Beszélőinvariáns akusztikus modellek létrehozása mély neuronhálók ellenséges multi-taszk tanításával

Tóth László[1], Gosztolya Gábor[2]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
{tothl, ggabor}@inf.u-szeged.hu

Kivonat Bár a mély neuronhálós technológia bevezetésével a beszédfelismerő rendszerek pontossága rengeteget javult, a környezeti tényezőkkel szembeni robusztusságuk növelése továbbra is az egyik legfontosabb kutatási terület. Cikkünkben egy nemrégiben javasolt eljárást, a neuronhálók ellenséges multi-taszk tanítását próbáltuk bevetni a beszélő személyére való érzékenység csökkentésére. Ehhez olyan tanító adatbázisra van szükség, ami a szöveges átirat mellett a beszélő személyére vonatkozó annotációt is tartalmaz. Bár a kiindulási alapként szolgáló cikkhez képest jóval több beszélővel, valamint teljesen kapcsolt neuronháló helyett konvolúciós hálóval dolgoztunk, ennek ellenére minden konfigurációban konzisztens 2-3% körüli relatív hibacsökkenést kaptunk. A módszert beszélőklaszterezéssel kiterjesztve arra az esetre is adunk egy megoldási javaslatot, amikor nem áll rendelkezésre beszélőannotáció. A kezdeti eredmények bíztatóak, ebben a felügyelet nélküli esetben is hibacsökkenést mértünk, habár a felügyelt esethez képest szerényebb mértékűt.
Kulcsszavak: beszédfelismerés, mély neuronhálók, multi-taszk tanulás, ellenséges tanulás

1. Bevezetés

A mély neuronhálókon alapuló beszédfelismerési technológia ma már széles körben elfogadott és elterjedt [1]. Azonban továbbra is kihívás, hogy ezeket a rendszereket robusztussá tegyük, azaz hatékonyságuk ne romoljon a legkülönfélébb felhasználási körülmények között sem. Sajnos ilyen zavaró tényező rengeteg létezik, a beszélő személy hangjának egyedi sajátosságaitól a mikrofonok eltérő átviteli karakterisztikáján át a beszűrődő háttérzajig. A neuronhálók általánosító képességének növelésére az egyik lehetőség a regularizációs módszerek használata a betanítás során. Általánosan megfogalmazva, a regularizáció célja, hogy a háló ne tanuljon rá nagyon specifikusan az aktuális tanítóadatokra, mert ez az ún. túltanulás az új adatokra való általánosítási képesség csökkenését okozhatja. A túltanulás csökkentésének egy lehetséges módja, ha a hálónak több feladatot kell megtanulnia egyszerre, ez az ún. multi-taszk tanítás [2]. Megfigyelték ugyanis, hogy ha ezek a feladatok kicsit eltérnek, de hasonló jellegűek, azaz hasonló

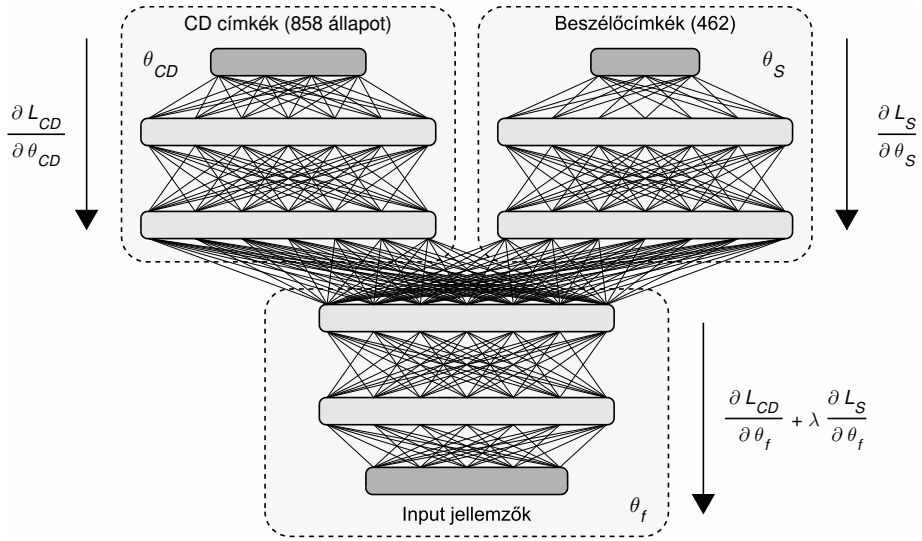
belső reprezentáció kialakítását igénylik, akkor a két feladat egyidejű tanulásának köszönhetően a háló robusztusabbá válik, és gyakran mindkét feladaton jobb pontosságot ér el, mint külön-külön tanításnál. A multi-taszk tanítást a beszédfelismerésben is többen sikeresen alkalmazták már [3,4].

