

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1043

**NEURALNE MREŽE U IZDVAJANJU  
GOVORNOG SIGNALA IZ ZVUČNOG ZAPISA**

Stjepan Henc

Zagreb, lipnja 2015.

*Posvećujem ovaj rad svim gluhim osobama koje sanjaju o danu  
kada će moći normalno komunicirati sa svim članovima društva.*

*Zahvaljujem se dragom Bogu za snagu  
i mudrost potrebnu za izradu ovog rada.*

## **Sažetak**

U ovom radu dan je pregled trenutnog stanja područja izdvajanja govora i pregled najuspješnijih strategija za izdvajanje govora. Poseban naglasak stavljen je na duboke neuronske mreže. Odabrana je BLSTM-RNN arhitektura neuronske mreže i CURRENNT kao programski paket za rad s tom mrežom. Uspješnost izdvajanja govora je ispitana na skupu podataka prvog zadatka drugog CHiME natjecanja (engl. CHiME 2nd challenge Task 1) i postignuto je apsolutno poboljšanje točnosti od 25% za prepoznavanje govora sa smetnjama koristeći sustav za prepoznavanje čistog govora s malim opsegom rječnika.

Ključne riječi: izdvajanje govora, duboke neuronske mreže, RNN, BLSTM, CUDA, CHiME, CURRENNT

## **Summary**

An overview of recent developments in the field of speech extraction is given, including a survey of most successful strategies. A special emphasis is put on approaches based on deep neural networks. The BLSTM-RNN neural network architecture is chosen along with the CURRENNT software package for working with the network. Success of speech extraction is evaluated on Task 1 of the CHiME 2nd challenge, and an absolute improvement in word accuracy of 25% is achieved on a noisy speech recognition task using a small-dictionary ASR system specialised for clean speech.

Keywords: speech extraction, deep neural networks, RNN, BLSTM, CUDA, CHiME, CURRENNT

## Sadržaj

1.	Uvod.....	8
2.	Pregled literature .....	9
2.1.	Stanje istraživanja.....	9
2.2.	CHiME .....	10
2.3.	Odabir strategije .....	13
2.4.	Odabir programskog paketa .....	15
3.	Metodologija.....	16
3.1.	Rekurzivne neuronske mreže .....	16
3.2.	Arhitektura sustava .....	20
3.3.	Metoda treniranja neuronske mreže .....	22
4.	Primjena .....	24
4.1.	Priprema podataka .....	24
4.2.	Treniranje mreže.....	25
4.3.	Rezultati.....	27
5.	Diskusija.....	29
6.	Zaključak .....	33
7.	Literatura .....	34

## Popis oznaka i kratica

BLSTM	dvosmjerna dugotrajno-kratkotrajna memorija (engl. Bidirectional Long-Short Term Memory)
BRNN	dvosmjerna rekurzivna neuronska mreža (engl. Bidirectional Recursive Neural Network)
BPTT	nadogradnja backpropagation algoritma za treniranje RNN, “prolaz u nazad kroz vrijeme” (engl. Backpropagation Through Time)
CHiME	natjecanje u računalnom prepoznavanju govora u okruženjima gdje je prisutno više izvora signala (engl. Computational Hearing in Multisource Environments)
CHiME2	druga iteracija CHiME natjecanja (engl. CHiME 2nd challenge)
CTC	klasifikacija vremenskih nizova pomoću neuronskih mreža (engl. Connectionist Temporal Classification)
DNN	duboka neuronska mreža (engl. Deep Neural Network)
engl	engleski
GPU	grafički procesor opće namjene (engl. Graphic Processor Unit)
ICA	analiza neovisnih komponenata (engl. ICA - Independent Component Analysis)
LSTM	dugotrajno-kratkotrajna memorija (engl. Long-Short Term Memory)
NMF	nenegativna faktorizacija matrica (engl. Non-negative Matrix Factorization)
PCA	analiza principalnih komponenata (engl. PCA - Principal Component Analysis)
RNN	rekurzivna neuronska mreža (engl. Recursive Neural Network)
SNR	omjer korisnog signala i signala smetnje (engl. Signal to Noise Ratio)

## Popis tablica

Tablica 1. Usporedba programskih paketa za rad s RNN .....	15
Tablica 2. Rezultati prepoznavanja s istreniranom mrežom .....	27
Tablica 3. Najbolji postignuti rezultati za odabrani pristup.....	27
Tablica 4. Trajanje epohe treniranja.....	28
Tablica 5. Trajanje obrade .....	28

## Popis slika

Slika 1. Primjer rekurzivne neuronske mreže .....	16
Slika 2. Dvosmjerna rekurzivna neuronska mreža .....	17
Slika 3. LSTM blok s jednom ćelijom.....	19
Slika 4. Arhitektura korištene BLSTM-RNN.....	20
Slika 5. Pregled sustava za izdvajanje i prepoznavanje govora .....	21
Slika 6. Krivulja učenja .....	26
Slika 7. Ilustracija izdvajanja govora iz parametriziranog signala.....	30

Sve korištene slike su djelo autora.

# 1. Uvod

U praksi je jako često da zvučni zapisi govora koje treba pretočiti u tekstualne zapise (engl. speech-to-text) sadrže i razne smetnje koje su prisutne jer govor nije bio sniman u idealnim uvjetima. Smetnje mogu biti razni šumovi, buka, muzika ili istovremeni govor i žamor. Sve takve smetnje uzrokuju jako veliki pad točnosti računalnog prepoznavanja govora [1].

Zadatak ovog diplomskog rada je pronaći odgovarajuću metodu i programski paket koji bi omogućio izdvajanje najčišćeg mogućeg govora iz takvih zvučnih zapisa s ciljem povećanja točnosti prepoznavanja govora.

Među mnogim opcijama kao metoda je odabrana BLSTM-RNN neuronska mreža, a od programskih paketa CURRENNT, koji podržava ubrzavanje treniranja pomoću GPU-a.

Efikasnost ovog pristupa je ispitana korištenjem CHiME2 skupa podataka i pripadnih alata.

Ovaj rad je organiziran na sljedeći način:

U 2. poglavlju iznesen je pregled literature i stanja istraživanja na ovom području, s posebnim naglaskom na ono što je korišteno u ovom radu. Opisan je CHiME2 skup podataka kao dobra platforma za usporedbu rješenja za izdvajanje govornog signala.

U 3. poglavlju opisan je rad RNN-BLSTM algoritma, odabrana je arhitektura mreže i opisane su metode treniranja mreže koje podržava programski paket CURRENNT.

U 4. poglavlju opisana je priprema podataka za treniranje, postupak treniranja mreže i rezultati dobiveni na CHiME2 skupu podataka.

U 5. poglavlju opisano je što bi ti rezultati mogli značiti za računalno prepoznavanje govora u praksi i mogući daljnji pravci istraživanja.



## 2. Pregled literature

### 2.1. Stanje istraživanja

U svim primjenama koje se bave govorom, prisutnost smetnji je neizbježna. Bilo da se radi o snimanju zvuka, telekomunikacijama ili ljudsko-računalnim sučeljima (engl. human-machine interfaces), mikrofoni koji snimaju govor uglavnom će snimiti i smetnje. Zbog toga se snimljeni govorni signal treba pročistiti digitalnom obradom signala prije daljnje upotrebe.

Taj proces pročišćavanja govora obično se naziva suzbijanje buke (engl. noise reduction), poboljšavanje govora (engl. speech enhancement) ili izdvajanje govora (engl. speech separation). To je područje koje se intenzivno proučava već nekoliko desetljeća, no problem još uvijek nije u potpunosti riješen [2].

Posebno važna primjena izdvajanja govora je računalno prepoznavanje govora. U usporedbi s računalnim sustavima za prepoznavanje govora, ljudsko uho radi nevjerojatno dobro. Kod ljudskih bića razumijevanje govora, ovisno o primjeni, počinje padati kada je SNR od -6 do 0 dB, a tek se od -25 do -20 dB sasvim gubi razumljivost. Računalno prepoznavanje govora počinje gubiti na točnosti već oko +20 dB, a na 0 dB (jednaka snaga signala govora i smetnje) se već približava nasumičnom pogađanju [1].

Činjenica da su ljudi tako sposobni u obavljanju tog zadatka potaknula je mnoge istraživače da u proučavanju ljudskog slušnog sustava pokušaju pronaći inspiraciju za nova tehnološka rješenja [3][4][5].

Postoje tri glavna pristupa izdvajanju govornog signala:

1. Tehnologija nizova mikrofona (engl. Microphone-Array Technology)
2. Slijepo razdvajanje signala (engl. Blind Signal Separation)
3. Razdvajanje temeljeno na modelu govora

Prvi od tih pristupa, temeljen na akustičnoj tehnologiji, je tradicionalan i već prisutan u praksi. No, iako radi dovoljno dobro u određenim uvjetima, manji sustavi su, u usporedbi s čovjekom, nefleksibilni i loši.

Drugi pristup se temelji na teoriji informacija, i trenutno je glavni trend u polju istraživanja razdvajanja signala.

Treći pristup pokušava donekle emulirati ljudski slušni sustav koristeći apriorno znanje, tj. model ciljnih signala, kao najvažniji faktor. U usporedbi sa slijepim razdvajanjem signala ovaj pristup je još u povojima, no njegovu opravdanost potvrđuju mnoga otkrića na području psihologije [6].

## **2.2. CHiME**

### **2.2.1. Motivacija**

U zadnjih nekoliko godina veoma je značajno CHiME natjecanje u razdvajanju i prepoznavanju govora (engl. CHiME Speech Separation and Recognition Challenge) kao platforma gdje različiti istraživači iz akademskog svijeta i industrije mogu usporediti svoja rješenja na skupu podataka koji je dobar pokazatelj koliko dobro bi ta rješenja mogla raditi na stvarnim podacima [7][8].

Iako je u tijeku već treća iteracija CHiME natjecanja, u ovom radu će se koristiti podaci za drugo CHiME natjecanje ili CHiME2, jer u ovom trenutku već postoji mnogo objavljenih rezultata za taj skup podataka, pa je moguće usporediti dobivene rezultate s već objavljenima [9].

