

ZENEI HANGOK PÓTLÁSA NEURÁLIS HÁLÓK SEGÍTSÉGÉVEL

BENEDEK GÁBOR – HORVÁTH CSILLA

Tanulmányunk a neurális hálók elméleti hátterét mutatja be három részben. Az első részben a neurális hálózatok elméletéről szólnunk, majd a második részben bemutatjuk, hogyan sikerült magyar népdalok esetében az elveszett hangok pótlására alkalmazni az eljárást. Végezetül összegezzük a cikkünk eredményeit. Bemutatjuk, hogy az eljárás még egy viszonylag egyszerű neurális háló esetén is segítséget nyújthat a magyar népdalok elveszett hangjainak pótlására.

TÁRGYSZÓ: Neurális hálók. Zenetudomány. Zenei hangok pótlása.

Sok zenemű hiányosan, vagy esetleg befejezetlenül maradt ránk. A legismertebb példa a hiányos kottaképre *Mozart* Requiemje, melynek befejezése előtt a zeneszerző meghalt, ezért a mű zenekari átíratát a zeneszerző tanítványa, *Franz Xaver Süssmayer* fejezte be és egészítette ki a Sanctus, a Benedictus és az Agnus Dei tételekkel. *Mozart* „nagy” c-moll miséje szintén hiányos formában maradt ránk, ám ebben az esetben a hiányosság oka kevésbé ismert. Avagy gondoljunk csak *Johann Sebastian Bach* Márkpassiójának kottájára, amely a második világháborúban a tűzvész martaléka lett, s melynek rekonstruálására *Dinnyés Soma* vállalkozott 1996-ban.

A hiányzó hangjegyek pótlása nem egyszerű feladat, hiszen minden zeneműnek sajátos szerkezete, hangzásvilága van, melyet a mű és a szerző hosszas tanulmányozása során ismerhetünk meg. A hiányzó hangjegyek pótlása már azért is kihívás, mert egyetlen hangjegy megváltoztatásával az egész zenemű hangsúlya és dallamvilága megváltozhat.

A leggyakrabban alkalmazott eljárás, amikor szakértőket kérnek fel arra, hogy speciális ismereteik felhasználása segítségével próbálják kitalálni, hogy vajon a szerző milyen zenei hangot, milyen zenei motívumot gondolt eredetileg a hiányzó hangjegyek helyére. Ebben az esetben a szakértő a mű ránk maradt résznek és a zeneszerzőről összegyűjtött egyéb információk, esetleg további műveiből nyerhető motívumok felhasználásával próbálja pótolni a hangjegyeket. A szakértő a zeneszerzőtől származó különböző zenerészletekből megtanulja a zeneszerző által gyakran használt motívumokat és képeket, felfedi, hogy milyen (talán maga a zeneszerző számára sem tudatos) szabályszerűségek találhatók meg művekben, és ezt a megtanult sémát felhasználva egészíti ki a művet.

Egy zeneműben vagy egy dallamban található szabályszerűségeket megfelelő statisztikai-ökonometriai módszerek segítségével is meg tudjuk ragadni. Célunk ebben a dolgozatban, hogy bemutassuk ennek az elméleti lehetőségnek egy gyakorlati megvalósítását, és pedig azt, hogyan lehet magyar népdalokban hiányzó hangokat neurális hálózatok segítségével pótolni. A neurális hálóra mint ökonometriai becslő eljárásra két okból esett a választásunk. Az első, hogy a neurális hálók működése közvetlen párhuzamba állítható a szakértő tanulási folyamatával, segítségével meg tudjuk fogni a feltételesen nemlineáris kapcsolatokat a különböző, egymást követő hangjegyek között. A második ok pedig az, hogy korábbi projektek kapcsán személyesen tapasztaltuk a neurális hálózatok ilyen téren történő alkalmazásának sikerességét. (2000-ben a hannoveri világkiállítás írországi pavilonjában az egyik érdekesség egy olyan zenemű volt, amelyet neurális hálózat és mesterséges intelligencia segítségével, kreatív emberi beavatkozás nélkül hoztak létre. Ennek a neurális hálónak a tanításához számos ír népdal digitális változata volt szükséges. Lásd: <http://www.softday.ie/info.htm>)

