Magyar nyelvű WaveNet kísérletek

Zainkó Csaba, Tóth Bálint Pál, Németh Géza,

Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
{zainko, toth.b, nemeth}@tmit.bme.hu

Kivonat: A gépi beszédkeltés legújabb iránya a mély neurális hálózat alapú közvetlen hullámforma generálás. A Google DeepMind kutatói által kidolgozott, ún. nyújtott konvolúció (dilated convolution) alapú WaveNet architektúra képes a hullámforma sajátosságait megtanulni és az így épített modell alapján új hullámformákat generálni. Ezzel az architektúrával magyar adatbázisokon végeztünk kísérleteket. Megvizsgáltuk a hálózat tanulási és generálási képességeit, majd különböző nyelvi jellemzőket felhasználva módosítottuk a tanulási és beszédhullámforma generálási folyamatot. A mondatok generálásához egyrészt természetes bemondásokból kinyert paraméterlistát használtunk, illetve szabály alapú beszédszintetizátor prozódiájával is végeztünk kísérleteket. A generált hangmintákat meghallgatásos teszt segítségével értékeltük, amelyben a WaveNet által generált hangmintákat hasonlítottuk össze természetes és szintetizált beszéddel.

1 Bevezetés

A gépi beszédkeltésnek, a beszédszintézisnek a fejlődését a tudományos eredmények mellett alapvetően meghatározzák az aktuálisan elérhető számítási és tárolási kapacitások. A formáns alapú szintézis a digitális szűrőkön és azok vezérlésén alapul, kihasználva az adott korban elérhető eszközök lehetőségeit [6]. A hullámforma összefűzéses eljárások a 90-es évek elején kezdtek elterjedni, amikor már lehetőség volt a szintézishez szükséges beszédhangminták időtartománybeli reprezentációjának tárolására és futás idejű feldolgozására. Később a háttértárak és memóriakapacitások növekedése lehetőséget nyitott a korpusz alapú beszédszintézisnek, amely akár több gigabájt mennyiségű előre rögzített hangfelvételből válogatja össze a szintetizáláshoz szükséges elemeket. A korpusz alapú beszédszintetizátorokhoz [2] szükséges nagy mennyiségű felvételek lehetővé tették, hogy a döntően szabály alapú gépi megoldások mellett fejlődésnek induljanak a statisztikai elveken működő megoldások [3]. A rejtett Markov-modell (*Hidden Markov Model, HMM*) alapú beszédszintetizátorok már nem a hullámforma szeletekből építik fel a szintetizált beszédet, hanem gépi tanulás útján meghatározott statisztikai paraméterek segítségével beszédkódolót vezérelnek.

1.1 Gépi tanulás alapú beszédszintézis

Már több mint egy évtizede aktívan foglalkoztatja a beszédkutatókat a gépi tanuláson alapuló beszédszintézis [20]. Ezen rendszerekben a beszéd hullámformáját beszédkó-

dolók segítségével paraméterekre bontjuk (alapfrekvencia, spektrális- és időzítési paraméterek), és a szöveges átirat segítségével ezeket a paramétereket tanítjuk be a gépi tanuló modellel. A beszéd generálása során pedig a bemeneti szöveg alapján a modell elkészíti a "legvalószínűbb" paraméter folyamokat, melyekből a beszédkódoló gépi beszédet állít elő.

Korábban rejtett Markov-modell alapú gépi tanulást használtak [15],[20] a paraméterfolyamok modellezésére. Az elmúlt években a HMM-el szemben előtérbe kerültek a nagyobb pontosságot és így jobb beszédminőséget nyújtó mély neurális hálózatok az ugrásszerűen növekedő számítás kapacitásnak – elsősorban a feladatra optimalizált grafikus kártyáknak (*Grapical Processing Unit, GPU*) – és az új tudományos eredményeknek köszönhetően [1], [16], [19].

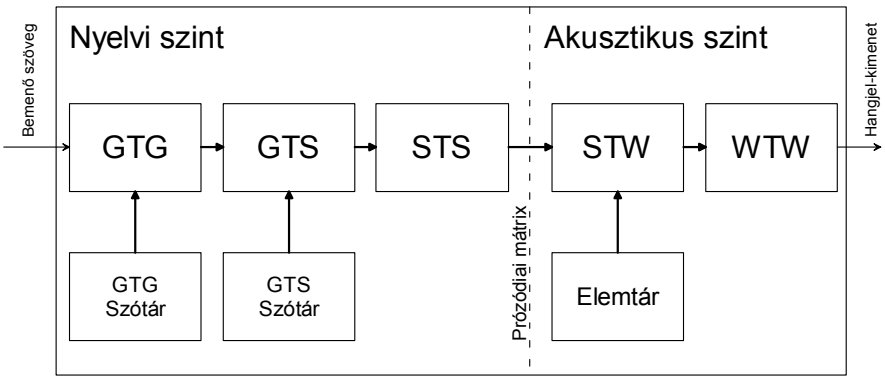
A gépi tanulás alapú beszédszintézisnek számos előnye van a korpusz alapú megoldással szemben: kötetlen témakörben közel azonos minőségű gépi beszédhangot képes nyújtani, kicsi a futásidejű adatbázisa és alkalmas viszonylag rövid (mintegy 10 perc) hangfelvétel alapján a célbeszélő hangjára emlékeztető gépi beszédhangot létrehozni. A beszédkódolók használata azonban hátrányokkal is jár: a paraméterfolyamok nem pontos modellezése esetén a generált beszéd gépiessé, vagy akár hibássá is válhat.

