és szakértők közreműködésével történik, akik az egyes számokat megfelelő elemzés után képesek besorolni valamelyik kategóriába, majd megmondani az egyes gének értékeit is. A program ezután a felhasználó pozitív és negatív értékeléseit figyelve próbálja meghatározni, melyek azok a jellemzők amelyek alapján egy szám tetszik neki, és melyek azok amelyeket nem kedvel. Az így felállított ízlés-becslés felhasználásával próbál aztán új számokat tenni a hallgató lejátszási listájába az algoritmus.

Emellett léteznek olyan szolgáltatások is, mint például a *Songza*, ahol a lejátszási listákat nem egy algoritmus állítja össze, hanem az adott cég által alkalmazott zenei szakértők. Itt a felhasználó ki tudja választani a programban, hogy milyen hangulatban van, és egy ennek megfelelő lejátszási lista indul el, illetve az egyes számokra is lehet pozitív / negatív értékelést adni. Ennek hatására az ahhoz a számhoz hasonlóakat a rendszer a továbbiakban javasolni, vagy kerülni fogja.

Ez a dolgozat egy olyan algoritmust mutat be, amellyel az előző két módszerhez felhasznált zeneelemzési és hasonlóságkeresési folyamat bizonyos mértékig automatizálhatóvá válik. Ehhez a program mintát vesz az egyes zeneszámokból, majd a mintákat a frekvencia- és időtartományban elemzi, és ennek eredményét egy jellegvektorban tárolja. Ezek a vektorok egy neurális háló bemenetei lesznek, ami megfelelő betanítás után képes lesz ismeretlen számokról is jó eséllyel megmondani, hogy milyen műfajba tartoznak.

Ez a rendszer például hasznos lehet egy olyan program létrehozására, amely háttértárolóra letöltött, meta adattal nem rendelkező zenei fájlokat képes valamilyen szubjektív jellemző (pl. műfaj) alapján csoportokba rendezni, majd ezekből több lejátszási listát összehozni. Ezzel a felhasználók képesek lennének műfajok szerint rendezni a saját zenei mappájukat, amit aztán később az így létrehozott lejátszási lista szerint tudnak majd a pillanatnyi hangulatuknak megfelelően hallgatni.

3. Korábbi megoldások

Irodalomkutatásunk során azt tapasztaltuk, hogy a fentebb leírt célra még alig létezik kész alkalmazás. Hasonló célú próbálkozásokról találtunk információt, bár ezek közül egyikből sem készült általánosan használható algoritmus. Ennek feltételezhető oka az, hogy erre a célra még nem létezik egy általánosan elterjedt, megfelelő pontossággal és sebességgel működő algoritmus, amely képes lenne megoldani ezt a feladatot. Az eddigi kísérleti próbálkozások abban egyet értenek, hogy a probléma megoldására valamilyen gépi tanuló, vagy mesterséges intelligencia algoritmust célszerű használni, mert ezek az eljárások a legalkalmasabbak csoportosítási problémák megoldására, amilyenre a dolgozatban tárgyalt feladat is visszavezethető.

A szakirodalomban leírt algoritmusok többsége 60-80% pontossággal működik, illetve találtunk egy jobb megoldást is, amelyben sikeresen alkalmaztak neurális hálózatot a számok 95% fölötti sikerességgel történő csoportosítására. Az irodalomban található módszerek mindegyikében 4 műfajba sorolták a számokat, melyek közt a pop, rock és klasszikus mindig szerepelt, negyedikként pedig az adott cikktől függően mást választottak, általában hip-hopot, rapet, népzenét, vagy egyszerűen egy egyéb kategóriát. Sajnos arról nem sikerült információt találnunk, hogy az egyes kísérletekhez konkrétan mely előadókat használták a rendszer tanításához, és melyeket a teszteléshez.

A fentebb leírt módszerek között, több megoldást is alkalmaztak az ismeretlen zeneszámok besorolására. M. Haggblade cikkében [3] k-közép módszert, legközelebbi szomszéd módszert, illetve neurális hálót is használt, míg C. Xiang

cikkében [4] egyirányú előrecsatolt neurális hálót használt, kifejezetten jó (89%) pontossággal. Az említett cikkekben leírt algoritmusok hatásfokai szép eredményekről számolnak be, viszont fontos említést tenni arról, hogy M. Haggblade cikkében mindössze 35-40 zeneszámot használtak bemenetként, C. Xiang pedig sajátos – kínai kultúrára jellemző – műfajokat is vizsgált, amelyek jelentős eltérést mutatnak a jelenlegi világzenei műfajoktól (ezáltal könnyebb elkülöníteni). Ezek tükrében, a dolgozatban a neurális háló alkalmazhatóságát vizsgáljuk, nagyobb számú mintán, az egyes műfajokon belül a műfajok határait is feszegető zeneszámok szerepeltetésével.

4. Vizsgált műfajok

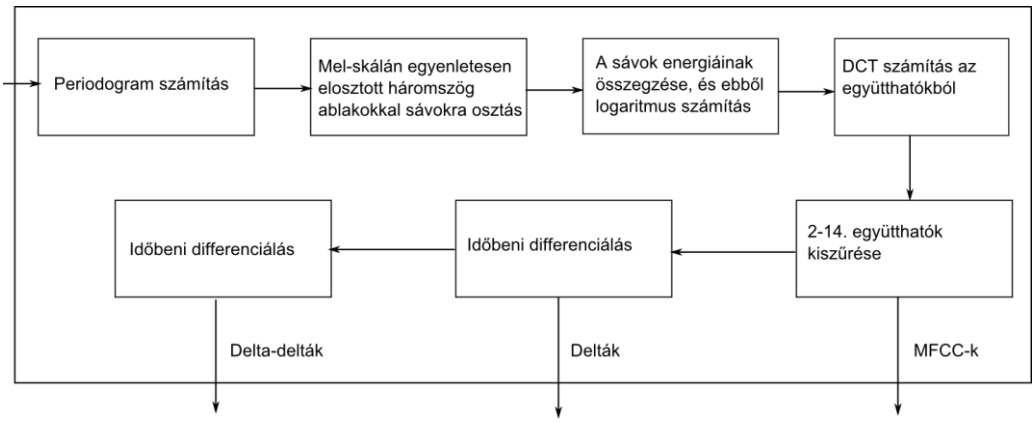
A vizsgálandó műfajokat négy csoportba osztottuk, amelyek a mai zenei világ leginkább elterjedt, sok művet magukba foglaló műfajai. A műfajokon belül az előadókat a saját ismereteink alapján választottuk ki, amelyek a következők:

Pop	Rap	Rock	Klasszikus
Adele	2Pac	Tankcsapda	Beethoven
Lady Gaga	Eminem	AC/DC	Mozart
Avicii	DMX	Deep Purple	Liszt
Kesha	Fankadeli	Billy Talent	Chopin
Miley Cyrus	Belga	Hammerfall	Bartók
ABBA	Hősök	Tyr	Bach
Boney M	Diw Antwort	SuperBus	Brahms
Neoton Família	Foreign Beggars	Supernem	Csajkovszkij
David Guetta	Wu Tang Clan	Republic	Schubert

Az előadók válogatása során különös figyelmet fordítottunk arra, hogy a kategóriákban a világhírű zenekarok mellett néhány hazai, illetve egyéb ország sajátos zenekarai is szerepeljenek, hogy még egy kategórián belül is a lehető legnagyobb változatosságot hozzuk létre el. A célunk ezzel az, hogy az így kiválasztott minta valóban megfelelően reprezentálja az adott stílust, teljes szélességében. Természetesen lehetetlen mindössze ennyi zeneszámmal tökéletesen leírni egy stílust, de úgy gondoljuk ezekkel az előadókkal a népszerű zenei stílusokat nagymértékben sikerült lefednünk.

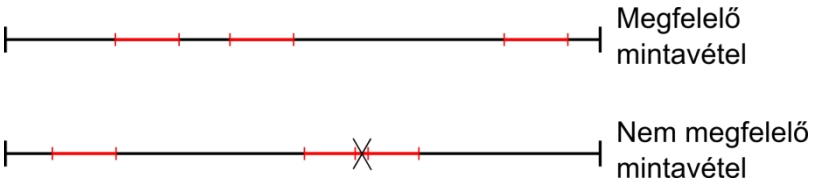
5. Hangfeldolgozás

Az első lépés a vizsgálandó számok neurális hálóba betáplálható formába hozása. Egy az egyben ugyanis egy 44,1 kHz-es frekvenciával mintavételezett 5-6 másodperces minta túl sok bemenő adatot jelentene a neurális hálónak, ami egyrészt a számítási idő megtöbbszöröződését okozta volna, másrészt a hibát is jelentősen megnövelné. Ehelyett a hangfájlnak inkább a frekvenciatartománybeli reprezentációját dolgoztuk fel, mivel itt létezik olyan kidolgozott módszer hangelemzésre, amely beszédfelismerésre sikeresen alkalmazható [5], és a szakirodalom szerint a zeneosztályozás területén is jelentős sikerek érhetőek el vele [6]. Ez a módszer az MFCC-kel, vagyis Mel-frekvenciás kepsztrális együtthatókkal való reprezentáció (1. ábra).



1. ábra | MFCC együtthatók számítása

Ahhoz, hogy a folyamat során létrehozott vektor minél valószínűbb legyen, hogy valóban az egész számot jellemzi, és nem csak szelektíven a mintaként használt néhány másodperces darabot, a feldolgozási folyamatot számonként háromszor végezzük el, három különböző - véletlenszerűen választott - időponttól kezdődő, egyenként 5 másodperces mintával (2. ábra). A kiválasztási algoritmus kizárja azt a lehetőséget, hogy ezek az 5 másodperces minták bármennyire is átfedésbe kerülhessenek egymással.