Cilj CHiME natjecanja je dobiti što veću točnost prepoznavanja govora izobličenog realističnim izvorima smetnji. Problem izdvajanja govora za automatsko prepoznavanje je različit od običnog pročišćivanja govora, jer veliki broj tehnika pročišćivanja govora samo poboljšava doživljaj kvalitete govora, ali ne povećava i njegovu razumljivost (bilo za ljudske ili računalne slušače) [10].

Računalno prepoznavanje govora je težak problem i u uvjetima savršeno čistog govornog signala. Faktori koji taj problem mogu dodatno otežati su:

1. promjenjivost položaja i udaljenost govornika u odnosu na mikrofonski
2. veličina rječnika
3. prirodnost govora

Budući da je fokus CHiME natjecanja na izdvajanju govora, autori su odlučili napraviti skup podataka s realnim signalima smetnje, snimljenim u pravoj dnevnoj

sobi. Tako je govoru superponirana izrazito realna smetnja, no kako bi zadatak ostao rješiv, govor kojeg treba prepoznati je nerealno jednostavan [7].

### 2.2.2. Grid korpus

Izvor čistog govora je Grid korpus govora, koji se sastoji od zvučnih zapisa jednostavnih komandi. Korištene su snimke 34 različita izvorna govornika engleskog jezika [11].

Zvučni zapisi su rečenice od šest riječi u obliku:

<naredba:4><boja:4><prijedlog.:4><slovo:25><znamenka:10><prilog:4>

pri čemu brojevi u uglatim zagradama označuju koliko opcija postoji za svaku kategoriju. Zadatak je prepoznati slovo i znamenku pa se točnost prepoznavanja mjeri samo na te dvije riječi.

Dakle, govor kojeg treba prepoznati se sastoji od malog rječnika i jednostavne gramatike, nije prirodan i govornik je uglavnom na istom položaju, što prepoznavanje čistog govora čini laganim. Točnost računalnog prepoznavanja za čisti govor je 97.25% [7], što je usporedivo s ljudskom točnošću koja je u tom slučaju oko 98.3% (točnost za slova je 99.05%, a za brojke 99.3%) [11].

### 2.2.3. Smetnje

Kako bi se dobio željeni raspon SNR, izgovorene rečenice su pozicionirane u odnosu na snimljenu pozadinsku buku tako da se dobiju željene vrijednosti (-6, -3, 0, 3, 6 i 9 dB). Snimka pozadinske buke je nasumično pretraživana i odabrani su oni vremenski intervali koji imaju željeni SNR u odnosu na zadanu rečenicu. Na 9 dB, najpovoljnijem odnosu željenog i neželjenog signala, smetnje su uglavnom kvazi-stacionarni šumovi (npr. šum ventilatora), dok su oni na -6 dB uglavnom iznenadni nestacionarni zvučni događaji (npr. dječje vrištanje).

Kako bi zadatak bio još realističniji, napravljena je konvolucija čiste izgovorene rečenice s vremenski promjenjivim binauralnim impulsnim odzivima sobe (engl. Binaural Room Impulse Response) koji simuliraju ograničeno pomicanje govornika i odjek prostorije [7].

### 2.2.4. Podaci

Sve snimke u CHiME2 skupu podataka su u 16-bitnom WAV formatu uzorkovanom na 16 kHz. Skup podataka za treniranje (engl. training set) sadrži

17000 rečenica, 500 za svakog od 34 govornika. Skup podataka za validaciju (engl. validation/development set) i skup podataka za testiranje (engl. test set) sadrže 600 rečenica na 6 različitih SNR-a [12].

### 2.2.5. Ocjenjivanje točnosti

Osim ispitnih podataka, u sklopu CHiME2 dostupni su i alati za mjerenje točnosti prepoznavanja govora. Ti alati su temeljeni na sustavima za prepoznavanje govora koji su dio HTK programskog paketa [13][14]. Iako CHiME dozvoljava korištenje vlastitog rješenja za prepoznavanje govora, na raspolaganje je stavljen osnovni sustav (engl. baseline recognizer) s nekoliko unaprijed istreniranih modela. To je učinjeno kako bi se moglo odrediti koji dio poboljšanja točnosti se može pripisati izdvajanju govora, a koji sustavu za prepoznavanje govora. Time se olakšava i usporedba različitih sustava [14].

Osnovni sustav za prepoznavanje je prilagođen sintaksi rečenica u Grid korpusu. Sustav je baziran na skrivenim Markovljevim modelima. Svaka od 51 riječi prisutne u Grid korpusu modelirana je pomoću skrivenog Markovljevog modela s 2 stanja po fonemu. Vjerojatnost izostavljanja nekog stanja je predstavljena pomoću mješovitog Gaussovog modela sa 7 komponenti i dijagonalnom kovarijancom.

Dana su tri unaprijed uvježbana modela za prepoznavanje:

1. "Čisti" model (engl. clean - treniran na čistom govoru)
2. "Odjek" model (engl. reverberated - treniran na govoru izobličenom jekom)
3. "Buka" model (engl. noisy - treniran na govoru izobličenom bukom)

Osim ta tri modela, dostupni su alati za lako uvježbavanje tj. prilagođavanje (engl. retraining) vlastitog modela. To se radi tako da se skup podataka za treniranje propusti kroz algoritam za izdvajanje govora i model za prepoznavanje se onda trenira na tom govoru. Cilj je poništiti utjecaj izobličenja govora koje je nastalo njegovim izdvajanjem [7].

Govor je parametriziran kao niz standardnih MFCC značajki. Svaki vektor značajke sadrži 39 parametara. Prvi parametri u vektoru su 13 mel-kepstralnih koeficijenata koji su normalizirani po srednjoj vrijednosti (ali ne i standardnoj devijaciji). Ta metoda se zove kepstralna metoda normalizacije srednje vrijednosti

(engl. CMN - Cepstral Mean Normalisation) i smanjuje utjecaj razlika u obliku vokalnog trakta na točnost prepoznavanja govora različitih govornika. Umjesto nultog MFCC parametra koristi se logaritamska energija okvira. Na tih 13 parametara dodaje se 13 diferencijalnih koeficijenata prvog reda i 13 drugog reda (engl. delta and acceleration coefficients). Standardna HTK šifra za te značajke je MFCC\_E\_D\_A\_Z i detaljno je opisana u literaturi [13][15].

MFCC značajke se standardno računaju na vremenskim okvirima od 25 ms, a korak je 10 ms. Budući da su zvučni podaci dani u stereo formatu, signal je pretvoren u mono signal uzimanjem srednje vrijednosti oba kanala [7].

### 2.3. Odabir strategije

Zanimljiva povijesna činjenica je da su neuronske mreže u području slijepog razdvajanja signala prisutne od njegovog samog početka 80-ih godina prošlog stoljeća.

Prvi algoritam koji je korišten je analiza principalnih komponentata ili PCA, gdje se parametrizirani oblik signala (najčešće spektar) pokušavala razdvojiti na komponente koje odgovaraju pojedinim izvorima signala korištenjem njihovih različitih statističkih svojstava.

Analiza neovisnih komponentata ili kraće ICA još je jedna metoda koja se može svrstati u neuronske mreže, a nastala je nadogradnjom originalnog PCA algoritma [16] [6].

U literaturi se mogu naći stvarno brojni i nerijetko vrlo složeni pristupi ovoj problematici, no valja izdvojiti dva koja su se pokazala posebno uspješnima i popularnima u posljednjih nekoliko godina.

To su nenegativna faktorizacija matrica ili NMF [16] i duboke neuronske mreže ili DNN. Oba pristupa su relativno jednostavna, no NMF ima nekoliko nedostataka u usporedbi s DNN.

NMF je isključivo linearan model, dok DNN u pravilu može modelirati i nelinearno preslikavanje iz izvora signala u mješavinu. Također, kod primjene istreniranog NMF modela mora se provoditi iterativni postupak koji uključuje množenje nekoliko velikih matrica, što je jako računski zahtjevno.

S druge strane, duboke neuronske mreže se u pravilu duže treniraju, ali se zato primjena istreniranog modela sastoji samo od jednokratnog množenja nekoliko matrica, što ih čini pogodnima za primjenu u stvarnom vremenu jer imaju linearnu složenost. Svi ti faktori čine duboke neuronske mreže moćnijim i bržim modelom [17].

No, zanimljivo je da je na CHiME2 pobijedio sustav koji, osim pregršt drugih algoritama, koristi i DNN i NMF [18], te su u literaturi poznate razne kombinacije ova dva pristupa [19][20][21].

Uglavnom svi visokorangirani sustavi koriste kombinaciju nekoliko složenih pristupa i, za razliku od ovog rada, nije im cilj doći do sustava koji bi bio dovoljno brz za primjenu u praksi, već pod cijenu brzine i jednostavnosti dobiti čim veće performanse na skupu podataka za testiranje [9].

Duboke neuronske mreže su dio jedne šire paradigme na području umjetne inteligencije pod nazivom duboko učenje. Glavna postavka te paradigme je da su korištenje veće količine podataka [22] i većih modela [23] glavni motori povećanja performansi u strojnom učenju. Cilj je iskoristiti sve veću raspoloživu računalnu moć i sve veću količinu podataka kako bi se stari algoritmi iskoristili za rješavanje dosad nezamislivih problema. Računalni resursi koji se koriste mogu biti tisuće servera u nekoj od velikih internetskih kompanija [24], ili pak grafički procesori [25] koji danas i običnim studentima čine dostupnom računalnu moć u rangu nekadašnjih superračunala. No, najmoćnijom se pokazala kombinacija više servera s nekoliko grafičkih procesora, što omogućuje treniranje neuronskih mreža s nekoliko milijardi parametara u roku nekoliko dana [26].

Povećani intenzitet istraživanja na ovom području doveo je i do novih algoritama i ideja, između ostalog i obećavajućih postignuća na području računalnog prepoznavanja govora [27][28].

Dosad je u ovom potpoglavlju pojam DNN korišten kao da se radi o jednom pristupu, no samo u području izdvajanja govora može se odnositi na mnogo različitih tipova mreža s različitim svojstvima [29][30][31][32][33][34].