Számos tudományterületen (pl. beszédfelismerés, képosztályozás) a paraméterek analitikus kinyerése (lényegkiemelés) helyett ma már hatékonyan alkalmazzák az ún. mély konvolúciós neurális hálózatokat (*Convolutional Neural Network, CNN*) a paraméterek tanulására [7]. Ez annyit jelent, hogy magukból a nyers adatokból tanulja meg a rendszer, hogy milyen absztrakció írja le legjobban azokat.

Beszédszintézisben először 2016. szeptemberében alkalmaztak a Google DeepMind kutatói CNN-eket a beszéd (és zene) pusztán hullámformából történő modellezésére és generálására. Az új architektúrát WaveNet-nek [11] nevezték el, mely a PixelCNN-ben [13] kidolgozott képgenerálás átültetése hang generálására.

1.2 WaveNet kísérletek

A jelen tanulmányban a WaveNet alapú hullámforma-generáló eljárás magyar nyelvű alkalmazására vonatkozó kísérleteinket mutatjuk be. A WaveNet önmagában nem alkalmas értelmes beszéd szintetizálására, mivel csak a beszédszintetizálás egyik lényeges elemére nyújt megoldást, a hullámforma generálásra, a többire nem.



1. ábra: Példa egy általános TTS felépítésére. A komponensekben használt rövidítések: T: konverzió (to) G: graféma, S: hangkód, W: hullámforma. (az ábra [10] alapján készült)

A beszédszintetizátorok működésének első lépése a nyelvi szint, amit az akusztikai szint követ (1. ábra). A WaveNet az akusztikai szintre ad egy megoldást. Ahhoz, hogy beszédet tudjunk generálni a WaveNet segítségével, a kísérletek során két megoldást használtunk: vagy egy meglévő TTS nyelvi szintjének kimenetét használtuk fel (prozódia mátrix), vagy természetes bemondásból nyertük ezeket a paramétereket.

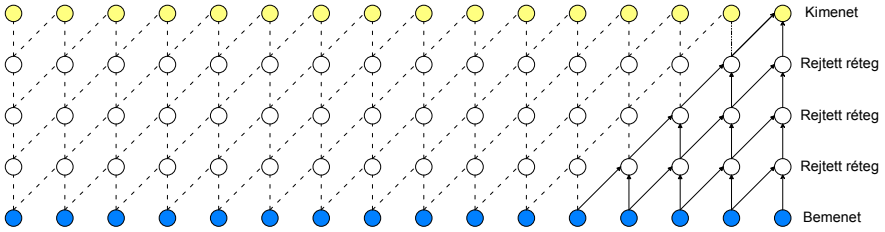
Cikkünk 2. fejezetében ismertetjük a WaveNet hullámformageneráló lényegi elemeit és azok működését. A 3. fejezetben bemutatjuk a kutatás során felhasznált környezetet és beszédadatbázisokat, majd ismertetjük a WaveNet-tel végrehajtott kísérleteinket. A 4. fejezetben bemutatjuk, hogy a kapott modelleket miként értékeltük, és hogy a meghallgatásos tesztek milyen eredményt hoztak. Az utolsó fejezetben pedig összefoglaljuk a tapasztalatainkat és bemutatjuk a továbbfejlesztési irányokat.

2 WaveNet

A WaveNet hálózat kialakítását Oord et al. [12][13] képekre és Józefowicz et al. [5] szövegre alkalmazott megoldásai inspirálták. Azt feltételezték, hogy ha a PixelRNN [13] hálózat képes 64x64 pixeles képeket modellezni, akkor az audió jelek finom struktúráját is lehetséges egy hasonló módszerrel kezelni. A WaveNet kialakításához a PixelCNN-nél [12] is használt felépítést vették alapul, ahol egy képpont generálását a korábbi képpontoktól függő feltételes valószínűségek segítségével adták meg. Az $x=\{x_1,\ldots x_T\}$ hullámformához tartozó feltételes valószínűségeket az (1) képlet adja meg.