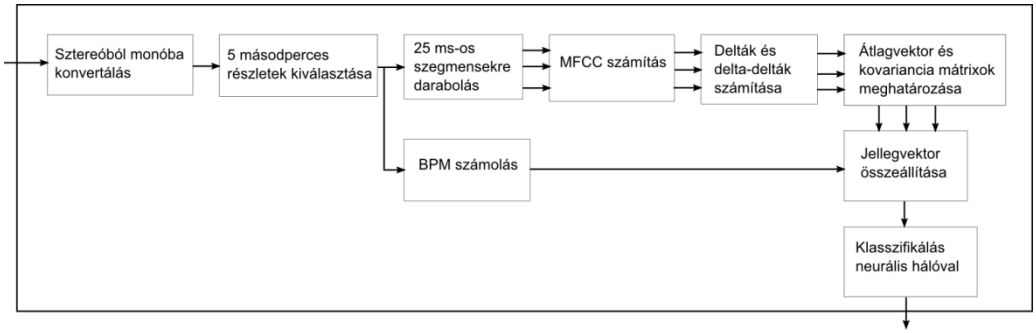


2. ábra | Mintavételi helyek

Ezt a teljes folyamatot azonban egy sztereó jel esetén - mivel a két hangsávban általában akkora eltérés nincs - indokolatlan kétszer elvégezni. Ezért egyszerűsítésként a sztereó jelből első lépésként egy sávot generálunk, ami

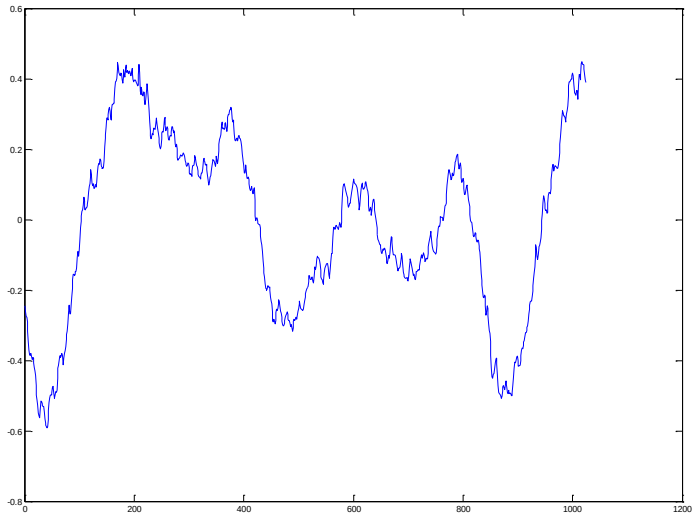
egyszerűen a jobb és a bal sáv átlagolásával történik. Bár ez a módszer bizonyos (ritka) számokon már igencsak észrevehető hibát okoz, általános esetben ezt a legcélszerűbb alkalmazni.

A számok feldolgozásának a teljes folyamát szemlélteti a 3. ábra.



3. ábra | Számok feldolgozásának folyamata

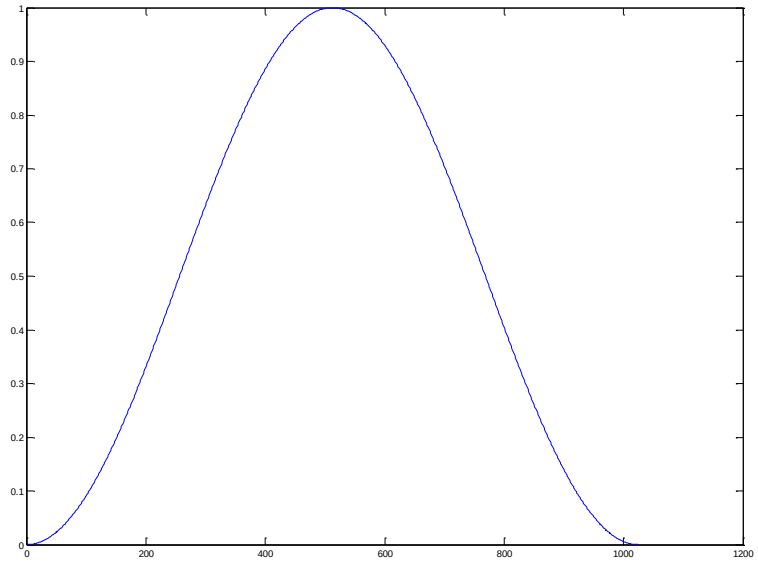
Frekvencia vizsgálat



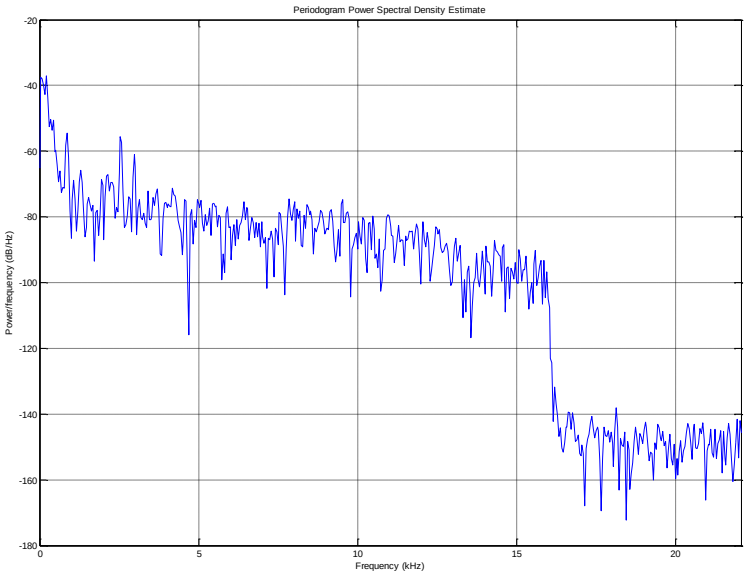
4. ábra | Egy 25 ms-os részlet időtartományban

A módszerhez a jel periodogramjára, vagyis teljesítmény spektrális sűrűség becslésére van szükség. Ennek számítása során figyeltünk arra, hogy az ezt számító függvénynek ne adjuk egyben át a teljes 5 másodperces mintát, mivel ez esetben teljesen elveszne az időfüggés az adatainkból. Ezért ezt a szegmenst először 25 ms-os darabokra bontjuk, amelyek 20%-os átfedésben vannak egymással. Egy 25 ms-os részletet szemléltet a 4. ábra. Ezek egyenkénti transzformálásával információt kapunk arról is, hogy a feldolgozandó jelben hogyan változnak az idő múlásával az egyes frekvenciákhoz tartozó amplitúdók. A transzformációhoz a szakirodalom ajánlása szerint Hanning-

ablakot használunk ablakozó-függvényként. Az 5. ábra a Hanning ablakot mutatja be, míg a 6. ábra az ablakozott jel periodogramját.



5. ábra | Hanning ablak



6. ábra | Periodogram ablakozás után

A következő lépés az MFCC-k számításához a sávokra bontás. A szokásos frekvencia sávokra bontással ellentétben itt az egyes sávok határait a Mel-skálán jelöljük ki a frekvencia skála helyett. A Mel-skála egy olyan, zenei hangok leírására alkalmazott mutató, amely empirikus alapon, az emberi hallásnak megfelelően lett kialakítva [7]. Ez azt jelenti, hogy azok a hangok, amiket az ember egymástól azonos magasságkülönbségként hall, azok ezen a skálán azonos távolságra is helyezkednek el egymástól. A frekvencia és a Mel-

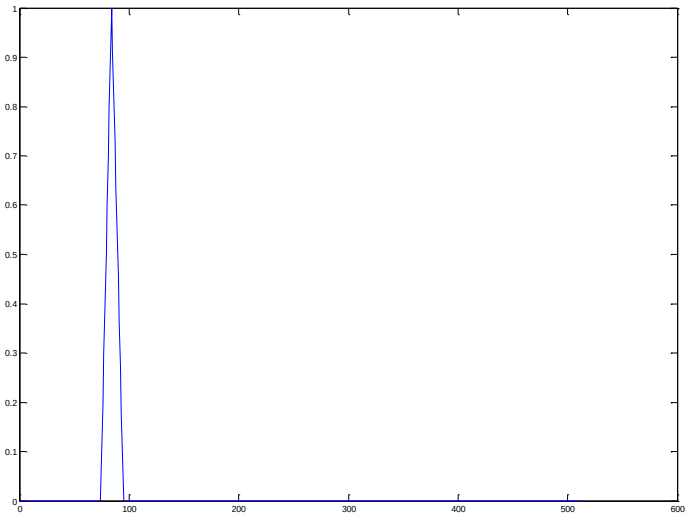
skála közötti átváltásra több képlet is létezik, ennél a feladatnál mi a legelterjedtebben alkalmazottat használtuk:

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right)$$

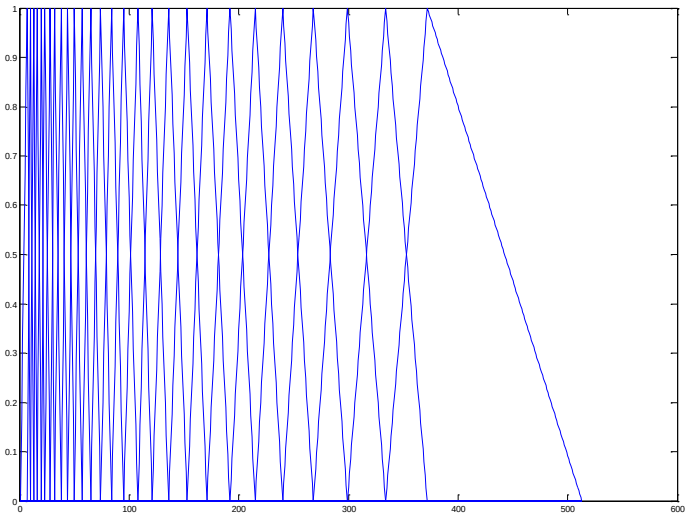
,ahol f az adott hang frekvenciája, m pedig az ennek megfelelő mel-skálán felvett érték. Látható a képletből, hogy a Mel érték a frekvencia tízes logaritmusával arányos, ami összhangban áll azzal a gyakori feltételezéssel, hogy az emberi hallás logaritmikus alapon működik.