Tip duboke neuronske mreže koji se pokazao najprikladniji za izdvajanje govora je rekurzivna neuronska mreža ili RNN s dvosmjernim dugotrajno-kratkotrajnim memorijskim ćelijama ili BLSTM [35][36][37][38].

Taj pristup će biti daljnji fokus ovog rada.

## 2.4. Odabir programskog paketa

Budući da su duboke neuronske mreže u zadnje vrijeme vrlo popularno područje istraživanja, pojavili su se mnogi programski paketi koji olakšavaju njihovu upotrebu.

Zato što je odabran BLSTM tip rekurzivne neuronske mreže, u obzir dolaze samo paketi koji podržavaju takve slojeve.

Odabrani paket također mora podržavati i ubrzavanje izvođenja na grafičkim procesorima i biti općenito dovoljno učinkovit, jer bi u suprotnom treniranje mreže moglo premašiti trajanje ljetnog semestra.

Tablica 1 daje usporedbu dostupnih paketa otvorenog koda i neke njihove karakteristike [39][40][41][42][43][44].

**Tablica 1. Usporedba programskih paketa za rad s RNN**

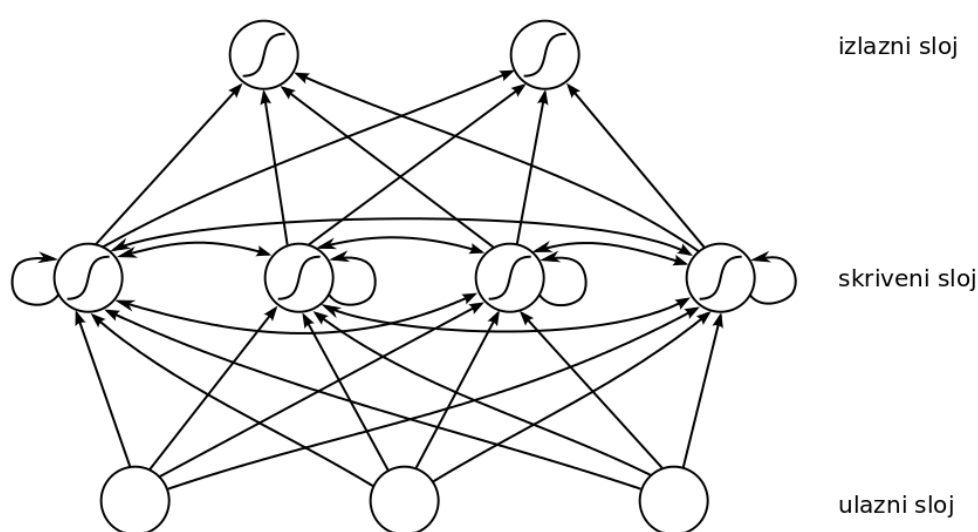
	pybrain	torch7	theano	rnnlib	CURRENNT
GPU	ne	da	da	ne	da
BLSTM	da	ne	ne	da	da
jezik	python	lua/c	python	c++	c++

Jedino programski paket CURRENNT zadovoljava sve zadane kriterije: podržava BLSTM-RNN neuronske mreže i ubrzavanje izvršavanja na grafičkim procesorima. Taj programski paket koristi biblioteku CUDA za rad s GPU-om [45] i napisan je u programskom jeziku C++, što mu omogućava da obavi treniranje mreže zadovoljavajućom brzinom.

### 3. Metodologija

#### 3.1. Rekurzivne neuronske mreže

Naziv rekurzivna neuronska mreža u užem smislu odnosi se na nadogradnju višeslojnog perceptrona [46]. U najčešćoj varijanti RNN-a, sloju se uz uobičajenu pobudu daju i izlazi iz tog sloja u prethodnom trenutku (pod izrazom trenutak se podrazumijeva pozicija bilo na vremenskoj ili na prostornoj osi).



**Slika 1. Primjer rekurzivne neuronske mreže**

Slika 1 daje primjer jedne takve mreže. Ovaj tip mreže najčešće se koristi kada je u problemu klasifikacije nekog niza podataka potrebno iskoristiti kontekst, npr. kod prepoznavanja rukopisa.

Ova nadogradnja višeslojnog perceptrona omogućava mreži da u svojem internom stanju pohrani informaciju o prethodnim ulazima i tako pamti što je bilo na ulazu u prethodnim koracima.

Prolaz unaprijed kod rekurzivne neuronske mreže izgleda isto kao kod višeslojnog perceptrona, no kod prolaza unazad koristi se BPTT algoritam. Ideja tog algoritma je da se mreža "razmota", tako da se mreži na ulaz odjednom postavi cijeli ulazni niz. Izračuna se izlaz mreže za cijeli niz, i izračunaju se greške

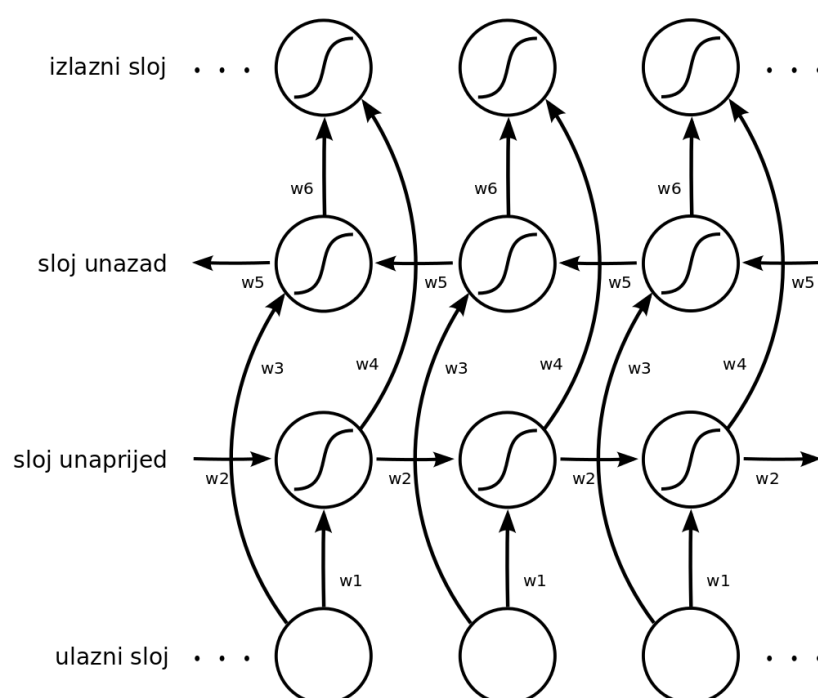


za svaki korak. Budući da se zbog razmatranja mreže težine ponavljaju, za svaku težinu se zbroje sve pripadne greške i s tom vrijednošću se osvježi njezina težina [46].

### 3.1.1. Dvosmjerna rekurzivna neuronska mreža

Budući da je u mnogim primjenama osim konteksta koji prethodi danom koraku korisno uzeti u obzir i ono što slijedi nakon njega, uvedene su i dvosmjerne rekurzivne neuronske mreže. To je nadogradnja rekurzivne neuronske mreže gdje jedna polovina rekurzivnog skrivenog sloja analizira ulazni niz u pozitivnom smjeru, a druga u negativnom.

Primjer mreže dan je na slici 2, gdje je prikazana dvosmjerna rekurzivna mreža s jednim skrivenim slojem.



**Slika 2. Dvosmjerna rekurzivna neuronska mreža**

Kako bi se izbjegli ciklusi u neuronskoj mreži dio koji računa unaprijed i dio koji računa unazad u istom sloju ne smiju biti međusobno povezani, već njihov izlaz služi kao ulaz višim slojevima.

Rad mreže je u osnovi isti kao kod obične rekurzivne neuronske mreže, no potrebno je malo prilagoditi algoritam za izračunavanje izlaza mreže i prolaz unatrag. Te izmjene su prikazane pseudokodom 1 i pseudokodom 2 [46].

```

za t = 1 do T :
    prolaz unaprijed za skriveni sloj koji računa unaprijed,
    za svaki korak se spremaju izlazi
za t = T do 1 :
    prolaz unaprijed za skriveni sloj koji računa unazad,
    za svaki korak se spremaju izlazi
za sve t, bilo kojim redoslijedom :
    prolaz unaprijed za izlazni sloj,
    koristeći spremljene izlaze iz oba skrivena sloja

```

### Pseudokod 1. BRNN prolaz unaprijed

```

za sve t, bilo kojim redoslijedom :
    prolaz unazad za izlazni sloj, spremajući  $\delta$  članove
    za svaki korak
za t = T do 1 :
    BPTT prolaz unazad za skriveni sloj koji računa unazad,
    koristeći  $\delta$  članove iz izlaznog sloja
za t = 1 do T :
    BPTT prolaz unazad za skriveni sloj koji računa unaprijed,
    koristeći  $\delta$  članove iz izlaznog sloja