NEURÁLIS HÁLÓK

A neurális háló két objektumból áll; *csomópontokból (neuronok)* és az őket összekötő *élekből (szinapszis)*. A bejövő információ (inger, impulzus) egy-egy neuronba érkezik. Amennyiben az adott inger egy *küszöbértéket* meghalad, a neuron továbbküld egy jelet abba a neuronba, amellyel összeköttetésben van. Egy ilyen neuronba több neuron is küldhet impulzust. Ha ezek összessége meghaladja e neuron ingerküszöbét, akkor ez a neuron is továbbküld egy impulzust a következő neuronba stb. Végül az impulzus eljut néhány végső neuronba, s attól függően, hogy melyikbe jutott el, úgy kötünk valamilyen válaszreakciót az adott inputhoz. Tanulás során az intenzíven használt szinapszisok megerősödnek, a keveset használtak pedig megszűnnek. Az emberi agy tanulásának ezt a mechanizmusát másolja le a számítógépes modell. Egyes elemeket azonban meg kellett változtatni, ugyanis még a leggyorsabb számítógépek sem képesek néhány száz neuronnál nagyobb hálózatot modellezni – szemben az aggyal, amelyben milliárdnyi neuron működik. Az első változtatás az volt, hogy az egyes neuronok több más neuronnak is adhatnak impulzust, akár visszafelé is. A második módosítás, hogy minden szinapszisnak meghatározott az impulzusátadási hatásfoka, azaz meghatározott erővel (*súlyal*) továbbítja az információt. A harmadik változtatás az, hogy az ingereket fogadó neuronokat különböző mértékű (intenzitású) ingerek érhetik, továbbá az ingerek továbbítására nem egy egyszerű küszöbérték-meghaladási kritérium létezik, hanem bonyolult függvények segítenek meghatározni a kimenő impulzus mértékét. Kivétel általában a beérkező ingereket fogadó neuron, ahonnan az inger „egy az egyben” továbbítódik. Végezetül, a tanulás mechanizmusát matematikai statisztikai módszerek szabályozzák, melyek közül a legismertebb és leginkább használatos a *hiba-visszaterjesztési (back-propagation)* eljárás.

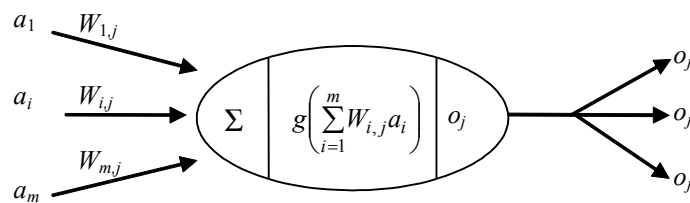
Vizsgáljuk meg először a neurális háló alapegységét, a neuront. Ezt az egységet gyakran nevezik *számítási* (vagy *processzáló*) elemnek is, mivel itt zajlanak le a nemlineáris számítási folyamatok. A neuron gyakorlatilag nem más, mint egy függvény, ami a bemeneti jeleket egyetlen kimeneti jellé alakítja. E függvényeknek általában meghatározott alakjuk van. A neuron előbb a bemeneti jeleket összegzi valamilyen módon (*additív, multiplikatív*), majd egy függvény segítségével transzformálja ezeket. A leggyakrabban alkalmazott függvények az úgynevezett ugrásfüggvény /1/ és a szigmoid függvény /2/:

$$U(x,t) = \begin{cases} 1, & \text{ha } x \geq t \\ 0, & \text{ha } x < t \end{cases} \quad /1/$$

$$S(x) = \frac{1}{1 + e^{-x}}. \quad /2/$$

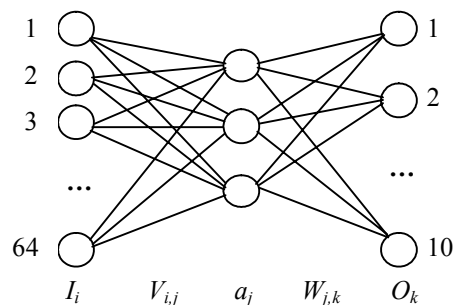
Az összegzésre általában additív módszert alkalmaznak. A neuronokat összekötő éleken súlyok szerepelnek, amelyek az adott jeleket felerősíthetik vagy gyengíthetik. Ha tehát egy j neuronba m számú más neurontól érkezik W_{ij} súlyokon keresztül a_i jel, akkor ezt a következőképpen ábrázolhatjuk:

1. ábra. A neuron működése



Ha többrétegű hálózatot kívánunk létrehozni, akkor a bemeneti és a kimeneti rétegek közé egy (vagy több) *köztes* vagy *rejtett réteget* kell beékelnünk. A 2. ábrán egy olyan neurális hálózat struktúrája látható, ahol a bemeneti réteg 64 darab, a köztes réteg 3 darab, a kimeneti réteg pedig 10 darab neuronból áll. Mivel a bemeneti neuronok nem számítógépes egységként vannak a hálózatban, ezért a hálózat egy olyan függvény, amelyben 13 nemlineáris függvény van meghatározott súlyokkal összekapcsolva. Mivel a súlyok első csoportja a 64 bemenetből a 3 köztes neuronba juttatja el a jelet, ezért itt összesen 192 súly szerepel. A 3 köztes neuronból a 10 kimeneti neuronhoz összesen 30 súly tartozik. Ez azt jelenti, hogy ha például szigmoid függvény segítségével dolgozunk, akkor a 2. ábrán egy 222 szabad paraméterrel rendelkező bonyolult nemlineáris függvényről van szó.