A sávokra bontáshoz létrehozunk tehát a Mel-skálán 26 háromszöges ablakozó függvényt, amelyek egyenletesen helyezkednek a beállított minimum és maximum Mel értékek között. A 7. ábra egy ilyen háromszöges szűrőfüggvényt ábrázol**.** Ezeket az értékeket a periodogramokból ránézésre megállapított alsó- és felső-frekvenciahatárok átszámításával kaptuk meg, melyek a 300 Hz-nek és 16000 Hz-nek adódtak.

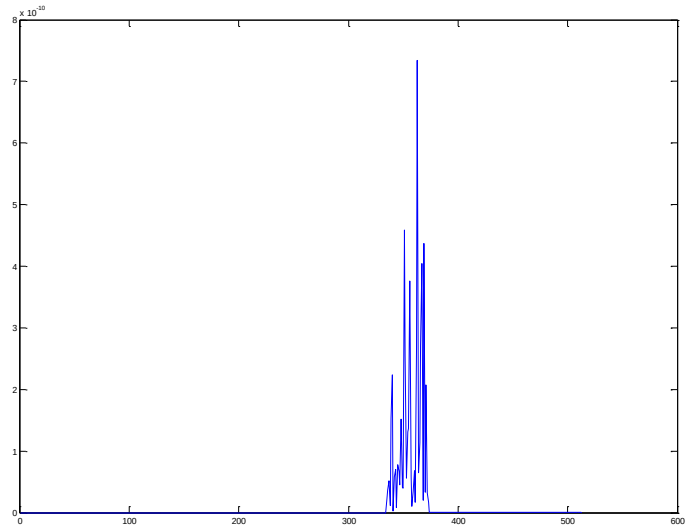
A 8. ábra az egyes szűrőfüggvények egymáshoz képesti elhelyezkedését mutatja frekvencia-tartományban. A Mel-skálás ablakokat visszaszámítjuk frekvenciára, majd elvégezzük egyesével mindegyikkel az ablakozást. A 9. ábra jól mutatja a jelnek a 26-os számú szűrőfüggvénnyel való szorzatát.



7. ábra | Háromszöges szűrőfüggvény



8. ábra | A szűrőfüggvények elhelyezkedése a különböző frekvenciákon



9. ábra | A jel az egyik szűrővel való beszorzás után

Következő lépésként az így kapott sávokban összegezzük az energiát, majd ennek a logaritmusát vesszük. Ezzel minden egyes 25 ms-os időszeletben egy 26 darab együtthatóból álló vektort kapunk eredményül, melyek az adott időben, az adott frekvenciasáv által tartalmazott energiával arányosak. A folyamat matematikai leírható a következő képlettel:

$$y[k] = \log_{10}\left(\sum_{i=1}^{n} \varphi_k[i] \cdot x[i]\right)$$

, ahol *y[k]* jelöli a k-adik szűrővel kapott tagot, φ_k pedig a k-adik szűrőt.

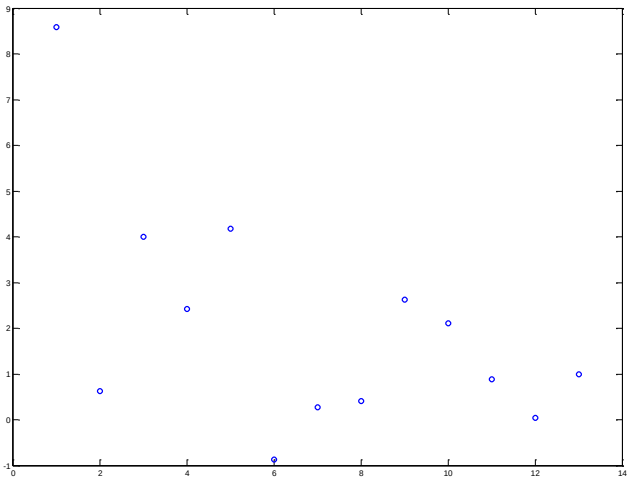
DCT

Az előzőleg kapott 26 együtthatót a következő lépésben egy számsorozatként kezelve őket diszkrét koszinusz transzformációt (DCT) hajtunk végre, amit a következő képlettel írhatunk le:

$$Y_n = \sum_{k=0}^{N-1} y_k \cos\left[\frac{\pi}{N}\left(k + \frac{1}{2}n\right)\right] \qquad n = 0, 1, \dots, N-1$$

Ezt a transzformáció általánosan alkalmazott például JPEG vagy MP3 tömörítésben is, mivel ez a Fourier-hez hasonlóan frekvenciainformációt szolgáltat az adott jelről, azonban annál jobban képes besűríteni a lényeges információt az elöl álló komponensekbe.

Az így kapott spektrum amplitúdóit nevezzük mel-frekvenciás kepsztrális kitevőknek, vagyis MFCC-knek. Ebből mi az alkalmazáshoz a felső 14-et vesszük figyelembe, a legelsőt elhanyagolva. A hátul lévők elhanyagolását maga a diszkrét koszinusz transzformáció indokolja, mivel a művelet tulajdonságaiból következik, hogy ezek a tagok már olyan kis hatással vannak a teljes jelre, hogy elhanyagolásukkal nem okozunk túl nagy hibát, bemenő adataink száma pedig jelentősen csökken. A legelső kitevőt azért nem vizsgáljuk, mert az a többivel ellentétben az nem egy lineárisan független adat, hanem a teljes mintadarabra eső energiával arányos mérőszám. Így végül 13 kitevőt kapunk eredményül, melyek időbeni differenciaszámításával megkapjuk a megfelelő *delta* értékeket is, melyek az adott hangszakasz dinamikáját jelzik. Ha a *delta* értékeket még egyszer differenciáljuk idő szerint, az a *delta-delta* értékeket adja eredményük. Az algoritmushoz ezeket is felhasználjuk. A 10. ábra egy számrészlet 13 MFCC együtthatóját szemlélteti.



10. ábra | Egy számrészlet MFCC együtthatói

A műveletsor tehát 3×13, vagyis 39 együtthatót ad eredményül minden egyes 25 ms-os részlethez. Mivel ennyiből még nehéz bármilyen következtetést is megfelelő bizonyossággal levonni, elvégezzük a folyamatot a többi mintavételezett részletre is.

Összefoglalva az elvégzett műveletek az MFCC együtthatók és azok időbeli változásának (delta értékeinek) meghatározásához szükséges lépések a következők:

1. Hangfájl sztereóból egy sávba (mono) konvertálása
2. Megfelelő hosszúságú minta kiválasztása a hangfájlból
3. Hangadatok részletekre osztása megfelelő átfedéssel
4. A kapott részletek ablakozó függvénnyel szorzása
5. Gyors Fourier-transzformáció alkalmazása (FFT)
6. Mel szűrőbázis alkalmazása, majd az egyes szűrők energiáinak összegzése
7. Összegek logaritmusának meghatározása
8. Diszkrét koszinusz transzformáció végrehajtása a logaritmikus összegeken
9. Az előző lépésben megkaptuk az MFCC együtthatókat, amelyekből az első 12 elemet használjuk a további számításhoz
10. Differenciaszámítás a delták és a delta-delták meghatározásához

Jellemvektor

Az adott hangfájlokon belül a minták helyének megfelelő kiválasztása dilemmát jelentett. Ha egy helyről veszünk mintát, akkor ugyan egy folytonos összefüggő részlet vizsgálható, ami pontosabb tulajdonságokkal bírna, viszont ha meg a fájlból több helyről veszünk kisebb mintákat, akkor nagyobb az esélye hogy ezek közül valamelyik valóban reprezentatív lesz a valós hangfájlt illetően. Végül 3 darab minta mellett döntöttünk, melyek helyének kiválasztása egyszerűen véletlenszerű, annyi feltétellel, hogy ne legyen köztük átfedés. Mindegyik kivágott minta 5 másodperc hosszú, de már ez is elég ahhoz, hogy az átszámolás után több ezer adatot kapjunk, amit még mindig sok lenne beadni a neurális hálónak.

A bemenő adatok számának csökkentése érdekében ezért nem az így kapott szeletenként 36 adatot adjuk be a hálóba, hanem ezekből egy jellemvektort számolunk. Ez két részből épül fel: a fent kapott eredményekből képzett átlagvektorból, és a kovariancia mátrixuk egyedi elemeivel.