```

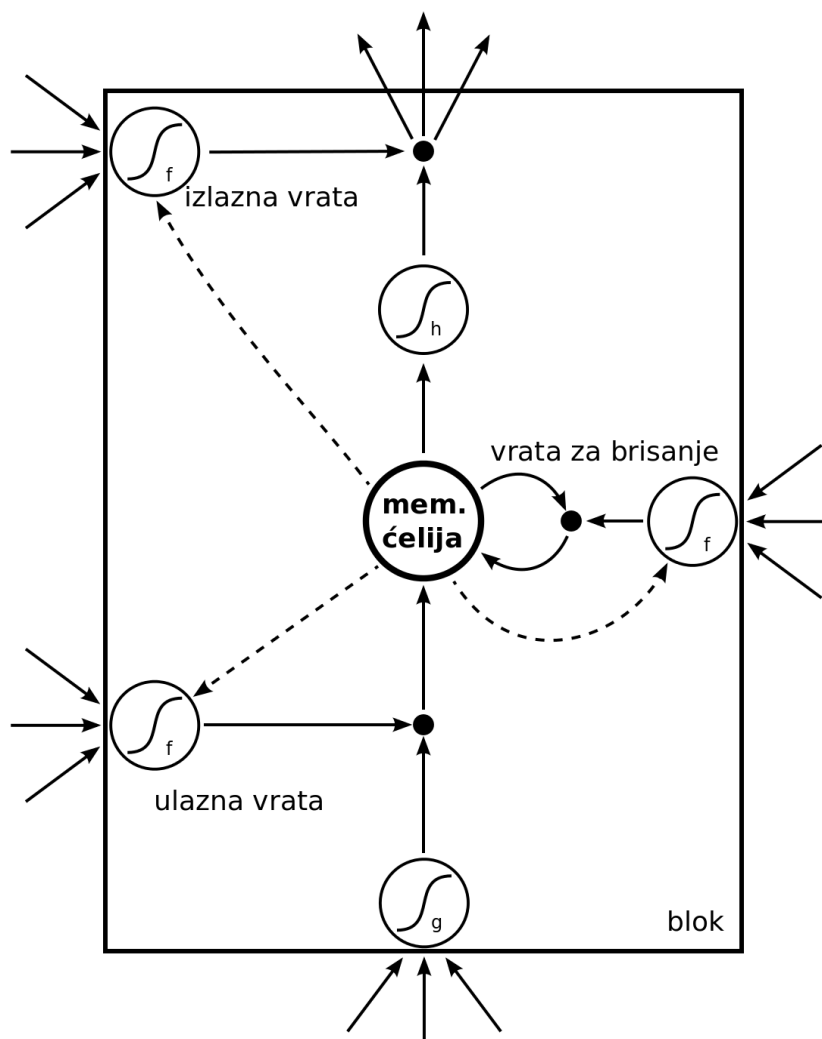
### Pseudokod 2. BRNN prolaz unazad

#### 3.1.2. Dugotrajno-kratkotrajna memorija

Rekurzivne neuronske mreže imaju bolju da kod treniranja pate od ili "eksplozivnog" ili "iščezavajućeg" gradijenta (engl. exploding and vanishing gradient), tj. greška pri prolazu unatrag kroz mrežu ili naglo raste sa svakim korakom ili se naglo smanjuje.

Problem eksplozivnog gradijenta može dovesti do nestabilnosti mreže, pa je jedan način da se tome doskoči smanjiti stopu učenja, što u skoro svim slučajevima vodi u drugu krajnost. Iščezavajući gradijent uzrokuje sporo treniranje mreže za pamćenje dužih vremenskih ovisnosti. Posljedica toga je da rekurzivne neuronske mreže teško pamte kontekst duže od nekoliko desetaka koraka.

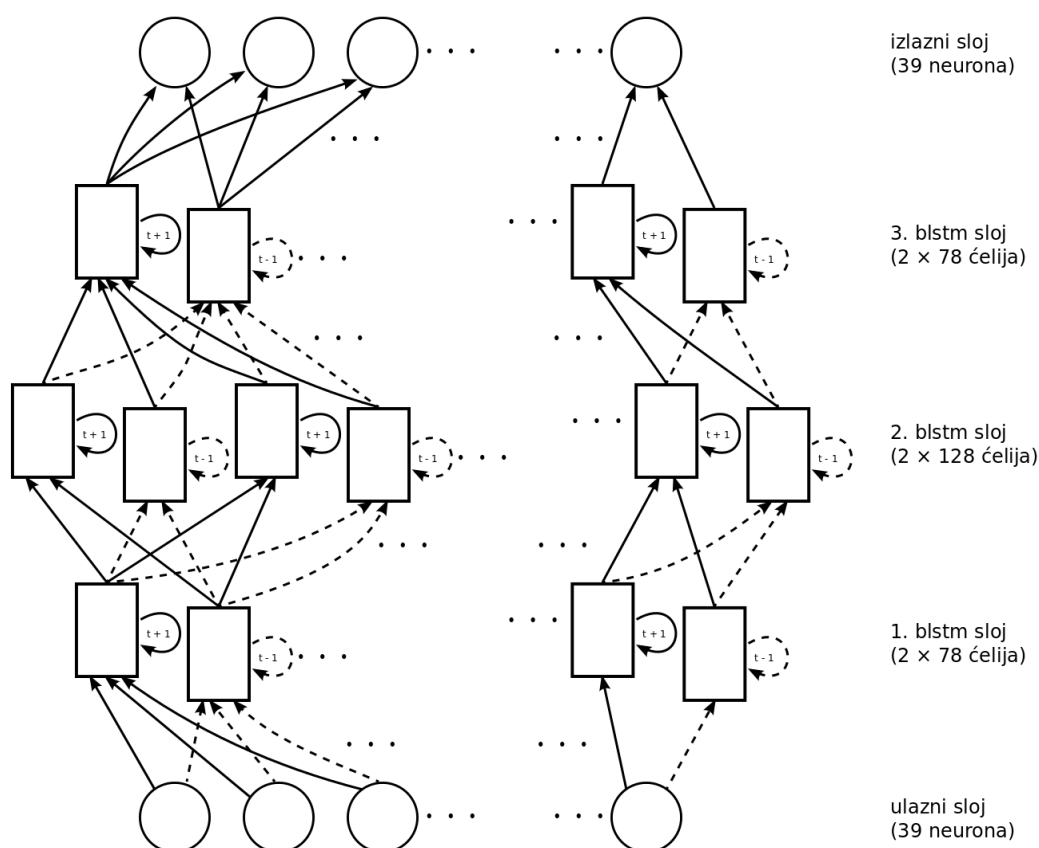
Nadogradnja na rekurzivne mreže koja rješava te probleme je dugotrajno-kratkotrajna memorija ili LSTM [47]. Na slici 3 je prikazana arhitektura LSTM ćelije. Najjednostavnije rečeno, LSTM je diferencijabilna memorijska ćelija koja se može integrirati u neuronsku mrežu.



**Slika 3. LSTM blok s jednom ćelijom.**

Troja vrata koja su prikazana na slici 3 su nelinearne sume koje skupljaju pobude izvan i unutar bloka, i kontroliraju aktivnost ćelije preko množenja (mali crni krugovi). Ulazna i izlazna vrata množe redom ulaz i izlaz ćelije, dok vrata za brisanje množe prethodno stanje ćelije. Sama ćelija nema aktivacijsku funkciju već pamti nepromijenjenu vrijednost koju dobije na ulaz. Aktivacijska funkcija vrata 'f' je obično sigmoidna funkcija, tako da joj je izlaz između 0 (vrata zatvorena) i 1 (vrata otvorena). Ulazna i izlazna aktivacijska funkcija ćelije ('g' i 'h') su obično tangens hiperbolni ili sigmoidna funkcija, iako 'h' nekada može biti i funkcija identiteta. Veze od memorijske ćelije prema vratima (engl. peephole connections) su prikazane isprekidanim strelicama, i one za razliku od ostalih veza unutar bloka imaju težinu [46]. Blok ima četiri ulaza i samo jedan izlaz. Tako svaki LSTM blok

ima sedam parametara : tri unutarnje veze s težinama, te još četiri pomaka (engl. bias) za svaki od ulaza. Izlaz svakog od N neurona na koji je ovaj blok spojen spaja se na sva četiri ulaza, tako da je broj ulaznih težina  $4 * N$ .



**Slika 4. Arhitektura korištene BLSTM-RNN**

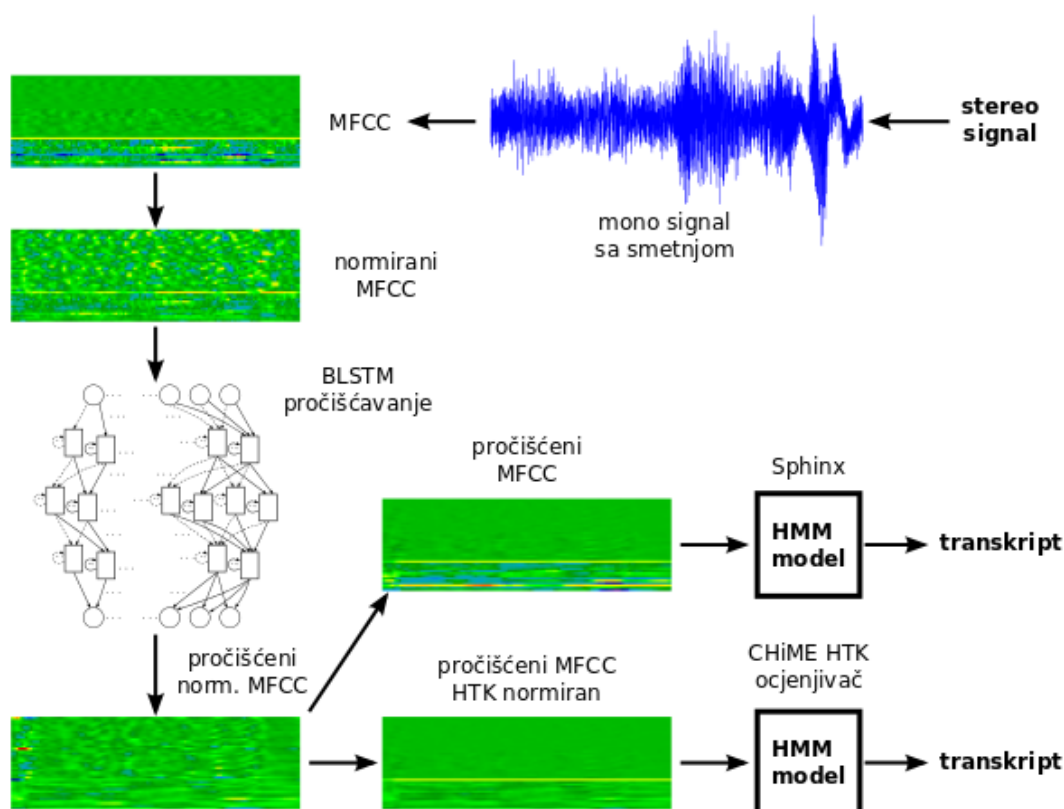
### 3.2. Arhitektura sustava

U ovom radu je korištena dvosmjerna LSTM mreža ili BLSTM mreža, koja je zapravo obična dvosmjerna rekurzivna mreža, samo su neuroni zamijenjeni sa LSTM blokovima.