Míg a sztenderd multi-taszk tanításnál arra törekszünk, hogy a háló a másodlagos feladaton is kis hibát érjen el, létezik a módszernek egy ellenséges (adversarial) multi-taszk tanítás nevű változata is, ahol a másodlagos feladat hibáját nem minimalizálni, hanem *maximalizálni* próbáljuk [5]. Ettől azt várjuk, hogy a háló olyan belső reprezentációt alakítson ki, amely a másodlagos feladatra nézve invariáns. A beszédtechnológiában az ellenséges multi-taszk tanítást eddig leginkább az akusztikus modellek felvételi környezetre, például a háttérzajra való robusztussá tevésére alkalmazták [6,7], de az akcentussal [8], illetve legújabban a beszélő személyével szemben való függetlenítésre is találunk példát [9]. Mi ez utóbbival fogunk itt próbálkozni, azaz az ellenséges multi-taszk tanítástól a modell beszélőinvariánssá, de legalábbis a beszélő személyére kevésbé érzékennyé válását reméljük. Cikkünkben bemutatjuk az ellenséges tanítás módszerét, és a kapott eredmények alapján kielemezzük a megoldás előnyeit-hátrányait. A módszer egyik hátránya az lesz, hogy beszélőket azonosító annotációt igényel, ezért a kiértékelést nem magyar adatbázison, hanem az angol TIMIT adatbázison végezzük, amelynek tanító része egyenletes eloszlásban 462 beszélőtől tartalmaz mintát (Meng és társai cikkükben jóval kevesebb beszélővel dolgoztak [9]). A kiindulási cikkhez képest további lényeges eltérés lesz, hogy teljesen kapcsolt háló háló helyett konvolúciós hálót fogunk használni. Mivel a konvolúció célja eleve a beszélő személyére való érzékenység csökkentése, kérdéses, hogy konvolúciós háló esetén is segít-e az ellenséges tanítás.

2. Multi-taszk és ellenséges multi-taszk tanítás

A multi-taszk neuronháló sematikus felépítését szemlélteti az 1. ábra. A hálózatnak mindkét (vagy esetleg több) feladathoz van egy-egy dedikált kimenőrétege, illetve opcionálisan lehetnek feladatspecifikus rejtett rétegei is. Az ábrán a két ág hibafüggvényét L_{CD} és L_S, az ágak paramétereit (súlyait) pedig θ_{CD} és θ_S jelölik (CD a környezetfüggő (context-dependent) állapotokat, S a beszélőket (speaker) kódolja). A hálózat inputja, valamint alsó rétegei közösek, ami technikailag annyi nehézséget okoz, hogy a hiba visszaterjesztése során a közös rétegekhez érve a két ágból érkező hibát össze kell adni (azaz az ábrán $\lambda = 1$). Ez arra kényszeríti a hálózatot, hogy ezekben a közös rétegekben olyan reprezentációt alakítson ki, amely mindkét feladat megoldását segíti.