Az átlagvektor egyszerűen az azonos sorszámú MFCC, delta illetve delta-delta adatok átlagolásával kapható meg, így ezzel a fentebb említett több

mintavételnyi MFCC érték lecsökkenthető egy darab 36 elemű vektorrá. Viszont ha kizárólag ezt használnánk, akkor az jelentős információvesztést jelentene, mivel ekkor minden adatot elvesztenénk arról, hogy idő közben mennyit, illetve hogyan változtak az MFCC-k értékei. Ezért bevezetjük a kovariancia-mátrixot is.

A kovariancia-mátrix egy olyan statisztikában alkalmazott mátrix, amelynek az *i,j* helyen álló eleme megmutatja két vektor *i*. és *j*. elemének kovarianciáját, vagyis együttmozgását. Ez gyakorlatilag annak a mértéke, hogy a két változó mennyire van összefüggésben egymással, értékeik mennyire mozognak együtt. Ha magas, akkor az egyik értéket megváltoztatva, ugyanúgy változik a másik is, míg ha 0 közeli, akkor bármit változtathatunk az egyik változón, a másik értéke nem fog változni. Számítási módja a definíciójából kiindulva:

$$\operatorname{cov}(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

A két vektor mi esetünkben a két különböző 25ms-os szegmensből vett MFCC, delta, illetve delta-delta érték. Az így felépített mátrix belátható, hogy szimmetrikus lesz, tehát nem kell a teljes mátrixot beadni a neurális hálónak, hanem elég helyette a főátlót, és az a fölött álló elemeket használni. Ezekből az egymástól független elemekből képezzük majd az adott zeneszámot jellemző jellemvektor maradék elemeit, ami így most már nem csak az átlagos MFCC értékekről tartalmaz információt, hanem azok változékonyságáról is.

Ezt a jellemvektort még azért célszerű alkalmazni, mert így már tetszőlegesen hosszú mintát vehetünk ki az adott számból, az nem fogja befolyásolni a hálóba bemenő adatok mennyiségét, hanem az már fix hosszúságú lesz.

BPM

Az MFCC-k mellett úgy gondoltuk jó mutató lehet még a *BPM*, vagyis *Beat Per Minute* érték, amely az adott szám tempójáról ad információt. Ezt az értéket kiszámoljuk szintén mind a három 5 másodperces mintára a számból, átlagoljuk őket, majd ezt adjuk meg a jellemvektor első elemeként.

Számítása időigényes a feldolgozás többi lépéséhez képest: a jellemvektor előállítása BPM számolás nélkül körülbelül 5-9 s, míg BPM számolással együtt ez az idő megnő körülbelül 50 s-ra. Hogy ezt ne kelljen minden egyes futtatáskor végigvárni, az egyes számokhoz tartozó jellemvektorokat az első számítás után egy CSV fájlba írjuk, így a következő futtatáskor elég ezeket a fájlokat beolvasni az időigényesebb számolás helyett.

Számítása

A számításhoz használt algoritmus Eric D. Scheirer munkája. [8] [9]

Egy beat a zenében nem más, mint egy a környezetéhez képest erős hangimpulzus. A BPM számításához mindössze össze kell számolni, hogy az adott mintában percenként mennyi ilyen impulzus van. Ha ez a szám magas, akkor a szám üteme gyors, ha kevés, akkor lassú. Az alapvető probléma a számításával az automatizált számításával annak a határértéknek a helyes megállapítása, ami fölött valami már beatnek számít, alatta pedig nem. Tovább nehezíti a problémát, hogy egy valós dalban több hangszer párhuzamosan szól, ami zavart okoz a bpm megállapításához.

Első lépésként tehát felbontjuk a mintát különböző frekvenciasávokra, az egyes jellemző hangszer-frekvenciáknak megfelelően. Ezek a szakirodalom szerint [0-200, 200-400, 400-800, 800-1600, 1600-3200] MHz-es sávoknak felelnek meg. Hogy ez a felbontás megtörténhessen, a jelet gyors Fourier-transzformáljuk, szűrjük a sávoknak megfelelően, majd visszatérünk időtartományba inverz Fourier-transzformáció segítségével.

A következő lépés az így kapott jel simítása, amiből majd jobban láthatóvá válnak a hirtelen ugrások, amelyek a beat-eknek felelhetnek meg. Ennek érdekében vesszük az előző lépésben kapott hat frekvenciasávot, és mindegyiket átengedjük egy alul áteresztő szűrőn. Ezzel tulajdonképpen megkapjuk a jelünk burkológörbéjét. Ennek a görbének számítjuk ki aztán az időbeni differenciáját, amiből kiszűrjük a negatív értékeket, hogy csak a pozitív változásokat kapjuk eredményül.

A folyamat utolsó lépése a legidőigényesebb, mivel itt különböző frekvenciájú fésű-szűrőket alkalmazunk az előző lépésben kapott jelen, és közben figyeljük, melyik frekvencián adja a legnagyobb energiát a szűrő és a jel szorzata. Ha ezt megtaláltuk, akkor az ehhez a frekvenciához tartozó tempó lesz a BPM értéke.

6. Neurális háló

A neurális hálózat a számítástechnikában elterjedt számítási modell, melyet az emberi, illetve az állati agy működése inspirált. Leggyakrabban gépi tanulásra, illetve alakfelismerésre alkalmazzák, de számos más - konkrét megoldással nem rendelkező - feladathoz is alkalmas.

Biológiai felépítés

A neurális hálózatok neuronokból épülnek fel, amelyek egyszerű számításokat végeznek (súlyozott összegzés). A neuronok közötti szinaptikus kapcsolatot a

kimenetek és a bemenetek megfelelő kapcsolata biztosítja. A kapcsolatban jelenlevő idegsejtek száma szerint beszélhetünk monoszinaptikus, illetve poliszinaptikus kapcsolatról, ahol az előbbi esetben egy neuron egy másikat informál, míg az utóbbinál egy neuron kimenete több neuronhoz is szolgálhat bemenetként. A szinapszisok lehetnek serkentő jellegűek, amin áthalad az ingerület, illetve lehetnek gátló jellegűek, amin elakad.

Mesterséges neurális hálózat

A mesterséges neurális hálózatokat hardveresen is próbálják megvalósítani, de a szoftveres megoldás a leggyakoribb. Alapvető építőeleme a mesterséges neuron, ami egy több bemenetű, egy kimenetű egyszerű számítási egység, amelyekből felépíthető egy teljes neurális hálózat. A neuronok között továbbított jelek binárisak, illetve 0 és 1 közötti értékek lehetnek. Az előző bekezdésben említett szinapszisoknak a hálózatban alkalmazott numerikus súlytényezők felelnek meg.

Csoportosításuk

Szervezettség szerint

- Rétegekbe szervezett
- Laza (szervezettség nélküli)

Neuronok kapcsolódása szerint

- Egyirányú (Feedforward Neural Network)
- Visszacsatolt (Recurrent Neural Network)

Időbeli viselkedés alapján

- Diszkrét idejű
- Folytonos idejű

Tanulási lehetőségek

A neurális hálózatok tanítására (leggyakrabban súlytényezők hangolásával) számos módszer létezik, amelyek a következők lehetnek:

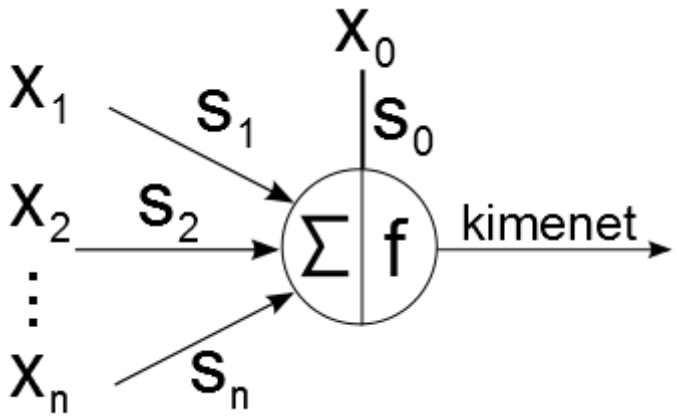
Felügyelt tanulás: Ez a tanítási módszer, mind a bemenő, mind a kimenő adatok ismeretében a súlytényezők hangolásával törekszik a hálózat által számított és a tanító bemenet közötti különbség minimalizálására.

Önálló tanulás: Néhány adat, illetve a minimalizálandó függvény ismert, a hálózat feladata az összefüggések, hasonlóságok keresése a bemenő adatok között. Ezt a módszert leggyakrabban becslési, illetve statisztikai feladatok megoldására alkalmazzák.

Megerősítő tanulás: Nem ismert a bemenethez tartozó kimenet, csak az dönthető el, hogy a kimeneti eredmény megfelelő-e, vagy nem. Arról, hogy mennyire jó az eredményünk nincs információnk. Leggyakrabban többdimenziós nemlineáris problémák megoldására alkalmazzák.

Mesterséges neuron modellek

A mesterséges neuronoknak több típusa létezik. Egy általános neuron felépítése az alábbi ábrán (11. ábra) látható:



11. ábra | Mesterséges neuron felépítése

McCulloch-Pitts neuron (MCP): Az MCP neuron bináris jelekkel működik, ha a bemeneteinek súlyozott (s_n) összege nagyobb mint az x_0 ingerküszöb, akkor a kimenete 1 lesz, egyébként 0. Ha $x_0 > 0$, akkor serkentő, $x_0 < 0$ esetén gátló típusú neuronról beszélünk. Az MCP neuron sajátsága, hogy a bemenetek azonos súlyúak. A neuron kimenete az alábbi egyenlettel határozható meg:

$$y = f\left(\sum_{k=0}^{n} w_i x_i\right)$$

A legegyszerűbb neuron, ami kizárólag bináris bemeneteket és kimeneteket kezel. Aktiváló függvénye az ideális relé jelleggörbéjével azonos.