Na slici 4 je nacrtana arhitektura mreže koja je korištena. Arhitektura je preuzeta iz drugog rada [35] jer zbog ograničenih računalnih resursa nije bilo vremena da se empirijski odredi optimalna veličina i broj slojeva.

Svaki BLSTM blok je povezan sa svim blokovima u slojevima ispod i iznad. BLSTM blok se sastoji od dva nepovezana LSTM bloka (zbog izbjegavanja ciklusa u mreži). Jedan je povezan s izlazom iz prošlog koraka od svih LSTM blokova koji računaju unaprijed u tom sloju. Drugi je povezan s izlazom iz idućeg koraka od svih LSTM blokova koji računaju unazad u tom sloju.

Mreža ima 39 neurona u ulaznom i izlaznom sloju jer toliko parametara ima standardni MFCC\_E\_D\_A\_Z vektor značajki koji se koristi u osnovnom prepoznavачu govora za CHiME2 natjecanje [7]. Ukupni broj parametara za korištenu mrežu je 582339.



**Slika 5. Pregled sustava za izdvajanje i prepoznavanje govora**

Slika 5 prikazuje shemu sustava. Ulazni stereo zvučni zapis se usrednjavanjem oba kanala prebacuje u mono zapis. Zatim se na temelju tog zapisa izračunavaju MFCC značajke metodom opisanom u poglavlju 2.2.5.

Dobivene značajke se normaliziraju s vrijednostima izračunatim na cijelom skupu podataka za treniranje. Tako se ne gubi nikakva informacija, ali se mreža brže trenira [35].

Zatim se izračunava izlaz mreže za cijeli zapis tj. niz značajki. Izlazne značajke iz mreže su također približno normalizirane tako da ih se za korištenje u uobičajenim sustavima za prepoznavanje govora treba ponovno denormalizirati.

CHiME2 osnovni prepoznavač koristi normalizirane značajke, no one su normalizirane na način koji nije sasvim jednoznačno objašnjen u dokumentaciji i drugačiji je od normalizacije korištene za treniranje mreže. Stoga se izlazne MFCC značajke moraju normalizirati tako da im statistička svojstva odgovaraju značajkama na kojima je treniran model za prepoznavanje. To je nužno kako bi se izbjegao pad performansi uslijed razlike između podataka na kojima je obavljeno treniranje modela za prepoznavanje i onima na kojima se ispituje točnost.

### 3.3. Metoda treniranja neuronske mreže

Treniranje i izvršavanje neuronske mreže obavljeno je korištenjem programskog paketa CURRENNT, trenutno jedinog javno dostupnog programskog paketa koji podržava treniranje BLSTM mreža pomoću grafičkih procesora. Korištenje GPU-a u nekim scenarijima omogućuje ubrzavanje treniranja i do 20 puta [43].

Kako bi se ubrzalo treniranje, CURRENNT obavlja treniranje na više ulaznih sekvenci paralelno i tako izračunava gradijent greške na tom podskupu (engl. mini-batch) ulaznih podataka. Zatim se nakon izračuna greške na svakom podskupu osvježavaju težine. Ta metoda se naziva stohastičko hibridno online-batch treniranje.

Kod dubokih neuronskih mreža ključna je dobra početna inicijalizacija, pa CURRENNT podržava podešavanje parametara distribucija za slučajnu inicijalizaciju [48].

Za duboke neuronske mreže također je veliki problem i pretreniranje (engl. overfitting). CURRENNT može koristiti sve tri uobičajene metode [46][48] da bi smanjio problem pretreniranja:

1. uranjeno zaustavljanje (engl. early stopping),

2. zašumljivanje ulaza (engl. input noise),
3. zašumljivanje težina (engl. weight noise).

U ovom radu su korištene sve tri navedene metode.

Zašumljivanje ulaza i težina provodi se tako da se pri treniranju svakom ulazu ili težini pribroji mala slučajna vrijednost. Ideja je da će to smanjiti osjetljivost mreže na nebitne detalje u ulaznim podacima i poboljšati sposobnost generalizacije mreže. Testiranje se provodi bez dodavanja tih slučajnih vrijednosti, jer bi to dovelo do pada performansi.

Za uranjeno zaustavljanje je osim uobičajenog skupa za treniranje i skupa za testiranje potrebno imati i skup za validaciju. Kod uranjenog zaustavljanja mreža se, naravno, trenira na skupu za treniranje. Tijekom treniranja računa se greška na skupu za treniranje i skupu za validaciju.

Skup podataka za validaciju nam omogućava da detektiramo kada je došlo do pretreniranja. Kada mreža počne biti pretrenirana, greška na primjerima za treniranje i dalje pada dok na neviđenim ispitnim primjerima (skupu za validaciju) stagnira ili počinje rasti. Obično se treniranje nastavi još nekoliko epoha nakon toga da bi se osiguralo da je to stvarno minimum, no onda se prekida (po tome je metoda dobila ime). Kao najbolja mreža odabire se ona koja daje najbolji rezultat na skupu za validaciju.

Za sprječavanje pretreniranja se zato u ovom slučaju ne koristi skup za testiranje, jer je odabir najbolje mreže po kriteriju greške na skupu za testiranje ekvivalentan treniranju nekog parametra modela na tom skupu podataka. Skup podataka za testiranje služi da se dobije procjena kako će se mreža ponašati ako na ulaz dobije još neviđene podatke [49], pa ne smije imati nikakvu ulogu u optimizaciji modela [48].

CHiME skup podataka je već podijeljen na skup za treniranje s 17000 zapisa (500 za svakog od 34 govornika), skup za testiranje s 3600 zapisa i skup za validaciju s 3600 zapisa (600 za svaku od 6 SNR vrijednosti) [7]. Skup za testiranje se ionako ne bi mogao koristiti za treniranje ili validaciju jer nisu dostupne snimke čistog govora za taj dio podataka.

## 4. Primjena

### 4.1. Priprema podataka

Prije treniranja mreže potrebno je pripremiti podatke, što je u mnogim primjenama strojnog učenja, a tako i ovdje, velik dio posla.

Za generiranje značajki korišten je openSMILE paket otvorenog koda [50] tvrtke audEERING, koji podržava generiranje HTK-kompatibilnih značajki. No budući da openSMILE ne podržava njihovo normiranje na način koji je potreban, generirane su MFCC\_E\_D\_A značajke, a normalizacija je provedena naknadno.

CURRENNT podacima pristupa isključivo preko NetCDF znanstvenog formata za razmjenu podataka [54], što znači da je sve podatke potrebno prebaciti u taj format [48]. Za to je korišten program 'htk2nc' koji je dio programskog paketa CURRENNT [51].

Normalizacija se obavlja nakon što se skupovi za treniranje, testiranje i validaciju obrade i pospreme u zasebne NetCDF datoteke. U sklopu programskog paketa CURRENNT dostupan je i 'nc-standardize' alat [51] koji izračunava srednje vrijednosti i standardne devijacije za ulazne i izlazne podatke u NetCDF datoteci, te ih sprema u istu datoteku. Ulazne i izlazne nizove u NetCDF datoteci može normalizirati s vrijednostima izračunatima na njima samima ili sa srednjim vrijednostima i standardnim devijacijama drugog skupa podataka.

Normalizacija je provedena tako da su skup podataka za validaciju i testiranje normalizirani sa srednjim vrijednostima i standardnim devijacijama skupa podataka za treniranje.

Skupovi podataka za treniranje i validaciju sastoje se od ulaznih nizova značajki dobivenih od zašumljenog signala i očekivanih nizova značajki koji odgovaraju signalu koji je izobličen samo simuliranim odjekom. Skup podataka za testiranje sadrži samo zašumljene signale. Simulirani odjek prostorije u ovom slučaju ne utječe značajno na točnost prepoznavanja, a probno treniranje je pokazalo da ova neuronska mreža ima problema s učenjem ako joj se daje zadatak da nauči i poništavati utjecaj jeke.



Budući da je priprema podataka bila toliko zahtjevna, u sklopu ovog rada je razvijeno nekoliko skripti koje automatiziraju taj proces. Razvijene Python skripte su javno dostupne [52].

## 4.2. Treniranje mreže

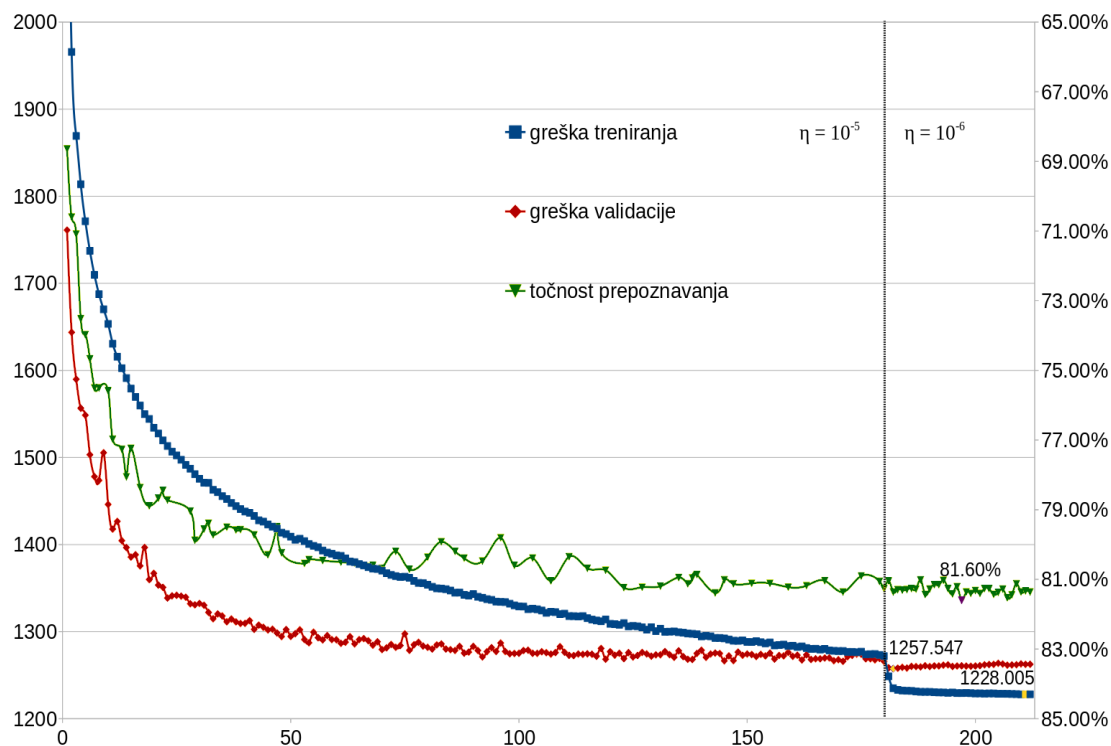
Za treniranje mreže korišteno je računalo s procesorom AMD Athlon II X3 450, s tri jezgre i radnim taktom od 3.2 GHz, te 8 GB radne memorije. Kao grafička kartica korištena je kineska kopija "Nvidia GeForce GT 630" ili sličnog modela kartice s 1 GB grafičke radne memorije i 96 CUDA procesnih jedinica. Iako je kartica nelegitimnog porijekla, podržava naredbe CUDA 2.1 arhitekture, što znači da bez problema može izvršavati sve algoritme za treniranje mreže.