2. ábra. Többrétegű neurális háló



Ha még azt is hozzá vesszük, hogy minden processzáló egységhez biztosítani szoktak egy konstans neuront is, ez azt jelenti, hogy a szabad paraméterek száma további 13-mal növekszik. Az ábrán látható neurális háló analitikusan is kifejezhető:

$$\begin{aligned} a_j &= g\left(C_j^a, \sum_{i=1}^{64} V_{i,j} I_i\right) = \frac{1}{1 + e^{-\sum V_{i,j} I_i + C_j^a}} \\ O_k &= g\left(C_k^o, \sum_{j=1}^3 W_{j,k} a_j\right) = \frac{1}{1 + e^{-\sum W_{j,k} a_j + C_k^o}}. \end{aligned} \quad /3/$$

Egyrétegű hálózat csak lineárisan szeparálható függvényt képes reprezentálni. Ha egy neurális hálózat legalább egy rejtett réteget tartalmaz, akkor tetszőleges folytonos függvény reprezentálására, ha egy hálózat két rejtett réteggel rendelkezik, akkor tetszőleges függvény reprezentációjára képes.

A neurális hálózatok alkalmazásának kritikus pontja a *tanulás*. Ez az a folyamat, amikor a minták által megadott inputokhoz (I_i) és outputokhoz (O_k) meg akarjuk találni a tökéletes hálózati struktúrát (*topológiát*) és súlyrendszert. (Vegyük észre, hogy itt tulajdonképpen egy nemlineáris regresszió építéséről van szó.)

A többretegű neuronhálózatok leggyakrabban alkalmazott súlytanulási algoritmus a *hiba-visszaterjesztés* módszer. A módszer lényege az, hogy úgy kell megváltoztatni a súlyokat, hogy az adott minta esetében az elkövetett hiba a lehető legkisebb legyen. Mivel több réteg működik együttesen, ezért a hibát hátulról kell visszaterjeszteni az egyre előbbi súlyokig. Azaz először a rejtett és output rétegek közötti súlyokat kell módosítani:

$$W_{j,k}^{t+1} = W_{j,k}^t + \lambda a_j (T_k - O_k) g' \left(\sum_j W_{j,k} a_j \right), \quad /4/$$

ahol g -vel jelöltük a processzáló függvényt, amelyet *aktiváló* vagy *reakciófüggvénynek* is nevez a szakirodalom. Látható, hogy a szigmoid függvény alkalmazása azért kedvező, mert könnyen számítható a deriváltja, ugyanis $g' = g(1 - g)$. A köztes és a bemeneti neuronok között szintén valami hasonló szabályt kellene megállapítani a súlyok megváltoztatása érdekében. Itt azonban nem egyszerű az elkövetett hiba mértékének meghatározása, hiszen amíg a kimeneti értéket össze lehetett hasonlítani a tényadattal, addig itt nem áll rendelkezésre a tényadathoz hasonló mennyiség. Az ötlet az, hogy minden j rejtett neuron valamilyen mértékben hozzájárul minden k kimeneti csomópontban elkövetett hibához. Ezért a már meghatározott kimeneti hibaértékeket a csomópontok közötti kapcsolat szorossága, azaz a súlyok függvényében osztjuk szét, és visszaterjesztjük azokat a rejtett csomópontokhoz. Azaz a j -edik rejtett neuron hibatényezője:

$$\Delta_j^t = \sum_k \left(W_{j,k}^t (T_k - O_k) g' \left(\sum_j W_{j,k}^t a_j \right) \right), \quad /5/$$

és így a bemeneti és a köztes neuronok közötti súlyok módosítása már hasonló a /4/ képlethez:

$$V_{i,j}^{t+1} = V_{i,j}^t + \lambda I_i \Delta_j^t g' \left(\sum_i V_{i,j} I_i \right). \quad /6/$$

Nézzünk a hiba-visszaterjesztési algoritmus matematikája mögé!

Tétel. A /4/ és /6/ képleteken alapuló hiba-visszaterjesztési eljárás nem más, mint egy gradiens-módszer segítségével történő numerikus optimalizálás.