Perceptron: A perceptronnal lehetőség van a bemenetek önálló súlyozására, vagyis az egységnyi bemenet az összes bemenet súlyozott összegében különböző súlyú szerepet játszhat (Hebb-szabály). A perceptron folytonos értékű bemenetet is kezel, illetve a bemenetek "erősebb" súlyozása is lehetséges vele.

A perceptron kimenete az alábbi összefüggéssel számolható:

$$z = \sum_{k=1}^{n} w_k x_k + b$$

Az aktiváló függvénye leggyakrabban az úgynevezett σfüggvény:

$$y = \sigma(z) = \sigma(\sum_{k=1}^{n} w_k x_k + b)$$

ahol $\sigma(z)$ leggyakrabban az ideális relé jelleggörbéjét leíró függvény:

$$f(x) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

illetve a másik gyakori aktiváló függvény, a szigmoid függvény:

$$P(t) = \frac{1}{1 + e^{-t}}$$

ADALINE (ADAptive LInear NEuron): Az ADALINE alapja az MCP neuron, a súlyokat, illetve ingerküszöböt és összegzést is alkalmaz, viszont nincs aktiváló függvénye. A súlytényezők módosítása a tanulási fázisban történik a bemenetek súlyozott összege alapján.

Felépítés, hálózatméretezés

A mesterséges neurális hálózat egy bemeneti rétegből, egy kimeneti rétegből, illetve tetszőleges számú rejtett rétegből áll. A szakirodalom szerint maximum 3-4 rejtett réteg elegendő bármilyen feladatra. Leggyakrabban egy rejtett réteget alkalmaznak, mivel több réteg használata nagyban növeli az igényelt számítási teljesítményt.

A megfelelő számú rejtett rétegbeli neuronok meghatározása lényeges feladat, viszont egyáltalán nem egyszerű. A túl kevés neuronból felépített hálózat nem közelíti megfelelően a célfüggvényt, míg a szükségesnél több neuron alkalmazásával túltaníthatjuk a hálózatot, amely esetén előfordulhat, hogy csak a tanító mintákkal fog működni. Általánosan elmondható, hogy a rejtett neuronok száma elsősorban a tanító minták számától, illetve a kimeneti függvény bonyolultságától függ.

Megvalósított hálózat

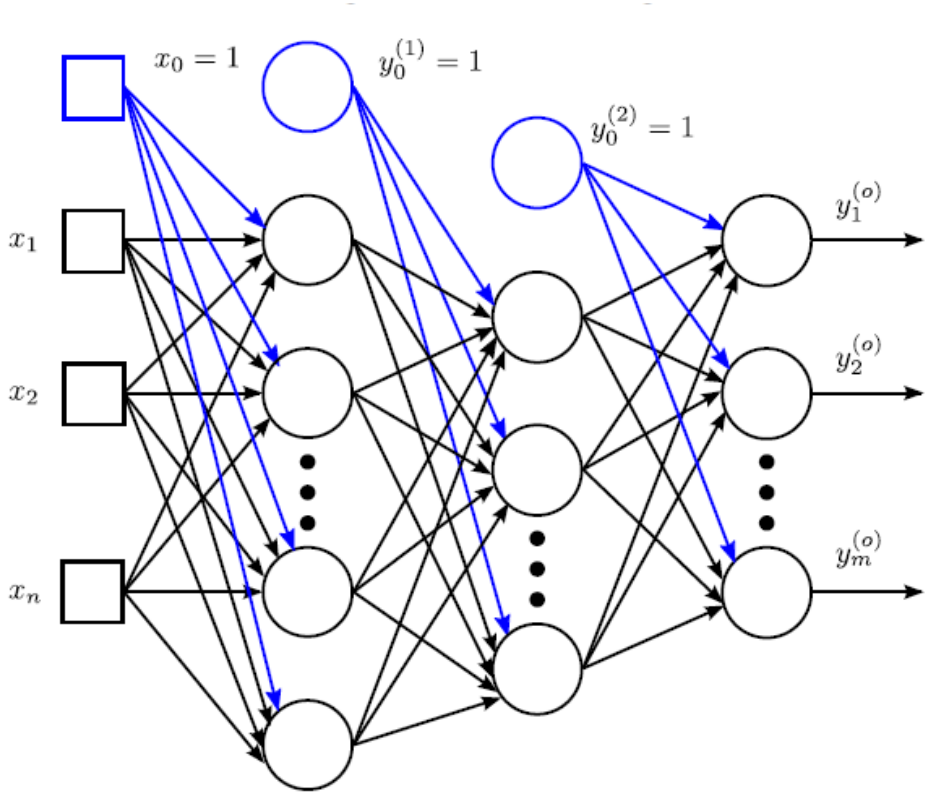
A feladat megoldásához egy előrecsatolt, egyirányú neurális hálózatot alkalmaztunk (Feedforward Neural Network), amit a PyBrain nevű Python modul segítségével hoztunk létre. A bemeneti réteg 180 neuronból áll, amelyek bemeneti értékeit a már fentebb tárgyalt módon állítjuk elő. A szakirodalom ajánlása alapján egy rejtett réteget alkalmaztunk, amely 90 neuront tartalmaz.

A hálózat négy kimeneti neuront tartalmaz, amik az általunk vizsgált 4 különböző műfajnak felelnek meg:

Pop	[1, 0, 0, 0]
Rap	[0, 1, 0, 0]
Rock	[0, 0, 1, 0]
Klasszikus	[0, 0, 0, 1]

Előrecsatolt hálózat kimenete

Egy előrecsatolt hálózat felépítése a következő ábrán (12. ábra) látható:



12. ábra | Előrecsatolt neurális hálózat (Feedforward network)

A továbbiakban egy 1 rejtett réteggel rendelkező, előrecsatolt hálózat neuronjainak számítási módszerét mutatjuk be, mátrix alakban:

$$\underline{z} = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z\_k \end{bmatrix} = \begin{bmatrix} w_{00} & w_{01} & w_{02} & \cdots & w_{0n} \\ w_{10} & w_{11} & w_{12} & \dots & w_{1n} \\ w_{20} & w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{k0} & w_{k1} & w_{k2} & \dots & w_{kn} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

, aminek a kimeneteit bevezetve az aktiváló függvénybe, megkapjuk a kimeneteket:

$$\underline{y} = \sigma(\underline{z}) \begin{bmatrix} 1 \\ \sigma(z_1) \\ \sigma(z_2) \\ \vdots \\ \sigma(z_k) \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

A tanító adatok ($\underline{t_i}$) és a hálózat $\underline{y}$ kimenetéből meghatározhatjuk az egyes bemenetekhez tartozó hibát:

$$\underline{e_i} = \sigma(\underline{z}) \begin{bmatrix} e_{i,1} \\ e_{i,2} \\ \vdots \\ e_{i,n} \end{bmatrix} = \underline{t_i} - \underline{y_i} = \begin{bmatrix} \underline{t}_{i,1} - \underline{y}_{i,1} \\ \underline{t}_{i,2} - \underline{y}_{i,2} \\ \vdots \\ \underline{t}_{i,n} - \underline{y}_{i,n} \end{bmatrix}$$

Hiba visszavezetés

A hálózatunk tanításához hiba visszavezetéses módszert alkalmaztunk, amely a felügyelt tanítási módszerek közé tartozik. Ez a módszer a szakirodalomban olvasottak szerint alkalmas különböző csoportosítási feladatokhoz.

Első lépésben a bemenő adatokhoz tartozó négyzetes hibákat számoljuk, majd azok alapján (gradiensükből) meghatározhatóak a neuronok bemeneteinek újabb súlyaik. Feltételezhető, hogy a rejtett réteg neuronjai is hibával számolnak, de előrecsatolt hálózatban ezekhez a hibákhoz nem férünk hozzá.

7. Eredmények vizsgálata

Az eredmények szemléltetéséhez MDS, azaz Multi-Dimensional Scaling módszert alkalmaztunk

Bemeneti adatok szemléltetése

A bemeneti vektorokat a jobb szemléltethetőség érdekében többdimenziós skálázás (Multidimensional scaling, MDS) segítségével, egy síkon ábrázoltuk. A többdimenziós skálázás egy olyan módszer, amellyel egy adathalmazt, egy síkban elhelyezkedő ponthalmazként tudunk ábrázolni, ahol a pontok közötti távolságok az eredeti adatok több dimenzióban mért távolsága közötti különbségekkel arányosak, a leképezés dimenziójától függő hibával (minél kisebb dimenzióba képezzük le a bemeneteket, annál nagyobb hibák jelentkeznek).