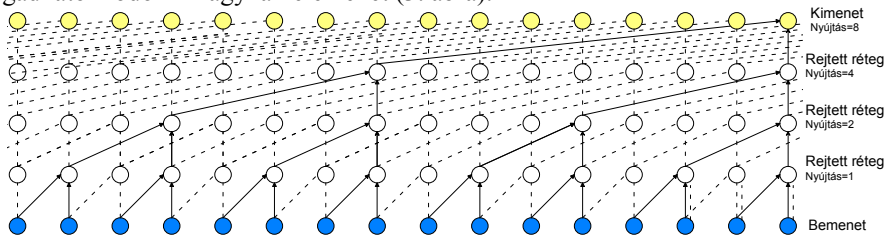
$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}) \quad \textbf{(1)}$$

Minden x_t minta függ a korábbi időpillanatok mintáitól.



2. ábra: Példa egy 5 rétegű konvolúciós hálózati megoldásra (az ábra [11] alapján készült)

A konvolúciós hálózatban ahhoz, hogy nagyszámú korábbi mintát figyelembe tudjunk venni, nagy számú rejtett réteg, vagy nagy méretű szűrők alkalmazása szükséges (2. ábra). Ezeknek viszont óriásira nőhet a számítási költségük mind a tanítás, mind a generálás során, ezért az ún. nyújtott konvolúciós (*dilated convolution*) architektúrát alkalmazták. Ennek lényege, hogy a rétegek nagyobb részénél, nem az előző időpillanat mintájához tartozó pontokat vonják be a konvolúcióba, hanem paraméterként megadható módon kihagynak elemeket (3. ábra).



3. ábra: Példa egy 5 rétegű nyújtott konvolúciós hálózati megoldásra (az ábra [11] alapján készült)

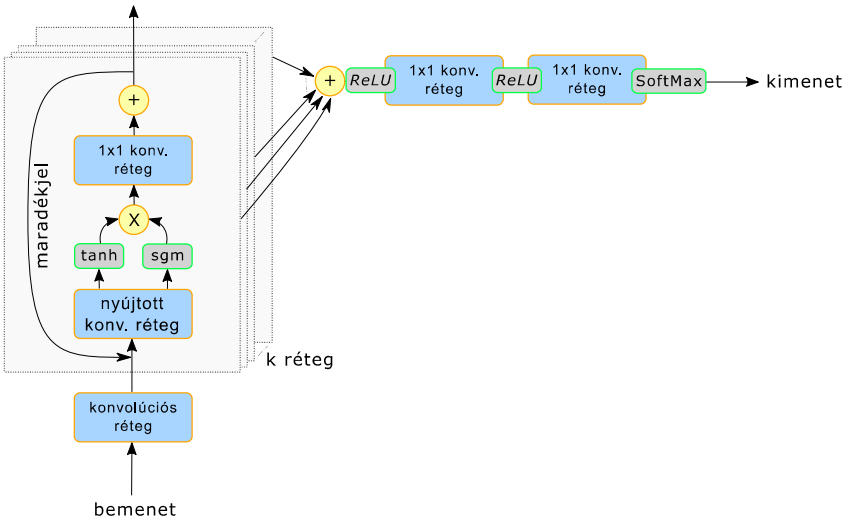
A 2. és a 3. képen is 5 rétegű hálózatot láthatunk. Míg az első hálózat esetében a kimenet az azt megelőző 5 mintától függ, addig nyújtott konvolúció esetén ugyanannyi számítás mellett, 16 mintától függ.

Az audió jelek előállítására tekinthetünk regressziós feladatként, de a digitális jelfeldolgozáshoz és átvitelhez széles körben használt logaritmikus kódolás segítségével osztályozási feladattá lehet átalakítani a problémát. A WaveNet esetében az ITU-T μ-law [4] kódolását használták. A beszédfeldolgozásban tipikusan használt 16 bites lineáris PCM jelet – amely 65536 különböző kvantálási szinttel rendelkezik – átalakítják egy 256 logaritmikus kvantálási szinttel rendelkező μ-law kódolásba. Ezt a 256 szintű reprezentációt utána „one-hot" kódolással adják a hálózat bemenetére, ahol a 256 bemenet közül mindig csak egy tartalmaz nullától eltérő értéket.

A WaveNet esetében ún. kapuzott aktivációs (*gated activation*) egységeket alkalmaznak két aktivációs függvény szorzataként:

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right) \tag{2}$$

A * jelöli a konvolúciót, az x a réteg bemenet, a $W_{f,k}$ a k-dik réteghez tartozó szűrő súlymátrixa, a $W_{g,k}$ pedig a kapu súlymátrixa. A σ a szigmoid függvényt jelöli, a kör pedig az elemenkénti szorzást.