Bizonyítás. Elegendő azt megmutatnunk, hogy alkalmas célfüggvényt választva a /4/ és /6/ képletekben a λ paraméter után szereplő kifejezés nem más, mint az alkalmasan választott célfüggvény súlyonként vett parciális deriváltja. A célfüggvény megválasztásánál a legkisebb négyzetek módszerét alkalmazzuk a tényadatok és neurális háló output adatainak összesímitására:

$$E = \frac{1}{2} \sum_k (T_k - O_k)^2. \quad /7/$$

Nyilvánvaló, hogy E minimalizálásával a legjobban illeszkedő hálózatot alakítjuk ki. Írjuk fel E -t a súlyok függvényében egy-egy rejtett réteggel rendelkező hálózati topológia esetén:

$$E = \frac{1}{2} \sum_k \left(T_k - g \left(\sum_j W_{j,k} a_j \right) \right)^2 = \frac{1}{2} \sum_k \left(T_k - g \left(\sum_j W_{j,k} g \left(\sum_i V_{i,j} I_i \right) \right) \right)^2. \quad /8/$$

Képezzük először a $W_{j,k}$ szerinti deriváltakat. Vegyük észre, hogy az a_j nem függ $W_{j,k}$ -től, továbbá a j szerinti összegben csupán egyszer fordul elő. Így:

$$\begin{aligned} \frac{\partial E}{\partial W_{j,k}} &= -a_j \left(T_k - g \left(\sum_j W_{j,k} a_j \right) \right) g' \left(\sum_j W_{j,k} a_j \right) = \\ &= -a_j (T_k - O_k) g' \left(\sum_j W_{j,k} a_j \right). \end{aligned} \quad /9/$$

Hasonló megfontolások alapján a $V_{i,j}$ szerinti deriváltak a következők:

$$\frac{\partial E}{\partial V_{i,j}} = -I_i g' \left(\sum_i V_{i,j} I_i \right) \sum_k \left(W_{j,k} (T_k - O_k) g' \left(\sum_j W_{j,k} a_j \right) \right). \quad /10/$$

Ha összevetjük a /9/ és a /10/ eredményeit láthatjuk, hogy pontosan gradiens irányokat adnak. Ezzel az állítást beláttuk.

Sokan kritizálják a gradiensmódszer alapján működő neurális hálózat tanítását, (lásd például *Benedek* [2003] 3.2. alfejezetét). A tanulás szempontjából kritikus az indulóérték

(induló súlyok) és a lépésköz. (Megjegyezzük, hogy az itt λ -val definiált lépésközt a neurális szakirodalom gyakran jelöli η -val és nevezi *bátorsági faktornak* vagy *tanulási tényezőnek* (*learning rate*). Értékét pedig nem konstans, hanem a tanulási idő függvényében változó mennyiségnek tekintik.) A tanulási folyamat bonyolultságával többen is foglalkoztak, és sikerült bebizonyítani, hogy a mintahalmaz optimális közelítésének megkeresése még abban az esetben is NP-teljes probléma¹, ha igen erős korlátozó feltételeket alkalmazunk, és ha legalább három processzáló egységet használunk fel. (A pontos tételeket és bizonyításokat lásd: Judd [1990] és Blum–Rivest [1992].)

A súlyok optimalizálásánál is nagyobb probléma lehet az optimális hálózati topológia megválasztása. Nyilvánvaló, hogy egy sok neuronból és köztes rétegből álló hálózat súlyoptimalizálása nehezebb, de bonyolultabb összefüggések feltárására is alkalmas. Egy egyszerű hálózatot könnyű tanítani, de reprezentációs képessége is szegényebb. A helyes topológia kiválasztásakor gyakran alkalmazható a genetikus algoritmus is.

A neurális hálózati modellezés esetén a fentiekén kívül néhány szempontot érdemes szem előtt tartani. Az egyik legáltalánosabban elfogadott eljárás az, hogy a minta-adatbázist három – nem feltétlenül egyenlő – független halmazra osztjuk, mégpedig *tanuló* (*train*), *tesztelő* (*test*) és *validáló* (*validation*) halmazra. Így a súlyok tanulásánál kizárólag a tanuló adatbázis mintáit használjuk, de közben párhuzamosan kiértékeljük a tesztadatbázis eseteit is. Amikor a tanuló adatbázis minden további súlymódosítással ront a tesztelő adatbázis eredményein, megállítjuk a hálózat tanulását. Elképzelhető ugyanis, hogy egy bizonyos tanulási mennyiség után már nem a mintában húzódó általános összefüggéseket fedezi fel a hálózat, hanem a minta specialitását, zajait. Ezt nevezi a hagyományos statisztika *túllillesztésnek*, míg a neurális hálózatokkal foglalkozó szakirodalom *túltanulásként* (*overtraining*) említi.