A bemeneteket több különböző módon határoztuk meg, a neurális háló hatékonyságának pontosabb vizsgálata érdekében. Az alábbi csoportokat vizsgáltuk:

Műfajonként egy előadó, egy albumról származó számai

Műfajonként több különböző, de műfajon belül hasonló előadók számai

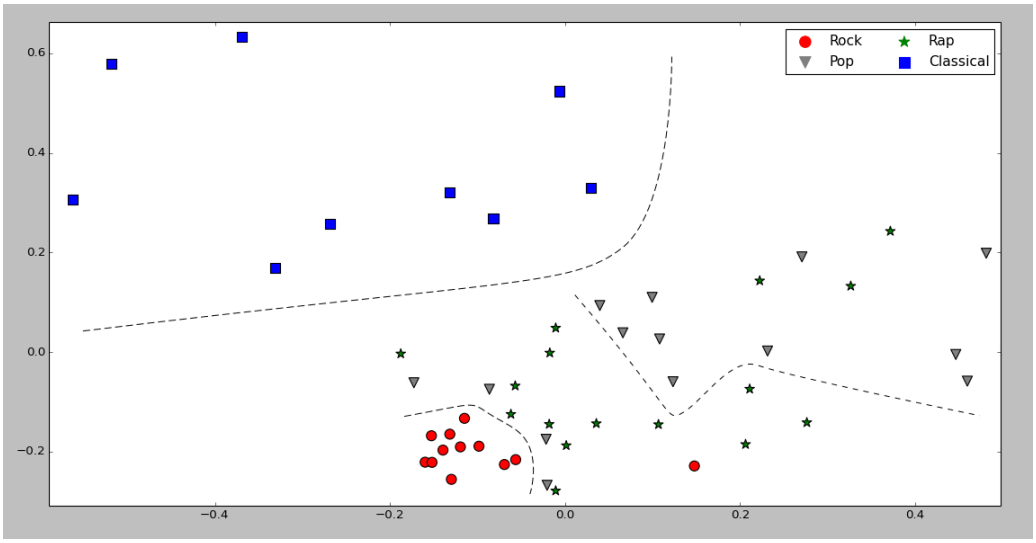
Műfajonként több különböző – műfajon belül is eltérő – előadó számai

Az általunk vizsgált összes zeneszám

Az MDS alkalmazása után a következő ábrákat kaptuk:

Műfajonként egy előadó, egy albumról származó számai

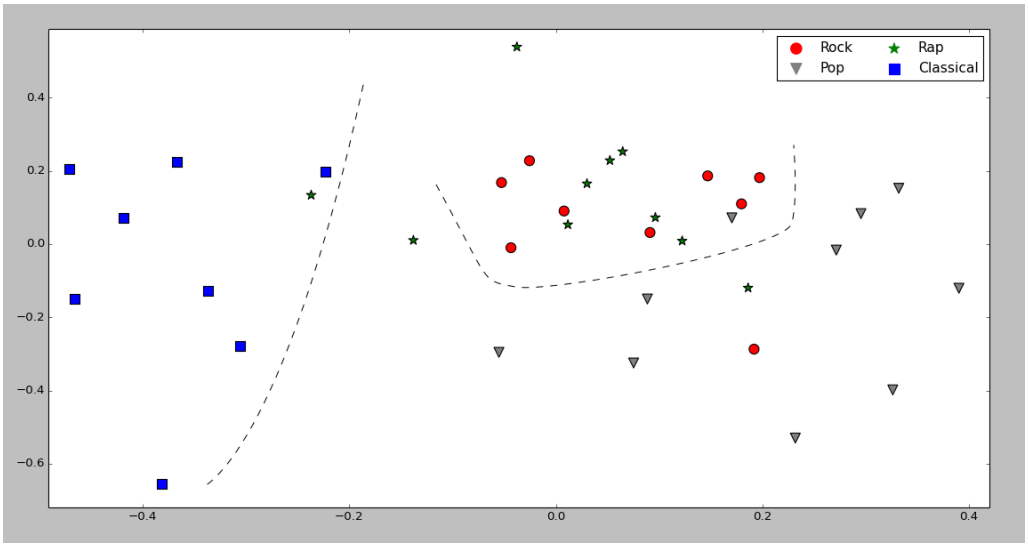
Az egy műfajon belüli, egy előadótól származó minták esetén, az alábbi ábrán (13. ábra) látható módon, viszonylag jól csoportosítható halmazokba lehet sorolni az egyes zeneszámokat (szaggatott vonallal jelölve a határokat). A pop illetve a rap esetében megfigyelhető hogy, azonos előadóktól származó minták esetén is összemosódnak a halmazok határai.



13. ábra | MDS egy előadó, egy albumról származó számai esetén

Műfajonként több különböző, de műfajon belül hasonló előadók számai

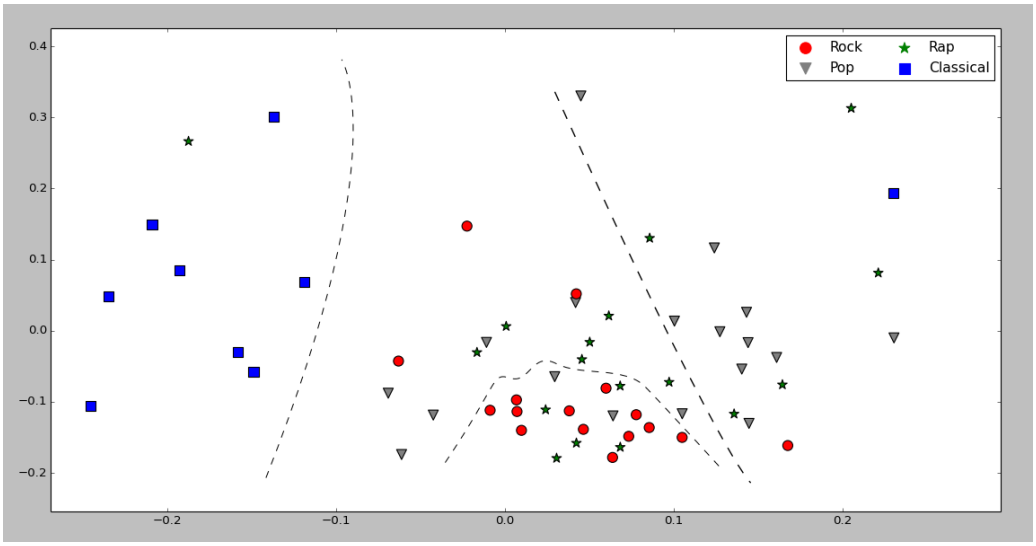
Amikor műfajonként több – a műfajon belül hasonló – zeneszámot osztályoztunk, már kevésbé kaptunk jól elhatárolható eredményeket. Mint az alábbi ábrán (14. ábra) is látható, a klasszikus, illetve a pop műfaj még jól elhatárolható (bár hibás helyre sorolt elemek találhatóak a halmazokban), a rock és a rap már egyáltalán nem választható szét egyszerű görbékkel.



14. ábra | MDS műfajonként több különböző, műfajonként hasonló előadók esetén

Műfajonként több különböző – műfajon belül is eltérő – előadó számai

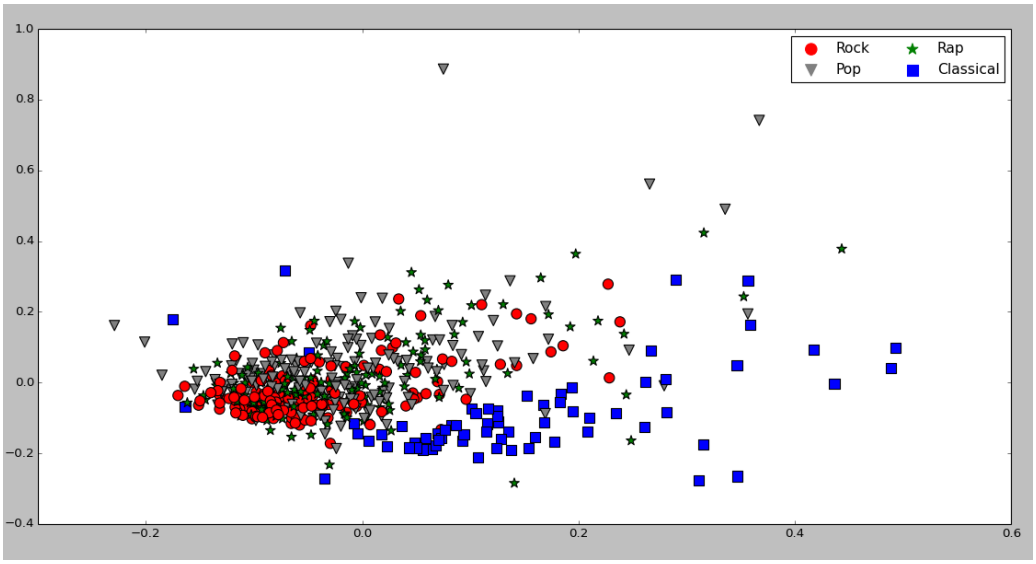
A különböző bemeneti adatok vizsgálatában, a következő lépésként, megvizsgáltuk műfajonként több különböző –műfajon belül is eltérő – előadótól származó minta esetén a többdimenziós skálázás által szolgáltatott eredményeket, amelyen jól látható, hogy a klasszikus műfaj még mindig jól elkülöníthető a többitől, viszont a rock, a pop, illetve a rap esetében már komoly összemosódás látható (15. ábra), bár nagyobb hibával még mindig osztályokba sorolható.