4. ábra: Rétegek kapcsolata egymáshoz és a kimenethez a WaveNet esetében (az ábra [11] alapján készült)

A 4. ábrán látható, hogy a kimenetekhez minden rétegből kivezetjük az adatokat, és azok összegzése és 1x1-es konvolúciója után egy szoftmax függvény adja meg a kimeneti kvantált amplitúdó osztályt.

A WaveNet hálózat ebben a formában csak feltétel nélküli hullámforma generálásra alkalmas. Ahhoz, hogy generáláskor paraméterek segítségével szabályozni tudjuk a generálási folyamatot, több megoldás lehetséges. Az egyik megoldás, hogy a bemenetek mellé párosítjuk a paramétereket. Ezzel a módszerrel végezett kísérleteinket a 3.3-as fejezetben mutatjuk be. A másik módszer – amelyet a Google kutatói is publikáltak [11] – az, hogy a hálózat rétegeibe vezetjük be ezeket az információkat. Ezt a módszert és a kapcsolódó kísérleteinket a 3.4-es fejezetben mutatjuk be.

3 WaveNet kísérletek

A kísérletek elvégzéséhez GPU alapú mély tanuló keretrendszert használtunk. Az adatbázisok előfeldolgozása C++ nyelven történt, a tanítást és a generálást pedig Python alapú TensorFlow (v0.9.0) keretrendszerrel végeztük. A keretrendszer 5.1-es cudnn-t és 7.5-ös CUDA drivert használt. A tanítást GeForce GTX TITAN X-en, a generálást GeForce GTX 970-en végeztük.

3.1 Felhasznált adatbázisok

A különböző tanításokhoz és kísérletekhez egy angol nyelvű több-beszélős és egy illetve több-beszélős magyar nyelvű adatbázisokat használtunk az alábbiak szerint.

VCTK-Corpus [17]: 109 angol anyanyelvű beszélő, beszélőnként kb. 400 felolvasott mondattal. A mondatok főleg újság szövegekből lettek válogatva.

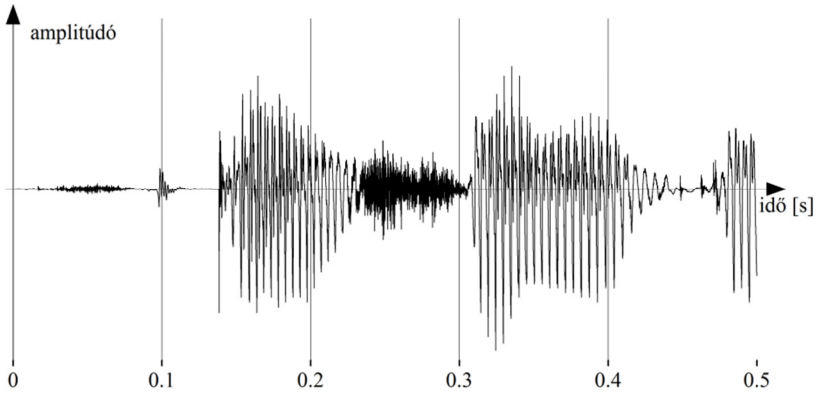
MK_MÁV [18]: Ez a korpusz egy egybeszélős felolvasott korpusz, amely a MÁV állomások hangos utastájékoztató rendszeréhez optimalizált mondatokból áll. A rögzített mondatok stúdióban készültek professzionális rádióbemondó közreműködésével. 3225 mondatot tartalmaz.

MK_RADIÓ [8]: A Nagy et al. [8] által is használt rádiós korpusz. A bemondója megegyezik a MK_MÁV bemondójával. Az adatbázis valós, rádióban elhangzott hírblokkok rögzített hanganyagaiból van összeállítva. Összesen 377 mondatot használtunk fel.

FONETIKA [9]: Egy 2000 mondatos párhuzamos adatbázis, amelyben a hangkapcsolatok fonetikailag kiegyenlítettek. Az adatbázisból 10 beszélőt használtunk, összesen 20000 mondatot.

3.2 Nyers hullámforma előállítás

A WaveNet nyers hullámforma generálásra önmagában is alkalmas, bemeneti minták, címkék vagy feltételek nélkül is. Ekkor nem értelmes szöveget állít elő, hanem hangsorozatokat. Generáláskor a modellt inicializálni kell, induló bemeneti adatként adhatunk valódi beszédmintát, vagy véletlenszerű értékekkel is feltölthetjük a bemenetet. A determinisztikus futás és a reprodukálhatóság miatt a generátort véletlen értékekkel inicializáltuk, de az álvéletlen számgenerátort rögzített értékről indítottuk.