Sajnos tudomásunk szerint jelenleg csak empirikus úton lehet kideríteni, hogy egy konkrét másodlagos feladat felvétele segíteni fogja-e vagy sem az eredeti feladat megoldását, az észszerűség azonban azt diktálja, hogy hasonló jellegű, de a fő feladattól némiképp eltérő másodlagos feladatot érdemes választani. Az is csak kísérleti úton deríthető ki, hogy mely rétegnél érdemes a hálózatot elágaztatni. A logika és a tapasztalat is azt mondja azonban, hogy minél eltérőbb a



1. ábra: Az (ellenséges) multi-taszk neuronháló struktúrája.

két feladat, annál kevesebb közös, és annál több feladatspecifikus rétegre lesz szükség [10].

Tudomásunk szerint a multi-taszk tanítást beszédtechnológiában elsőként Green és társai alkalmazták, ahol a felismerés mellett a másodlagos feladat a beszéd háttérzajtól való megtisztítása volt [11]. A mély neuronhálós világban a multi-taszk tanítás Seltzer és Droppo munkájában bukkan fel újra, akik az aktuális beszédhang felismerése mellé a kontextus, azaz a szomszédos hangok felismerését vették fel második feladatnak [3]. Nagyon hasonló ehhez Bell és Renals megoldása, akik a környezetfüggő állapotcímkék mellé a környezetfüggetlen címkék megtanulását tekintették másodlagos feladatnak [4]. Lényegében ezt a megoldást ismételtük meg korábban magyar nyelvre, és a korábban említett munkákkal egybevágóan néhány százalékos relatív hibacsökkenést értünk el [12].

Bár logikusan hangzik, hogy a közös reprezentáció egy másodlagos feladatra való érzékenyítése segíthet, ennek épp az ellenkezője, azaz a reprezentáció valamilyen szempontból invariánssá tételének hasznossága is éppen annyira indokolható. Ez utóbbi a célja az ún. ellenséges (adversarial) multi-taszk tanításnak [5], ami a beszédtechnológiában tudomásunk szerint 2016-ban bukkant fel először [6]. Ellenséges tanítás esetén a multi-taszk háló struktúrája ugyanaz marad, viszont a tanítás során a másodlagos feladathoz tartozó hibát nem minimalizálni, hanem *maximalizálni* próbáljuk. Technikailag ezt úgy oldjuk meg, hogy a másodlagos feladathoz tartozó feladatspecifikus ágban továbbra is minimalizálást végzünk; azonban a hibavisszaterjesztési folyamat során a közös jellemzőkinyerő rétegekhez érve az λ paraméternek *negatív* értéket adunk. Ennek hatására a hálózat olyan közös reprezentáció kialakítására fog törekedni, amely alapján a feladat-

specifikus ágak a elsődleges feladatot minél pontosabban, a másodlagos feladatot viszont minél kevésbé tudják megoldani. Az így kialakított közös reprezentáció optimális esetben tehát nem fog a második feladat megoldását segítő információt tartalmazni, azaz invariáns lesz arra. A módszert a beszédtechnológiában eddig főleg arra próbálták használni, hogy a neuronhálót az aktuális környezetre érzéketlenné, "domain-invariánssá" tegyék, ahol a környezeten alapvetően a különféle háttérzajok értendők, de van példa az akcentussal szembeni robusztusság növelésére is [8]. Vizsgálatainkban Meng és társai "beszélőinvariáns" modellt ígérő módszertanát próbáltuk reprodukálni, ahol a másodlagos feladatot a beszélő felismerése képezte [9].

Shinohara cikkében azt javasolja, hogy az ellenséges tanítást csak fokozatosan vezessük be, azaz az λ paraméter értékét fokozatosan növeljük a tanítási iterációk során [6]. Tanácsát követve az k-adik iterácóban a paraméter értékét az alábbi képlet szerint állítottuk be:

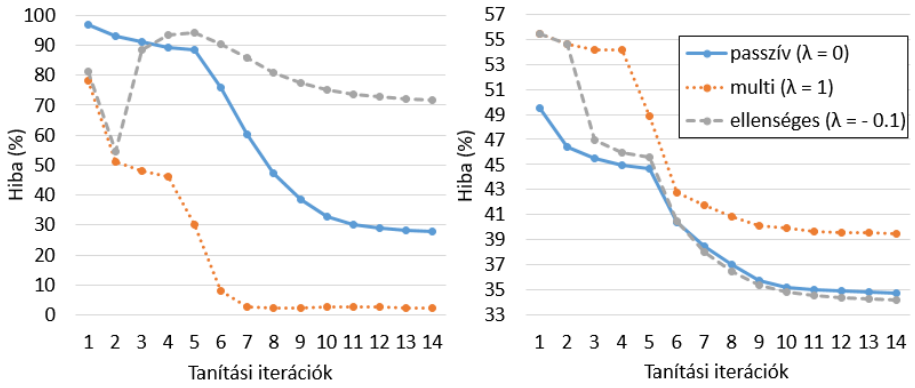
$$\lambda_k = min\Big(\frac{k}{c}, 1\Big) \cdot \lambda,$$

azaz λ a végleges értékét c iteráció után veszi fel. Shinohara cikkében $c = 10$ szerepel, de mi a $c = 7$ értékkel is kísérleteztünk, mivel tapasztalatunk szerint a tanulási ráta felezése tipikusan 6-7 iteráció után indul be.

3. Kísérleti beállítások

Vizsgálatainkat az angol nyelvű TIMIT beszédadatbázison végeztük. Míg ez az adatbázis beszédfelismerési szempontból már nagyon kicsinek számít, beszélőfelismerési kísérletekre ideális, mivel sok beszélőtől tartalmaz mintákat egyenletes eloszlásban. A tanító mintahalmazban 462 beszélőtől szerepel 8-8 mondat, míg a "core" teszthalmaz 24, a tanító adatoktól független beszélőből áll. Fejlesztési (development) mintaként a tanító adatokból véletlenszerűen kivett 10% mintát használtunk, így erre a halmazra a beszélőfüggetlenség nem teljesült.

Kísérleteinkben egy olyan neuronhálót alkalmaztunk, amely a legalsó rétegében konvolúciós neuronokat tartalmaz, melyek a frekvenciatengely mentén végeznek konvolúciót [13]. Megjegyezzük, hogy egy bonyolultabb hálóstruktúrával a konvolúciót az időtengelyre is kiterjeszthetjük, amivel kicsit jobb eredményeket kaphatnánk [14], de itt most a célunk nem a maximális teljesítmény elérése volt, hanem az ellenséges tanítási módszer működésének elemzése. Konvolúciós neuronhálónk a legalsó, konvolúciós rétegen kívül még két további teljesen kapcsolt réteget tartalmazott a legalsó, közös blokkban, míg a feladatspecifikus blokkok mindkét ágon 1-1 rejtett réteget használtak. A rejtett rétegek mindegyike 2000 darab "egyenirányított" (ReLU) neuronból állt. A kimentő réteg a beszédfelismerési feladathoz tartozó ágon a hibrid rejtett Markov-modelles beszédfelismerő 858 állapotának megfelelően 858 kimenő neuront tartalmazott, míg a másodlagos, beszélőfelismerési ágon a beszélők számának megfelelő 462 neuron került a kimenő rétegbe. A hálót az adatvektor-szintű keresztentrópia hibafüggvény minimalizálásával tanítottuk mindkét ágon.



2. ábra: A másodlagos feladat hibájának alakulása a tanítás során a tanítóhalmazon (bal oldal), illetve az elsődleges feladat hibája a development halmazon (jobb oldal).