15. ábra | MDS több különböző, műfajon belül is eltérő előadó esetén

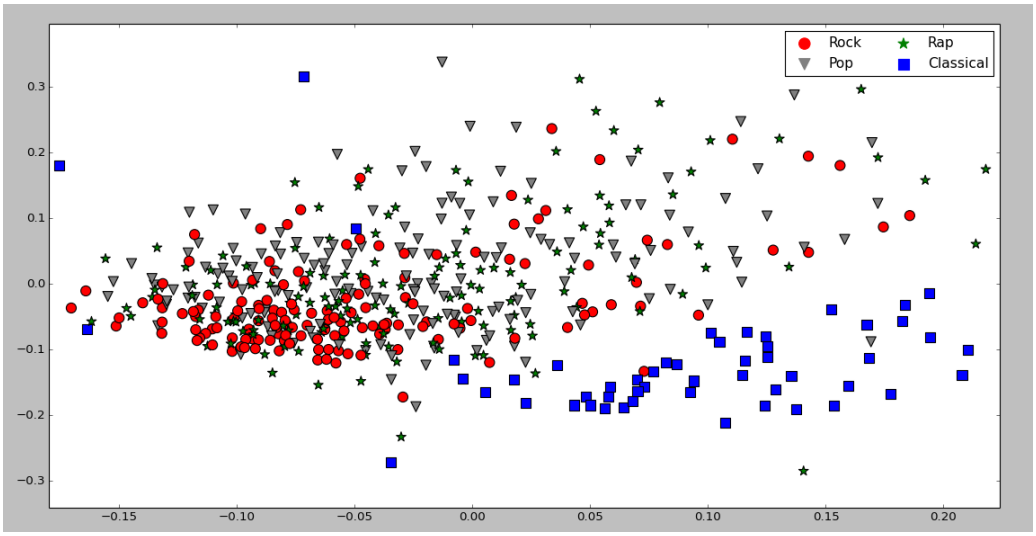
Az általunk vizsgált összes zeneszám

Az utolsó vizsgált mintában, az összes általunk eredetileg is vizsgálatra szánt számot szerepeltettük, majd ezt is MDS segítségével ábrázoltuk (16. ábra) egy síkon, amely a következő ábrán látható:



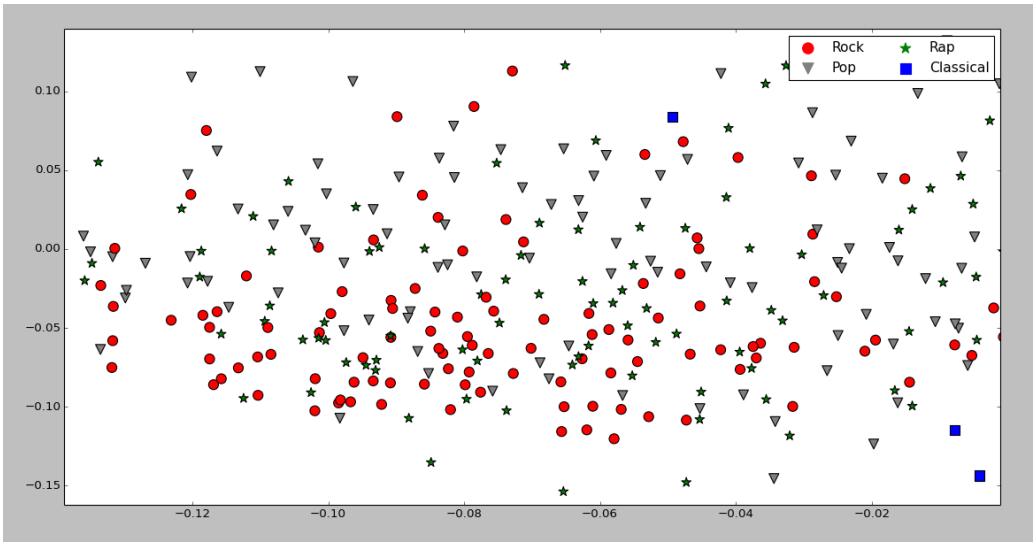
16. ábra | MDS az általunk vizsgált összes zeneszám esetén

Már első ránézésre is látható, hogy a klasszikus műfaj jól láthatóan elkülönül a másik három vizsgált műfajtól, viszont a többi osztály határainak a meghatározása már nem lehetséges egyszerű görbékkel (17. ábra).



17. ábra | MDS az általunk vizsgált összes zeneszám esetén (nagyított) – 1

Tovább nagyítva a fenti ábrát, még jobban látható a műfajok keveredése (18. ábra).



18. ábra | MDS az általunk vizsgált összes zeneszám esetén (nagyított) – 2

Külön figyelmet érdemel, hogy a klasszikus műfaj minden vizsgált esetben jól megkülönböztethetőnek bizonyult, amely a műfajra jellemző, tradicionális hangszerek használata, illetve az egyéb műfajokban előszeretettel alkalmazott elektronikus hangszerek mellőzése miatt lehetséges. Ezen indokok miatt már a mel-frekvencia kepsztrum számítás eredményeiből – illetve annak dinamikus tulajdonságaiból – látható, hogy sokkal nagyobb energiaváltozások lépnek fel ebben a műfajban, aminek következtében a többdimenziós skálázás eredménye is nagyobb távolságokra helyezi el az ilyen típusú darabokat.

A kapott eredmények jól szemléltetik, hogy minél több műfajból, és több előadótól vett vektort választunk bemenetként, annál bonyolultabb műveletek szükségesek a halmazokba rendezéshez. Az egyre zavarosabb kép kialakulásához nagymértékben hozzájárul a stílusok határainak összemosódása, illetve a különböző alműfajok hasonlósága. A mai zenei világban rengeteg stílusirányzat létezik, amelyeknek a megkülönböztetése sokszor gyakorlott zenészeknek is problémát okoz. Erre ad bizonyítékot az is, hogy az interneten található zenemegosztó oldalak, illetve zeneáruházak kínálatában is találhatóak olyan zeneszámok, amelyek különböző osztályokba vannak sorolva az egyes oldalakon.

Többdimenziós skálázás (Multidimensional scale, MDS)

Az többdimenziós skálázás különböző adathalmazok hasonlóságának szemléltetésére alkalmas statisztikai módszer. A módszer lényege, hogy egy adathalmazok között definiált távolságfüggvény alapján egy N dimenziós

térben ábrázoljuk a halmazokat. Az általunk használt esetben N=2 értéket választottunk, így adathalmazonként két koordinátát kaptunk, amely szépen ábrázolható egy X-Y diagramon.

A számítás első lépéseként elő kell állítanunk egy távolság mátrixot, amely az adathalmazok közötti Euklideszi távolságokat tartalmazza. Ehhez az alábbi összefüggést használjuk fel:

$$d_{1,2} = \sqrt{\left(a_{1_1} - a_{2_1}\right)^2 + \left(a_{1_2} - a_{2_2}\right)^2 + \cdots + \left(a_{1_n} - a_{2_n}\right)^2}$$

ahol a_{1_n} az egyik halmaz n-edik tagja, a_{2_n} pedig a másik halmaz azonos sorszámú tagja (n-edik tagja). Az összefüggést az összes adathalmazra alkalmazva a következő mátrixot kapjuk:

$$\boldsymbol{D} = \begin{bmatrix} d_{1_1} & \dots & d_{1_n} \\ \vdots & \ddots & \vdots \\ d_{n_1} & \dots & d_{n_n} \end{bmatrix}$$

ahol $d_{i_i} = 0$

Következő lépésként a $\boldsymbol{D}$ mátrixot elemenként négyzetre emeljük, majd két irányból középpontosítjuk a $\boldsymbol{J}$ mátrix segítségével:

$$\boldsymbol{D_c} = -\frac{1}{2}\boldsymbol{J}\boldsymbol{D}^2\boldsymbol{J}$$

A $\boldsymbol{J}$ mátrixot az alábbi módon kapjuk meg:

$$\boldsymbol{J} = \boldsymbol{I} - n^{-1}\boldsymbol{I}\boldsymbol{I}^{-1}$$

Ahol $\boldsymbol{I}$ az egységmátrix, n pedig az összes szerepeltetett adathalmaz száma.

A harmadik lépésben a $\boldsymbol{D_c}$ mátrix legnagyobb m sajátértékét ($\lambda_1 \dots \lambda_m$), illetve az azokhoz tartozó sajátvektorokat ($e_1 \dots e_m$) határozzuk meg.

Végül az m-dimenzós (esetünkben $m = 2$) koordináták előállnak a következő összefüggés használatával:

$$\boldsymbol{X} = \boldsymbol{E}_m \boldsymbol{\Lambda}_m^{\frac{1}{2}}$$

, ahol $\boldsymbol{E}_m$ az m darab sajátvektor, $\boldsymbol{\Lambda_m}$ pedig a $\boldsymbol{D_c}$ mátrix sajátértékeiből álló diagonális mátrix.

Végeredményként az $\boldsymbol{X}$ mátrix oszloponként tartalmazza az ábrázoláshoz szükséges koordinátákat.

Neurális háló eredményei

Műfajonként egy előadó, egy albumról származó számai

Amikor egy műfajba egy előadótól választottuk a bemeneti adatokat, 81%-os pontossággal sikerült megállapítani a zeneszámok műfajait. Ez az érték megfelel az irodalomkutatásunk során talált értékeknek. Az eredmények az 1. táblázatban láthatóak. A táblázat sorai az adott szám valódi műfaját jelölik, míg az oszlopok az algoritmus által kiadott eredmény. A cellákban szereplő számok a gyakoriságot jelölik.