Za treniranje na grafičkim procesorima CURRENNT treba biblioteku CUDA verzije 5 ili više, a korištena je verzija 6.5 [48]. Korišten je operativni sustav Ubuntu Linux 14.04, na kojem su lako dostupni i besplatni svi potrebni paketi za pripremanje izvršnih verzija CURRENNT (verzija 0.2-rc1) i OpenSMILE (verzija 2.1) paketa iz izvornog koda.

Jedna epoha na ovom računalu i u tom programskom okruženju trajala je oko 4850 sekundi, tj. oko 1 sat i 20 minuta. Za treniranje finalne mreže trebalo je 211 epoha, tj. oko 12 dana. No ukupno vrijeme treniranja, s neuspješnim pokušajima, je bilo oko 24 dana. Zbog dugog vremena treniranja mreže i kratkog trajanja semestra nije bilo dovoljno vremena da se eksperimentira s arhitekturom mreže i parametrima treniranja, pa su uzete vrijednosti iz literature [35], što je garantiralo da će se mreža dobro istrenirati i u prvom pokušaju.

Arhitektura mreže je već opisana u poglavlju 3.2. Stopa učenja iznosi  $\eta = 10^{-5}$ , a moment  $m = 0.9$ . Težinama i ulazima se dodaju slučajne vrijednosti iz distribucije sa srednjom vrijednosti  $\mu = 0$  i standardnom devijacijom  $\sigma = 0.1$ . Veličina mini-serije (engl. mini-batch) koja se paralelno obrađuje je 100 ulaznih nizova. Treniranje se zaustavlja kada se u zadnjih 30 epoha nije smanjila greška na skupu za validaciju. CURRENNT je također bio konfiguriran da sprema težine mreže nakon svake epohe, što omogućava naknadno proučavanje svojstava mreže.

Obično se kod korištenja stohastičke inicijalizacije mreža nekoliko puta trenira ispočetka, pa se odabire mreža koja postigne najmanju grešku na validacijskom skupu podataka. To nije napravljeno zbog vremenskih ograničenja.



**Slika 6. Krivulja učenja**

Na slici 6 prikazana je krivulja učenja. Prikazane greške treniranja i validacije su kvadratne sredine razlike između izlaznog vektora značajki i očekivanog vektora značajki. Posebno su istaknute najbolje vrijednosti za sve tri krivulje.

Treniranje mreže sa stopom učenja  $\eta = 10^{-5}$  daje najmanju grešku na validacijskom skupu za epohu 182. Eksperiment je pokazao da smanjivanje stope učenja na  $\eta = 10^{-6}$  i nastavljanje treniranja od epohe 180 dodatno smanjuje grešku na skupu za treniranje i skupu za validaciju.

Na slici 6 prikazana je i točnost prepoznavanja (na obrnutoj skali) na validacijskom skupu podataka korištenjem "Odjek" osnovnog modela. Tijekom provjeravanja uspješnosti rada mreže uočeno je da smanjivanje mjere razlike između izlaznih i očekivanih značajki ne rezultira u svakom slučaju i proporcionalnim smanjivanjem pogreške prepoznavanja.

Stoga je kao finalni kriterij za odabir najbolje mreže uzeta točnost prepoznavanja na validacijskom skupu.

### 4.3. Rezultati

Rezultati za odabranu mrežu (epoha 197) prikazani su u tablici 2. Za usporedbu su dani i rezultati prepoznavanja na zašumljenom govoru. Točnost prepoznavanja čistog govora izobličenog jekom na podacima za validaciju iznosi 93.8%, a na podacima za testiranje je vjerojatno 1 do 2% veća, no to nije moguće provjeriti jer čisti govor nije javno dostupan. Točnost prepoznavanja na čistom govoru je gornja granica točnosti koju može postići teoretski idealni sustav za izdvajanje govora.

Validacija		Test						
WA [%]	sr.vr.	-6dB	-3dB	0dB	3dB	6dB	9dB	sr.vr.
“Odjek” model, govor s bukom	56,9	32,2	38,3	52,1	62,7	76,1	83,8	57,5
“Buka” model, govor s bukom	68,6	49,3	58,7	67,5	75,1	78,8	82,9	68,7
“Odjek” model, izdvojeni govor	81,6	71,3	76,9	82,3	84,9	89,3	90,5	82,5
Prilagođeni model, izdvojeni govor	83,3	73,3	78,8	84,1	87,2	89,1	91,6	84,0
“Odjek” model, govor s jekom	93,8							

**Tablica 2. Rezultati prepoznavanja s istreniranom mrežom**

U tablici 3 dani su rezultati iz rada u kojem je korištena ista strategija za izdvajanje govora [35]. U usporedbi s tim rezultatima dobiveni rezultati su samo 1 do 1.5 % lošiji.

Validacija		Test						
WA [%]	sr.vr.	-6dB	-3dB	0dB	3dB	6dB	9dB	sr.vr.
“Odjek” model, izdvojeni govor	83,1	71,6	78,3	83,1	86,7	89,3	91,1	83,4
Prilagođeni model, izdvojeni govor	84,8	74,8	78,7	86,0	88,3	90,0	92,3	85,0

**Tablica 3. Najbolji postignuti rezultati za odabrani pristup**

U tablici 4 je prikaz prosječnog trajanje epohe za treniranje mreže na običnom i na grafičkom procesoru.

	CPU	GPU
t [s]	12524	4843
ubrzanje	1,0	2,6

**Tablica 4. Trajanje epohe treniranja**

U tablici 5 prikazani su rezultati mjerenja brzine obrade 18 minuta zvučnih zapisa pomoću dobivene mreže i CURRENNT paketa.

	CPU	GPU
t [min]	10	31
RTF	0,56	1,72

**Tablica 5. Trajanje obrade**

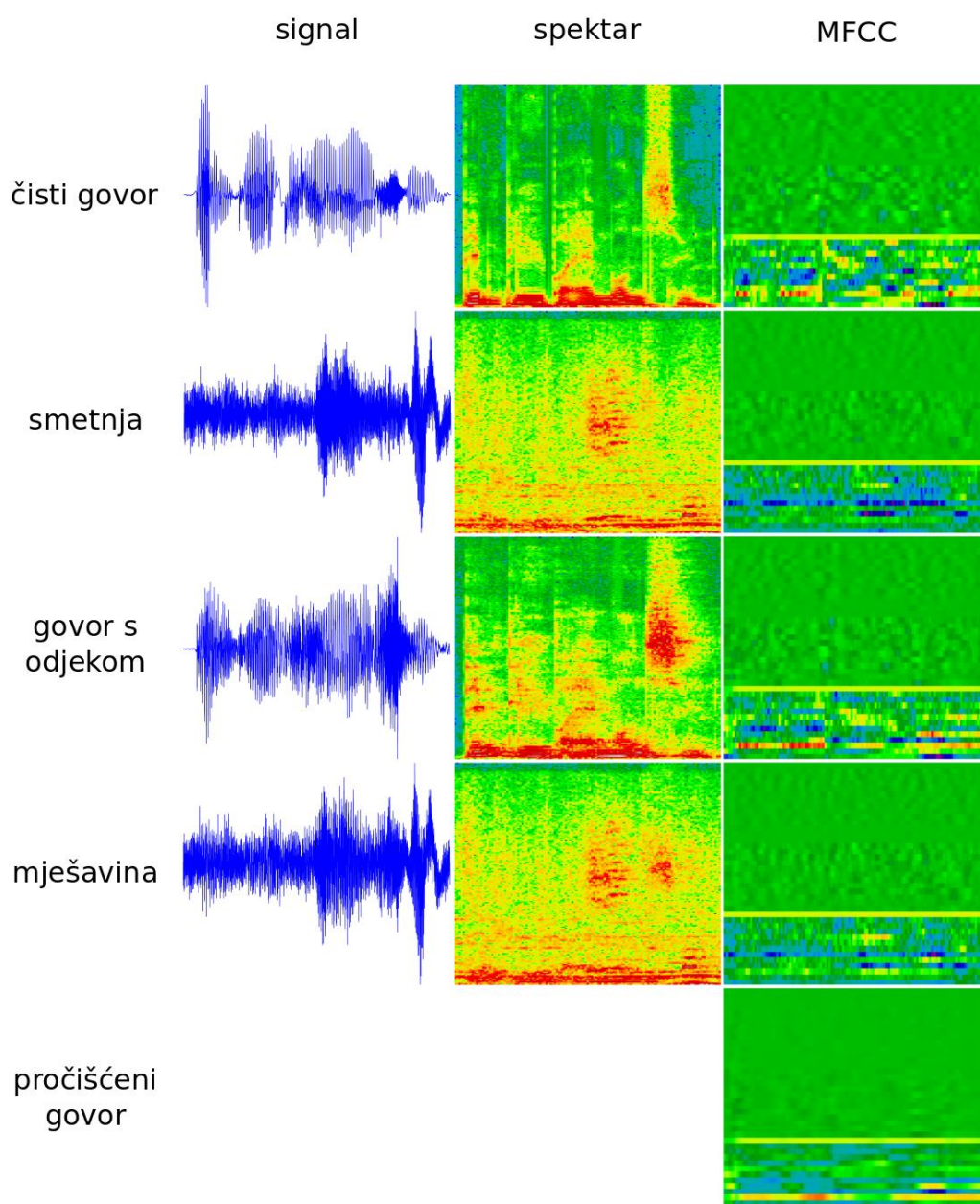
## 5. Diskusija

Dobiveni rezultati su očekivani. Kod sustava za prepoznavanje govora koji je specijaliziran za čisti govor dobiveno je poboljšanje točnosti prepoznavanja od 25% dodavanjem izdvajanja govora. Najveće poboljšanje je dobiveno u slučaju kada je signal govora za 6dB tiši od smetnje, gdje je postignuto poboljšanje od 39.1%. Ovaj rezultat nam govori kakav utjecaj bi dodavanje ovakvog izdvajanja govora moglo imati na performanse sustava za automatsko prepoznavanje govora specijaliziranog za čisti govor bez ikakvih promjena na modelu sustava za prepoznavanje.