Végül a validáló halmazra szintén meg szokták vizsgálni a hálózat teljesítését, hiszen ezt az adatbázist az optimalizálási folyamat során egyáltalán nem használta fel az algoritmus. Abban az esetben tartjuk elfogadhatónak a végső eredményeket, ha a tanuló és validáló halmazon hasonló eredményeket ér el a hálózat.

ALKALMAZÁS

Jelen esetben egy viszonylag egyszerű feladatra vállalkozunk, arra, hogy olyan esetben pótoljuk a hangot, amikor csak egyetlenegy hiányzik a kottaképről, ráadásul ez is csak elméleti feltételezés, mert a valóságban ezek a hangok ismertek. A magyar népdalok elveszett hangjainak pótlását bemutató elemzés a következő feladatot tűzte ki célul. Minden népdal esetén feltételeztük, hogy egyetlen hang hiányzik a kottaképről és az általunk a fentiekben bemutatott neurális hálózat segítségével megpróbálkoztunk ennek a hangnak a rekonstruálására. A hiányzó hangot a következőképpen változtattuk a népdalokon belül. A népdal negyedik hangjától kezdve (az első és az utolsó három hangra nem adtunk becslést) elhagytunk egy hangot és a maradék hangokat és a neurális hálózatot felhasználva becslést

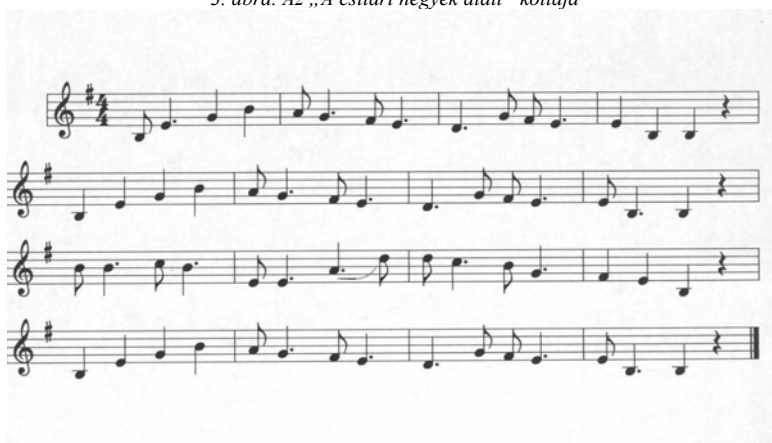
¹ A kiszámíthatóságelmélet és az algoritmusok bonyolultságelmélete szerint a problémákat különböző bonyolultsági osztályokba sorolhatjuk. Az NP-osztályba (a nem determinisztikus időbonyolultságú problémák osztályába) tartozó problémák megoldásához egy ún. ötletre van szükség. Az ötlet vagy a ráhibázás helyességét ellenőrizhetjük egy polinomiális algoritmusmal. Ezen belül van még egy kisebb csoport, amelyet NP-teljes osztálynak neveznek, ennek feladatai a „legnehezebbek”, a sejtés az, hogy nem létezik rájuk alkalmazható polinomiális megoldó algoritmus.

adtunk ennek a hangnak a magasságára. Ezután a teljes kottaképről a következő hangot hagytuk el és becsültük meg, majd a következőt stb., egészen a négygel az utolsó előtti hangig. A végleges becslést azonban több lépés is megelőzte. Először is össze kellett gyűjteni azokat a népdalokat, amelyek az illesztés és tesztelés alapadataiként szolgálnak. Ezt követően az analóg (kotta), vagy digitális (midi) formátumban összegyűjtött adatokból adatbázist kellett létrehozni, amelyen már statisztikai elemzések és becslések végezhetők. Ezután ki kellett találni, hogy pontosan mi legyen a magyarázott változó és mi legyen a magyarázó változók köre. Ezután végezhető el a modellezés, majd végül a tesztelés.

Adatgyűjtés

Összesen 144 magyar népdalt töltöttünk le a <http://www.kapisztran.hu/ismeretek/nepdal.html> honlapról. A letöltött népdalokról rendelkezésre állt egy dalszövegfájl (html), egy kotta- (pdf), és egy médiafájl (mid). A továbbiakban az „A csitári hegyek alatt” dal példáján mutatjuk be az alkalmazott eljárást. Nézzük meg a dal kottaképét:

3. ábra. Az „A csitári hegyek alatt” kottája



Adatbázis-készítés

A következő lépés egy olyan adatbázis készítése, amelyen már statisztikai elemzések és modellezés végezhető. Ezt a kották segítségével MS-Excelben manuálisan állítottuk elő.²

Az első épített adatbázisban minden dal külön állományban szerepelt. Az adatbázis rekordjai egy-egy hangot tartalmaztak, a mezők pedig a következők voltak.