5. ábra: A generált hullámforma időtartományban

A generált minták esetében azt tudtuk vizsgálni, hogy mennyire tartalmaznak beszédhang jellegű részeket (lásd 5. ábra), illetve a hullámforma hangszíne mennyire áll közel a tanító adatbázis beszélőjéhez. A képen látható, hogy a hullámforma a beszédhangokra jellemző képet mutat, különböző zöngés, zöngétlen jellegű szakaszok váltakoznak rajta.

3.3 Bemenet bővítése

A WaveNet bemeneti rétegénél alapesetben a μ-law kódolás eredményeként mintánként 256 különböző bemenet található. Ezeknek a bemeneteknek a száma módosítható, így első lépésként a minták mellé az aktuális beszédhang kódját is beadtuk a hálózatnak, szintén „one-hot" kódolással. A célunk a magyar nyelvű beszédgenerálás, így a további kísérleteket már magyar nyelvű adatbázisokkal végeztük.

Az első kísérletben csak a hullámforma adott részéhez tartozó aktuális hang kódjával bővítettük a bemenetet. A generáláskor azonos módon a hangminták mellé illesztettük a hangkódokat, és így futattuk le a hullámforma generálást. A generált beszéd nem követte a megadott hangkódokat, de megjelent a hangkódok által közvetve megadott időstruktúra. Mivel egy hangkódot annyi mintán keresztül adunk be a hálózat bemenetére, amennyi ideig az adott hang tart, ezért így közvetve hangidőtartam információkat is megadunk.

A bemenetet később 5-ös hangkörnyezetre bővítettük: minden hang esetében az adott hang kódja mellett, az azt megelőző és azt követő két hang kódját is megadtuk. Ekkor már a generált beszédben azonosíthatóak voltak a bemenetre adott hangkódokhoz tartozó hangok.

Az 5 hangos bemeneti kódolást tovább bővítettük a beszéd alapfrekvenciájával. A frekvencia értékek logaritmusát véve osztályokba soroltuk ezt a paramétert és a beszédhang kódokhoz hasonló „one-hot" kódolással vezettük a hálózat bemenetére. Az alapfrekvencia megadása javított a beszéd minőségén, de továbbra is maradtak kevésbé jó minőségű beszédrészletek.

A hálózat bemenetére adott hangkód és alapfrekvencia információk nem elegendőek a megfelelő minőség eléréséhez. Azért választottuk mégis a kezdeti lépésekhez ezt a formát, mert egyrészt a bemenet nagyon egyszerűen bővíthető volt, másrészt így néhány gyorsan kivitelezhető kísérletben meg tudtuk vizsgálni a hangkörnyezet és az alapfrekvencia felhasználhatóságát.

3.4 Mély rétegek szabályozása (*Local Conditon*)

A WaveNet hálózat nem csak a bemeneti rétegen keresztül vezérelhető, hanem minden rétegben módosíthatjuk a szűrő és a kapu súlyok hatását [11]. Az (1)-es képletet módosítva a feltételes eloszlásunk a következő formába írható át:

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h}) \tag{3}$$

A plusz bemeneteket ***h***-val jelöltük. A ***h*** lehet egy globális paraméter, amely hosszú időn keresztül állandó, például a beszélő azonosítója. Amennyiben egy *y=f(h)* függvénnyel a bemeneti mintákhoz illesztett paraméterlistát generálunk (például hangkódok vagy egyéb nyelvi jellemzők), akkor a konvolúciós egységekben lévő aktivációt - (2)-es képlet - a következőképpen módosíthatjuk (ahol $V_{f,k}$*y és $V_{g,k}$*y egy-egy 1x1-es konvolúció):

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y}) \tag{4}$$

A kísérleteinkben az előző fejezethez hasonlóan a 2-2 beszédhangos környezettel kibővített beszédhangot kódoltuk, illetve az alapfrekvencia logaritmusát.

Több-beszélős adatbázis esetében nem használtuk a $\boldsymbol{h}$ globális paraméterezés lehetőségét, a beszélő azonosítóját is az y bemenetre konvertáltuk át.

3.5 Mondatok generálása

A WaveNet tanítása időigényes, de a nagy mennyiségű feldolgozott adatot figyelembe véve hatékonynak tekinthető. A modell paramétereitől függően egy iterációs lépés kb. 1,4-2 mp-ig tart, amely során 100000 mintát, 6,25 mp hanganyagot használunk fel. A tanítás és a generálás is 16kHz-es mintavételi frekvenciával történt. A hanggenerálás ezzel szemben lassabb művelet, mivel 1 db minta legenerálásához egy teljes forward lépést végre kell hajtanunk, ami alig tart kevesebb ideig, mint egy tanítási lépés. Mivel a következő minta generálásához szükséges az előzőleg generált minta, ezért nem tudjuk a GPU-k párhuzamos számítását kihasználni és egyszerre több mintát előállítani. A mérések szerint egy minta generálása, kb. 0,25 mp-ig tart. Így nagyságrendileg 1 mp hanganyag előállítása kb. 2 óráig tart.