Za slučaj kada je prilagođavanje modela moguće, imamo druge rezultate. U usporedbi s prepoznavачem koji je već uvježban na govoru sa smetnjom, prosječno poboljšanje korištenjem izdvajanjem govora i prilagodbom modela je 14.3%, a u najgorim uvjetima prepoznavanja čak 24%.

Ovi rezultati su impresivni jer ovaj sustav svrstavaju među najboljih pet sustava na CHiME2 natjecanju s malim vokabularom [9][12], a sustav je još uvijek relativno male složenosti.

Dobiveni rezultati su 1 do 1.5% lošiji od onih koji su dani za isti pristup u literaturi [35]. Za reproduciranje tog rezultata je vjerojatno bilo potrebno nekoliko puta trenirati mrežu kako bi se dobro inicijalizirala i konvergirala bliže globalnom minimumu. Pristup većim računalnim resursima omogućio bi više eksperimentiranja i sasvim sigurno i bolje rezultate.



**Slika 7. Ilustracija izdvajanja govora iz parametriziranog signala**

Na slici 7 dana je ilustracija izlaza iz mreže i izdvajanja značajki govora. Uzet je primjer signala govora 6 dB tišeg od smetnje. Dane su ilustracije kako jeka i dodavanje buke izobličavaju obični, spektralni i MFCC prikaz govora. Također je vidljivo kako mreža uspijeva proizvesti nešto što je slično MFCC značajkama govora s jekom, iako je prikaz dosta glađi od originala. To po mišljenju autora

može značiti ili da model dobro generalizira ili da bi složeniji model mogao dati još bolje performanse.

Ideja koja stoji iza ovog diplomskog rada bila je istražiti postoji li metoda koja bi omogućila da se iskoristi govor iz brojnih radio i televizijskih emisija na hrvatskom jeziku za razvoj prepoznavanja govora na našem jeziku. U emitiranim emisijama govor često prati muzika, ponekad i slabiji šumovi (ako je govor snimljen van studija) ili govor više osoba koje govore istovremeno. No, zato je snimljeni govor uglavnom jako dobre kvalitete i glasniji je od svih pozadinskih smetnji. Utjecaj jeka je također malen jer je mikrofoni u pravilu blizu govornika. Izdvajanje takvog govora trebalo bi biti lakši problem od izdvajanja govora u CHiME2 skupu podataka, gdje je jeka prisutna u svim snimkama, a i odnos snage govora i smetnje je mnogo nepovoljniji.

S druge strane, u literaturi se iznosi činjenica da dobar rezultat u izdvajanju i prepoznavanju govora s malim rječnikom ne mora nužno biti prenosiv na prepoznavanje govora sa srednjim i velikim rječnicima. CHiME2 natjecanje uključuje i prepoznavanje govora sa srednjim rječnikom.

Čisti govor je uzet iz Wall Street Journal korpusa čitanog govora s rječnikom od 5000 riječi [7]. Nažalost nije bilo moguće dobiti rezultat na tom skupu podataka jer nije javno dostupan [53]. Ipak, rezultati iz literature [35] pokazuju da se metodom korištenom u ovom radu može dobiti slično poboljšanje i na skupu podataka sa srednjim rječnikom.

Svakako je zanimljiva mogućnost izdvajanja i prepoznavanja govora u stvarnom vremenu, i stoga je izmjeren RT faktor za izvršavanje mreže pomoću CURRENNT-a. Zanimljivo je da iako primjena grafičkih procesora ubrzava treniranje mreže za 2.6 puta (tablica 4), to nije slučaj kod primjene mreže uživo. U scenariju kada se mreža primjenjuje serijski na jednom po jednom ulaznom nizu, izvršavanje je 3 puta sporije u usporedbi s izvršavanjem tog algoritma na običnom procesoru (tablica 5). Razlog tome je vjerojatno puno veća latencija uslijed prebacivanja podataka iz glavne memorije u memoriju grafičke kartice.

Zanimljiv je i podatak da openSMILE podržava obradu zvučnog signala uživo pomoću dvosmjernih rekurzivnih neuronskih mreža, no trenutno ne podržava

dvosmjerne rekurzivne mreže. Korištenjem LSTM-RNN u takvom scenariju vjerojatno bi se mogli dobiti usporedivi rezultati, iako vjerojatno nešto lošiji.

Još uvijek treba istražiti utjecaj korištenja drugih značajki za parametriziranje ulaznog govora. Pokazano je da primjena jednostavnijih značajki koje imitiraju svojstva ljudske pužnice (engl. log-filterbank features) daje još bolje rezultate jer omogućava mreži da sama nauči najbolju parametrizaciju signala [36].

Jedan relativno novi doprinos polju dubokog učenja je algoritam za vremensku klasifikaciju pomoću neuronskih mreža ili CTC algoritam [28] [46], koji omogućava treniranje neuronskih mreža koje obavljaju cijeli proces od pročišćavanja govora do njegove transkripcije. To bi riješilo problem koji je bio prisutan u ovom radu, tj. činjenicu da funkcija greške koja se koristi za treniranje mreže ne odgovara onome što se zapravo želi optimizirati (točnost prepoznavanja govora).

Glavna i vjerojatno neizbježna boljka svih spomenutih pristupa temeljenih na dubokom učenju je što zahtijevaju velike računalne resurse. Velike svjetske kompanije koje imaju pristup upravo spomenutim velikim računalnim resursima postižu zavidne rezultate, između ostalog i na računalnom prepoznavanju govora u neidealnim uvjetima [27].

Složenost i „računalna“ moć ljudskog slušnog sustava, koji je svojevrsni ideal kojemu teži računalno prepoznavanje govora, svakako opravdava korištenje barem dijela tih resursa. No isto tako je moguće da se mudrim korištenjem skromnije dostupne računalne moći može postići slične rezultate. Ipak mišljenje autora je da će za to vjerojatno biti potrebno razviti algoritme za brže treniranje dubokih neuronskih mreža.



## 6. Zaključak

U ovom radu dan je uvod u problem računalnog izdvajanja govora. Velika razlika u osjetljivosti ljudskog i računalnog prepoznavanja govora na smetnje dana je kao motivacija za računalno izdvajanje govora.

Nakon kratkog pregleda povijesnih pristupa izdvajanju govora, među najuspješnijim novijim pristupima odabran je pristup korištenjem dubokih neuronskih mreža.

BLSTM-RNN tip neuronske mreže je pokazan kao trenutno najbolji izbor za arhitekturu mreže u ovoj primjeni.

Od pet javno dostupnih paketa za rad s rekurzivnim mrežama odabran je CURRENNT, koji podržava odabrani tip mreže i ubrzavanje treniranja korištenjem grafičkih procesora.

Za ispitivanje uspješnosti izdvajanja govora odabrani su javno dostupni podaci i alati drugog CHiME natjecanja (engl. CHiME 2nd challenge).

Mreža je istrenirana i izmjerena je uspješnost prepoznavanja. Najvažniji rezultat je koliko poboljšanje se dobiva u prepoznavanju govora sa smetnjama korištenjem sustava koji je specijaliziran za prepoznavanje čistog govora. U tom scenariju je dobiveno apsolutno poboljšanje točnosti prepoznavanja od 25% za sve odnose govora i smetnje i čak 39.1% za najnepovoljniji odnos (6 dB u korist smetnje).

Time je demonstrirana učinkovitost izdvajanja govora korištenjem dubokih neuronskih mreža. Dana je analiza koliki je značaj tog rezultata za povećanje otpornosti na smetnje kod sustava za prepoznavanja govora s velikim i srednjim rječnikom i koji su mogući pravci daljnjeg istraživanja.

## 7. Literatura

1. Allen, Jont B. : "Articulation and Intelligibility", Morgan & Claypool, SAD, 2005., str. 2,
2. Benetsy, Jacob; Sondhi, M. Mohan; Huang, Yiteng : "Springer Handbook of Speech Processing", Springer-Verlag, Berlin, 2008., str. 843-845
3. Bregman, Albert S. : "Auditory Scene Analysis : The Perceptual Organization of Sound", MIT Press; Cambridge, Massachusetts; 1994.
4. Wang, DeLiang; Brown; Guy J. : "Computational Auditory Scene Analysis: Principles, Algorithms, and Applications", John Wiley & Sons, Inc., New Jersey, 2006.
5. Divenyi, Pierre: "Speech Separation by Humans and Machines", Kluwer Academic Publishers, Boston, 2005.
6. Hu, Yu Hen; Hwang, Jenq-Neng: "Handbook of Neural Network Signal Processing", CRC Press, Boca Raton, 2002., str. 184, 180
7. Vincent, E.; Barker, J.; Watanabe, S.; Le Roux, J.; Nesta, F; Matassoni, M. : The second CHiME Speech Separation and Recognition Challenge: Datasets, tasks and baselines, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vancouver, 2013.
8. Barker, J.P.; Vincent, E.; Ma, N.; Christensen, H.; Green, P.D.: "The PASCAL CHiME Speech Separation and Recognition Challenge", Computer Speech and Language, Elsevier, 2013, str. 621-633
9. Vincent, E.; Barker, J. ; Watanabe, S. ; Le Roux, J. ; Nesta, F. ; Matassoni, M. : "The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes", IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), IEEE, Olomouc, 2013, str. 162-167
10. Loizou, Philipos C.: "Speech Enhancement: Theory and Practice, 2nd Edition", CRC Press, Boca Raton, 2013, str. 609
11. Cooke, Martin; Barker, Jon; Cunningham, Stuart; Shao, Xu: "An audio-visual corpus for speech perception and automatic speech recognition", The

Journal of the Acoustical Society of America, Acoustical Society of America, SAD, 2006.