- Sorszám: az adott dalban az adott hang hányadik pozícióban (helyen) szerepel.
- Ütem: az adott dalban az adott hang hányadik ütemben szerepel.
- Pozíció: az adott dalban, az adott ütemben, az adott hang hányadik pozícióban szerepel.
- Hang: zenei hang ABC-s jelöléssel, magasság-helyiértékkel (például: C5).
- Rítmus: az adott hang hossza (például: egynegyed = 0,25).

² Itt szeretnénk köszönetet mondani *Winkler Barbarának* a sok-sok segítségért, amelyet az adatbázis-építés nem túl hálás feladatában vállalt.

Szemléltetésül nézzük meg, hogyan is épül fel az „A csitári hegyek alatt” című dal első négy ütemének az adatbázisa.

Az „A csitári hegyek alatt” dal adatbázisának egy részlete

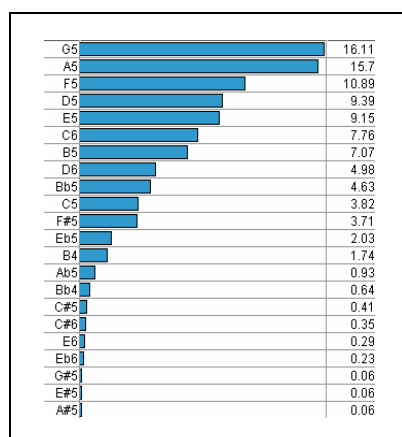
Sorszám	Ütem	Pozíció	Hang	Ritmus
1	1	1	B 4	0,125
2	1	2	E 5	0,375
3	1	3	G 5	0,250
4	1	4	B 5	0,250
5	2	1	A 5	0,125
6	2	2	G 5	0,375
7	2	3	F#5	0,125
8	2	4	E 5	0,375
9	3	1	D 5	0,375
10	3	2	G 5	0,125
11	3	3	F#5	0,125
12	3	4	E 5	0,375
13	4	1	E 5	0,250
14	4	2	B 4	0,250
15	4	3	B 4	0,250
16	4	4	–	0,250

Természetesen az egyes dalok különálló adatbázisát össze kellett fűzni, és ezért egy további oszlopra volt szükség a dal azonosítása céljából.

Transzformáció

Az általunk előre jelezni kívánt változó az a hangmagasság, amelynek az adatbázisbeli eloszlása a 4. ábrán látható. E szerint az általunk vizsgált népdalok esetén a leggyakrabban előforduló hang a G5 hang volt, mely az összes előforduló hang 16,11 százalékát tette ki.

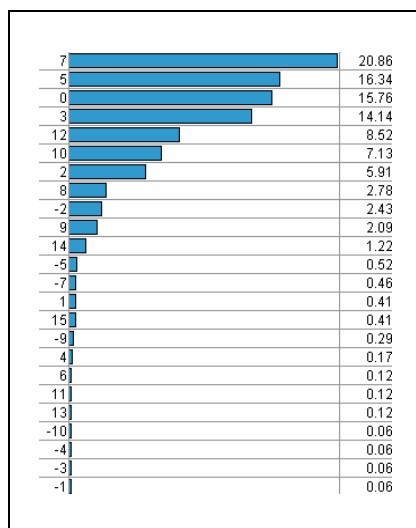
4. ábra. Hangok eloszlása az adatbázisban



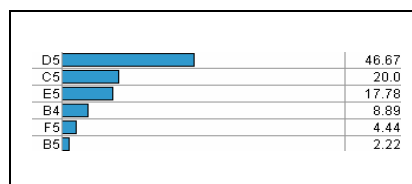
Számos előrejelző modell alkalmazása után azonban úgy döntöttünk, hogy a hangok abszolút helyének előrejelzése helyett a hangok relatív elhelyezkedését becsüljük. A relatív elhelyezkedés egy numerikus érték, amely a záró hang és az adott hang közötti félhang eltérést mutatja. Ez gyakorlatilag azt jelenti, hogy a különböző hangnemben levő dalokat azonos skálára hoztuk a szolmizációs sémához hasonlóan. A dúr és moll hangnemeket elválasztottuk.

A záró hangok természetesen nem feltétlenül egyeznek meg a hangnemmel, de ettől a jelen elemzésben eltekintünk. A záró hangok és a relatív hangok eloszlását mutatja az 5. és a 6. ábra.

5. ábra. A relatív hangok eloszlása



6. ábra. A záró hangok eloszlása



Azaz a kiválasztott dalok 46,67 százaléka D hangon zárul, valamint a leggyakrabban szereplő hangok a záró hang felett pontosan 7 félhanggal (kvárttal) helyezkednek el. Az 5. ábrából jól látható, hogy ebben az esetben a moll jellegű dalokat vizsgáltuk, hiszen a kis terc (3) elhelyezkedés jóval gyakoribb (14,14%), mint a nagy terc (4: 0,17%), nem is lesz könnyű ez utóbbit eltalálni!