A generálás gyorsítható, amelyre Le Paine [14] adott egy módszert, a Fast-WaveNet-et. A forward lépéseknél olyan számításokat végzünk el minden egyes lépésnél, amit már korábban egyszer kiszámoltunk. La Paine rámutatott, hogy a részeredmények eltárolásával a generálás $O(2^L)$ –ről O(L)-re gyorsítható, ahol L a rejtett rétegek száma. A saját méréseink először nem támasztották alá ezt a gyorsulást. A generálás folyamatát elemezve megállapítottuk, hogy a Fast-WaveNet esetében a numerikus számítás mennyisége annyira lecsökken, hogy a futási idő legnagyobb részét a GPU-ra történő adatátvitel és a számítások után az eredmények memóriába való visszaolvasása adta. Így a GPU-t kihagyva, csak CPU-n futtatva a Fast-WaveNet-et, 1 mp hanganyag előállítása csak kb. 4 percet vett igénybe.

4 Az eredmények értékelése

4.1 Hiba mérése

A neurális hálózatok tanítása során a túltanítás elkerülése céljából a leállási feltételt gyakran a hibafüggvény értékének alakulásához kötjük. Például, ha adott tanítási cikluson keresztül nem csökken a hiba (vagy elkezd növekedni) egy tanító adatoktól elkülönített, ún. validációs halmazon, akkor leállítjuk a tanítást. A WaveNet hálózat tanítása két szempontból is speciális. Egyrészt a generált hangminták esetén jellemző, hogy nem minden esetben a legkisebb hibát produkáló hálózatok adják a legjobb szubjektív értékelést a tesztelőknél. A másik tényező az, hogy a validációs halmaz elemeivel való összehasonlításhoz a hangmintákat le kell generálni. Mivel a generálási idő nagy, egy nagyon kicsinek mondható 2-3 mondatos validációs halmaz legenerálása is 1 teljes napba kerül normál WaveNet generálással (kb. 12 mp hanganyag esetén). Ez alatt a tanítás kb. 50-60 ezer iterációt is el tud végezni, ezért egyelőre inkább a rövid, párhuzamos tesztminták generálása és azok szubjektív értékelése alapján

határoztuk meg, hogy meddig fusson egy tanítás. Fast-WaveNet esetén gyorsabb lenne a validáció, de ezt nem minden hálózaton tudjuk még alkalmazni.

4.2 A szubjektív teszt felépítése

A meghallgatásos teszt három részből állt. Az első részben a tesztelők a magyar nyelvű WaveNet által generált tartalom nélküli hangsorozatot hasonlították össze az eredeti beszélőtől rögzített mondattal. A tesztelőknek azt kellett eldönteniük, hogy menynyire adja vissza a generált hangsorozat az eredeti beszélő hangszínét.

A második részben generált mondatrészleteket kellett összehasonlítani a természetes mondattal. A generált mondatrészekben kétféle módon keltett hangsorozatok szerepelnek. Az első esetben a 3.3-as fejezetben ismertetett módon a bemeneti rétegnél bővítettük a mintákat különböző egyéb információkkal. A mondatok másik részét a 3.4-es fejezetben leírtak szerinti hálózattal generáltuk, ahol a bemeneti paraméterek a rétegek szűrőit és kapuit módosították.

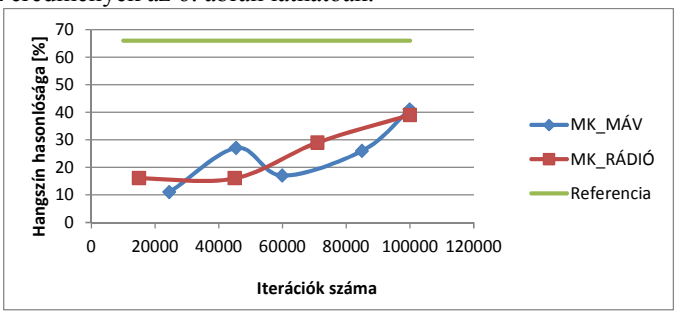
A harmadik részben egy HMM-TTS [8] és egy korpuszos TTS [18] által generált mondatot hasonlítottunk össze a WaveNet MK_MÁV-os valamint a több-beszélős modelljével generált mondattal.