12. "The 2nd 'CHiME' Speech Separation and Recognition Challenge: Small vocabulary track",  
[http://spandh.dcs.shef.ac.uk/chime\\_challenge/chime2013/chime2\\_task1.html](http://spandh.dcs.shef.ac.uk/chime_challenge/chime2013/chime2_task1.html),  
25.6.2015

13. Young, Steve; Evermann, Gunnar; Gales, Mark; Hain, Thomas; Kershaw, Dan; Liu, Xunying (Andrew); Moore, Gareth; Odell, Julian; Ollason, Dave; Povey, Dan; Valtchev, Valtcho; Woodland, Phil : "The HTK Book", Cambridge University Engineering Department, <http://htk.eng.cam.ac.uk/docs/docs.shtml>, 2009., str.80

14. Vincent, Emanuel: "The 2nd CHiME Challenge Baseline scoring, decoding and training scripts",  
[http://spandh.dcs.shef.ac.uk/chime\\_challenge/chime2013/grid/README.pdf](http://spandh.dcs.shef.ac.uk/chime_challenge/chime2013/grid/README.pdf),  
2012., 25.6.2015

15. Eyben, Florian; Weninger, Felix; Wollmer, Martin; Schuller, Bjorn: "openSMILE: open-Source Media Interpretation by Large feature-space Extraction", <http://www.audeering.com/research-and-open-source/files/openSMILE-book-latest.pdf>, str. 32, 25.6.2015.

16. Comon, Pierre; Jutten, Christian: "Handbook of Blind Source Separation: Independent Component Analysis and Applications", Academic Press, Oxford, 2010., str. 7-9, 515

17. Liu, Ding; Smaragdis, Paris; Kim, Minje: "Experiments on Deep Learning for Speech Denoising", Proceedings of the annual conference of the International Speech Communication Association (INTERSPEECH), Singapore, 2014.

18. Geiger, Jürgen T.; i drugi: "The TUM+ TUT+ KUL approach to the 2nd CHiME challenge: Multi-stream ASR exploiting BLSTM networks and sparse NMF.", Proc. of CHiME, 2013.

19. Kang, Tae Gyoan; i drugi: "NMF-based Target Source Separation Using Deep Neural Network", Signal Processing Letters, IEEE, 2015.

20. Huang, Po-Sen; Kim, Minje; Hasegawa-Johnson, Mark; Smaragdis, Paris: "Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural

Source Separation", IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, 2015.

21. Le Roux, Jonathan; Hershey, John R.; Wenginger, Felix: "Deep NMF for Speech Separation", <http://www.jonathanleroux.org/pdf/LeRoux2015ICASSP04DeepNMF.pdf>, 2015.

22. Banko, Michele; Brill, Eric: "Scaling to very very large corpora for natural language disambiguation", Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2001.

23. Coates, Adam; Ng, Andrew Y.; Lee, Honglak: "An analysis of single-layer networks in unsupervised feature learning", International conference on artificial intelligence and statistics, 2011.

24. Dean, Jeffrey; Corrado, Greg; Monga, Rajat; Chen, Kai; Devin, Matthieu; Mao, Mark; Senior, Andrew; Tucker, Paul; Yang, Ke; Le, Quoc V.; i drugi: "Large scale distributed deep networks", Advances in Neural Information Processing Systems, 2012.

25. Chetlur, Sharan; Woolley, Cliff; Vandermersch, Philippe; Cohen, Jonathan; Tran, John; Catanzaro, Bryan; Shelhamer, Evan: "cudnn: Efficient primitives for deep learning", arXiv preprint arXiv:1410.0759, 2014.

26. Coates, Adam; Huval, Brody; Wang, Tao; Wu, David; Catanzaro, Bryan; Andrew, Ng: "Deep learning with COTS HPC systems", Proceedings of the 30th international conference on machine learning, 2013.

27. Hannun, Awni; i drugi: "DeepSpeech: Scaling up end-to-end speech recognition.", arXiv preprint arXiv:1412.5567, 2014.

28. Graves, Alex; Jaitly, Navdeep: "Towards end-to-end speech recognition with recurrent neural networks." Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014.

29. Grais, Emad M.; Sen, Mehmet Umut; Erdogan, Hakan: "Deep neural networks for single channel source separation." Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014.

- 
30. Narayanan, Arun; Wang, DeLiang: "Investigation of speech separation as a front-end for noise robust speech recognition", Audio, Speech, and Language Processing, IEEE/ACM Transactions on, IEEE, 2014.
31. Huang, Po-Sen; Kim, Minje; Hasegawa-Johnson, Mark; Smaragdis, Paris: "Deep learning for monaural speech separation", Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 2014.
32. Xu, Yong; Du, Jun; Dai, Li-Rong; Lee, Chin-Hui: "An experimental study on speech enhancement based on deep neural networks", Signal Processing Letters, IEEE, 2014.
33. Weng, Chao; Yu, Dong; Seltzer, Michael L.; Droppo, Jasha: "Single-channel mixed speech recognition using deep neural networks", Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE. 2014.
34. Huang, Po-Sen; Chen, Scott Deeann; Smaragdis, Paris; Hasegawa-Johnson, Mark: "Singing-voice separation from monaural recordings using robust principal component analysis", Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012.
35. Weninger, Felix; Geiger, Jurgen; Wollmer, Martin; Schuller, Bjorn; Rigoll, Gerhard: "The Munich feature enhancement approach to the 2nd CHiME challenge using BLSTM recurrent neural networks", Proceedings of the 2nd CHiME workshop on machine listening in multisource environments, Citeseer, 2013.
36. Weninger, Felix; Geiger, Jurgen; Wollmer, Martin; Schuller, Bjorn; Rigoll, Gerhard: "Feature enhancement by deep LSTM networks for ASR in reverberant multisource environments", Computer Speech & Language, Elsevier, 2014
37. Weninger, Felix; Hershey, John R.; Le Roux, Jonathan; Schuller, Bjorn: "Discriminatively trained recurrent neural networks for single-channel speech separation", Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on, IEEE, 2014.
38. Weninger, Felix; Geiger, Jurgen; Wollmer, Martin; Schuller, Bjorn; Rigoll, Gerhard: "The Munich 2011 CHiME Challenge Contribution: BLSTM-NMF Speech Enhancement and Recognition for Reverberated Multisource Environments",

CHiME 2011 Workshop on Machine Listening in Multisource Environments, Citeseer, 2011.

39. Schaul, Tom; Bayer, Justin; Wierstra, Daan; Sun, Yi; Felder, Martin; Sehnke, Frank; Rückstieß, Thomas; Schmidhuber, Jürgen: "PyBrain", Journal of Machine Learning Research, JMLR. org, 2010.

40. Bastien, Frederic; Lamblin, Pascal; Pascanu, Razvan; Bergstra, James; Goodfellow, Ian; Bergeron, Arnaud; Bouchard, Nicolas; Warde-Farley, David; Bengio, Yoshua : "Theano: new features and speed improvements", arXiv preprint arXiv:1211.5590, 2012.

41. Bergstra, James; Breuleux, Olivier; Bastien, Frederic; Lamblin, Pascal; Pascanu, Razvan; Desjardins, Guillaume; Turian, Joseph; Warde-Farley, David; Bengio, Yoshua : "Theano: a CPU and GPU math expression compiler", Proceedings of the Python for scientific computing conference, SciPy, 2010.

42. Collobert, Ronan; Kavukcuoglu, Koray; Farabet, Clement: "Torch7: A matlab-like environment for machine learning", BigLearn, NIPS Workshop, 2011.

43. Weninger, Felix; Bergmann, Johannes; Schuller, Bjorn: "Introducing CURRENNT: The Munich Open-Source CUDA RecurREnt Neural Network Toolkit", Journal of Machine Learning Research, 2015.

44. Alex Graves, RNNLIB: A recurrent neural network library for sequence learning problems, <http://sourceforge.net/projects/rnnl/>, 25.6.2015

45. "Parallel Programming and Computing Platform - CUDA - NVIDIA", [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html), 25.6.2015

46. Graves, Alex: "Supervised Sequence Labelling with Recurrent Neural Networks", Springer, <http://www.cs.toronto.edu/~graves/preprint.pdf>, 2012., str. 20, 21, 26

47. Hochreiter, Sepp; Schmidhuber, Jurgen: "Long short-term memory", MIT Press, 1997.

48. "CURRENNT README", [sourceforge.net/projects/currennt/](http://sourceforge.net/projects/currennt/), 25.6.2015

49. Elkan, Charles: "Evaluating Classifiers", 20.1.2012, <http://cseweb.ucsd.edu/~elkan/250Bwinter2012/classifiereval.pdf>, 25.6.2015

50. Eyben, Florian; Weninger, Felix; Gross, Florian; Schuller, Bjoern: "Recent developments in opensmile, the munich open-source multimedia feature extractor", Proceedings of the 21st ACM international conference on Multimedia, ACM, 2013.

51. "CURRENNT tools README", [sourceforge.net/projects/currennt/](https://sourceforge.net/projects/currennt/), 25.6.2015

52. Henc, Stjepan: "Scripts for creating netCDF databases for CURRENNT", [https://github.com/sthenc/nc\\_packer](https://github.com/sthenc/nc_packer), 25.6.2015

53. "The 2nd 'CHiME' Speech Separation and Recognition Challenge: Medium vocabulary track", [http://spandh.dcs.shef.ac.uk/chime\\_challenge/chime2013/chime2\\_task2.html](http://spandh.dcs.shef.ac.uk/chime_challenge/chime2013/chime2_task2.html), 25.6.2015

54. "Network Common Data Form (NetCDF)", <http://www.unidata.ucar.edu/software/netcdf/>, 25.6.2015