4. Eredmények és diszkusszió

A módszer működésének megértéséhez első lépésben elvégeztünk egy kísérletet, amely a multi-taszk és az ellenséges multi-taszk tanítás hatását hasonlítja össze. Az 2. ábra bal oldali része szemlélteti, hogy tipikusan hogyan alakul a másodlagos (beszélőfelismerési) feladat hibája a tanítóhalmazon a tanítási iterációk függvényében. Elsőként λ értékét nullára állítottuk. Ez azt jelenti, hogy a fő feladat mellett a másodlagos ág is tud ugyan tanulni, de a közös rétegekben kialakuló rejtett reprezentációba nem szólhat bele (ezért címkéztük ezt az esetet 'passzív' tanulásként). A főágon kapott eredmény fog viszonyítási alapként szolgálni, hiszen ilyenkor ez az ág ugyanazt az eredményt adja, mint egy egyfeladatos hálókonfiguráció. Az ábrán azt láthatjuk, hogy a másodlagos ág ilyenkor is tud tanulni, 30% körüli pontosságot ér el a beszélők felismerésében. Második lépésben hagyományos multi-taszk tanítást futtatunk, azaz λ értéke 1 volt. Az ábra azt mutatja, hogy ilyenkor a beszédfelismerő ág 3% körüli pontossággal képes azonosítani a train halmaz beszélőit. Végezetül, λ értékét $-0,1$-re, azaz ellenséges tanulásra állítottuk, és a látványosabb hatás kedvéért két iteráció multi-taszk tanulás után váltottunk át ellenséges multi-taszk tanulásra. A másodlagos ág hibája ekkor gyorsan felszalad 90% fölé, és végig 70% fölött marad.

A 2. ábra jobb oldala mutatja ezzel párhuzamosan az elsődleges, beszédfelismerési ágon kapott hibaértékeket (ezúttal a development halmazon, mert itt már az általánosítási képeség is fontos, hiszen ezt a kimenetet fogjuk felhasználni a beszédfelismerőben). Azt láthatjuk, hogy az alaprendszerhez képest a multi-taszk esetben lényegesen megnövekszik a hiba, míg ellenséges tanítás mellett ha szerény mértékben is, de csökken.

Az eredmények számszerű kiértékelése során az λ (és részben a c) paraméterek optimális értékét igyekeztünk megtalálni. A kezdeti próbálkozások alapján c ér-

Paraméterek		Keretszintű hiba		Felism. hiba
λ	c	1. ág (dev)	2. ág (train)	(teszthalmaz)
0 (passzív)	–	34,7%	36%	18,6%
-0,03	7	34,3%	57%	18,3%
-0,06	7	**34,1%**	73%	18,1%
-0,10	7	34,3%	82%	18,1%
-0,10	10	34,2%	79%	17,9%
-0,15	10	34,4%	85%	**17,8%**
-0,20	10	34,6%	90%	18,1%

1. táblázat. Beszédhang-felismerési hibaarányok különböző paraméterértékekkel.

tékét 7-re állítottuk, λ-t pedig 0,03 és 0,1 között változtattuk. Az 1. táblázat mutatja a kapott hibaértékeket – az összehasonlítás alapjául szolgáló 'passzív' konfigurációt az első sorban tüntettük fel. Az első eredményoszlop a neuronháló keretszintű hibáját mutatja a development halmazon, a másodikban érdekességképp a másodlagos feladat keretszintű hibáját tüntettük fel (ezt csak a tanítóhalmazon mértük), végül a felismerő lefuttatása után a teszthalmazon kapott beszédhang-felismerési hibaarányokat az utolsó oszlop mutatja. Rögzített $c = 7$ érték mellett szépen látszik, hogy λ növelésével a másodlagos feladat hibája is nő, miközben a fő feladat hibája konzisztensen alatta marad az alaprendszerének. Az is az elvártnak megfelelő viselkedés, hogy c értékét 10-re növelve λ értékét is növelni lehetett. A development halmazon a keretszintű hiba $c = 7, \lambda = -0,06$ esetén érte el a minimumát, míg a teszthalmazon a felismerési hiba $c = 10, \lambda = -0,15$ mellett. Ennek az lehet az oka, hogy a development halmazunk beszélői a tanítópéldák között is szerepeltek. A megbízhatóbb kiértékeléshez meg kell majd ismételnünk a kísérletet a development halmaz beszélőfüggetlen újratervezésével. Azt a tanulságot azonban mindenképpen le tudtuk vonni, hogy a módszer valóban segít, hiszen konzisztensen minden esetben hibacsökkenést tapasztaltunk. A csökkenés mértéke átlagosan 3% körüli volt, a teszthalmazon kapott legjobb érték 3,8% relatív hibacsökkenésnek felel meg. Összevetésképp, Meng és társai 5%-os javulásról számoltak be [9]. Az eltérés oka az lehet, hogy mi konvolúciós hálót használtunk, ami eleve csökkenti a háló beszélő személyére való érzékenységét. Korábbi méréseink szerint a TIMIT adatbázison a felismerési eredmények beszélők szerinti szórását a konvolúció bevezetése 5,7%-kal csökkentette [15].