Az értékelét egy internetes MUSHRA (MUlti-Stimulus test with Hidden Reference and Anchor) teszttel végeztük, ahol a résztvevők egy referencia mintához hasonlították a különböző variációkat. A teszt sajátossága, hogy a minták közé a referencia mintát is elhelyeztük, így könnyítve meg a minták skálán való elhelyezését. A tesztelő személy a mintákat egy csúszka segítségével 0-100-as skálán értékelhette, így árnyalatnyinak vélt különbség is megadható volt.

A tesztet 5 nő és 12 férfi végezte el. A legfiatalabb 23 éves volt, a legidősebb 74 éves, az átlag életkor 42 év volt. Valamennyien ép hallásúak és magyar anyanyelvűek.

4.3 A szubjektív teszt eredményei

Az első részben az értelmetlen hangsorozatokat értékelték. A teszt nehézsége az volt, hogy a tesztelőknek a hangszínt kellett értékelni, viszont a szubjektív véleményeket a hangminőség biztosan befolyásolja, csak szakértők tudják ezt megbízhatóan szétválasztani. Az eredmények az 6. ábrán láthatóak.



6. ábra. Az értelmetlen hangsorozatok értékelése két tanító adatbázis esetén.

Az iteráció számával növekedett a hasonlóság a természetes mintához, valószínűleg túl hamar lett leállítva a tanítás, 100000 iteráció után a grafikon alapján még elképzelhető lett volna javulás. Rejtett referenciának a referencia beszéd kevert hangsorozatát raktuk be, amely hangszín szempontjából megegyezik azzal, amihez hasonlították a tesztelők a mintákat, mégis csak 66 %-osra értékelték. Sajnos a vártnak megfelelően nem tudták a hangszínt a minőségtől és értelemtől függetlenül értékelni.

A második részben a különböző módszerrel generált mondatokat hasonlítottuk össze egy természetes mondattal. Az eredmények releváns részei az 1. táblázatban találhatók.

1. táblázat: Generált mondatok minősége összehasonlítva a referenciával

Bemenet	Paraméterek	Iterációk száma	Hasonlóság	Szórás
bem. réteg	1 hang	145k	13 %	3,9
bem. réteg	5 hang	50k	28 %	3,8
bem. réteg	5 hang + logF0	70k	41 %	5,6
mély réteg	5 hang + logF0 + pp	50k	46 %	4,5
mély réteg	5 hang + logF0	55k	53 %	4,8
mély réteg	5 hang + logF0 + pp	200k	54 %	4,6
Referencia			86 %	3,8

Az első oszlop adja meg, hogy a bemeneteket a bemeneti rétegre (3.3 fejezet) vagy a mély rétegekbe (3.4 fejezet) vezetjük be. A bemeneti paraméterek esetében 1 hang, vagy 2-2 hangos környezettel együtt adtuk meg (5 hang). Bizonyos esetekben az alapfrekvencia logaritmusát (logF0) és a prozódiai egységen belüli pozíciót (pp) is megadtuk paraméterként. Az egyhangos bemeneti paraméter egyáltalán nem működött, ezt az eredmények is alátámasztották. A bemeneti rétegre adott plusz információk javítottak a minőségen, de a mély rétegek módosításával jobb eredményeket értünk el.

A teszt harmadik részében korpuszos és HMM technológiával is összehasonlítottuk a generált mondatokat, az eredmények a 2. táblázatban láthatóak.

2. táblázat: Generált mondatok minősége összehasonlítva a referenciával

WaveNet	Korpuszos	HMM	Referencia
40 %	56 %	70 %	81 %

A WaveNet-tel készített mondatot ítélték a leggyengébb minőségűnek, majd a korpusz technológiával készült következett. A korpuszos esetében a szöveg nem a korpusz témakörének megfelelő volt, ezért a mondat minősége rosszabb volt a szokásosnál. A HMM 70 %-ot ért el, a referencia minta pedig 81 %-ot.

5 Összefoglalás

Az első magyar nyelvű WaveNet kísérletek megmutatták, hogy a Google DeepMind kutatói által kidolgozott architektúra alkalmazható magyar nyelvre is. Érthető beszéd már minimális paraméterezéssel is előállítható, 2-2 hangos környezet már elegendő

arra, hogy jól azonosíthatóan megtanulja a hálózat a magyar beszédhangokat. További címkék segítségével javítható a generált beszéd minősége. A meghallgatásos tesztek támpontot adnak a további munkához, de a megbízható értékeléshez több mintát kell generálni, amely most még technológia okokból nehezen kivitelezhető.

Az eljárás legnagyobb hátránya a generálás futásideje, amely nem teszi lehetővé, hogy valós idejű alkalmazásokba ezt a technológiát beintegráljuk.