	Pop	Rap	Rock	Klasszikus	Pontosság
Pop	8	1	0	1	80%
Rap	1	7	2	0	70%
Rock	1	1	9	0	82%
Klasszikus	1	0	0	5	83%

1. táblázat | Műfajonként egy előadó eredményei

Műfajonként több különböző, de műfajon belül hasonló előadók számai

A második kísérletnél, a bemeneti adatok már több előadótól származtak az adott műfajon belül, viszont úgy választottuk meg az előadókat, hogy azok a műfajon belül hasonlóak legyenek egymáshoz. Az átlagos pontosság ebben az esetben 74,6 %, amit a több különböző előadótól származó mintával indokolhatunk. Eredmények a 2. táblázatban találhatóak.

	Pop	Rap	Rock	Klasszikus	Pontosság
Pop	15	2	2	1	75%
Rap	3	13	3	0	68%
Rock	2	2	16	1	76%
Klasszikus	1	1	0	9	82%

2. táblázat | Műfajonként több hasonló előadó eredményei

Műfajonként több különböző – műfajon belül is eltérő – előadó számai

A harmadik esetben, már olyan előadókat választottunk bemenetként, amelyek között már a műfajon belül is hallható különbségek adódnak, másik alműfajba tartoznak. Ilyenkor a pontosságunk tovább csökkent, körülbelül 67,5 %-os pontosságot sikerült elérni (3. táblázat). Ez a jelentős különbség szerintünk a még nagyobb műfajbeli eltéréseknek köszönhető.

	Pop	Rap	Rock	Klasszikus	Pontosság
Pop	15	3	2	1	71%
Rap	5	15	2	2	63%
Rock	4	3	16	2	64%
Klasszikus	1	1	1	10	77%

3. táblázat | Műfajonként több különböző előadó eredményei

Az általunk vizsgált összes zeneszám

Az utolsó esetben a bemeneti adatokat nem válogattuk, a már említett összes előadó, több albumából származó számokat használtuk bemenetként. Ebben az esetben a pontossága az osztályozásnak már csak 58 %-os volt (4. táblázat), amit a nagyobb, illetve jóval változatosabb bemeneti mintának tudhatunk be.

	Pop	Rap	Rock	Klasszikus	Pontosság
Pop	23	5	6	2	64%
Rap	10	18	9	2	46%
Rock	6	4	24	5	62%
Klasszikus	1	3	1	12	71%

4. táblázat | Az összes vizsgált zeneszám eredményei

Értékelés

Az eredmények alapján arra a következtetésre jutottunk, hogy kis bemeneti mintaszám, illetve az adott stíluson belüli kis változatosság esetén a módszer jól használható, viszont ahogy egyre több, illetve egyre szélesebb rétegből származó zenéket vizsgálunk, a pontosság egyre csökken. A bemeneti adatok kinyerési módszerének fejlesztésével, illetve több – lehetőleg az adott stílusra minél jobban jellemző – tulajdonság figyelembevételével a pontosság növelhető.

8. Továbbfejlesztési lehetőségek

Látható, hogy a neurális hálózatok alkalmasak lehetnek a különböző stílusú zeneszámok megkülönböztetésére, ehhez viszont még további fejlesztések szükségesek. A továbbiakban tervezzük azt megfigyelni, hogy az egyes paraméterek megváltoztatása milyen hatással van a kapott végeredményekre. Így például érdekes lenne azzal próbálkozni, hogy milyen hatása van annak, ha számonként több rövidebb, vagy kevesebb, de hosszabb szegmenseket vennénk. Esetleg az változtat-e az eredményeken, ha módosítjuk az MFCC számítás előtti daraboláskor a darabok átfedését vagy hosszát. Ha ezeket az értékeket sikerülne megfelelően optimalizálni, akkor úgy véljük még pontosabb eredményeket érhetnénk el.

Kérdés továbbá, hogy ha a szegmensek kiválasztásakor pont olyan szakaszt választunk, ahol hirtelen választás van a dallamban (pl. refrén vége), az okoz-e zavart a rendszerben. Ennek az esélye mindenesetre csökkenthető lenne azzal, ha először az egész számon végigcsinálnánk egy feldolgozó eljárást, és a szegmenseket olyan helyről vennénk, ahol például a tempó viszonylag állandó.

A neurális hálózat struktúrájának módosítása is pontosabb eredményeket szolgáltathat, így további hálózati struktúrák alkalmazása is szükséges lehet, az

előrecsatolt hálózaton kívül. Ehhez a rejtett rétegen belüli visszacsatolások megvalósítása megfelelő megoldás lehet, mivel így a rejtett neuronok hibái is elérhetővé válnának, amellyel tovább pontosítható a kimenet.

Az osztályozó rendszer tovább pontosítható lenne fuzzy logika bevezetésével, amellyel már nem az adott szám konkrét stílusát tudjuk megállapítani, hanem a különböző stílusokhoz való tartozás mértékét is kifejezhetjük. Ezzel a fejlesztéssel a mai zenei világ, kisebb, kevésbé népszerű stílusirányzataira is következtethetünk, az ismertebb stílusok jellemzőinek jelenlétének függvényében, így egy még pontosabb, szélesebb körben alkalmazható módszerhez juthatnánk.

Felmerülhet a kérdés továbbá, hogy a kiszámolt BPM érték mennyire segíti a műfajokba rendezést, mivel az összes kimeneti műfajban léteznek lassabb és gyorsabb tempójú művek is. Ennek kihagyásával is érdemes lenne tesztelni az elkészült rendszert, ellenőrizni, valóban szükséges-e ez az érték is.

További fejlesztési lehetőség lehet, ha a tanításhoz felhasznált számoknak a stílusát nem mi adjuk meg kézzel, hanem internetről szedjük össze. Ehhez például használható a *last.fm* API-ja, amivel az egyes meghallgatott számokról lehet olyan adatokat kinyerni, mint az egyes műfajokhoz tartozás mértéke százalékban. Ezeket az adatokat aztán fel tudnánk használni a rendszer pontosabb betanításához.

9. Összefoglalás

A dolgozatban bemutattunk egy módszert, amely képes ismeretlen zeneszámokból, pusztán a rögzített jelet használva jó eséllyel képes megmondani a szám műfaját. Ehhez a számból képeztünk MFCC-ket, deltákat, delta-deltákat illetve tempót, amit aztán a rendszer neurális hálójának adtuk át bemenetként.

Az eredmény négy műfajba (pop, rock, rap, klasszikus) sorolás esetén a véletlenszerű 25%-hoz képest, jelentős sikernek mondható. A tanításhoz és a teszteléshez használt számlista összeállításától függően a sikerességi esély 58 % és 81 % között változik, ami feltételezésünk szerint tovább javítható a neurális háló bemeneti értékeinek számolásához használt paraméterek további finomhangolásával.

Ez az eredmény sajnos még nem elég jó ahhoz, hogy alkalmazható legyen egy automatikusan lejátszási listákat összeállító programhoz, viszont a fentebb leírt fejlesztéseket elvégezve, feltételezzük, hogy sikerül elérni egy 80-90 %-os pontosságot, ami már használható lenne erre a célra is.

10. Irodalomjegyzék

[1] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35

[2] US Patent Number 7003515 — Consumer item matching method and system

[3] Michael Haggblade, Yang Hong, Kenny Kao: Music Genre Classification

[4] Changsheng Xiang, ZiYing Zhou: A New Music Classification Method based on BP Neural Network, International Journal of Digital Content Technology and its Applications. Volume 5, Number 6, June 2011

[5] Kshitiz Kumar, Chanwoo Kim, Richard M. Stern: Delta-Spectral Cepstral Coefficients for Robust Speech Recognition

[6] Meinard Müller: Information Retrieval for Music and Motion, Springer, p. 65., 2007.

[7] S. S. Stevens, J. Volkmann, E. B. Newman: A Scale for the Measurement of the Psychological Magnitude Pitch; Journal of Acoust. Soc. Am. 8, p. 185, 1937

[8] Eric D. Scheirer: Tempo and beat analysis of acoustic musical signals, 1998 Acoustical Society of America

[9] Chang, Nazer, Uppuluri, Verret: Beat Detection Algorithm, https://www.clear.rice.edu/elec301/Projects01/beat_sync/beatalgo.html

[10] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk: Speech Recognition using MFCC, International Conference on Computer Graphics, Simulation and Modeling, 2012 Pattaya (Thailand)

[11] Katariina Mahkonen: Mel frequency cepstral coefficients (MFCCs)

[12] Florian Wickelmaier: An Introduction to MDS, Sound Quality Research Unit, Aalborg University, Denmark, May 4, 2003

[13] Wei HAN, Cheong-Fat CHAN, Chiu-Sing CHOY and Kong-Pang PUN: An Efficient MFCC Extraction Method in Speech Recognition

[14] S. H. Kim: CALCULATING BPM COEFFICIENTS WITH GREEN'S RECIPROCATION THEOREM, Proceedings of the 2001 Particle Accelerator Conference

[15] Miguel Alonso, Bertrand David , Gäel Richard: TEMPO AND BEAT ESTIMATION OF MUSICAL SIGNALS

[16] Avery Li-Chun Wang: An Industrial-Strength Audio Search Algorithm

[17] Aradi Petra: Biomechatronikai modellezés és szimuláció tárgy diái, neurális hálózat fejezet, BME

[18] S. H. Kim: CALCULATING BPM COEFFICIENTS WITH GREEN'S RECIPROCATION THEOREM, Proceedings of the 2001 Particle Accelerator Conference

[19] Miguel Alonso, Bertrand David , Gäel Richard: TEMPO AND BEAT ESTIMATION OF MUSICAL SIGNALS

[20] Avery Li-Chun Wang: An Industrial-Strength Audio Search Algorithm

11. Ábrajegyzék

12. Táblázatjegyzék