A javulás szerény mértéke miatt az is felvetődött bennünk, hogy az eredmények esetleg pusztán a tanulásba bevezetett 'zajnak' köszönhetően javultak – ismert ugyanis, hogy némi zaj hozzáadása a tanításhoz javítani tudja a neuronhálók általánosítási képességét. Ennek ellentmond azonban, hogy λ előjelét megfordítva egyértelmű romlást tapasztaltunk ($\lambda = 0,1$ esetén is). A biztonság kedvéért kiszámoltuk a felismerési hiba beszélőkre nézve vett szórását is. Azt találtuk, hogy a 17,8%-ot elérő modell szórása az alaprendszeréhez képest kb. 10%-kal alacsonyabb. Ez igazolja, hogy az ellenséges tanításnak valóban olyan

Paraméterek		Keretszintű hiba		Felism. hiba
λ	c	1. ág (dev)	2. ág (train)	(teszthalmaz)
0 (passzív)	–	34,7%	36%	18,6%
-0,10	10	34,1%	70%	18,4%
-0,06	7	34,3%	65%	18,3%

2. táblázat. Beszédhang-felismerési hibaarányok beszélőklaszterezéssel.

hatása volt, mint amit vártunk tőle. Ennek ellenére a kapott modellt beszélőinvariánsnak nevezni erős túlzás – például Meng és társai további komoly javulást kaptak a modellen beszélőadaptációt alkalmazva [9].

4.1. Felügyelet nélküli eset

A Meng és társai által javasolt módszer komoly hátulütője, hogy beszélők szerint annotált adatbázist igényel. Bár a TIMIT esetén rendelkezésre áll ilyen annotáció, a legtöbb, beszédfelismerők betanításához összeállított korpusz nem tartalmaz ilyen információt. Az ilyen esetek kezelésére valamilyen felügyelet nélküli tanítási módszert kell bevetnünk. Mi azzal próbálkoztunk, hogy a tanító adatbázis fájljait klaszterezés segítségével csoportokra bontottuk. A klaszterezésre egy hierarchikus beszélőklaszterezési módszert alkalmaztunk [16,17,18]. A klaszterek számát 50-re állítottuk, λ-t pedig a korábban legjobb eredményeket adó értékre állítottuk be. A 2. táblázatban látható kezdeti eredmények biztatóak, mivel a keretszintű hiba a validációs halmazon a korábbiakhoz hasonló módon csökkent; a teszthalmazon kapott felismerési eredmények azonban szerényebb javulást mutatnak, mint a valódi beszélőcímkék használata esetén. Ezért további, alaposabb kiértékelést tervezünk a klaszterméret változtatásával, valamint más klaszterező algoritmusok kipróbálásával.

5. Összegzés

Cikkünkben egy nemrégiben javasolt gépi tanulási technikát, a mély neuronhálók ellenséges multi-taszk tanítását vizsgáltuk, a módszerrel a gépi beszédfelismerők akusztikus modelljének beszélőkre való érzékenységét akartuk csökkenteni. Kísérleteinkben a módszer konzisztensen 2-3% körüli relatív hibacsökkenést hozott. Ez kisebb, mint a kiindulási alapként felhasznált publikációban szereplő 5%, aminek oka az lehet, hogy az eredeti cikkel szemben mi konvolúciós hálót használtunk, ami eleve kevésbé érzékeny a beszélők közti eltérésekre. A módszert kiterjesztve egy megoldási lehetőséget javasoltunk arra az esetre is, amikor a tanítókorpuszhoz nem áll rendelkezésre beszélőkre vonatkozó annotáció. A módszer ebben az esetben is működni látszik, bár a kapott hibacsökkenés szerényebb. A jövőben ennek a felügyelet nélküli megoldásnak az alaposabb kivizsgálását tervezzük, a klaszterméretek és a klaszterezési algoritmusok széles körű vizsgálatával.