Új magyarázó változók

Új magyarázó változó szerepeltetésén azt értjük, hogy egy t -edik hang becsléséhez a t -edik sorszám, ütem, pozíció és ritmus mellett mennyi $t-k$ és $t+k$ pozícióban levő hangot és ritmust használunk fel. Végző modellünkben azt feltételeztük, hogy minden elveszett hang pótlása esetén ismerjük az előtte szereplő három és az utána szereplő három hangot és annak ritmusát. Végül ezek közül is a sorszámra, az ütemre, valamint az előző és a követő hangok ritmusára vonatkozó információk irrelevánsnak bizonyultak, ezért nem kerültek bele a modellbe. A neurális hálózat minden input változója numerikus volt, azonban az output változót kategóriaváltozónak választottuk, ponto-

san azért, mert a 3-at 4-gyel összetéveszteni éppen akkora hiba, mint a 3-at 5-tel, azaz csak a tökéletes egyezést fogadjuk el jó előrejelzésnek. Így a neurális hálózat 15 beemeneti neuront (2x3 neuront az adott hangot megelőző és követő 3-3 hang magasságára és 2x3-at az adott hangot megelőző és követő 3-3 hang ritmusára vonatkozóan, plusz a keresett hang ütemére, pozíciójára és ritmusára vonatkozó információt hordozó neuronok) 30 köztes neuront (a végleges „optimális” topológiát az elméleti részben található leírás alapján választottuk, melynek során több lehetséges topológiának az előrejelző erejét összehasonlítva jutottunk el a neurális háló végleges felépítéséhez) és 24 végső neuront tartalmazott.

Modelltanítás és -ellenőrzés

A modellek tanításához átlagosan 50 dalt használtunk fel, amelyek hossza 24 és 92 hang között változott³. Adott topológia, nagy számú különböző induló érték mellett a végső, legjobban illeszkedő hálót választottuk. A változók fontosságának sorrendje (legfontosabbtól a legkevésbé fontosig) a következő volt: 1-gyel előző hang, 1-gyel utána levő hang, 3-mal utána levő hang, 3-mal előző hang, 2-vel előző hang, 2-vel utána levő hang, pozíció, ritmus. Az illeszkedés és a független tesztelés pontossága 82 százalékos volt. Az egyezőségi mátrix (lásd a 7. ábrát) alapján látható, hogy a leggyakoribb hiba az 5 helyett 7 (0,97%), a 10 helyett 12 (0,89%) és a 3 helyett 7 (0,81%) hangok voltak. Különösen érdekes volt megvizsgálni, hogy mely népdalok esetében mekkora az elkövetett hiba. Legpontosabban becsülhető népdalunk a „Sárgul már” volt, amelyet hiba nélkül jelzett előre a modell, de például a „Csak azt szánom”, az „A csitári hegyek alatt” vagy az „Elindultam szép hazámból” dalokban elkövetett hiba sem érte el a 8 százalékos értéket. A másik végletet a „Most szép” és a „Túlso soron” képviselték, ahol a hiba 39 és 36 százalékos volt.