5.1 Jövőbeli tervek

A beszédminőség javítása érdekében további nyelvi jellemzőkkel érdemes bővíteni a tanításkor és generáláskor használt címkehalmazt. Mivel nem biztos, hogy a több jellemző jobb minőséget eredményez, ezért ezen címkék optimális halmazának kiválasztása az egyik cél.

Mivel a generálási idő komoly korlátot jelent a felhasználás szempontjából, ezért a sebesség növelése a másik prioritás a jövőben. Ahhoz, hogy széles körben használható legyen, legalább valós idejű működés szükséges, amely azt jelenti, hogy a most használt Fast-WaveNet generálási módszernél is legalább 240-szer gyorsabb eljárás szükséges.

A tesztmondatok generálása során azt tapasztaltuk, hogy a betanított modellek minőségén túl a generálás inicializálása is jelentősen befolyásolja a generált beszéd minőségét. A további kutatásaink során ezzel a részterülettel is behatóbban szeretnénk foglalkozni.

5.2 Köszönetnyilvánítás

Köszönjük Bartalis István Mátyásnak a meghallgatásos teszt létrehozásában nyújtott segítségét. Tóth Bálint Pál köszöni az NVIDIA vállalat támogatását, a kutatási célokra rendelkezésére bocsájtott NVidia Titan X GPU kártyát.

A generált magyar nyelvű WaveNet minták a http://smartlab.tmit.bme.hu/wavenet oldalon meghallgathatók.

Bibliográfia

1. Fan, Y., Qian, Y., Xie, F. L., & Soong, F. K. TTS synthesis with bidirectional LSTM based recurrent neural networks. In Interspeech (2014). pp. 1964-1968.
2. Fék M, Pesti P, Németh G, Zainkó C, Olaszy G. Corpus-based unit selection TTS for Hungarian. In: International Conference on Text, Speech and Dialogue (2006 Sep 11) pp. 367-373. Springer
3. Heiga, Zen., et al. "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005." IEICE transactions on information and systems 90.1 (2007): 325-333.
4. ITU-T. Recommendation G. 711. Pulse Code Modulation (PCM) of voice frequencies, (1988)

5. Jozefowicz R, Vinyals O, Schuster M, Shazeer N, Wu Y. Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410. (2016 Feb 7).
6. Klatt, Dennis H. "Review of text-to-speech conversion for English." The Journal of the Acoustical Society of America 82.3 (1987): 737-793.
7. Le Cun, Y., & Bengio, Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), (1995)
8. Nagy, Péter, Csaba Zainkó, and Géza Németh. "Synthesis of speaking styles with corpus-and HMM-based approaches." Cognitive Infocommunications (CogInfoCom), (2015) 6th IEEE International Conference on. IEEE.
9. Olaszy, G., "Precíziós, párhuzamos magyar beszédadatbázis fejlesztése és szolgáltatásai Beszédkutatás (2013), pp. 261–270, 2013.
10. Olaszy, G., Németh G., Olaszi, P., Kiss, G., Gordos , G.: "PROFIVOX - A Hungarian Professional TTS System for Telecommunications Applications", International Journal of Speech Technology, Volume 3, Numbers ¾ , (December 2000), pp. 201-216.
11. Oord AV, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499. (2016 Sep 12.)
12. van den Oord A, Kalchbrenner N, Kavukcuoglu K. Pixel Recurrent Neural Networks. arXiv preprint arXiv:1601.06759. (2016 Jan).
13. Oord AV, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K. Conditional image generation with pixelcnn decoders. arXiv preprint arXiv:1606.05328. (2016 Jun 16.)
14. Tom Le Paine: Fast Wavenet: An efficient Wavenet generation implementation https://github.com/tomlepaine/fast-wavenet (2016.nov.10)
15. Tóth Bálint Pál, Németh Géza, Rejtett Markov-modell alapú szövegfelolvasó adaptációja félig spontán magyar beszéddel, In: VI. Magyar Számítógépes Nyelvészeti Konferencia], Szeged, Magyarország, (2009), pp. 246-256
16. Wu, Z., Valentini-Botinhao, C., Watts, O., & King, S.. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2015, April) pp. 4460-4464. IEEE.
17. Yamagishi, Junichi. English multi-speaker corpus for CSTR voice cloning toolkit, (2012). URL http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html
18. Zainkó, Csaba, et al. "A Polyglot Domain Optimised Text-To-Speech System for Railway Station Announcements." Sixteenth Annual Conference of the International Speech Communication Association. (2015).
19. Zen, H., Senior, A., & Schuster, M. Statistical parametric speech synthesis using deep neural networks. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (2013, May). pp. 7962-7966. IEEE.
20. Zen, H., Tokuda, K., & Black, A. W. Statistical parametric speech synthesis. Speech Communication, 51(11), (2009). 1039-1064