Köszönetnyilvánítás

Tóth Lászlót az MTA Bolyai János Kutatási Ösztöndíja, valamint az Emberi Erőforrások Minisztériuma ÚNKP-18-4 kódszámú Új Nemzeti Kiválóság Programja támogatta. A kutatást az Emberi Erőforrások Minisztériuma Emberi Erőforrások Minisztériuma 20391-3/2018/FEKUSTRAT kódjelű pályázata támogatta. A kutatáshoz használt grafikus kártyát az NVIDIA Corporation ajándékozta csoportunknak.

Hivatkozások

1. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29**(6) (2012) 82–97
2. Caruana, R.: Multitask learning. Journal of Machine Learning Research **17**(1) (1997) 41–75
3. Seltzer, M., Droppo, J.: Multi-task learning in deep neural networks for improved phoneme recognition. In: Proc. ICASSP. (2013) 6965–6969
4. Bell, P., Renals, S.: Regularization of deep neural networks with context-independent multi-task training. In: Proc. ICASSP. (2015) 4290–4294
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, H., Larochelle, H., Laviolette, F., Marchand, M., Lempitzky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(59) (2016) 1–35
6. Shinohara, Y.: Adversarial multi-task learning of deep neural networks for robust speech recognition. In: Proc. Interspeech. (2016) 2369–2372
7. Denisov, P., Vu, N., Font, F.: Unsupervised domain adaptation by adversarial learning for robust speech recognition. In: Proc. ITG Conference of Speech Communication. (2018)
8. Sun, S., Yeh, C., Hwang, M., Ostendorf, M., Xie, L.: Domain-adversarial training for accented speech recognition. In: Proc. ICASSP. (2018) 4854–4858
9. Meng, Z., Li, J., Chen, Z., Zhao, Y., Mazalov, V., Gong, Y., Juang, B.: Speaker-invariant training via adversarial learning. In: Proc. ICASSP. (2018) 5969–5973
10. Tóth, L., Grósz, T., Markó, A., Csapó, T.: Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces. In: Proc. Interspeech. (2018) 3172–3176
11. Lou, Y., Lu, Y., Seghal, S., Gupta, S., Du, J., Tham, C., Green, P., Vincent, W.: Multitask learning in connectionist speech recognition. In: Proc. Australian International Conference on Speech Science and Technology. (2004)
12. Tóth, L., Gosztolya, G.: Adaptation of DNN acoustic models using KL-divergence regularization and multi-task training. In: Proc. SPECOM. (2016) 108–115
13. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.: Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition. In: Proc. ICASSP. (2012) 4277 – 4280
14. Tóth, L.: Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition. In: Proceedings of ICASSP. (2014) 190–194
15. Tóth, L.: Phone recognition with hierarchical convolutional deep maxout networks. EURASIP Journal on Audio, Speech and Music Processing **25** (2015)

16. Han, K.J., Kim, S., Narayanan, S.S.: Strategies to improve the robustness of Agglomerative Hierarchical Clustering under data source variation for speaker diarization. IEEE Transactions on Audio, Speech and Language Processing **16**(8) (2008) 1590–1601
17. Wang, W., Lu, P., Yan, Y.: An improved hierarchical speaker clustering. Acta Acustica **33**(1) (2008) 9–14
18. Kaya, H., Karpov, A., Salah, A.: Fisher Vectors with cascaded normalization for paralinguistic analysis. In: Proceedings of Interspeech. (2015) 909–913