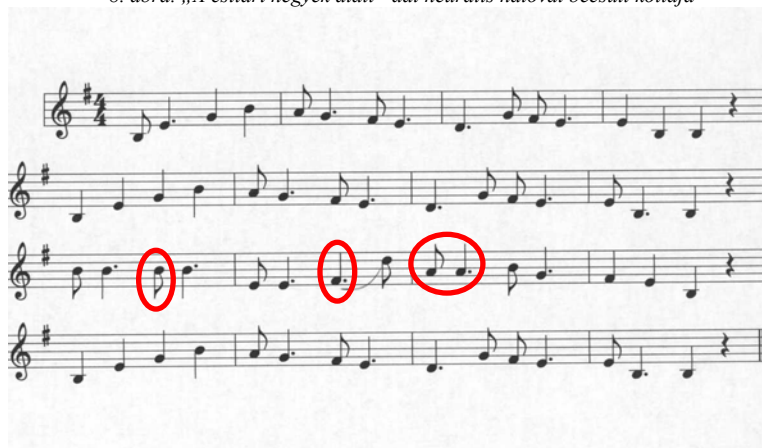
7. ábra. Egyezőségi mátrix

Relativ	-2	0	1	10	12	14	15	2	3	5	7	8	9
-1	0.000	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-2	1.292	0.404	0.000	0.000	0.000	0.000	0.000	0.081	0.081	0.162	0.081	0.000	0.000
-5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.162	0.000	0.000	0.000	0.000	0.000
0	0.000	9.532	0.081	0.000	0.000	0.000	0.000	0.162	0.404	0.000	0.404	0.000	0.000
1	0.000	0.000	0.162	0.000	0.000	0.000	0.000	0.081	0.081	0.000	0.000	0.000	0.000
10	0.000	0.242	0.000	5.654	0.889	0.000	0.000	0.000	0.081	0.242	0.727	0.081	0.323
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.081	0.000
12	0.000	0.000	0.000	0.162	7.987	0.000	0.081	0.081	0.081	0.081	0.646	0.000	0.081
13	0.000	0.000	0.000	0.081	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.323	0.646	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.081	0.323	0.000	0.162	0.000	0.000	0.000	0.000	0.000	0.000
2	0.242	0.565	0.000	0.162	0.162	0.000	0.000	3.393	0.404	0.242	0.242	0.000	0.000
3	0.081	0.242	0.000	0.000	0.323	0.000	0.000	0.000	13.651	0.495	0.808	0.000	0.152
4	0.000	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.565	0.000	0.000	0.323	0.000	0.000	0.000	0.323	16.640	0.889	0.081	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.081	0.000	0.000
7	0.000	0.162	0.000	0.323	0.727	0.000	0.000	0.000	0.162	0.565	20.355	0.000	0.323
8	0.000	0.000	0.000	0.081	0.323	0.000	0.000	0.000	0.404	0.323	1.616	0.000	0.000
9	0.000	0.000	0.000	0.404	0.162	0.000	0.000	0.000	0.000	0.081	0.323	0.081	0.889

Végezetül nézzük meg ismét az „A csitári hegyek alatt” dalban hol milyen hibát vétett a neurális hálózat.

³ Mint említettük, az „optimális” neurális háló megtalálásához több lehetséges felépítést is összehasonlítottunk. Ezek során azt találtuk, hogy a legjobban viselkedő háló az, amelyik a dūr és a moll hangnemű népdalokat külön, egymástól függetlenül kezeli. A vizsgált dalok között 48 dūr és 52 moll hangnemű volt.

8. ábra. „A csitári hegyek alatt” dal neurális hálóval becsült kottája



Mint a 8. ábrán látható kotta mutatja, összesen 4 hiba történt. Az első egy C6 helyett egy B5, a második egy A5 helyett egy F#5, a harmadik és a negyedik pedig D6 és C6 helyett mindkét esetben A5 volt. Érdekes megfigyelni, hogy mind a négy hiba a dal harmadik, az A-A-B-A szerkezetű új stílusú népdal szabadabb, B sorában történt.

A TOVÁBBFEJLESZTÉS LEHETSÉGES IRÁNYAI

Az általunk használt eljárás több irányban is továbbfejleszthető. Egyrészt, az alkalmazás során csupán a becsülni kívánt hangot megelőző és az azt követő 3-3 hangot vettük figyelembe. Egy lehetséges kiegészítés, mely várhatóan javítaná a neurális háló teljesítményét, több „lead” és „lag” figyelembevétele lenne, vagy egy olyan beállítás, mely képes a magyar népdalok sajátos szerkezetét (például az új stílusú népdalok – amilyen a szemléltetésként használt „A csitári hegyek alatt” is – szerkezetének az A-A-B-A sémáját) megtanulni és azt figyelembe venni. Egy másik érdekes kutatási irány a neurális háló és egyéb lehetséges statisztikai és ökonometriai módszerek felhasználhatóságának az összehasonlítása az elveszett hangok megtalálásának problémájára. A neurális háló által nyújtott megoldást érdekes lenne összehasonlítani szakértői válaszokkal. Ennek a nehézsége egyrészt olyan zenemű megtalálásában rejlik, mely a szakértők számára még nem ismert, viszont létezik a teljes kottaképe, másrészt pedig a szakértők valódi hozzáértésének a felmérésében.

IRODALOM

- BENEDEK, G. [2003]: *Evolúciós gazdaságok szimulációja*. PhD-értekezés. BKAE. <http://www.lib.bkae.hu/>
 BLUM, A. L. – RIVEST, R. L. [1992]: Training a 3node neural network is NP-complete. *Neural Networks*. 5. évf. 1. sz. 117–127. old.
 JUDD, J. S. [1990]: *Neural Network Design and the Complexity of Learning*. MIT Press. Cambridge. Massachusetts.

SUMMARY

In our article we deal with the theory of neural networks. After the introduction of the theoretical background of neural networks we provide an interesting application. In this application we use neural networks to fill in missing notes of Hungarian folk songs. Our results suggest that even a relatively simple neural network can perform quite well in filling in missing notes.